UNIVERSITY OF CALIFORNIA SAN DIEGO

From Information Theory to Machine Learning Algorithms:
A Few Vignettes

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Electrical Engineering
(Communication Theory and Systems)

by

Jongha Jon Ryu

Committee in charge:

Professor Young-Han Kim, Chair
Professor Sanjoy Dasgupta, Co-Chair
Professor Ery Arias-Castro
Professor Nikolay Atanasov
Professor Yoav Freund
Professor Piya Pal

2022

The Dissertation of Jongha Jon Ryu is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2022

DEDICATION

To my soulmate Kyungeun,
and my babies Jian and Yeonu.

For their endless support and love.

EPIGRAPH

"This is OK, but I think things could be done better.
I think there is a neater way to do this.
I think things could be improved a little."
In other words, there is continually a slight irritation
when things don't look quite right;
and I think that dissatisfaction in present days
is a key driving force in good scientists.

Claude E. Shannon[1]

[1]Claude Shannon, "Creative Thinking," March 20, 1952, in *Claude Shannon's Miscellaneous Writings*, ed. N. J. A. Sloane and Aaron D. Wyner (Murray Hill, NJ: Mathematical Sciences Research Center, AT&T Bell Laboratories, 1993), 528–39.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I came to San Diego alone with B.S. degrees seven years ago, but now I begin a new journey in my career, leaving San Diego with a Ph.D. degree and three family members. Like everyone else going through this long journey, it would not have been possible for me to get to the end without help from others.

My first and foremost thanks should go to my advisors. Not only being an exemplar of a great information theorist, Prof. Kim has taught me almost everything to be an independent researcher, step-by-step, as a father fosters a baby. He has taught me how to think, how to write, how to talk, how to solve, and how to find a good research problem. He did not always give a fish, however, but often waited for me to fish one by myself with unlimited patience. I realized how hard it is only after being a father of little kids. Words cannot express my gratitude to him for all the support he has provided me during the seven years of my Ph.D., from every aspect of my life. Prof. Dasgupta has been always a great mentor and teacher for me. His one-of-a-kind style and taste in research have greatly influenced my viewpoint on how to approach to a research problem. I feel extremely fortunate that I learned a lot of fascinating topics on machine learning directly from him. I should also thank for his great patience with me, as I have been a very slow learner and worker with him. When I decided to go to a graduate school, my biggest wish was to have an advisor whom I can sincerely respect. I am really fortunate that I ended up having two, not even just only one.

I am also deeply indebted to my committee members Prof. Ery Arias-Castro, Prof. Yoav Freund, Prof. Nikolay Atanasov, Prof. Piya Pal, for graciously serving on the committee, their teaching, invaluable feedback on my research, career advice, as well as emotional supports. I would like to thank my senior collaborators and mentors in academia, especially Prof. Yung-Kyun Noh for his advice in my research and collaboration, and one of my undergraduate advisors Prof. Jungwoo Lee for his constant encouragement and warm hospitality whenever I met him at conferences.

I was very fortunate to spend two summers as a research intern with great mentors at Samsung and Qualcomm. The work I started working at Samsung with Dr. Yoojin Choi in Summer 2018 has become the main piece of my thesis, which is the first of my talk. I am extremely grateful to Dr. Choi for his time, work, advices, and great patience to help me gradually develop the work into the final form for the last 4 years. Many thanks goes to Dr. Yang Yang at Qualcomm, who helped me explore premature information-theoretic ideas during the internship and taught me how to write a code systematically for a large project. I also thank to Dr. Jungwon Lee at Samsung and Dr. Jilei Hou at Qualcomm for giving me the internship opportunities. My dissertation would have not been possible without them and the summers I spent there.

I would like to thank everyone I met in Prof. Kim's group, particularly Shouvik Ganguly, Pinar Sen, Alankrita Bhatt, Nadim Ghaddar, Jiun-Ting Huang, who have been together for the most of my Ph.D. years, not only for their research collaboration and intellectual stimulus, but also for their great friendship. I should make another word to Shouvik for our long collaboration we went through, our intimacy, the intellectual rapport, and the friendship he has offered to me. I thank Alanrkita for our countless time we spent together reading lecture notes and papers and trying to decipher them. I would like to extend my sincere thanks to Prof. Lele Wang and Prof. Yu Xiang, who have been great mentors giving me endless amount of career advice and emotional support.

I am also grateful for the members of Prof. Dasgupta's group for their exciting research discussions and reading groups I had with them. Though I tended to be a rather quiet listener there, I learned a lot from all of them. Especially I thank to Geelon for being a wonderful research collaborator and a good friend.

I am grateful for NAVER, Samsung, and NSF for their funding that have supported my Ph.D. research. Sincere thanks should also go to Kwanjeong educational foundation, for the gracious support for the first five years through their scholarship, as

well as the support for my undergraduate study. It would have been impossible for me to survive along this journey without their support.

Needless to say, I am indebted to many friends from San Diego, other areas of US, and Korea for their emotional support and the time we spent together in the course of my Ph.D study. I especially would like to thank Moojin, Jinhyung, Wonmin, Jaehyun, and Minchul for their friendship during my first two years of the graduate study. I will not be able to forget the trips to the Joshua Tree National Park. I am also grateful to thank my good old friend Joonwon for his friendship and endless generosity to listen to my anxieties on a daily basis during my Ph.D. I would also like to thank Dogyoon, Jeongyeol, Heewoo, Younghoon, and Dongkwun, who spent the time together preparing to study abroad, for their emotional support from a variety of regions in the United States.

I should also thank to my parents Sihyun Ryu and Youngnam Choi. This endeavor would not have been possible without all the love and support from them throughout my life, especially their consistent encouragement during my Ph.D. years.

I would like to express my deepest appreciation to my wife Kyungeun for not only the boundless love and support during my Ph.D., but also being a soulmate for the last 14 years since we met at a college as freshmen. She sacrificed her career and decided to be with me during my Ph.D. I cannot imagine how I would have survived with all the ups and downs without her holding me tight. Last but not least, special thanks to my two little babies Jian and Yeonu. Unlike the conventional saying, my research productivity has substantially increased for the last few years, by their unconditional love and cuteness, and I finally graduate thanks to them. This dissertation is dedicated to my wife and the kids.

Chapter 1, in part, is a reprint of the material in the paper: J. Jon Ryu, Yoojin Choi, Young-Han Kim, Mostafa El-Khamy, and Jungwon Lee, "Learning with Succinct Common Representation Based on Wyner's Common Information," arXiv:1905.10945v2,

July 2022, submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence*. The dissertation author was the primary investigator and author of this paper. This work was supported in part by the National Science Foundation under Grant CCF-1911238.

Chapter 2, in part, is a reprint of the material in the paper with permission: © 2021 IEEE. J. Jon Ryu, Jiun-Ting Huang, and Young-Han Kim, "On the role of eigen-decomposition in kernel embedding," In *Proceedings of IEEE International Symposium on Information Theory*, pp. 2030–2035, Melbourne, Australia, July 2021. The dissertation author was the primary investigator and author of this paper. This work was supported in part by the National Science Foundation under Grant CCF-1911238.

Chapter 3, in part, is a reprint of the material in the paper: J. Jon Ryu and Young-Han Kim, "One-Nearest-Neighbor Search Is All You Need for Minimax Regression and Classification," February 2022, arXiv:2202.02464, submitted to *The Thirty-Sixth Annual Conference on Neural Information Processing Systems*. The dissertation author was the primary investigator and author of this paper. This work was supported in part by the National Science Foundation under Grant CCF-1911238.

Chapter 4, in part, is a reprint of the material in the paper with permission: © 2022 IEEE. J. Jon Ryu, Shouvik Ganguly, Young-Han Kim, Yung-Kyun Noh, Daniel Lee, "Nearest neighbor density functional estimation from inverse Laplace transform," *IEEE Transactions on Information Theory*, vol. 68, Issue 6, pp. 3511–3551, June 2022. The dissertation author was the primary investigator and author of this paper. This work was supported in part by the National Science Foundation under Grant CCF-1911238.

Chapter 5, in part, is a reprint of the material in the paper with permission: © 2018 IEEE. Jongha Ryu and Young-Han Kim, "Conditional distribution learning with neural networks and its application to universal image denoising," In *Proceedings of IEEE International Conference on Image Processing*, pp. 3214–3218, Athens, Greece, October 2018. The dissertation author was the primary investigator and author of this paper.

Chapter 6, in part, is a reprint of the material in the paper: J. Jon Ryu, Alankrita

Bhatt, and Young-Han Kim, "Parameter-Free Online Linear Optimization with Side Information via Universal Coin Betting," In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, A Virtual Conference, March 2022. The dissertation author was the primary investigator and author of this paper. This work was supported in part by the National Science Foundation under Grant CCF-1911238.

VITA

2015	Bachelor of Science in Electrical and Computer Engineering Seoul National University
2015	Bachelor of Science in Mathematical Science Seoul National University
2011–2012	R.O.K. Army (mandatory service)
2016–2022	Graduate Student Researcher, University of California San Diego
2018	Master of Science in Electrical Engineering (Communication Theory and Systems) University of California San Diego
2022	Doctor of Philosophy in Electrical Engineering (Communication Theory and Systems) University of California San Diego

PUBLICATIONS

* indicates equal contributions. † indicates that the author ordering is alphabetical.

Jongha Ryu and Young-Han Kim, "Conditional distribution learning with neural networks and its application to universal image denoising," In *Proceedings of IEEE International Conference on Image Processing*, pp. 3214–3218, Athens, Greece, October 2018.

Alanrkita Bhatt†, Jiun-Ting Huang†, Young-Han Kim†, J. Jon Ryu†, and Pinar Sen†, "Monte Carlo methods for randomized likelihood decoding," In *Proceedings of the 56th Annual Allerton Conference on Communication, Control, and Computing*, pp. 204–211, Monticello, Illinois, October 2018.

Alankrita Bhatt†, Jiun-Ting Huang†, Young-Han Kim†, J. Jon Ryu†, and Pinar Sen†, "Variations on a theme by Liu, Cuff, and Verdú: The power of posterior sampling," In *Proceedings of IEEE Information Theory Workshop*, pp. 290–294, Guangzhou, China, November 2018.

Yang Yang, Guillaume Sautière, J. Jon Ryu, and Taco S. Cohen, "Feedback Recurrent Autoencoder," In *Proceedings of 2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, Barcelona, Spain, May 2020.

J. Jon Ryu, Jiun-Ting Huang, and Young-Han Kim, "On the role of eigendecomposition in kernel embedding," In *Proceedings of IEEE International Symposium on Information*

Theory, pp. 2030–2035, Melbourne, Australia, July 2021.

J. Jon Ryu and Young-Han Kim, "An information-theoretic proof of the Kac–Bernstein theorem," arXiv:2202.06005, February 2022.

J. Jon Ryu and Young-Han Kim, "One-Nearest-Neighbor Search Is All You Need for Minimax Regression and Classification," arXiv:2202.02464, February 2022, Submitted to *The Thirty-Sixth Annual Conference on Neural Information Processing Systems*.

Alankrita Bhatt*, J. Jon Ryu*, and Young-Han Kim, "On Universal Portfolios with Continuous Side Information," arXiv:2202.02431, February 2022.

J. Jon Ryu, Alankrita Bhatt, and Young-Han Kim, "Parameter-Free Online Linear Optimization with Side Information via Universal Coin Betting," In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, A Virtual Conference, March 2022.

J. Jon Ryu*, Shouvik Ganguly*, Young-Han Kim, Yung-Kyun Noh, Daniel Lee, "Nearest neighbor density functional estimation from inverse Laplace transform," *IEEE Transactions on Information Theory*, vol. 68, Issue 6, pp. 3511–3551, June 2022.

J. Jon Ryu, Yoojin Choi, Young-Han Kim, Mostafa El-Khamy, and Jungwon Lee, "Learning with Succinct Common Representation Based on Wyner's Common Information," *arXiv:1905.10945v2*; Submitted to *IEEE Transactions on Patter Analysis and Machine Intelligence*; A preliminary work was presented in *NeurIPS Workshop on Bayesian Deep Learning*, Montreal, Canada, December 2018. An extended abstract was presented in *NeurIPS Workshop on Bayesian Deep Learning*, A Virtual Conference, December 2021.

ABSTRACT OF THE DISSERTATION

From Information Theory to Machine Learning Algorithms:
A Few Vignettes

by

Jongha Jon Ryu

Doctor of Philosophy in Electrical Engineering
(Communication Theory and Systems)

University of California San Diego, 2022

Professor Young-Han Kim, Chair
Professor Sanjoy Dasgupta, Co-Chair

This dissertation illustrates how certain information-theoretic ideas and views on learning problems can lead to new algorithms via concrete examples. The three information-theoretic strategies taken in this dissertation are (1) to abstract out the gist of a learning problem in the infinite-sample limit; (2) to reduce a learning problem into a probability estimation problem and plugging-in a "good" probability; and (3) to adapt and apply relevant results from information theory. These are applied to three topics in machine learning, including representation learning, nearest-neighbor methods, and universal information processing, where two problems are studied from each topic.

Introduction

A high-level goal of this dissertation is to investigate how to design a provably efficient machine learning algorithm based on an information-theoretic insight. Since this research program is rather generic at this level, we motivate our perspective below, by examining the goals of information theory and machine learning.

Since initiated by C. Shannon's seminal work on a mathematical theory of communication (Shannon, 1948), information theory has virtually underpinned almost all modern information technologies over the last seven decades, having grown as an independent, impactful field of engineering. Roughly put, information theory studies how far efficient we can (*fundamental limits*) and how we should (*coding schemes*) communicate over a channel (a *conditional distribution*), or compress a source (a *probability distribution*). In the course of developing important fundamental limits and practical coding schemes, information theory has also proposed a number of interesting information measures and genuine mathematical tools, which have been successfully applied to various different areas such as economics, statistics, and computer science. For an overview of the important results in information theory, we refer an interested reader to the standard textbooks on information theory by Cover and Thomas (2006) and El Gamal and Kim (2011). See also (Csiszár and Körner, 2011; Csiszár and Shields, 2004; Duchi, 2019; MacKay et al., 2003; Mezard and Montanari, 2009; Polyanskiy and Wu, 2014).

Machine learning is the field of study on how to *learn* or *learn about* a probability distribution from their *samples*. For example, in the standard problem setting of clas-

sification, a learner is provided a set of training samples, where each sample is a pair of an instance and its label, assumed to be drawn from some (unknown) underlying distribution; the goal of a learner here is to make a guess on the label of a new test instance as correctly as possible, under a choice of performance criterion. Depending on the form of distributions and the task, we can define a wide range of different machine learning problems, such as supervised learning (e.g., classification and regression), unsupervised learning (e.g., clustering, density estimation, image generation), online learning (e.g., bandit learning, reinforcement learning). In any case, the ultimate goal of machine learning is to design a practical algorithm for a given learning problem that is computationally fast, sample-efficient, and provably optimal.

Clearly, at the very heart, information theory and machine learning have *probability distributions* as their central object of study in common: information theory aims to deal with a fundamental property of a probability distribution, and machine learning tries to infer about an unknown distribution from its samples. Hence, several connections have been already established in the literature. For example, see (MacKay et al., 2003) for a nice introduction to machine learning from an information-theoretic perspective and see (Duchi, 2019) for the role of information theory as the source of a variety of technical tools in modern statistics.

One natural question that arises at the interface between the two fields, which has driven the development of this dissertation, is:

How can we use tools and lessons from information theory to develop machine learning algorithms?

While there may exist many other alternatives that can be qualified as "information-theoretic" strategies, we take the following, high-level strategies throughout this dissertation:

1. *Abstract the gist of a learning problem in the infinite-sample limit.* Information theorists are used to contemplating in an asymptotic regime, rooted in the spirit of Shannon's characterization of a channel capacity. Although it might sound too simplistic to get rid of the finiteness of samples in a learning problem, oftentimes it suffices for providing a good, first-order principle.

2. *Reduce a learning problem to a probability estimation problem and plug-in a "good" probability.* This strategy is in the spirit of *universal compression* (see, e.g., (Cover and Thomas, 2006, Section 11.3)); a source compression problem can be reduced to a probability estimation problem under the log loss, via the use of arithmetic coding (Rissanen and Langdon, 1979). This reduction-based modular approach often gives a simple and elegant solution to a learning problem.

3. *Adapt and apply relevant ideas from information theory.* After a simplification of a learning problem at hand by the two strategies above (or possibly something else), one may seek an analogue of the problem from information theory and adapt the existing solution back to the learning problem.

In this dissertation, we illustrate how these information-theoretic strategies can lead to new machine learning algorithms, via concrete examples. We study two problems from three selected topics in machine learning, including *representation learning* (Part I), *nearest-neighbors methods* (Part II), and *universal information processing* (Part III), as summarized below. A reader can the chapters independently. We conclude each chapter with some concluding remarks and future directions to pursue.

Part I. Representation Learning

In Part I, we study the problem of representation learning. The ultimate goal in representation learning is to compute a *succinctly good* representation of given data, in that it makes a learner *easier* to learn from the representation in place of the raw data, by

preferably having a much lower dimension than the ambient dimension and having a "good" structure.

In Chapter 1, we study the problem of *cross-domain disentanglement*, where the goal is to learn a good joint representation with a certain disentanglement structure of a pair of two high-dimensional random objects. Thinking in the infinite-sample limit (Strategy 1) and inspired by network information theory (Strategy 3), we propose a new definition for an optimal common representation based on the notion of Wyner's common information (Wyner, 1975); it is the first principled attempt to define an optimality of cross-domain disentanglement in the literature. We then propose a new generative model framework based on the definition of the optimal structure, and present an adversarial training method for various learning tasks such as joint generation, conditional generation, and cross-domain retrieval.

In Chapter 2, we study how to *efficiently* compute a good representation using kernels. While being widely used in practice due to their simplicity and decent performance, the standard kernel-based embedding methods such as kernel PCA (Schölkopf et al., 1998) and Laplacian eigenmaps (Belkin and Niyogi, 2003) inherently suffer a demanding computational complexity. The computational bottleneck is at an eigendecomposition step which is crucial in the class of methods. In this work, we revisit the existing methods, examine their optimization-problem characterizations in the infinite-sample limit (Strategy 1), and propose a new embedding algorithm which only requires density estimation rather than the cumbersome eigendecomposition.

Part II. Nearest-Neighbor Methods

One of the simplest class of nonparametric algorithms for such problems is the class of k-nearest-neighbor (k-NN) based algorithms, which is appealing due to their simplicity, decent performance, and rich understanding of their statistical properties. We remark, however, two general limitations of the NN-based methods. First, while

the number of neighbors k needs to grow to infinity in the sample size to achieve statistical consistency in general for such procedures, small k is highly preferred in practice to avoid possibly demanding time complexity of large-k-NN search. Second, k-NN based algorithms are often deemed to be inherently infeasible for large-scale data, as they need to store and process the entire data in a single machine for NN search. The central question in this part of dissertation is whether we can design an NN-based algorithm that circumvents such practical issues. Towards the goal, we specifically focus on developing and analyzing various algorithms using fixed-k-NNs.

In Chapter 3, we show how to perform minimax rate-optimal classification and regression for any fixed k-NN search, especially even for $k = 1$. To derive the proposed algorithm, we make two information-theoretic observations (Strategy 3) in the infinite-sample limit (Strategy 1). First, the 1-NN classifier behaves as the randomized likelihood detector which outputs a random draw from the posterior distribution, in the infinite-sample limit (Cover and Hart, 1967). Second, by aggregating multiple random draws from the posterior distribution instead of a single draw, the detection error can be shown to be exponentially close in the number of draws (Bhatt et al., 2018). Consequently, fusing the insights, we study a distributed nearest-neighbor classification method, in which a massive dataset is split into smaller groups, each processed with a k-nearest-neighbor classifier, and the final class label is predicted by a majority vote among these groupwise class labels. As expected, we indeed show that the distributed algorithm with $k = 1$ over a sufficiently large number of groups attains a minimax optimal error rate up to a multiplicative logarithmic factor under some regularity conditions, for both regression and classification problems. Roughly speaking, distributed 1-nearest-neighbor rules with M groups has a performance comparable to standard $\Theta(M)$-nearest-neighbor rules. More importantly, the proposed algorithm is fully parallelizable and thus naturally suitable for large-scale data.

In Chapter 4, we study how to estimate a single-density or double-density

functional using fixed-k-NNs, where the functional under consideration is in the form of the expectation of some function f of the densities at each point. We propose a new approach to L_2-consistent estimation of a general density functional using k-nearest neighbor distances. The estimator is designed to be asymptotically unbiased, using the convergence of the normalized volume of a k-nearest neighbor ball to a Gamma distribution in the infinite-sample limit (Strategy 1), and naturally involves the inverse Laplace transform of a scaled version of the function f. Some instantiations of the proposed estimator recover existing k-NN based estimators of Shannon and Rényi entropies and Kullback–Leibler and Rényi divergences, and discover new consistent estimators for many other functionals such as logarithmic entropies and divergences. We establish the L_2-consistency of the proposed estimator for a broad class of densities for general functionals, and the convergence rate in mean squared error as a function of the sample size for smooth, bounded densities.

Part III. Universal Information Processing

Many, if not most, existing data processing algorithms are designed upon some assumptions on unknown data generating processes and are sometimes guaranteed to be optimal under such premises. With the ever increasing amount of more complex and high-dimensional data, however, such assumptions neither are verifiable, nor could accurately reflect the reality (Wasserman, 2011); i.e., "all models are wrong", as famously quoted. Towards building intelligent systems without potential risks from misspecified assumptions, we are interested in developing an algorithm that provably works well under minimal statistical requirements. In this part of dissertation, we especially demonstrate the power of the reduction-and-plug-in approach (Strategy 2) for the discrete sequence denoising and online learning with side information problems.

In Chapter 5, we propose a simple and scalable discrete denoising algorithm that can be applied to a wide range of source and noise models. At the core of the proposed

CUDE algorithm is symbol-by-symbol universal denoising used by the celebrated DUDE algorithm (Weissman et al., 2005), whereby the optimal estimate of the source from an unknown distribution is computed by inverting the empirical distribution of the noisy observation sequence by a deep neural network, which naturally and implicitly aggregates multiple contexts of similar characteristics and estimates the conditional distribution more accurately (Strategy 2). The performance of CUDE is evaluated for grayscale images of varying bit depths, which improves upon DUDE and its recent neural network based extension, Neural DUDE (Moon et al., 2016).

In Chapter 6, we propose a class of parameter-free online linear optimization algorithms that harnesses the structure of an adversarial sequence by adapting to some side information. These algorithms combine the reduction technique of Orabona and Pál (2016) for adapting coin betting algorithms for online linear optimization with universal compression techniques in information theory (Strategies 2 and 3) for incorporating sequential side information to coin betting. We study concrete examples in which the side information has a tree structure and consists of quantized values of the previous symbols of the adversarial sequence, including fixed-order and variable-order Markov cases. By modifying the context-tree weighting technique of Willems et al. (1995) (Strategy 3), we further refine the proposed algorithm to achieve the best performance over all adaptive algorithms with tree-structured side information of a given maximum order in a computationally efficient manner.

Bibliography

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396, 2003.

Alankrita Bhatt, Jiun-Ting Huang, Young-Han Kim, J. Jon Ryu, and Pinar Sen. Variations on a theme by Liu, Cuff, and Verdú: The power of posterior sampling. In *Proc. IEEE Inf. Theory Workshop*, pages 1–5. IEEE, 2018.

Thomas M Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory*, 13(1):21–27, 1967.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, New York, second edition, 2006.

Imre Csiszár and János Körner. *Information theory: coding theorems for discrete memoryless systems*. Cambridge University Press, 2011.

Imre Csiszár and Paul C Shields. Information theory and statistics: A tutorial. 2004.

John C. Duchi. Information theory and statistics. 2019. URL https://web.stanford.edu/class/stats311/lecture-notes.pdf.

Abbas El Gamal and Young-Han Kim. *Network Information Theory*. Cambridge University Press, Cambridge, 2011.

David JC MacKay, David JC Mac Kay, et al. *Information theory, inference and learning algorithms*. Cambridge University Press, 2003.

Marc Mezard and Andrea Montanari. *Information, physics, and computation*. Oxford University Press, 2009.

T. Moon, S. Min, B. Lee, and S. Yoon. Neural universal discrete denoiser. In *Proc. Adv. Neural Info. Proc. Syst.*, pages 4772–4780, 2016. URL http://papers.nips.cc/paper/6497-neural-universal-discrete-denoiser.pdf.

Francesco Orabona and Dávid Pál. Coin betting and parameter-free online learning. In *Proc. Adv. Neural Info. Proc. Syst.*, volume 29. Curran Associates, Inc., 2016.

Yury Polyanskiy and Yihong Wu. Lecture notes on information theory. *Lecture Notes for MIT (6.441), UIUC (ECE 563), and Yale (STAT 664).*, 2014.

Jorma Rissanen and Glen G Langdon. Arithmetic coding. *IBM J. Res. Dev.*, 23(2):149–162, 1979.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10(5):1299–1319, 1998.

Claude Elwood Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3):379–423, 1948.

Larry Wasserman. Low assumptions, high dimensions. *Ration., Mark. Morals*, 2(11): 201–209, 2011.

T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu, and M. J. Weinberger. Universal discrete denoising: known channel. *IEEE Trans. Inf. Theory*, 51(1):5–28, Jan 2005. ISSN 0018-9448. doi: 10.1109/TIT.2004.839518. URL http://ieeexplore.ieee.org/stampPDF/getPDF.jsp?arnumber=1377489{&}isnumber=30067.

Frans MJ Willems, Yuri M Shtarkov, and Tjalling J Tjalkens. The context-tree weighting method: Basic properties. *IEEE Trans. Inf. Theory*, 41(3):653–664, 1995.

Aaron Wyner. The common information of two dependent random variables. *IEEE Trans. Inf. Theory*, 21(2):163–179, 1975.

Part I

Representation Learning

Chapter 1

Learning with Succinct Common Representation

1.1 Introduction

Over the last decade, we have witnessed an uncountable number of successes of deep learning, which are mostly attributed to its power of learning a good, low-dimensional representation of data (Bengio et al., 2013). The importance of representation learning has become more significant than ever in the last few years, as represented by a recent paradigm shift towards a task-agnostic learning framework (Bommasani et al., 2021) and the emerging successes in self-supervised learning (Chen et al., 2020). Despite the practical breakthroughs, however, answers to the fundamental questions like "what is a good representation?" and "how can we find such a representation?" are still unsatisfactory.

In this context, we study how to learn a good joint representation of a pair of random vectors $(\mathbf{X}, \mathbf{Y})$ with complex dependence from data, with the following structure: we wish to learn a *structured* representation $(\mathbf{Z}, \mathbf{U}, \mathbf{V})$ of $(\mathbf{X}, \mathbf{Y})$ such that $(\mathbf{Z}, \mathbf{U})$ and $(\mathbf{Z}, \mathbf{V})$ capture the information of $\mathbf{X}$ and $\mathbf{Y}$, respectively. Here, $\mathbf{Z}$ captures the commonality of $(\mathbf{X}, \mathbf{Y})$, which we thus call a *common representation* of $(\mathbf{X}, \mathbf{Y})$; $\mathbf{U}$ and $\mathbf{V}$, which we call *local representations*, correspond to the remaining information on $\mathbf{X}$ and $\mathbf{Y}$, respectively. See Fig. 1.1 for a Venn-diagram schematic of the structured representation.

This problem is often referred to as the *cross-domain disentanglement* problem (Gonzalez-Garcia et al., 2018) in the machine learning literature and has numerous applications including joint and conditional generative tasks (also known as domain transfer or image-to-image translation) and cross-domain retrieval tasks (Gonzalez-Garcia et al., 2018; Huang et al., 2018; Lee et al., 2018; Liu et al., 2018; Press et al., 2019; Yu et al., 2019; Zhu et al., 2017).

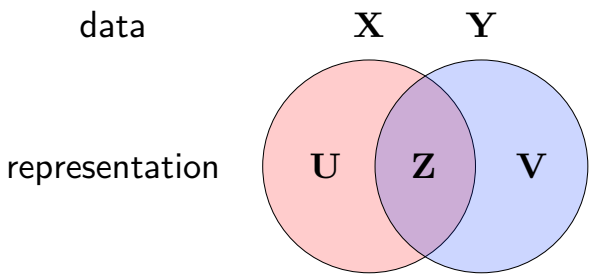


Figure 1.1. A Venn-diagram schematic for cross-domain disentanglement.

The main difficulty in learning such joint distributions with disentangled representations is that there have been no proper criterion for cross-domain disentanglement that defines an *optimal* common representation. Indeed, existing approaches which are mostly from the deep learning literature focus on developing a network architecture and/or a set of ad-hoc loss functions that promote the degree of disentanglement, not defining an optimal common representation of a joint distribution. Even a few existing information-theoretic proposals on learning a good bottleneck representation such as the famous information bottleneck principle (Tishby et al., 1999) and a recent proposal (Hwang et al., 2020) based on interactive information (McGill, 1954) do not define what an optimal representation is and what they aim to look for.

Observe that there are two bad extremes for the common representation. On one hand, we can use raw data as the common representation $\mathbf{Z} = \mathbf{X}$ or $\mathbf{Z} = \mathbf{Y}$, which contain maximal information of the pair, but may not be helpful from the view of a user who wishes to perform a downstream task based on it, as there exists a large

degree of *redundancy*. On the other hand, one may choose a common representation $\mathbf{Z}$ as a constant; albeit being the simplest, it discards essential information about the pair and is thus not useful representation. Hence, it is natural to assume that an optimal representation must lie somewhere in between, i.e., capturing the most *succinct* possible representation, as well as maintaining all the commonality of the pair.

In this paper, as a first proposal to the missing definition of an optimal common representation, we propose a new representation learning principle inspired by network information theory. To motivate our perspective, consider the following game between Alice ("encoder") and Bob ("decoder") that captures the problem setting of *conditional generation*. Given an image of a child's photo $\mathbf{X}$, Alice is asked to encode $\mathbf{X}$ and send its description $\mathbf{Z}$ to Bob who draws a portrait $\mathbf{Y}$ of how the child will grow up based on it. In this game, Bob wishes to draw nice adulthood portraits, as various as possible, given a child's photo. In this cooperative game, Alice needs to help Bob in the process by providing a *good* description $\mathbf{Z}$ of the child's photo $\mathbf{X}$. Intuitively, seeking the *most succinct description* $\mathbf{Z}$ that contains information *common in* $\mathbf{X}$ *and* $\mathbf{Y}$ may be beneficial in their guessing processes, since Alice need not describe any extra information beyond that is contained in $\mathbf{X}$ and Bob need not filter out any redundant information from $\mathbf{Z}$.

P. Cuff (2013) formulated this game of conditional generation as the *channel synthesis* problem in network information theory depicted in Fig. 1.2. Given a joint distribution $q_{\mathsf{data}}(\mathbf{x}, \mathbf{y}) = q_{\mathsf{data}}(\mathbf{x})q_{\mathsf{data}}(\mathbf{y}|\mathbf{x})$, Alice and Bob want to generate $\mathbf{Y}$ according to $q_{\mathsf{data}}(\mathbf{y}|\mathbf{x})$ based on a sample from $q_{\mathsf{data}}(\mathbf{x})$. In this problem, Alice wishes to find the most succinct description $\mathbf{Z}$ of $\mathbf{X}$ (a child's photo) such that $\mathbf{Y}$ (her adulthood portrait) can be simulated by Bob according to the desired distribution using this description and local randomness $\mathbf{V}$ (new features to draw a portrait of adults that are not contained in photos of children). The minimum description rate for such conditional generation is characterized by *Wyner's common information (CI)* (El Gamal and Kim, 2011; Wyner, 1975), which is denoted by $J(\mathbf{X}; \mathbf{Y})$ and defined as the optimal value of the following

optimization problem, which we will call *Wyner's optimization problem* hereafter:

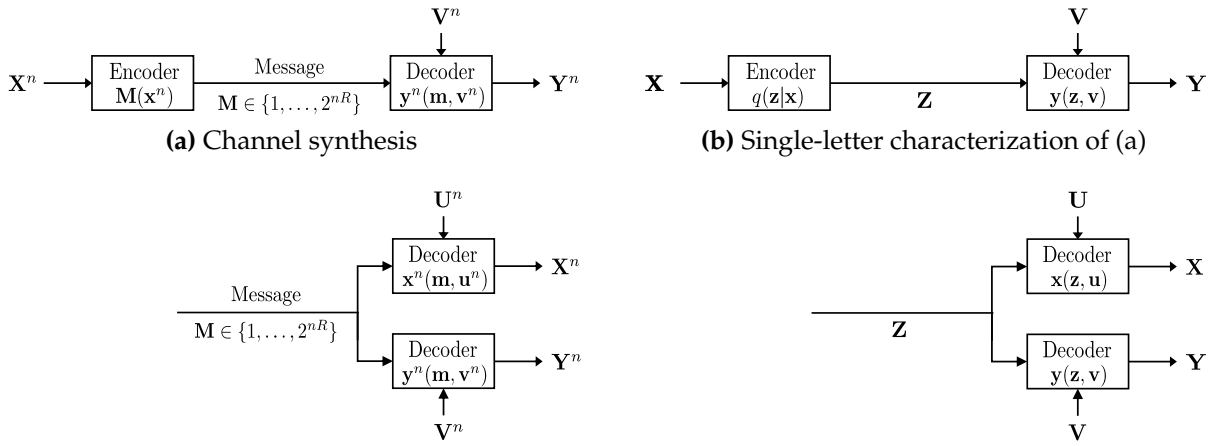
$$\begin{aligned} &\text{minimize } I(\mathbf{X}, \mathbf{Y}; \mathbf{Z}) \\ &\text{subject to } (\mathbf{X}, \mathbf{Y}, \mathbf{Z}) \sim q_{\mathsf{data}}(\mathbf{x}, \mathbf{y}) q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}) \\ &\qquad\qquad \mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y} \\ &\text{variables } q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}). \end{aligned} \tag{1.1}$$

Here, $I(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$ is the mutual information between $(\mathbf{X}, \mathbf{Y})$ and $\mathbf{Z}$, and $\mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y}$ denotes that $\mathbf{X}, \mathbf{Z}, \mathbf{Y}$ form a Markov chain, i.e., $\mathbf{X}$ is independent of $\mathbf{Y}$ given $\mathbf{Z}$ (Cover and Thomas, 2006).

Notably, the same quantity $J(\mathbf{X}; \mathbf{Y})$ arises as the fundamental limit of the *distributed simulation* of correlated sources studied originally by A. Wyner (1975) in which two distributed agents wish to simulate a target distribution $q_{\mathsf{data}}(\mathbf{x}, \mathbf{y})$ based on the least possible amount of shared common randomness; see Fig. 1.2 (c,d). As the channel synthesis problem can be viewed as an information-theoretic counterpart of conditional generation, the distributed simulation corresponds to joint generation.

Thus motivated from these observations, in this paper, we suggest to define an optimal common representation of a given joint distribution $q(\mathbf{x}, \mathbf{y})$ as the optimal solution of Wyner's optimization problem (1.1), and use the probabilistic model with a succinct common representation for both joint and conditional generation tasks.

Towards its application in generative modeling, in Section 1.2, we first propose a probabilistic model that finds a common representation, based on the single-letter characterizations of the distributed simulation and channel synthesis problems; see Fig. 1.2(b),(d). Note that the resulting probabilistic models, which we call the *variational Wyner model* as a whole, follow the Markov chain $\mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y}$, which also appear in Wyner's optimization problem (1.1) as a constraint. Here, the mutual information $I(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$ emerges as a measure of the complexity of the common representation $\mathbf{Z}$



(a) Channel synthesis

(b) Single-letter characterization of (a)

(c) Distributed simulation

(d) Single-letter characterization of (c)

Figure 1.2. Schematics for channel synthesis from $\mathbf{X}$ to $\mathbf{Y}$ (a,b), and distributed simulation of $(\mathbf{X}, \mathbf{Y})$ (c,d). (a,c) and (b,d) correspond to the operational definition and the single-letter characterization of each problem, respectively. The local randomness $\mathbf{U}$ and $\mathbf{V}$ make the decoders stochastic.

characterized by $q(\mathbf{z}|\mathbf{x}, \mathbf{y})$; see Remark 1.2.4.

Now in the learning setting, where we only have access to the joint distribution $q(\mathbf{x}, \mathbf{y})$ via its samples, we propose to train the probabilistic model based on Wyner's optimization problem (1.1). We will first derive from (1.1) a set of distribution matching losses and CI regularization losses as the main objectives. To learn with samples more effectively, we further propose auxiliary objectives such as reconstruction losses and latent matching losses. See Section 1.3.

In Section 1.4, we discuss how to train the variational Wyner model based on the proposed training objective. As an effective training trick, we specifically adopt an approximate training method using an variational density ratio estimation technique (Pu et al., 2017). With this training trick, after all, the proposed generative model can be viewed as an adversarially learned bimodal autoencoder.

Section 1.5 discusses related work on Wyner's CI from the information theory literature, existing information-theoretic approaches such as (Tishby et al., 1999) and (Hwang et al., 2020), and other bimodal generative models and cross-domain disentan-

glement approaches.

In Section 1.6, we justify this framework (the model, the training objectives, and the training method as a whole) by empirically showing that learning with its deep generative model manifestation can indeed improve an *empirical quality* in generative tasks and various downstream tasks for synthetic and real-world dataset, demonstrating that the amount of CI captured in $\mathbf{Z}$ can be controlled to improve the quality of the model. We defer the details of training schemes and network architectures to Appendix.

1.2 Probabilistic Models

In this section, we define all probabilistic model components for joint and conditional sampling tasks based on the Markov chain $\mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y}$ and the single-letter characterizations in Fig. 1.2 (b,d).

1.2.1 Joint Model

As a generative model for modeling the joint distribution $q_{\mathsf{data}}(\mathbf{x}, \mathbf{y})$, we consider the latent variable model $p_\theta(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{y}|\mathbf{z})$. Here, $\mathbf{Z} \sim p_\theta(\mathbf{z})$ signifies the common randomness fed into the *probabilistic* decoders $p_\theta(\mathbf{x}|\mathbf{z})$ and $p_\theta(\mathbf{y}|\mathbf{z})$. We parameterize the probabilistic decoders $p_\theta(\mathbf{x}|\mathbf{z})$ and $p_\theta(\mathbf{y}|\mathbf{z})$ by (deterministic) functions $\mathbf{x}_\theta(\mathbf{z}, \mathbf{u})$ and $y_\theta(\mathbf{z}, \mathbf{v})$ with independent *local randomness* $\mathbf{U} \sim p_\theta(\mathbf{u})$ and $\mathbf{V} \sim p_\theta(\mathbf{v})$, as depicted in the single letter characterization of distributed simulation (Fig. 1.2 (d)). With a slight abuse of notation, we use $\mathbf{x}_\theta(\mathbf{z}, \mathbf{u})$ for a shorthand for the degenerate distribution $\delta(\mathbf{x} - \mathbf{x}_\theta(\mathbf{z}, \mathbf{u}))$.

1.2.2 Conditional Models

To model the conditional distribution $q_{\mathsf{data}}(\mathbf{y}|\mathbf{x})$, we consider the bottleneck conditional model $q_\theta(\mathbf{z}|\mathbf{x})p_\theta(\mathbf{y}|\mathbf{z})$ that follows $\mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y}$; note that the decoder $p_\theta(\mathbf{y}|\mathbf{z})$ is shared by the joint model. The other direction for modeling $q_{\mathsf{data}}(\mathbf{x}|\mathbf{y})$ is symmetric

and thus omitted.

1.2.3 Variational Encoders

In addition to the base components introduced so far from which we can draw joint and conditional samples, we introduce three additional encoders:

- A joint encoder $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})$: it plays a key role of an anchor during training, tying the joint and conditional models.

- Local encoders $q_\phi(\mathbf{u}|\mathbf{z},\mathbf{x})$, $q_\phi(\mathbf{v}|\mathbf{z},\mathbf{y})$: these can be viewed as *style extractors* for each modality $\mathbf{x}$ and $\mathbf{y}$: if we learn a succinct common representation $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})$ (e.g., a shared concept) from $(\mathbf{x},\mathbf{y})$, then $q_\phi(\mathbf{u}|\mathbf{z},\mathbf{x})$ captures the remaining randomness $\mathbf{U}$ of $\mathbf{X}$ (e.g., texture and style).

We will call these encoders *variational* due to a technical reason to be justified when training objectives are introduced below in Section 1.3.1.These encoders can be used in training by allowing us to enforce the reconstruction consistency of the model as shown in the next section, as well as in several inference tasks such as domain translation; see Remark 1.2.2.

1.2.4 Variational Wyner Model

We call the entire model with all the components introduced above, i.e., in Sections 1.2.1–1.2.3, as the (bimodal) *variational Wyner model*. We remark that in this general framework we may learn multiple models sharing common components, or learn a single model without training the others, depending on the task at hand. For example, if one is interested in captioning an image, we may only require learning a conditional model from image to caption, without learning the joint distribution and the conditional distribution of image given caption. For the cross-domain retrieval task (see Remark 1.2.3 and Section 1.6.3), we require to learn both conditional models. When

multiple models are trained simultaneously, common components such as the joint encoder $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})$ and local encoders $q_\phi(\mathbf{u}|\mathbf{z},\mathbf{x}), q_\phi(\mathbf{v}|\mathbf{z},\mathbf{y})$ are *shared*.

Remark 1.2.1 (Conditional independence structure)**.** The components of the variational Wyner model may naturally arise considering the cross-domain disentanglement problem, and indeed similar models have been studied in the literature; see e.g., (Gonzalez-Garcia et al., 2018; Hwang et al., 2020; Wang et al., 2016). The key structural difference of our model compared to existing ones is the conditioning with the common representation $\mathbf{Z}$ in the local encoders $q_\phi(\mathbf{u}|\mathbf{z},\mathbf{x})$, $q_\phi(\mathbf{v}|\mathbf{z},\mathbf{y})$, which are designed to satisfy the conditional independence structure implied by the joint model $p_\theta(\mathbf{z})p_\theta(\mathbf{u})p_\theta(\mathbf{v})\mathbf{x}_\theta(\mathbf{z},\mathbf{u})\mathbf{y}_\theta(\mathbf{z},\mathbf{v})$, i.e.,

$$q_\phi(\mathbf{z},\mathbf{u},\mathbf{v}|\mathbf{x},\mathbf{y}) = q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})q_\phi(\mathbf{u}|\mathbf{z},\mathbf{x})q_\phi(\mathbf{v}|\mathbf{z},\mathbf{y}),$$

(see Proposition 1.B.1 in Appendix), while the existing models ignore the conditioning with $\mathbf{z}$, that is, use variational encoders of the form $q(\mathbf{u}|\mathbf{x})$ and/or $q(\mathbf{v}|\mathbf{y})$ (Gonzalez-Garcia et al., 2018; Hwang et al., 2020; Wang et al., 2016). In experiments, we empirically validate that this conditioning with $\mathbf{Z}$ indeed helps learn disentangled representations; see the cross-domain retrieval task experiment in Section 1.6.

Remark 1.2.2 (Sampling with style control)**.** Once the Wyner model is properly trained to fit $q(\mathbf{x},\mathbf{y})$, joint or conditional sampling can be done to simulate sample generation from $q(\mathbf{x},\mathbf{y})$ or $q(\mathbf{y}|\mathbf{x})$ in a straightforward manner as depicted in Fig. 1.3(a,c). The variational encoders $q_\phi(\mathbf{u}|\mathbf{z},\mathbf{x}), q_\phi(\mathbf{v}|\mathbf{z},\mathbf{y})$ can be used as a local representation extractor in sampling tasks *with style control*, such as joint stochastic reconstruction (Fig. 1.3(b)) or conditional sampling with style control (Fig. 1.3(d)); here, we elaborate the latter use case. Suppose that $(\mathbf{X},\mathbf{Y})$ is a pair of correlated images generated from the common concept but from different domains. Similar to conditional sampling (Fig. 1.3(c)), we first draw $\mathbf{Z}_j$ from $q_\phi(\mathbf{z}|\mathbf{x}_j)$. Given an image $\mathbf{y}_0$, we then extract the style information $\mathbf{V}_{0,j}$

from $q_\phi(\mathbf{v}|\mathbf{Z}_j, \mathbf{y}_0)$ (Fig. 1.3(d)). Finally, we generate $\mathbf{Y}_{0,j}$ from an image $\mathbf{x}_j$ while replacing the randomly drawn local representation $\mathbf{V} \sim p_\theta(\mathbf{v})$ with the previously extracted style $\mathbf{V}_{0,j}$, thereby the generated images $\mathbf{Y}_{0,j}$ is of the same style as the reference image $\mathbf{y}_0$. In a similar manner, we can also perform joint sampling with a fixed style given a style reference data pair $(\mathbf{x}_0, \mathbf{y}_0)$, by mixing a randomly drawn common representation $\mathbf{Z}$ from the prior $p_\theta(\mathbf{z})$ with the extracted style variables $(\mathbf{u}_0, \mathbf{v}_0)$.

Remark 1.2.3 (Cross-domain retrieval)**.** Beyond the joint and conditional generation tasks, there is another closely related task which is called the *cross-domain retrieval* (Gonzalez-Garcia et al., 2018; Hwang et al., 2020). In this task, we are given a reference set $\{\mathbf{y}_i\}$. For a query $\mathbf{x}_o$, instead of aiming to draw a fresh sample $\mathbf{y}$ from $q(\mathbf{y}|\mathbf{x}_o)$, we wish to *retrieve* relevant $\mathbf{y}$'s from $\{\mathbf{y}_i\}$. We can solve the retrieval task using a trained variational Wyner model over the common representation space, similar to (Gonzalez-Garcia et al., 2018; Hwang et al., 2020). That is, we first find and keep the common representations $\{\mathbf{z}_i\}$ of reference points $\{\mathbf{y}_i\}$ using the model encoder $q_\theta(\mathbf{z}|\mathbf{y})$. Then, given a query $\mathbf{x}_o$, we find the common representation $\mathbf{z}_o \sim q_\theta(\mathbf{z}|\mathbf{x}_o)$ to retrieve the K-nearest neighbors of $\mathbf{z}_o$ from $\{\mathbf{z}_i\}$ with respect to, say, the cosine similarity. We remark that for this task, we only require to learn the conditional models of both directions. See Section 1.6.3 for our experimental results.

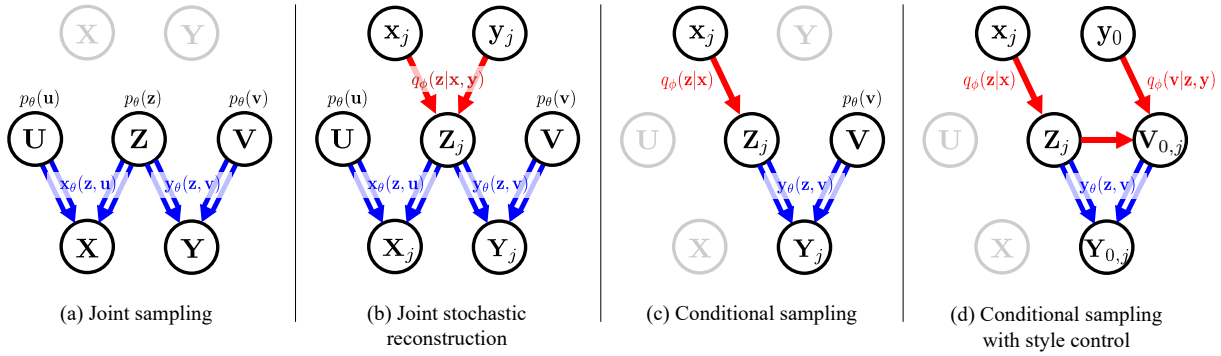


Figure 1.3. Schematics for selected sampling tasks. Double-line arrows are used to emphasize the deterministic mappings.

1.2.5 Induced Distributions

The variational Wyner model defines four different distributions over the extended set of variables $(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v})$. We explicitly write down the distributions below, as we will match the pairs of distributions to train the generative models of our interest in the next section.

The first one is the *variational distribution*, which is defined by the underlying data distribution and the variational encoders:

$$q_{\mathsf{xy}\rightarrow}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v}) := q_{\mathsf{data}}(\mathbf{x}, \mathbf{y}) q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}) q_\phi(\mathbf{u}|\mathbf{z}, \mathbf{x}) q_\phi(\mathbf{v}|\mathbf{z}, \mathbf{y}). \tag{1.2}$$

The other three distributions are the *model* distributions, which correspond to the joint and conditional generative models. The joint model induces

$$p_{\rightarrow\mathsf{xy}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v}) := p_\theta(\mathbf{z}) p_\theta(\mathbf{u}) p_\theta(\mathbf{v}) \mathbf{x}_\theta(\mathbf{z}, \mathbf{u}) \mathbf{y}_\theta(\mathbf{z}, \mathbf{v}), \tag{1.3}$$

and the conditional model that *maps* $\mathbf{x}$ to $\mathbf{y}$ induces

$$p_{\mathsf{x}\rightarrow\mathsf{y}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v}) := q_{\mathsf{data}}(\mathbf{x}) q_\theta(\mathbf{z}|\mathbf{x}) q_\phi(\mathbf{u}|\mathbf{z}, \mathbf{x}) p_\theta(\mathbf{v}) \mathbf{y}_\theta(\mathbf{z}, \mathbf{v}). \tag{1.4}$$

Symmetrically, the other direction from $\mathbf{y}$ to $\mathbf{x}$ induces

$$p_{\mathsf{y}\rightarrow\mathsf{x}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v}) := q_{\mathsf{data}}(\mathbf{y}) q_\theta(\mathbf{z}|\mathbf{y}) q_\phi(\mathbf{v}|\mathbf{z}, \mathbf{y}) p_\theta(\mathbf{u}) \mathbf{x}_\theta(\mathbf{z}, \mathbf{u}). \tag{1.5}$$

Hereafter, for each model $p_{\mathsf{model}} \in \{p_{\rightarrow\mathsf{xy}}, p_{\mathsf{x}\rightarrow\mathsf{y}}, p_{\mathsf{y}\rightarrow\mathsf{x}}\}$ and a subset of variables $\mathbf{w} \subseteq \{\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v}\}$, we use $p_{\mathsf{model}}(\mathbf{w})$ to denote the induced distributions over $\mathbf{w}$. For example, for $\mathbf{w} = \{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$, the induced distribution is

$$p_{\mathsf{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z}) := \int p_{\mathsf{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v}) \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{v}.$$

We remark that while the variational distribution $q_{\mathsf{xy}\to}$ is always consistent with the data distribution but may fail to satisfy $\mathbf{X}-\mathbf{Z}-\mathbf{Y}$, the model distributions $p_{\to\mathsf{xy}}, p_{\mathsf{x}\to\mathsf{y}}, p_{\mathsf{y}\to\mathsf{x}}$ may not be consistent with $q_{\mathsf{data}}(\mathbf{x},\mathbf{y})$ but always follow the Markov chain.

Remark 1.2.4 (Common information)**.** In (1.1), we call the mutual information term $I_{\mathsf{xy}\to}(\mathbf{X},\mathbf{Y};\mathbf{Z})$ induced by the variational distribution $q_{\mathsf{xy}\to}(\mathbf{x},\mathbf{y},\mathbf{z})$, i.e.,

$$I_{\mathsf{xy}\to}(\mathbf{X},\mathbf{Y};\mathbf{Z}) := \mathbb{E}_{q_{\mathsf{data}}(\mathbf{x},\mathbf{y})}[D_{\mathsf{KL}}(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}) \parallel q_{\mathsf{xy}\to}(\mathbf{z}))],$$

the *variational CI*. Here, $D_{\mathsf{KL}}(p \parallel q)$ denotes the Kullback–Leibler (KL) divergence between two distributions p and q. Now, for each model distribution $p_{\mathsf{model}} \in \{p_{\to\mathsf{xy}}, p_{\mathsf{x}\to\mathsf{y}}, p_{\mathsf{y}\to\mathsf{x}}\}$, which is designed to follow $\mathbf{X}-\mathbf{Z}-\mathbf{Y}$ above, we call the the corresponding mutual information between $(\mathbf{X},\mathbf{Y})$ and $\mathbf{Z}$

$$\begin{aligned} I_{\mathsf{model}} &:= I_{\mathsf{model}}(\mathbf{X},\mathbf{Y};\mathbf{Z}) \\ &:= D_{\mathsf{KL}}(p_{\mathsf{model}}(\mathbf{x},\mathbf{y},\mathbf{z}) \parallel p_{\mathsf{model}}(\mathbf{x},\mathbf{y})p_{\mathsf{model}}(\mathbf{z})) \end{aligned} \tag{1.6}$$

the *model CI* under p_{model}.

1.3 Training Objectives

In this section, we describe a set of training objectives for effectively training the proposed variational Wyner model.

1.3.1 Main Objectives

Recall Wyner's optimization problem from the introduction:

$$\begin{aligned} &\text{minimize } I_{\mathsf{xy}\to}(\mathbf{X},\mathbf{Y};\mathbf{Z}) \\ &\text{subject to } \mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y}, \end{aligned} \tag{1.7}$$

where the variable is the joint encoder $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})$. Hence, this optimization problem seeks a joint encoder $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})$ that captures the minimal common information under the Markovity constraint.

For each model distribution $p_{\mathsf{model}} \in \{p_{\to\mathsf{xy}}, p_{\mathsf{x}\to\mathsf{y}}, p_{\mathsf{y}\to\mathsf{x}}\}$, our main learning principle is to train it by seeking a succinct common representation characterized by Wyner's optimization problem, and we reformulate the optimization problem (1.7) for each model. First, since each model distribution $p_{\mathsf{model}}(\mathbf{x},\mathbf{y},\mathbf{z})$ follows the Markov chain $\mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y}$, we replace the Markovity constraint with the following model consistency

$$p_{\mathsf{model}}(\mathbf{x},\mathbf{y},\mathbf{z}) \equiv q_{\mathsf{data}}(\mathbf{x},\mathbf{y})q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}),$$

i.e., the model p_{model} is *consistent* with the target data distribution $q_{\mathsf{data}}(\mathbf{x},\mathbf{y})$ and the variational distribution $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})$. Under this model consistency, we can further replace the variational CI $I_{\mathsf{xy}\to}(\mathbf{X},\mathbf{Y};\mathbf{Z})$ with the model CI $I_{\mathsf{model}}(\mathbf{X},\mathbf{Y};\mathbf{Z})$, which is defined in (1.6). Hence, for each model p_{model}, we obtain

$$\begin{aligned} &\text{minimize } I_{\mathsf{model}}(\mathbf{X},\mathbf{Y};\mathbf{Z}) \\ &\text{subject to } p_{\mathsf{model}}(\mathbf{x},\mathbf{y},\mathbf{z}) \equiv q_{\mathsf{data}}(\mathbf{x},\mathbf{y},\mathbf{z}). \end{aligned} \tag{1.8}$$

The model consistency can be imposed by introducing a distribution matching term

$$\mathcal{D}^{\mathsf{xyz}}_{\mathsf{model}} := D(p_{\mathsf{model}}(\mathbf{x},\mathbf{y},\mathbf{z}), q_{\mathsf{data}}(\mathbf{x},\mathbf{y},\mathbf{z}))$$

and constraining it to be 0 for a choice of positive definite divergence function $D(p,q)$ such as f-divergences, Wasserstein distance, or maximum mean discrepancy (Zhao et al., 2018). Note that each choice of $D(p,q)$ requires different training methods and

training procedures. In this paper, we specifically choose the *symmetric KL divergence*

$$D_{\mathsf{sym}}(p(\mathbf{s}), q(\mathbf{s})) := D_{\mathsf{KL}}(p(\mathbf{s}) \parallel q(\mathbf{s})) + D_{\mathsf{KL}}(q(\mathbf{s}) \parallel p(\mathbf{s}))$$

which is also known as the *Jeffreys divergence* (Jeffreys, 1998). Pu et al. (2017) originally suggested its use in generative modeling as an alternative of the one-sided (reverse) KL diveregence $D_{\mathsf{KL}}(q \parallel p)$ of VAEs, since it can encourage mode-seeking and mass-covering simultaneously. Unlike the typical VAE training, however, the symmetric KL divergence necessitates an additional trick to deal with intractable density ratios. We illustrate an approximate training method in Section 1.4.

Therefore, Wyner's optimization problem (1.8) for each model can be written as

$$\begin{aligned} &\text{minimize } I_{\mathsf{model}}(\mathbf{X}, \mathbf{Y}; \mathbf{Z}) \\ &\text{subject to } \mathcal{D}^{\mathsf{xyz}}_{\mathsf{model}} = 0. \end{aligned} \tag{1.9}$$

We now show that this can be further relaxed as

$$\text{minimize } \ \mathcal{D}^{\mathsf{xyzuv}}_{\mathsf{model}} + \lambda^{\mathsf{CI}}_{\mathsf{model}} I_{\mathsf{model}}(\mathbf{X}, \mathbf{Y}; \mathbf{Z}), \tag{1.10}$$

where

$$\mathcal{D}^{\mathsf{xyzuv}}_{\mathsf{model}} := D_{\mathsf{sym}}(q_{\mathsf{xy}\rightarrow}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v}), p_{\mathsf{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v})) \tag{1.11}$$

is the divergence between the variational distribution and the model distribution over $(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v})$ and $\lambda^{\mathsf{CI}}_{\mathsf{model}} > 0$ is a hyperparameter that trades off matching distributions and seeking succinct representation.

We first relax the equality constraint $\mathcal{D}^{\mathsf{xyz}}_{\mathsf{model}} = 0$ with an inequality constraint $\mathcal{D}^{\mathsf{xyz}}_{\mathsf{model}} \leq \epsilon$ for some $\epsilon > 0$ and as in Zhao et al. (2018) to convert the problem (1.9) into

an unconstrained Lagrangian form

$$\text{minimize} \;\; \mathcal{D}^{\mathsf{xyz}}_{\mathsf{model}} + \lambda^{\mathsf{CI}}_{\mathsf{model}} I_{\mathsf{model}}(\mathbf{X}, \mathbf{Y}; \mathbf{Z}), \tag{1.12}$$

where the reciprocal of a Lagrange multiplier $\lambda^{\mathsf{CI}}_{\mathsf{model}} > 0$ controls the model CI of $p_{\mathsf{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z})$. We then introduce an additional variational relaxation step to train the style extractors $q_\phi(\mathbf{u}|\mathbf{z}, \mathbf{x})$ and $q_\phi(\mathbf{v}|\mathbf{z}, \mathbf{y})$. That is, by the monotonicity of f-divergences (see Proposition 1.B.2 in Appendix), we have

$$\begin{aligned} \mathcal{D}^{\mathsf{xyz}}_{\mathsf{model}} &:= D_{\mathsf{sym}}(q_{\mathsf{data}}(\mathbf{x}, \mathbf{y}, \mathbf{z}), p_{\mathsf{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z})) \\ &\leq D_{\mathsf{sym}}(q_{\mathsf{xy}\rightarrow}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v}), p_{\mathsf{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v})) \\ &=: \mathcal{D}^{\mathsf{xyzuv}}_{\mathsf{model}}, \end{aligned} \tag{1.13}$$

where the equality holds if and only if the composite variational encoders

$$q_{\mathsf{xy}\rightarrow}(\mathbf{z}, \mathbf{u}, \mathbf{v}|\mathbf{x}, \mathbf{y}) = q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}) q_\phi(\mathbf{u}|\mathbf{z}, \mathbf{x}) q_\phi(\mathbf{v}|\mathbf{z}, \mathbf{y})$$

match to the model posterior $p_{\mathsf{model}}(\mathbf{z}, \mathbf{u}, \mathbf{v}|\mathbf{x}, \mathbf{y})$. After all, we obtain (1.10) as the final relaxed optimization problem for each model distribution $p_{\mathsf{model}} \in \{p_{\rightarrow \mathsf{xy}}, p_{\mathsf{x}\rightarrow\mathsf{y}}, p_{\mathsf{y}\rightarrow\mathsf{x}}\}$.

Remark 1.3.1 (On the variational common information)**.** An acute reader may suggest to simply control the CI regularization by a single term $I_{\mathsf{xy}\rightarrow}(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$, the variational CI. Indeed, given that $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$ is succinctly learned based on (1.7), one could expect that the rest of the model components would be encouraged to be consistent with the succinctly learned $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$, via the distribution matching term $\mathcal{D}^{\mathsf{xyzuv}}_{\mathsf{model}}$ (1.13). We empirically found, however, that $I_{\mathsf{xy}\rightarrow}(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$ as a regularization term is not sufficiently effective to control the model CI's and leads to unstable training, compared to directly using the model CI $I_{\mathsf{model}}(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$ for regularization. Intuitively, this phenomenon may

be attributed to imperfect distribution matching, which is enforced by minimizing $\mathcal{D}^{\mathsf{xyzuv}}_{\mathsf{model}}$ in our framework, due to a lack of samples, a limited expressivity of the parametric models, an imperfect training, or their combinations.

1.3.2 Auxiliary Objectives

Note that learning a succinct common representation becomes meaningful only when a good degree of consistency between the models and data can be assured. In principle, solving the optimization problem (1.10) may suffice for training the target generative models with a succinct common representation. Since, however, such non-convex optimization problems are hard to solve in general, distribution matching may not take place to begin with. Hence, in this section, we additionally introduce a set of auxiliary objectives that can considerably improve the degree of distribution matching.

Reconstruction Losses

With the variational encoders, we can further guide the training by imposing certain *reconstruction consistency* in the model, so that the optimization of the encoders and decoders is over a restricted function space that conforms to the consistency. Note that the model trained with the reconstruction loss terms below can be viewed as a form of *autoencoders*.

For the joint model $p_{\to\mathsf{xy}}$, similar to the reconstruction in autoencoders, it is natural to desire that the decoders $\mathbf{x}_\theta(\mathbf{z},\mathbf{u}), \mathbf{y}_\theta(\mathbf{z},\mathbf{v})$ map the inferred representations $(\mathbf{z}_o,\mathbf{u}_o,\mathbf{v}_o) \sim q_{\mathsf{xy}\to}(\mathbf{z},\mathbf{u},\mathbf{v}|\mathbf{x}_o,\mathbf{y}_o)$ for a given pair $(\mathbf{x}_o,\mathbf{y}_o)$ back to $(\mathbf{x}_o,\mathbf{y}_o)$; hence, we aim to minimize the *joint data reconstruction losses* defined as

$$\mathcal{R}_{\mathsf{xy}\to\mathsf{x}} := \mathbb{E}_{q_{\mathsf{xy}\to}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v})p_\theta(\hat{\mathbf{x}}|\mathbf{z},\mathbf{u})}[d_{\mathsf{x}}(\mathbf{x},\hat{\mathbf{x}})] \tag{1.14}$$

and the symmetrically defined $\mathcal{R}_{\mathsf{xy}\to\mathsf{y}}$ for some dissimilarity functions $d_{\mathsf{x}}(\mathbf{x},\hat{\mathbf{x}})$ and $d_{\mathsf{y}}(\mathbf{y},\hat{\mathbf{y}})$.

For the conditional model $p_{\mathsf{x}\to\mathsf{y}}$, we consider the following consistency on the data space. Given a data pair $(\mathbf{x}_o, \mathbf{y}_o)$, we first draw a common representation $\mathbf{z}_o \sim q_\phi(\mathbf{z}|\mathbf{x})$ only from $\mathbf{x}_o$ and find a local representation of $\mathbf{y}_o$ conditioned on $\mathbf{z}_o$, i.e., $\mathbf{v}_o \sim q_\phi(\mathbf{v}|\mathbf{z}_o, \mathbf{y}_o)$. Then, we expect the decoder $\mathbf{y}_\theta(\mathbf{z}, \mathbf{v})$ to reconstruct $\mathbf{y}_o$ from the representation $(\mathbf{z}_o, \mathbf{v}_o)$, which leads to the definition of the *conditional reconstruction loss*

$$\mathcal{R}_{\mathsf{x}\to\mathsf{y}} := \mathbb{E}_{q_{\mathsf{data}}(\mathbf{x},\mathbf{y})q_\theta(\mathbf{z}|\mathbf{x})q_\phi(\mathbf{v}|\mathbf{z},\mathbf{y})p_\theta(\hat{\mathbf{y}}|\mathbf{z},\mathbf{v})}[d_{\mathsf{y}}(\mathbf{y}, \hat{\mathbf{y}})] \tag{1.15}$$

from $\mathbf{x}$ to $\mathbf{y}$; the other direction $\mathcal{R}_{\mathsf{y}\to\mathsf{x}}$ is symmetric.

When Learning Joint and Conditional Models Simultaneously: Common Latent Space Matching Losses

When we wish to train a single model that can perform every direction of inference (i.e., joint and both ways of conditional generation), it is important to enforce the induced aggregated posteriors of the model encoders $q_\theta(\mathbf{z}|\mathbf{x})$ and $q_\theta(\mathbf{z}|\mathbf{y})$ in the conditional models to be consistent with the prior distribution $p_\theta(\mathbf{z})$, so that they can share the common latent space over $\mathbf{z}$. That is, we wish to match the aggregated posterior $p_{\mathsf{model}}(\mathbf{z})$ to the prior $p_\theta(\mathbf{z})$ for $p_{\mathsf{model}} \in \{p_{\mathsf{x}\to\mathsf{y}}, p_{\mathsf{y}\to\mathsf{x}}\}$. Since it is only enforced indirectly by the distribution matching losses $\mathcal{D}^{\mathsf{xyzuv}}_{\mathsf{model}}$, we further introduce the *latent matching objectives*

$$\mathcal{M}_{\mathsf{model}} := D_{\mathsf{sym}}(p_{\mathsf{model}}(\mathbf{z}), p_\theta(\mathbf{z})). \tag{1.16}$$

We remark that this consistency is also enforced by minimizing the distribution matching objective $\mathcal{D}^{\mathsf{xyzuv}}_{\mathsf{model}}$. More precisely, by the monotonicity of f-divergences (Proposition 1.B.2), we have $\mathcal{M}_{\mathsf{model}} \leq \mathcal{D}^{\mathsf{xyzuv}}_{\mathsf{model}}$. Hence, additionally introducing the term $\mathcal{M}_{\mathsf{model}}$ should be understood as further encouraging the consistency between the model aggregated posterior $p_{\mathsf{model}}(\mathbf{z})$ and the prior, so that it improves the quality of downstream

tasks. We empirically found this objective especially helpful when learning with a paired data from two different domains such as an image-caption dataset, where the hardness of learning the target modalities is imbalanced.

When Learning Both Conditional Models Simultaneously: Cross Matching Loss and Marginal Reconstruction Losses

In some applications such as the cross-domain retrieval task (Remark 1.2.3), we are only interested in learning conditional models of both directions such that they share the same common latent space, without learning the joint distribution. In this case, we find that matching the two conditional distributions, i.e., minimizing

$$\mathcal{D}^{\mathsf{xyzuv}}_{\mathsf{x}\leftrightarrow\mathsf{y}} := D_{\mathsf{sym}}(p_{\mathsf{x}\to\mathsf{y}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v}), p_{\mathsf{y}\to\mathsf{x}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v})), \tag{1.17}$$

can improve the quality of representation.

Another auxiliary term we find helpful in this scenario is the *marginal* reconstruction losses

$$\mathcal{R}_{\mathsf{y}\to\mathsf{y}} := \mathbb{E}_{q_{\mathsf{data}}(\mathbf{y})q_\theta(\mathbf{z}|\mathbf{y})q_\phi(\mathbf{v}|\mathbf{z},\mathbf{y})p_\theta(\hat{\mathbf{y}}|\mathbf{z},\mathbf{v})}[d_{\mathsf{y}}(\mathbf{y},\hat{\mathbf{y}})] \tag{1.18}$$

and the symmetrically defined $\mathcal{R}_{\mathsf{x}\to\mathsf{x}}$, which naturally arise when matching two conditional models $p_{\mathsf{x}\to\mathsf{y}}$ and $p_{\mathsf{y}\to\mathsf{x}}$.

1.3.3 The Final Objective

All the losses introduced so far are summarized in Table 1.1. In experiments, we optimized a weighted combination of those objectives, and tune the weights as hyperparameters based on the *hardness* of learning the corresponding modalities.

Table 1.1. Summary of the objectives for training the variational Wyner model. The objectives in the square brackets $[\cdot]$ with tilde notation are the corresponding discriminator objectives; see Section 1.4.1. For the sake of easy reference, we indicate the definition for each objective term. The shaded objectives are the main objectives introduced in Section 1.3.1, which are derived from the relaxed Wyner optimization problem (1.10). The rest are auxiliary objectives defined in Section 1.3.2.

Type (key)	Distribution matching	CI regularization	Reconstruction	Latent matching
Joint ($\rightarrow \mathbf{xy}$)	$\mathcal{D}^{\mathsf{xyzuv}}_{\rightarrow \mathsf{xy}}$ (1.11) $[\tilde{\mathcal{D}}^{\mathsf{xyzuv}}_{\rightarrow \mathsf{xy}}$ (1.21)]	$I_{\rightarrow \mathsf{xy}}$ (1.6) $[\tilde{I}_{\rightarrow \mathsf{xy}}$ (1.23)]	$\mathcal{R}_{\mathsf{xy}\rightarrow \mathsf{x}}, \mathcal{R}_{\mathsf{xy}\rightarrow \mathsf{y}}$ (1.14)	-
Cond. ($\mathbf{x} \rightarrow \mathbf{y}$)	$\mathcal{D}^{\mathsf{xyzuv}}_{\mathsf{x}\rightarrow \mathsf{y}}$ (1.11) $[\tilde{\mathcal{D}}^{\mathsf{xyzuv}}_{\mathsf{x}\rightarrow \mathsf{y}}$ (cf. 1.21)]	$I_{\mathsf{x}\rightarrow \mathsf{y}}$ (1.6) $[\tilde{I}_{\mathsf{x}\rightarrow \mathsf{y}}$ (1.23)]	$\mathcal{R}_{\mathsf{x}\rightarrow \mathsf{y}}$ (1.15)	$\mathcal{M}_{\mathsf{x}\rightarrow \mathsf{y}}$ (1.16) $[\tilde{\mathcal{M}}_{\mathsf{x}\rightarrow \mathsf{y}}$ (1.22)]
Cond. ($\mathbf{x} \rightarrow \mathbf{y}$)	$\mathcal{D}^{\mathsf{xyzuv}}_{\mathsf{y}\rightarrow \mathsf{x}}$ (1.11) $[\tilde{\mathcal{D}}^{\mathsf{xyzuv}}_{\mathsf{y}\rightarrow \mathsf{x}}$ (cf. 1.21)]	$I_{\mathsf{y}\rightarrow \mathsf{x}}$ (1.6) $[\tilde{I}_{\mathsf{y}\rightarrow \mathsf{x}}$ (1.23)]	$\mathcal{R}_{\mathsf{y}\rightarrow \mathsf{x}}$ (1.15)	$\mathcal{M}_{\mathsf{y}\rightarrow \mathsf{x}}$ (1.16) $[\tilde{\mathcal{M}}_{\mathsf{y}\rightarrow \mathsf{x}}$ (1.22)]
Cond. ($\mathbf{x} \leftrightarrow \mathbf{y}$)	$\mathcal{D}^{\mathsf{xyzuv}}_{\mathsf{x}\leftrightarrow \mathsf{y}}$ (1.17) $[\tilde{\mathcal{D}}^{\mathsf{xyzuv}}_{\mathsf{x}\leftrightarrow \mathsf{y}}$ (cf. 1.21)]	-	$\mathcal{R}_{\mathsf{x}\rightarrow \mathsf{x}}, \mathcal{R}_{\mathsf{y}\rightarrow \mathsf{y}}$ (1.18)	-

1.4 Training Method

As alluded to earlier, we assume implicit generative models with deterministic decoders, the densities of the model distributions are not computable. Hence, minimizing the objective functions introduced above requires a GAN-like adversarial technique. In particular, we adopt a technique proposed by Pu et al. (2017), which we call the *variational density ratio estimation*. After all, the proposed training scheme can be viewed as an adversarial learning method of the variational Wyner model. We also illustrate some tricks that were empirically effective for training in our experiments.

1.4.1 Training with Variational Density Ratio Estimation

Note that all the objective terms proposed above are in the form of either $D_{\mathsf{sym}}(p(\mathbf{s}), q(\mathbf{s}))$ for distribution matching and $D_{\mathsf{KL}}(p(\mathbf{s}) \,\|\, q(\mathbf{s}))$ for mutual information, except the reconstruction losses. To training with these divergence terms, we approximate the divergence terms by estimating the density ratio $p(\mathbf{s})/q(\mathbf{s})$ via an adversarial technique based on a variational characterization of the Jensen–Shannon divergence; namely, we use the optimal solution $r(\mathbf{s})$ of the following maximization problem

$$D_{\mathsf{JS}}(p(\mathbf{s}), q(\mathbf{s})) = \max_{r(\mathbf{s})} \psi_{\mathsf{JS}}(r(\mathbf{s}); p(\mathbf{s}), q(\mathbf{s})), \tag{1.19}$$

where we define

$$\psi_{\mathsf{JS}}(r(\mathbf{s}); p(\mathbf{s}), q(\mathbf{s})) := \mathbb{E}_{p(\mathbf{s})}[\log \sigma(\log r(\mathbf{s}))] + \mathbb{E}_{q(\mathbf{s})}[\log \sigma(-\log r(\mathbf{s}))], \tag{1.20}$$

to estimate the density ratio $p(\mathbf{s})/q(\mathbf{s})$, since the maximum of (1.19) is attained if and only if $r^*(\mathbf{s}) \equiv p(\mathbf{s})/q(\mathbf{s})$. Here, $\sigma(x) = 1/(1 + e^{-x})$ denotes the sigmoid function. Note that this is equivalent to the discriminator objective of the original generative adversarial networks (GANs) (Goodfellow et al., 2014), where $D(\mathbf{s}) := \sigma(\log r(\mathbf{s})) \in [0, 1]$ is called

the *discriminator*; in this work we view $r(\mathbf{s}) = \exp(\sigma^{-1}(D(\mathbf{s}))) \in (0, \infty)$ as a density ratio estimator but also call a discriminator, slightly abusing the terminology. While, in principle, the variational characterization of any f-divergence by Nguyen et al. (2010) may be used to train a density estimator in a similar spirit of f-GANs (Nowozin et al., 2016), we empirically observed that other choices of f-divergences such as one-sided KL divergences and χ^2-divergences result in unstable training (data not shown).

As in a standard GAN training procedure, we alternate between training the variational Wyner model components and training the discriminators batch-by-batch, freezing one while training the other. When training the variational Wyner model, we freeze the density ratio estimators and estimate $D_{\mathsf{sym}}(p(\mathbf{s}), q(\mathbf{s}))$ by plugging in the approximate ratio $r(\mathbf{s})$ assuming that $r(\mathbf{s}) \approx p(\mathbf{s})/q(\mathbf{s})$, i.e.,

$$D_{\mathsf{sym}}(p(\mathbf{s}), q(\mathbf{s})) \approx \mathbb{E}_{p(\mathbf{s})}[\log r(\mathbf{s})] - \mathbb{E}_{q(\mathbf{s})}[\log r(\mathbf{s})].$$

Hence, for each distribution matching objective

$$\mathcal{D}^{\mathsf{xyzuv}}_{\mathsf{key}} \text{ for } \mathsf{key} \in \{\to \mathsf{xy}, \mathsf{x} \to \mathsf{y}, \mathsf{y} \to \mathsf{x}, \mathsf{x} \leftrightarrow \mathsf{y}\},$$

we introduce a corresponding density ratio estimator $r_{\mathsf{key}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v})$ and optimize it by the discriminator objective $\tilde{\mathcal{D}}^{\mathsf{xyzuv}}_{\mathsf{key}}$, where, e.g.,

$$\tilde{\mathcal{D}}^{\mathsf{xyzuv}}_{\to \mathsf{xy}} := \psi_{\mathsf{JS}}(r_{\to \mathsf{xy}}; q_{\mathsf{xy}\to}, p_{\to \mathsf{xy}}). \tag{1.21}$$

For a latent matching objective $\mathcal{M}_{\mathsf{model}}$, we train a discriminator $r^{\mathsf{latent}}_{\mathsf{model}}(\mathbf{z})$ by maximizing

$$\tilde{\mathcal{M}}_{\mathsf{model}} := \psi_{\mathsf{JS}}(r^{\mathsf{latent}}_{\mathsf{model}}(\mathbf{z}); p_{\mathsf{model}}(\mathbf{z}), p_\theta(\mathbf{z})), \tag{1.22}$$

for each $\mathsf{model} \in \{\mathsf{x} \to \mathsf{y}, \mathsf{y} \to \mathsf{x}\}$. The mutual information $I_{\mathsf{model}}(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$ term can be

handled by the same technique, i.e., training a discriminator $r^{\mathsf{CI}}_{\mathsf{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z})$ by maximizing

$$\tilde{I}_{\mathsf{model}} := \psi_{\mathsf{JS}}(r^{\mathsf{CI}}_{\mathsf{model}}; p_{\mathsf{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z}), p_{\mathsf{model}}(\mathbf{x}, \mathbf{y}) p_{\mathsf{model}}(\mathbf{z})), \tag{1.23}$$

so that $r^{\mathsf{CI}}_{\mathsf{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z}) \approx p_{\mathsf{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z}))/p_{\mathsf{model}}(\mathbf{x}, \mathbf{y}) p_{\mathsf{model}}(\mathbf{z})$, and approximate the mutual information by the same plug-in approach:

$$\begin{aligned} I_{\mathsf{model}}(\mathbf{X}, \mathbf{Y}; \mathbf{Z}) &= \mathbb{E}_{p_{\mathsf{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z})} \left[\log \frac{p_{\mathsf{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z})}{p_{\mathsf{model}}(\mathbf{x}, \mathbf{y}) p_{\mathsf{model}}(\mathbf{z})} \right] \\ &\approx \mathbb{E}_{p_{\mathsf{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z})} [\log r^{\mathsf{CI}}_{\mathsf{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z})]. \end{aligned}$$

In the minibatch training of density ratio estimators for CI estimation, in order to to sample from a product distribution $p_{\mathsf{model}}(\mathbf{x}, \mathbf{y}) p_{\mathsf{model}}(\mathbf{z})$, we first draw $(\mathbf{x}, \mathbf{y}, \mathbf{z}) \sim p_{\mathsf{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z})$ from the joint distribution and then simply permute $\mathbf{z}$ over the batch dimension as a proxy to independent sampling.

1.4.2 The Final Discriminator Objective

In experiments, to train the discriminators, we simply added the corresponding objective terms without additional weights.

1.4.3 Additional Tricks for Training

To make the training scheme more computationally efficient and stable, we made several important design choices, including (1) a shared joint data feature map among discriminators, (2) deterministic parameterization of encoders, and (3) the instance noise trick (Sønderby et al., 2017).

Shared Feature Map in Discriminators

In principle, for each pair of distributions whose density ratio is required to be estimated, we need a density ratio estimator $r_{\mathsf{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v})$ for distribution matching or $r^{\mathsf{CI}}_{\mathsf{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z})$ for CI regularization or $r^{\mathsf{latent}}_{\mathsf{model}}(\mathbf{z})$ for latent matching. To reduce the size of the discriminator network, in our implementation we use a single joint feature map $f(\mathbf{x}, \mathbf{y})$ which maps the pair $(\mathbf{x}, \mathbf{y})$ to a feature vector, and every density ratio estimator that takes $(\mathbf{x}, \mathbf{y})$ as an argument is of the form either $r_{\mathsf{model}}(f(\mathbf{x}, \mathbf{y}), \mathbf{z}, \mathbf{u}, \mathbf{v})$ or $r^{\mathsf{CI}}_{\mathsf{model}}(f(\mathbf{x}, \mathbf{y}), \mathbf{z})$.

Deterministic Encoders

Following the standard practice, we approximate the proposed objectives in Table 1.1 which are expectations over model distributions by a Monte Carlo approximation and plug-in it to a gradient-based optimization algorithm. Note, however, that sampling distributions and taking gradients with respect to a parameter of the sampling distribution may cause *biased* gradient estimates, since, in general,

$$\nabla_\theta \mathbb{E}_{p_\theta(\mathbf{s})}[f_\theta(\mathbf{s})] \neq \mathbb{E}_{p_\theta(\mathbf{s})}[\nabla_\theta f_\theta(\mathbf{s})].$$

A possible detour is to deploy diagonal Gaussian encoders used in VAEs to invoke the reparametrization trick (Kingma and Welling, 2014). In this work, even simpler, we parameterize all the encoders (i.e., variational encoders $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$, $q_\phi(\mathbf{u}|\mathbf{z}, \mathbf{x})$, $q_\phi(\mathbf{v}|\mathbf{z}, \mathbf{y})$ and model encoders $q_\theta(\mathbf{z}|\mathbf{x})$, $q_\theta(\mathbf{z}|\mathbf{y})$) by *deterministic* mappings, which can be viewed as the limiting version of the reparameterization trick with vanishing variances.

Instance Noise Trick

Since our encoders and decoders are all deterministic, they define *degenerate* model distributions, on which the divergences and mutual information terms may not

be properly defined due to disjoint support of paired distributions. This is a well-known issue of implicit generative models, and we adopt the well-known *instance noise trick* of Sønderby et al. (2017) from the GAN literature. Namely, we add small Gaussian noise to all inputs of the discriminators (density estimators), which is equivalent to replacing any distribution $p(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v})$ of our consideration with that convolved with a Gaussian kernel. We empirically observed that this trick is effective, but a proper tuning of the level of Gaussian noise is crucial in stabilizing the training procedure while not blurring out the distributions of our interest.

1.5 Related Work

1.5.1 On Wyner's CI and Related Measures

Wyner's CI was first studied by Wyner (1975) to investigate the problem of distributed simulation of two discrete random sources and distributed compression in the so-called Gray–Wyner network (Gray and Wyner, 1974). Witsenhausen (1976) established a lower bound on this quantity and studied its computability. Cuff (2013) established the role of Wyner's CI in its conditional counterpart, i.e., the channel synthesis problem. Later, Xu et al. (2016) studied the quantity for a pair of continuous random variables, and provided its operational justification in the distributed *lossy* compression setting.

Recently, Wyner's common information has received a lot of attention in the information theory literature, especially in the context of its application for extracting correlation between dependent variables. A recent line of theoretical work includes a local characterization of Wyner's CI (Huang et al., 2020) and an alternative, Wyner's-CI-based procedure for canonical correlation analysis (Sula and Gastpar, 2021).

There exist several other related dependence measures for a pair of random variables (X, Y) in information theory. The mutual information $I(X;Y)$ has significant

roles and concrete operational meanings in information theory and statistics, including source and channel coding problems and hypothesis testing problems; see, e.g., (Cover and Thomas, 2006). We remark, however, that Wyner's common information $J(X;Y)$ is in general different from the more famous quantity of mutual information $I(X;Y)$. It is easy to prove that $0 \leq I(X;Y) \leq J(X;Y)$. In general, the inequalities can be strict, but when X and Y are independent, $I(X;Y) = J(X;Y) = 0$.

The Gác–Körner–Witsenhausen common information $K(X;Y)$ (Gács and Körner, 1973) is defined to be the maximum number of common bits per symbol that can be independently extracted from X and Y. While it has several applications in secret key generation, it is known that the notion is rather restrictive in the sense that $K(X;Y)$ becomes positive only for limited cases (Gács and Körner, 1973; Witsenhausen, 1975). Moreover, this quantity can be defined only for discrete random variables.

The Hirschfeld–Gebelein–Rényi (HGR) maximal correlation (Gebelein, 1941; Hirschfeld, 1935; Rényi, 1959) is a nonlinear generalization of Pearson correlation coefficient. The HGR maximal correlation is originally defined for a pair of scalar random variables, but it was generalized to quantify a measure of dependence between high-dimensional random vectors; see (Michaeli et al., 2016). There also exists a line of work on the maximal correlation and its applications in machine learning from an information-theoretic view; see a recent paper by Huang et al. (2019) for an overview on the recent theoretical breakthroughs.

For a more broad treatment on this subject, we refer an interested reader to a recent monograph by Yu et al. (2022).

1.5.2 Existing Information-Theoretic Approaches

In this section, we provide an in-depth discussion on two information theoretic approaches (Hwang et al., 2020; Tishby et al., 1999), elaborating the philosophical differences compared to our approach.

Information Bottleneck Principle

The information bottleneck (IB) principle (or method) (Tishby et al., 1999) is a widely known information theoretic approach in representation learning. This approach is usually considered for *discriminative tasks*, i.e., when the target variable $\mathbf{Y}$ is a function of $\mathbf{X}$ and/or even discrete. Motivated by lossy compression, the IB principle proposes to find a *compressed* representation $\mathbf{Z}$ from the input variable $\mathbf{X}$ (i.e., $q_\theta(\mathbf{z}|\mathbf{x})$) while maximizing the relevance of $\mathbf{Z}$ in *predicting* the target variable $\mathbf{Y}$ as the minimizer of the optimization problem

$$\underset{q_\theta(\mathbf{z}|\mathbf{x})}{\text{minimize}} \; I(\mathbf{X};\mathbf{Z}) - \beta I(\mathbf{Y};\mathbf{Z}),$$

where $(\mathbf{X},\mathbf{Y},\mathbf{Z}) \sim q_{\mathsf{data}}(\mathbf{x},\mathbf{y})q_\theta(\mathbf{z}|\mathbf{x})$ and $\beta > 0$.

Indeed, the IB principle and the proposed framework are suitable to discriminative tasks and generative tasks, respectively, while they fail to define a good representation structure in the other respective cases. First, consider a discriminative task such as classification, where typically there is a near functional relationship $\mathbf{Y} \approx f(\mathbf{X})$ between $\mathbf{X}$ and $\mathbf{Y}$. The proposed framework principle does not posit an interesting structure in this case, since the trivial choice $\mathbf{Z} = \mathbf{Y}$ makes $\mathbf{X}$ and $\mathbf{Y}$ independent and achieves the minimum $I(\mathbf{X},\mathbf{Y};\mathbf{Z}) = H(\mathbf{Y})$, whereas the IB principle defines a series of representations of different levels of compression controlled by β. Secondly, for a generative task where the pair $(\mathbf{X},\mathbf{Y})$ has many-to-many relationship, guessing $\mathbf{Y}$ based on $\mathbf{Z}$ as a representation of $\mathbf{X}$, the symmetric Markov assumption $\mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y}$ of our approach is more appropriate than $\mathbf{Z} \leftrightarrow \mathbf{X} \leftrightarrow \mathbf{Y}$ of IB; crucially, under the Markov chain $\mathbf{Z} \leftrightarrow \mathbf{X} \leftrightarrow \mathbf{Y}$, $\mathbf{Y}$ is not conditionally independent of $\mathbf{X}$ given $\mathbf{Z}$ in general. We summarize the differences in Table 1.2.

In Appendix 1.A, we discuss a connection from the notion of minimal sufficient statistics (Lehmann and Scheffé, 1950) to Wyner's optimization problem and the IB principle.

Table 1.2. The variational Wyner model vs. the IB principle (Tishby et al., 1999).

	The variational Wyner model	**The IB principle**
Motivating problem	channel synthesis, distributed simulation	lossy compression, minimal sufficient statistics
Probabilistic model	$\mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y}$	$\mathbf{Z} \leftrightarrow \mathbf{X} \leftrightarrow \mathbf{Y}$
Direction of inference	bidirectional	unidirectional
Measure of succinctness	$I(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$	$I(\mathbf{X}; \mathbf{Z})$
Measure of fit/relevance	$D(p, q)$	$I(\mathbf{Y}; \mathbf{Z})$
Optimal quantity	$J(\mathbf{X}; \mathbf{Y})$	N/A

We finally remark that Alemi et al. (2017) proposed to train a neural network classifier with a variational relaxation of the IB objective to seek a robust representation and a few variations of this work were proposed to find an invariant factors of a target $\mathbf{X}$ given an attribute $\mathbf{Y}$ (Gao et al., 2019; Song et al., 2019).

Interactive Information Maximization

Recently, Hwang et al. (2020) proposed a new information-theoretic regularization principle to tackle the cross-domain disentanglement problem. To seek a disentangled representation $(\mathbf{Z}, \mathbf{U}, \mathbf{V})$ under the variational distribution

$$q'_{\mathsf{xy}\rightarrow}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v}) := q_{\mathsf{data}}(\mathbf{x}, \mathbf{y}) q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}) q_\phi(\mathbf{u}|\mathbf{x}) q_\phi(\mathbf{v}|\mathbf{y}),$$

they propose to maximize the *interactive information among* $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ to enforce $\mathbf{Z}$ to capture a commonality of $(\mathbf{X}, \mathbf{Y})$, while minimizing $I(\mathbf{Z}; \mathbf{U})$ and $I(\mathbf{Z}; \mathbf{V})$ to enforce the representations $(\mathbf{Z}, \mathbf{U}, \mathbf{V})$ to be independent. Here, the interactive information among $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ is defined as

$$I(\mathbf{X}; \mathbf{Y}; \mathbf{Z}) := I(\mathbf{X}; \mathbf{Z}) - I(\mathbf{X}; \mathbf{Z}|\mathbf{Y})$$

and it is symmetric in $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ (McGill, 1954). After all, they proposed to minimize a variational upper bound of a weighted combination of a distribution matching term

$$D_{\mathsf{KL}}\big(q'_{\mathsf{xy}\rightarrow}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v}) \parallel p_{\rightarrow\mathsf{xy}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v})\big)$$

and $I(\mathbf{Z}; \mathbf{U}) + I(\mathbf{Z}; \mathbf{V}) - 2I(\mathbf{X}; \mathbf{Y}; \mathbf{Z})$. We remark that since $I(\mathbf{Z}; \mathbf{U})$ and $I(\mathbf{Z}; \mathbf{V})$ terms should become 0 when the paired distributions exactly match, the essential driving force of cross-domain disentanglement in this framework is from the *maximization* of the interactive information. In a stark contrast, we propose to *minimize* the information captured in $\mathbf{Z}$. Further, this interactive-information maximization framework does not define an optimality criterion for a good representation as alluded to earlier, and the choice of weights in the objective term $I(\mathbf{Z}; \mathbf{U}) + I(\mathbf{Z}; \mathbf{V}) - 2I(\mathbf{X}; \mathbf{Y}; \mathbf{Z})$ is rather ad-hoc, based on a computational aspect of the final objective.

1.5.3 Other Cross-Domain Disentanglement Models and Bimodal Generative Models

One of the most closely related work in the deep learning literature is the cross-domain disentanglement networks proposed by Gonzalez-Garcia et al. (2018). Similar to the variational Wyner model, their model also aims to decompose a joint representation into *shared* (common) and *exclusive* (local) representations explicitly. The crucial difference is that, in (Gonzalez-Garcia et al., 2018), one of the key components for disentanglement is the use of a *gradient reversal layer* (Ganin and Lempitsky, 2015), while the variational Wyner model forces to learn succinct information using the CI regularization terms, towards learning the optimally succinct representation characterized by (1.1).

The Wyner model can be viewed as a generalization of the probabilistic model (i.e., encoder and decoder) assumed in two existing joint variational autoencoders (VAEs)— JVAE (Vedantam et al., 2018) and JMVAE (Suzuki et al., 2016)—as alluded in

the previous paragraph. These models implement a similar idea of performing joint and conditional generation tasks via a symmetric Markov chain $\mathbf{X} \leftrightarrow \mathbf{W} \leftrightarrow \mathbf{Y}$, where $\mathbf{W}$ is the *joint representation* of $(\mathbf{X}, \mathbf{Y})$. In other words, these models can be derived by removing the local variables $\mathbf{U}$ and $\mathbf{V}$ in the variational Wyner model.

The same decoder structure of the variational Wyner model with the "shared" ($\mathbf{Z}$) and the "private" ($\mathbf{U}, \mathbf{V}$) latent variables has been also studied in the context of multi-view learning (Damianou et al., 2012; Ek et al., 2008; Salzmann et al., 2010; Shon et al., 2006) mostly based on a linear analysis such as canonical correlation analysis (CCA). More recently, variational CCA-private (VCCA-private) (Wang et al., 2016) was proposed to learn the decoder model with variational encoders $q_\theta(\mathbf{z}|\mathbf{x})$, $q_\phi(\mathbf{u}|\mathbf{x})$, and $q_\phi(\mathbf{v}|\mathbf{y})$, with the encoder model $q_\phi(\mathbf{z}, \mathbf{u}, \mathbf{v}|\mathbf{x}, \mathbf{y}) = q_\theta(\mathbf{z}|\mathbf{x})q_\phi(\mathbf{u}|\mathbf{x})q_\phi(\mathbf{v}|\mathbf{y})$ to directly capture the conditional model from $\mathbf{X}$ to $\mathbf{Z}$ to $\mathbf{Y}$. On the other hand, the variational Wyner model relies on the conditional independence structure $q_\phi(\mathbf{z}, \mathbf{u}, \mathbf{v}|\mathbf{x}, \mathbf{y}) = q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})q_\phi(\mathbf{u}|\mathbf{z}, \mathbf{x})q_\phi(\mathbf{v}|\mathbf{z}, \mathbf{y})$, which is naturally induced by the decoder model; see Remark 1.2.1. We argue that this choice of encoder model in our variational Wyner model may capture better semantic meaning of the local (private) random variables $\mathbf{U}$ and $\mathbf{V}$, thereby leading a better generative performance; see, e.g., Fig. 1.4.

Conditional VAE (CVAE) (Sohn et al., 2015) directly models the conditional distribution $q_{\mathsf{data}}(\mathbf{y}|\mathbf{x})$, obtained by simply conditioning every component in the vanilla VAE for $q_{\mathsf{data}}(\mathbf{y})$ with the conditioning variable $\mathbf{X}$. If $\mathbf{Y}$ is an image and $\mathbf{X}$ is an attribute of the image, a latent representation $\mathbf{V}$ in CVAE needs to capture the redundant information of $\mathbf{Y}$, which is not contained in $\mathbf{X}$, i.e., style information of $\mathbf{Y}$ given $\mathbf{X}$. The variational Wyner model can be viewed as a combination of two CVAEs with $\mathbf{Z}$ as a common conditioning variable, being capable of bidirectional sampling in its nature. Yet, if $\mathbf{X}$ is high-dimensional, the conditional models like CVAE in general tend to overfit the input data of $\mathbf{X}$ (Dutordoir et al., 2018). To address this problem, a subsequent related work, bottleneck conditional density estimation (BCDE) (Shu et al., 2017), proposed to learn

joint and conditional VAE models simultaneously by softly tying the parameters of the two models for regularization. We note that the variational Wyner model naturally addresses such problem by using a unified single probabilistic model for both joint and conditional distribution learning, finding a succinct common representation $\mathbf{Z}$ for regularization.

1.6 Experiments

We empirically demonstrate the power of the proposed approach with synthetic and real-world datasets. We parameterized all model components by deep neural networks. Details of the experiments such as training schemes, network architectures, and evaluation metrics are deferred to Appendix 1.C. We performed most of the experiments over the Triton Shared Computing Cluster (San Diego Supercomputer Center, 2022). The code is available online [1].

1.6.1 MNIST–SVHN Add-One Dataset

To show the effect of information decomposition in our model, we first considered a synthetic image-image pair dataset constructed from MNIST (LeCun, 1998) and SVHN (Netzer et al., 2011) datasets, similar to Shi et al. (2019). Here, we randomly picked an MNIST image $\mathbf{X}_i$ of label $\ell_i \in \{0, \dots, 9\}$ and paired with four randomly picked SVHN images of label $(\ell_i + 1) \mod 10$; we call the resulting dataset the MNIST–SVHN add-one dataset. Note that the images are paired only through their labels, and clearly the common information structure we seek is the underlying label of a pair.

We trained all the joint and conditional models simultaneously, with the objectives

$$\mathcal{D}^{\mathsf{xyzuv}}_{\to \mathsf{xy}} + \mathcal{D}^{\mathsf{xyzuv}}_{\mathsf{x} \to \mathsf{y}} + \mathcal{D}^{\mathsf{xyzuv}}_{\mathsf{y} \to \mathsf{x}}$$

[1] https://github.com/jongharyu/wyner-model

$$+ \lambda^{\mathsf{CI}}(I_{\to \mathsf{xy}} + I_{\mathsf{x}\to\mathsf{y}} + I_{\mathsf{y}\to\mathsf{x}})$$
$$+ \mathcal{R}_{\mathsf{xy}\to\mathsf{x}} + \mathcal{R}_{\mathsf{xy}\to\mathsf{y}} + \mathcal{R}_{\mathsf{x}\to\mathsf{y}} + \mathcal{R}_{\mathsf{y}\to\mathsf{x}}$$

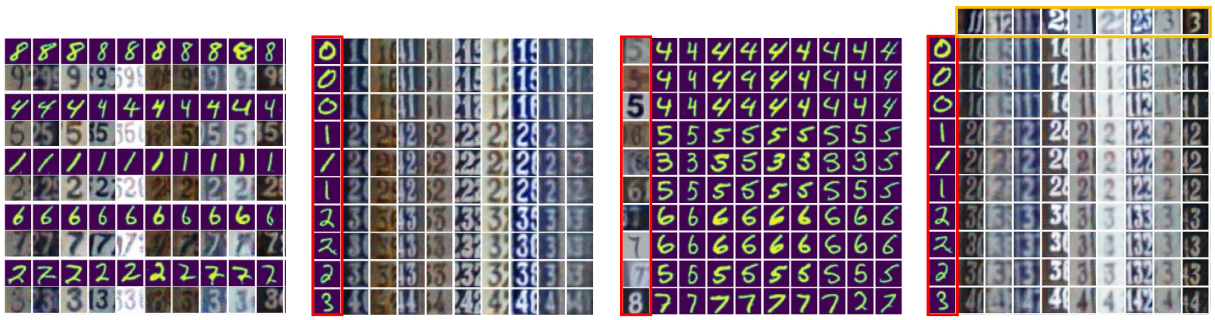
for training the variational Wyner model and

$$\tilde{\mathcal{D}}^{\mathsf{xyzuv}}_{\to\mathsf{xy}} + \tilde{\mathcal{D}}^{\mathsf{xyzuv}}_{\mathsf{x}\to\mathsf{y}} + \tilde{\mathcal{D}}^{\mathsf{xyzuv}}_{\mathsf{y}\to\mathsf{x}} + \tilde{I}_{\to\mathsf{xy}} + \tilde{I}_{\mathsf{x}\to\mathsf{y}} + \tilde{I}_{\mathsf{y}\to\mathsf{x}}$$

for training the discriminator. We tried four different CI regularization weight $\lambda^{\mathsf{CI}} \in \{0, 0.1, 0.2, 0.5, 1\}$ to demonstrate the effect of the regularization for 25 epochs and the averaged ℓ_1-distance over dimensions was used for the reconstruction loss functions. The dimension of the latent space $(\mathbf{Z}, \mathbf{U}, \mathbf{V})$ was $(16, 16, 16)$.

In Fig. 1.4, we present a few joint and conditional samples generated from the trained model with $\lambda^{\mathsf{CI}} = 0.5$ at the end of training. In the figure, each row shares the same $\mathbf{z}$, and each column shares the same $\mathbf{u}$ and/or $\mathbf{v}$. In particular, the top row of the last panel (d) shows the reference samples whose style are transferred downward along each column. The samples clearly indicate that the learned model successfully disentangles the common and local representations. For example, in Fig. 1.4(b), in the first three rows, regardless of the specifics of the input MNIST images independent to their label 0, the generated samples coherently present the correct label 1 as well as sharing the same style fixed along each column. Fig. 1.4(d) illustrates that using the local variational encoder $q_\phi(\mathbf{u}|\mathbf{z}, \mathbf{x})$, we can generate conditional samples given a fixed style extracted from a reference image; recall Remark 1.2.2.

In order to numerically examine the effect of CI regularization in the model, we computed two metrics (1) custom Frechet distance (FD) scores and (2) classification accuracy of generated samples. For the Frechet distance scores, unlike the original proposal by Heusel et al. (2017) using the Inception network, we separately trained two fully convolutional autoencoders with MNIST and SVHN datasets, respectively,



(a) →(MNIST,SVHN) (b) MNIST→SVHN (c) SVHN→MNIST (d) MNIST→SVHN with style transfer

Figure 1.4. MNIST–SVHN add-one samples from the variational Wyner model. In (b)-(d), the images in the red boxes are inputs to the conditional models. In (d), the yellow box highlights the style reference images.

and used the bottleneck features to compute the first and second order statistics for computing the Frechet distance. The digit classification errors were computed by pretrained classifiers for MNIST and SVHN. We refer the reader to the details for computing these metrics to Appendix. While achieving lower values under both metrics are ideal, note that there exists a natural trade-off between them. For example, a joint generative model that only generates good images of the digit pairs (1,2) could achieve zero error in the classification error, but may suffer a large FD score. On the other extreme, as a model tends to generate more diverse samples with different styles, it may be more prone to suffer a large error in the digit consistency.

The results are summarized in Fig. 1.5. As shown in the figure, in general, increasing λ^{CI} improves the quality of generated samples in terms of the smaller FD scores and improved the (estimated) digit accuracy. Note that the effect of the CI regularization is clear in →(MNIST, SVHN) and MNIST→SVHN, while it is not clear in SVHN→MNIST. As an explanation to this phenomenon in SVHN→MNIST, we remark that the learned model even with $\lambda^{\mathsf{ci}}_{\mathsf{y}\to\mathsf{x}} = 0$ achieved a decent degree of disentanglement, which can be justified indirectly via the generated samples as shown in Fig. 1.4(c). In general, for conditional models, we observed in our experiments that the CI regularization becomes more effective for the direction from a simpler modality (e.g., MNIST) to a more complex

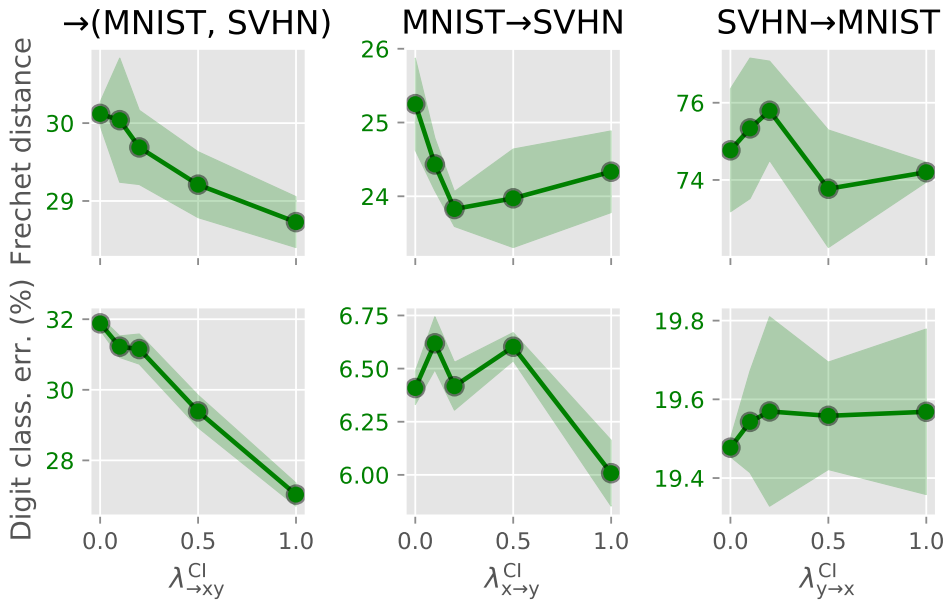


Figure 1.5. A summary of numerical evaluations for MNIST–SVHN add-one dataset. We ran five experiments with different random seeds and report the average scores. The shaded areas indicate the standard deviations.

one (e.g., SVHN).

1.6.2 CUB Image-Caption Dataset

We further demonstrate that the proposed model can even learn a complex real-world image-caption dataset, following the same setting of (Shi et al., 2019). We used the Caltech-UCSD Birds (CUB) dataset (Wah et al., 2011) that consists of 11,788 photos of birds, each of which is paired with 10 captions. To simplify the learning task, we translate the images into 2048-dimensional ResNet-101 features (He et al., 2016); to reconstruct a real image from feature, we retrieved the nearest neighbor in the feature space with respect to the Euclidean distance.

For this dataset, we found that learning the image modality $(\mathbf{x})$ from captions $(\mathbf{y})$ is harder than the other way around, and thus puts larger weights on learning the image modality. In particular, we used the following objective function with imbalanced

weights

$$\begin{aligned}
&\mathcal{D}^{\mathsf{xyzuv}}_{\to\mathsf{xy}} + \mathcal{D}^{\mathsf{xyzuv}}_{\mathsf{x}\to\mathsf{y}} + \mathcal{D}^{\mathsf{xyzuv}}_{\mathsf{y}\to\mathsf{x}} \\
&+ 0.5(I_{\to\mathsf{xy}} + I_{\mathsf{y}\to\mathsf{x}}) \\
&+ 256(\mathcal{R}_{\mathsf{xy}\to\mathsf{x}} + \mathcal{R}_{\mathsf{y}\to\mathsf{x}} + \mathcal{M}_{\mathsf{y}\to\mathsf{x}}) \\
&+ 8(\mathcal{R}_{\mathsf{xy}\to\mathsf{y}} + \mathcal{R}_{\mathsf{x}\to\mathsf{y}} + \mathcal{M}_{\mathsf{x}\to\mathsf{y}})
\end{aligned}$$

for training the variational Wyner model and

$$\tilde{\mathcal{D}}^{\mathsf{xyzuv}}_{\to\mathsf{xy}} + \tilde{\mathcal{D}}^{\mathsf{xyzuv}}_{\mathsf{x}\to\mathsf{y}} + \tilde{\mathcal{D}}^{\mathsf{xyzuv}}_{\mathsf{y}\to\mathsf{x}} + \tilde{I}_{\to\mathsf{xy}} + \tilde{I}_{\mathsf{y}\to\mathsf{x}}$$

for training the discriminator. Note that we did not use the CI regularization term $I_{\mathsf{x}\to\mathsf{y}}$ (from image to caption). We used the ℓ_2^2-distance for $\mathbf{x}$ (image-feature space) and $(\mathbf{z}, \mathbf{u}, \mathbf{v})$ (latent space), and the categorical cross-entropy loss for $\mathbf{y}$ (sentence space), all averaged over dimensions, for the reconstruction loss functions. The dimension of the latent space $(\mathbf{Z}, \mathbf{U}, \mathbf{V})$ was $(16, 8, 8)$.

As a numerical evaluation, we computed the correlation scores of jointly and conditionally generated samples with respect to the training samples, via a canonical correlation analysis, following (Massiceti et al., 2018; Shi et al., 2019); we refer an interested reader to Appendix for details. After 20 epochs of training, we attained correlation scores **(0.303, 0.327, 0.318)** for joint, conditional generation ($\mathbf{x} \to \mathbf{y}$ and $\mathbf{y} \to \mathbf{x}$), respectively. Note that the correlation scores computed with the test dataset is 0.273 and the reported scores attained by Shi et al. (2019) was (0.263, 0.104, 0.135).

We further report that without the CI regularization terms $0.5(I_{\to\mathsf{xy}} + I_{\mathsf{y}\to\mathsf{x}})$ the model fails to disentangle the representations (data not shown). We present some examples of generated samples in Fig. 1.6, which show that the trained variational Wyner model indeed generates a variety of samples of high coherence.

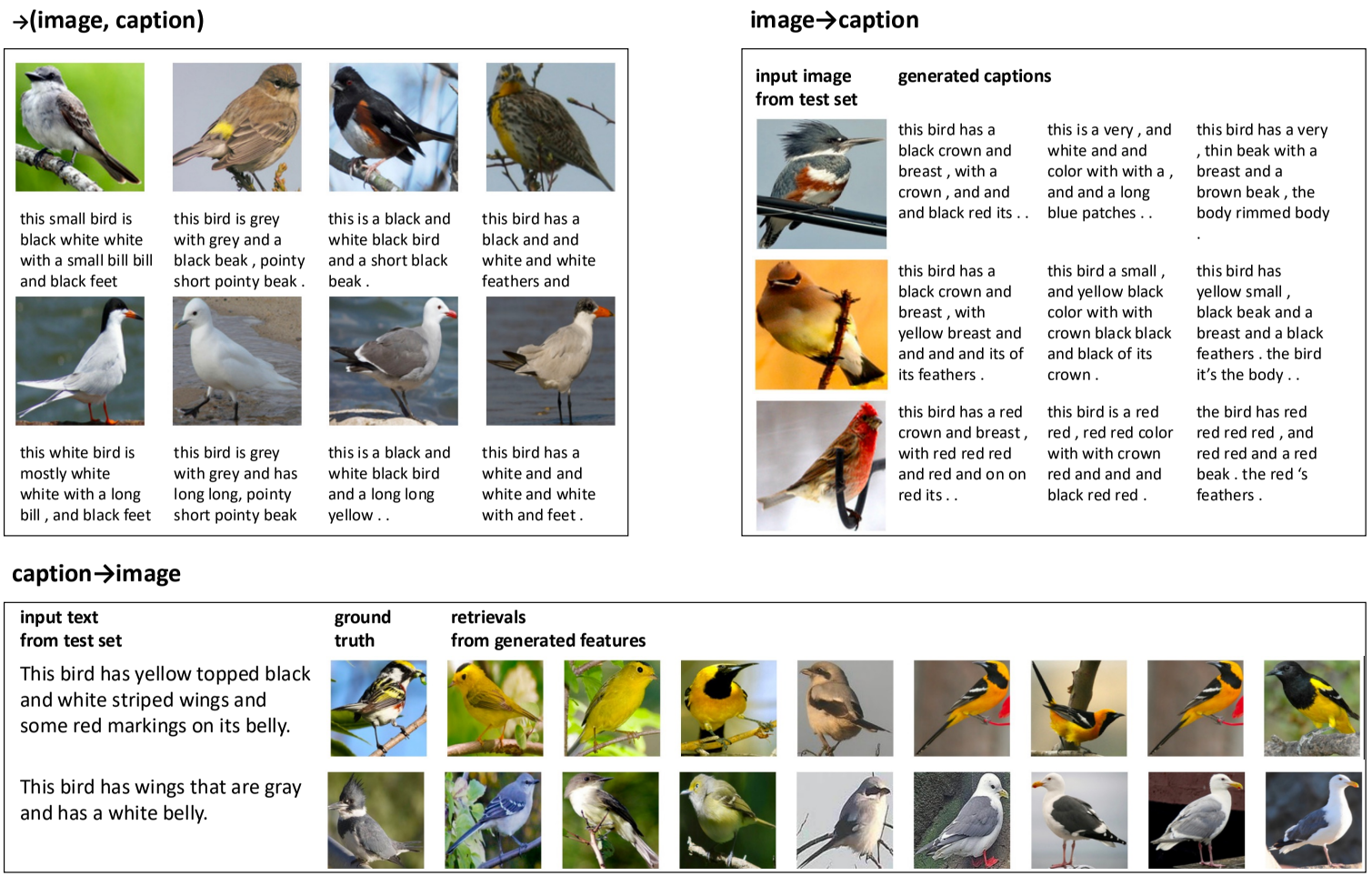


Figure 1.6. Samples from the variational Wyner model trained with CUB Image-Caption dataset. Note that we generated image features and the shown images are retrieved based on the nearest features from the test data.

1.6.3 Zero-Shot Sketch Based Image Retrieval

Lastly, to demonstrate the utility of learned representations beyond generative modeling, we consider the *zero-shot sketch based image retrieval* (ZS-SBIR) task proposed by Yelamarthi et al. (2018), where the goal is to construct a good retrieval model that retrieves relevant photos from a sketch, with a training set of no overlapping classes with a test set.

For this experiment, we borrowed the the same setting from Hwang et al. (2020). We trained and evaluated our model with the Sketchy Extended dataset (Liu et al., 2017; Sangkloy et al., 2016), which consists of total 75,479 sketches ($\mathbf{X}$) and 73,002 photos ($\mathbf{Y}$) from 125 different classes. During training, we constructed a random pair of a photo and a sketch from a same class. For training and evaluation, a pretrained VGG16 network (Simonyan and Zisserman, 2015) was used to extract features of the images. We used the pretrained VGG network and train-test splits for evaluation from the codebase[2] of Dutta and Akata (2019). After training, we performed the retrieval task by the procedure illustrated in Remark 1.2.3.

We trained our model only with conditional model components, as we only need to learn good model encoders $q_\theta(\mathbf{z}|\mathbf{x})$ and $q_\theta(\mathbf{z}|\mathbf{y})$. Specifically, we used the objectives

$$\begin{aligned}
&\mathcal{D}^{\mathsf{xyzuv}}_{\mathsf{x}\to\mathsf{y}} + \mathcal{D}^{\mathsf{xyzuv}}_{\mathsf{y}\to\mathsf{x}} + \mathcal{D}^{\mathsf{xyzuv}}_{\mathsf{x}\leftrightarrow\mathsf{y}} \\
&+ \lambda^{\mathsf{CI}}(I_{\mathsf{x}\to\mathsf{y}} + I_{\mathsf{y}\to\mathsf{x}}) \\
&+ \lambda^{\mathsf{rec}}(\mathcal{R}_{\mathsf{x}\to\mathsf{y}} + \mathcal{R}_{\mathsf{y}\to\mathsf{x}} + \mathcal{R}_{\mathsf{x}\to\mathsf{x}} + \mathcal{R}_{\mathsf{y}\to\mathsf{y}})
\end{aligned}$$

for training the variational Wyner model and

$$\tilde{\mathcal{D}}^{\mathsf{xyzuv}}_{\mathsf{x}\to\mathsf{y}} + \tilde{\mathcal{D}}^{\mathsf{xyzuv}}_{\mathsf{y}\to\mathsf{x}} + \tilde{I}_{\mathsf{x}\to\mathsf{y}} + \tilde{I}_{\mathsf{y}\to\mathsf{x}}$$

[2]https://github.com/AnjanDutta/sem-pcyc

for training the discriminator. We used the ℓ_2^2-distance averaged over dimensions for the reconstruction loss functions. The dimension of the latent space $(\mathbf{Z}, \mathbf{U}, \mathbf{V})$ was $(64, 64, 64)$.

As a quantitative evaluation, we computed the Precision@100 (P@100) and mean average precision (mAP) scores on the test split; see Table 1.3.

Table 1.3. Evaluation of the ZS-SBIR task with the Sketchy Extended dataset.

Models	P@100	mAP
LCALE Lin et al. (2020)	0.583	0.476
IIAE Hwang et al. (2020)	0.659	0.573
Variational Wyner	**0.703**	**0.629**

The reported scores for the adversarially learned Wyner model was obtained with $\lambda^{\mathsf{CI}} = 0.1$ and $\lambda^{\mathsf{rec}} = 8$. We outperform the scores reported by Hwang et al. (2020), who already demonstrated that their scores significantly improved upon the existing work tailored to extra information; for example, LCALE Lin et al. (2020) incorporated word embedding during training. The improvement corroborates the power of our approach in learning disentangled representations.[3] For an ablation study, we trained our model with degenerate local encoders $q_\phi(\mathbf{u}|\mathbf{x})$ and $q_\phi(\mathbf{u}|\mathbf{y})$, i.e., without conditioning with $\mathbf{z}$, and achieved suboptimal scores (0.670,0.591); it justifies the design of our local encoders $q_\phi(\mathbf{u}|\mathbf{z}, \mathbf{x})$ and $q_\phi(\mathbf{v}|\mathbf{z}, \mathbf{y})$.

Some examples of retrieved photos are shown in Fig. 1.7. Note that even the falsely retrieved photos share visual similarity with the query sketches.

1.7 Concluding Remarks

Cuff's channel synthesis and Wyner's distributed simulation provide an information-theoretic characterization of the simplest probabilistic structure that connects one

[3] To make a fair comparison as possible, the latent dimension and network architecture of the variational Wyner model part were also chosen almost identical to the one used in (Hwang et al., 2020).

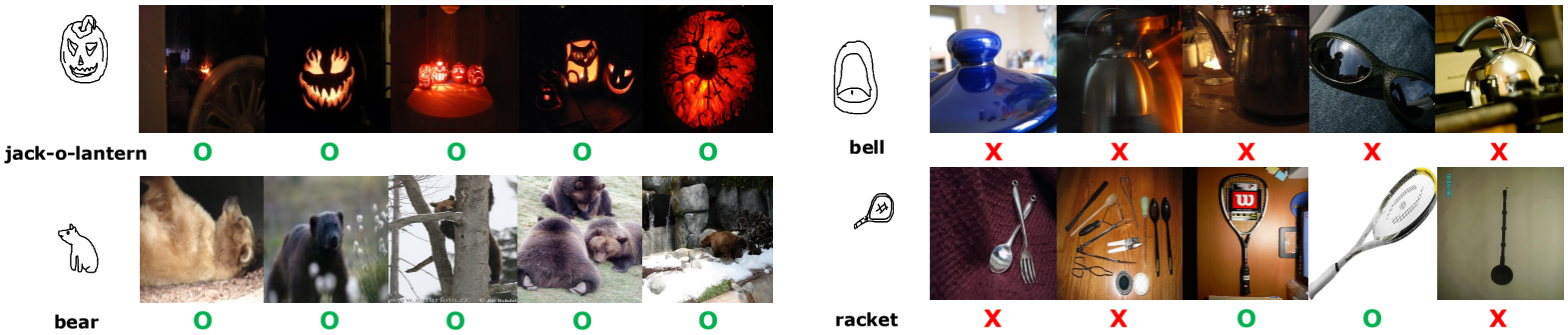


Figure 1.7. A few examples of retrieved samples from the Sketchy Extended dataset. For each query sketch, the top-5 retrieved images are shown, where the top-1 is in the leftmost. The O/X's indicate whether the retrievals belong to the same class of the query.

random object to another. The proposed variational Wyner model finds this succinct structure in a disciplined manner, and provides a theoretically sound alternative to the information bottleneck principle (Tishby et al., 1999). As alluded to earlier in Section 1.5, our approach is the first to define an optimal common representation and learn a generative model towards the optimality. The experimental results demonstrated the potential of our approach as a new way of learning joint and conditional generation tasks with optimal representation learning that can be further developed and refined for more complex dataset such as auditory, text, or a pair of those.

Albeit its potentially wide applicability, we remark that the proposed model and the accompanied training method may suffer slow training and memory inefficiency, as each divergence and mutual information term requires a separate density ratio estimator. While we reduce the number of parameters by sharing a joint feature map in the discriminator, it might be crucial to devise a more efficient way to implement the proposed framework with less parameters.

We conclude with future directions. First, we assumed fully paired data throughout the paper. In practice, however, paired data are limited and we have a plenty of unpaired data. Investigating on how to incorporate such unpaired data in the current learning framework and studying the role and effect of common information regularization in the *semi-supervised* setting may be a fruitful direction, which will make the developed framework applicable in a much richer context. Second, in this paper, we motivated the use of Wyner's common information via a heuristic justification from the resemblance between the generative tasks and the information theory problems. Hence, it would be also interesting to formally establish an operational meaning of the common information $I(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$ when learning distributions from samples. For example, can we develop a theory that relates the common information $I(\mathbf{X}, \mathbf{Y}; \mathbf{Z})$ with its "generalization error" for a generative model as in Xu and Raginsky (2017)?

Appendix

1.A From Minimal Sufficient Statistics to the Information Bottleneck Principle and Wyner's Optimization Problem

In this section, starting from the notion of minimal sufficient statistics Lehmann and Scheffé (1950) from the statistics literature, we derive the information bottleneck (IB) principle and Wyner's optimization problem from its relaxation, respectively. We hope that this discussion highlights similarities and dissimilarities between the IB principle and Wyner's optimization problem.

1.A.1 Minimal Sufficient Statistics

Consider a pair of random variables $(\mathbf{X}, \mathbf{Y})$. A function $\mathbf{Z} = \mathbf{z}(\mathbf{X})$ of $\mathbf{X}$ is said to be a *minimal sufficient statistic* of $\mathbf{X}$ for $\mathbf{Y}$ if (1) (sufficiency) $\mathbf{Z}$ is a sufficient statistic of $\mathbf{X}$ for $\mathbf{Y}$, i.e., $\mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y}$, and (2) (minimality) $\mathbf{Z}$ is a function of any other sufficient statistics. It can be easily shown that $\mathbf{Z} = \mathbf{z}(\mathbf{X})$ is a minimal sufficient statistic if and only if it is an optimal solution of the following optimization problem

$$\begin{aligned} \text{minimize} \quad & I(\mathbf{X}; \mathbf{Z}) \\ \text{subject to} \quad & \mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y} \\ \text{variable} \quad & \mathbf{Z} = \mathbf{z}(\mathbf{X}). \end{aligned} \tag{1.24}$$

Here, the optimization is over all possible functions $\mathbf{z}(\cdot)$ over $\mathbf{x}$.

1.A.2 The IB Principle

First, note that the Markovity constraint $\mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y}$ in (1.24) can be relaxed as $I(\mathbf{Y};\mathbf{Z}) \geq I(\mathbf{X};\mathbf{Y})$ by the data processing inequality. Second, optimization over functions $\mathbf{z}(\mathbf{x})$ can be relaxed by considering optimization over a probabilistic mapping $q(\mathbf{z}|\mathbf{x})$, as it subsumes deterministic functions. Finally, introducing a Lagrangian multiplier $\beta > 0$ to get rid of the inequality constraint on $I(\mathbf{Y};\mathbf{Z})$, we obtain a relaxed version of (1.24) in the unconstrained form

$$\begin{aligned} \text{minimize} \quad & I(\mathbf{X};\mathbf{Z}) - \beta I(\mathbf{Y};\mathbf{Z}) \\ \text{variable} \quad & q(\mathbf{z}|\mathbf{x}). \end{aligned} \tag{1.25}$$

Note that this is the optimization problem that characterizes the IB principle. A similar argument can be found in (Shamir et al., 2010).

1.A.3 Wyner's Optimization Problem

As done above, we first relax optimization over $\mathbf{z}(\mathbf{x})$ by optimization over $q(\mathbf{z}|\mathbf{x})$. Then, observe that optimizing over $q(\mathbf{z}|\mathbf{x})$ is equivalent to $q(\mathbf{z}|\mathbf{x},\mathbf{y})$ under an additional constraint $\mathbf{Z} \leftrightarrow \mathbf{X} \leftrightarrow \mathbf{Y}$. Hence, we can relax the optimization problem (1.24) as

$$\begin{aligned} \text{minimize} \quad & I(\mathbf{X};\mathbf{Z}) \\ \text{subject to} \quad & \mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y} \\ & \mathbf{Z} \leftrightarrow \mathbf{X} \leftrightarrow \mathbf{Y} \\ \text{variable} \quad & q(\mathbf{z}|\mathbf{x},\mathbf{y}). \end{aligned} \tag{1.26}$$

By removing the Markovity constraint $\mathbf{Z} \leftrightarrow \mathbf{X} \leftrightarrow \mathbf{Y}$, we can further relax it as

$$\begin{aligned} \text{minimize} \quad & I(\mathbf{X};\mathbf{Z}) \\ \text{subject to} \quad & \mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y} \\ \text{variable} \quad & q(\mathbf{z}|\mathbf{x},\mathbf{y}). \end{aligned} \tag{1.27}$$

We claim that this is equivalent to Wyner's optimization problem (1.1). Indeed, note that we have $I(\mathbf{X},\mathbf{Y};\mathbf{Z}) = I(\mathbf{X};\mathbf{Z}) + h(\mathbf{Y}|\mathbf{X})$ under the Markov chain $\mathbf{X} \to \mathbf{Z} \to \mathbf{Y}$, since

$$\begin{aligned} I(\mathbf{X},\mathbf{Y};\mathbf{Z}) &= I(\mathbf{X};\mathbf{Z}) + I(\mathbf{Y};\mathbf{Z}|\mathbf{X}) \\ &= I(\mathbf{X};\mathbf{Z}) + h(\mathbf{Y}|\mathbf{X}) + h(\mathbf{Y}|\mathbf{X},\mathbf{Z}) \\ &= I(\mathbf{X};\mathbf{Z}) + h(\mathbf{Y}|\mathbf{X}). \end{aligned}$$

Here, $h(\mathbf{Y}|\mathbf{X},\mathbf{Z}) = 0$ follows from the Markov chain $\mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y}$. Since $h(\mathbf{Y}|\mathbf{X})$ is constant given the target distribution $q(\mathbf{x},\mathbf{y})$, (1.27) is equivalent to (1.1).

1.A.4 Discussion

We remark that we can derive an optimization problem from (1.27) that is directly comparable to the IB principle (1.25). By applying the same argument in Appendix 1.A.2 to (1.27), we can relax $\mathbf{X} \leftrightarrow \mathbf{Z} \leftrightarrow \mathbf{Y}$ to $I(\mathbf{Y};\mathbf{Z}) \geq I(\mathbf{X};\mathbf{Y})$ and convert it into an unconstrained problem

$$\begin{aligned} \text{minimize} \quad & I(\mathbf{X};\mathbf{Z}) - \beta I(\mathbf{Y};\mathbf{Z}) \\ \text{variable} \quad & q(\mathbf{z}|\mathbf{x},\mathbf{y}). \end{aligned} \tag{1.28}$$

Interestingly, this has a close resemblance to the IB principle 1.25. In particular, (1.28) can be viewed as a relaxation of 1.25, since optimizing over $q(\mathbf{z}|\mathbf{x},\mathbf{y})$ subsumes optimizing

over $q(\mathbf{z}|\mathbf{x})$.

1.B Deferred technical statements

The following justifies the form of local variational encoders $q_\phi(\mathbf{u}|\mathbf{z},\mathbf{x})$ and $q_\phi(\mathbf{v}|\mathbf{z},\mathbf{y})$, as noted in Remark 1.2.1.

Proposition 1.B.1. *If* $(\mathbf{X},\mathbf{Y},\mathbf{Z},\mathbf{U},\mathbf{V}) \sim p(\mathbf{z})p(\mathbf{u})p(\mathbf{v})p(\mathbf{x}|\mathbf{z},\mathbf{u})p(\mathbf{y}|\mathbf{z},\mathbf{v})$*, then*

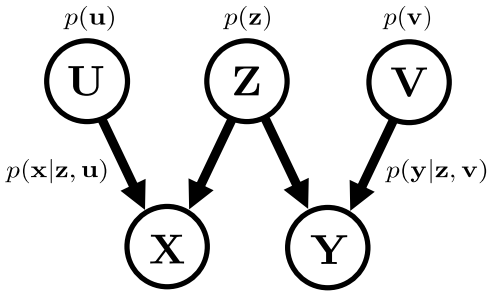
$$p(\mathbf{u},\mathbf{v}|\mathbf{z},\mathbf{x},\mathbf{y}) = p(\mathbf{u}|\mathbf{z},\mathbf{x})p(\mathbf{v}|\mathbf{z},\mathbf{y}),$$

i.e., $\mathbf{U}$ *and* $\mathbf{V}$ *are conditionally independent given* $(\mathbf{Z},\mathbf{X},\mathbf{Y})$*,* $\mathbf{U}$ *and* $\mathbf{Y}$ *are conditionally independent given* $(\mathbf{Z},\mathbf{X})$*, and* $\mathbf{V}$ *and* $\mathbf{X}$ *are conditionally independent given* $(\mathbf{Z},\mathbf{Y})$.

Proof. The conditional independence of the joint distribution

$$p(\mathbf{z})p(\mathbf{u})p(\mathbf{v})p(\mathbf{x}|\mathbf{z},\mathbf{u})p(\mathbf{y}|\mathbf{z},\mathbf{v})$$

is encoded as the following directed graphical model.



The desired conditional independences now follow from checking the d-separation (see, e.g.,, (Koller and Friedman, 2009)) between the nodes. That is, $\mathbf{U}$ and $\mathbf{V}$ are d-separated by $(\mathbf{Z},\mathbf{X},\mathbf{Y})$, $\mathbf{U}$ and $\mathbf{Y}$ are d-separated by $(\mathbf{Z},\mathbf{X})$, and $\mathbf{V}$ and $\mathbf{X}$ are d-separated by $(\mathbf{Z},\mathbf{Y})$. □

Proposition 1.B.2 (Monotonicity). *For a convex function $f : \mathbb{R}_+ \to \mathbb{R}$, let $D_f(\cdot\|\cdot)$ denote the f-divergence. For any joint distributions $p_1(x, y)$ and $p_2(x, y)$ over $\mathcal{X} \times \mathcal{Y}$, we have*

$$D_f(p_1(x)\|p_2(x)) \leq D_f(p_1(x, y)\|p_2(x, y)).$$

Here, $p_i(x)$ is the marginal distribution over $\mathcal{X}$ induced by $p_i(x, y)$. In particular, the equality holds if $p_2(y|x) \equiv p_1(y|x)$ for all $x \in \mathcal{X}$. When f is strictly convex, the condition becomes also necessary for the equality.

Proof. Consider

$$\begin{aligned}
D_f(p_1(x, y)\|p_2(x, y)) &= \int p_1(x, y) f\Big(\frac{p_2(x, y)}{p_1(x, y)}\Big)\,\mathrm{d}x\,\mathrm{d}y \\
&= \int p_1(x) p_1(y|x) f\Big(\frac{p_2(x) p_2(y|x)}{p_1(x) p_1(y|x)}\Big)\,\mathrm{d}x\,\mathrm{d}y \\
&\geq \int p_1(x) f\Big(\int p_1(y|x) \frac{p_2(x) p_2(y|x)}{p_1(x) p_1(y|x)}\,\mathrm{d}y\Big)\,\mathrm{d}x \\
&= \int p_1(x) f\Big(\frac{p_2(x)}{p_1(x)}\Big)\,\mathrm{d}x = D_f(p_1(x)\|p_2(x)).
\end{aligned}$$

Here, we use the convexity of f and Jensen's inequality. The equality condition follows from that of the Jensen's inequality. □

1.C Experiment Details

1.C.1 Common Settings

We used NVIDIA TITAN X (Pascal) for our experiments. We implemented all models using PyTorch (Paszke et al., 2017). For the inference phase, we applied the exponential moving average with decay parameter 0.999. As alluded to earlier, all distributions are parameterized by deterministic models.

Notation. Let `fc(c_in, c_out)` denote a fully-connected layer with `c_in` input

units and c_out output units. Let

$$\texttt{conv2d(c,k,s,p)}$$

$$(\texttt{deconv2d(c,k,s,p)})$$

denote a two-dimensional convolutional (transposed convolutional) layer with c filters, kernel of size $\texttt{k} \times \texttt{k}$, stride $(\texttt{s}, \texttt{s})$, and zero-padding of size $(\texttt{p}, \texttt{p})$. Let

$$\texttt{conv2d(c,(k1,k2),(s1,s2),(p1,p2))}$$

$$(\texttt{deconv2d(c,(k1,k2),(s1,s2),(p1,p2))})$$

denote a two-dimensional convolutional (transposed convolutional) layer with c filters, kernel of size $\texttt{k1} \times \texttt{k2}$, stride $(\texttt{s1}, \texttt{s2})$, and zero-padding of size $(\texttt{p1}, \texttt{p2})$. We use dropout1d and dropout2d to denote 1D and 2D dropout layers, and use

$$\texttt{maxpool2d(k)}$$

to denote a 2D max pooling layer with kernel of size (k,k).

1.C.2 MNIST–SVHN Add-One

In this dataset, $\mathbf{x} =$ MNIST, $\mathbf{y} =$ SVHN. For the MNIST–SVHN add-one dataset, we constructed $50,000$ add-one pairs from the MNIST and SVHN training datasets. For testing, we similarly constructed $1,000$ paired images from MNIST and SVHN test datasets in each case. We padded zeros around the 28×28 MNIST images to make them of size 32×32. All pixel values were linearly translated to range between $[-1, 1]$.

Table 1.C.1. The neural network architecture of the symmetric decoder in the MNIST and SVHN autoencoders. We used 1 and 3 for `c_out`, respectively for MNIST and SVHN datasets. This architecture was used to evaluate the Frechet distance; see Section 1.C.2.

$\psi_{\text{feature}\to\text{image}}$
`deconv2d(128,5,2,2)-bn2d-LReLU(0.2)`
`deconv2d(64,5,2,2)-bn2d-LReLU(0.2)`
`deconv2d(32,5,2,2)-bn2d-LReLU(0.2)`
`deconv2d(c_out,5,2,2)`

Evaluation Metrics

Frechet distance

To compute the Frechet distance (FD) score for the joint and conditional distributions over (MNIST, SVHN) pairs, we implemented a customized Frechet distance based on the PyTorch implementation of the Frechet Inception distance (FID) score (Heusel et al., 2017) by Seitzer (2020)[4]. Essentially, to be better tailored to the digit images of MNIST and SVHN datasets, we replaced the Inception-v3 model with pretrained feature extractors trained from autoencoders for MNIST and SVHN, respectively. We used the network architectures defined in Tables 1.C.1 and 1.C.3 and defined autoencoders

$$\mathbf{x}_{\texttt{image}} \mapsto \psi^{\texttt{image}}_{\text{feature}\to\text{image}}((f^{\texttt{image}}_{\text{image}\to\text{feature}}(\mathbf{x}_{\texttt{image}}))$$

for $\texttt{image}\in\{\texttt{mnist},\texttt{svhn}\}$. We trained the MNIST and SVHN autoencoders for 200 and 25 epochs, respectively, with Adam optimizer with learning rate 10^{-4} and batch size 64. Note that we used the "extra" split of the SVHN dataset for training. After training, we used the encoders as feature extractors. To evaluate the joint distribution, we concatenated the two feature vectors to compute the mean and covariance for each dataset. To evaluate the conditional distribution, we computed the FD score for each digit class separately and reported the averaged values over the 10 classes. Note that

[4]https://github.com/mseitzer/pytorch-fid

Table 1.C.2. The neural network architecture of the MNIST and SVHN classifiers. Note that we used the identical architecture, and it only differs in the bottleneck dimension due to the difference in the numbers of channels.

MNIST classifier
`conv2d(10,5,1,0)-maxpool2d(2)-ReLU`
`conv2d(20,5,1,0)-dropout2d-maxpool2d(2)-ReLU`
`reshape(batch_size, 320)`
`fc(320, 50)-maxpool2d(2)-ReLU-dropout1d`
`fc(50, 10)-softmax`

SVHN classifier
`conv2d(10,5,1,0)-maxpool2d(2)-ReLU`
`conv2d(20,5,1,0)-dropout2d-maxpool2d(2)-ReLU`
`reshape(batch_size, 500)`
`fc(500, 50)-maxpool2d(2)-ReLU-dropout1d`
`fc(50, 10)-softmax`

the FD score was evaluated with respect to the test datasets of MNIST and SVHN, so that a low FD score implies that the model generates similar images to (unseen) test images.

Digit classification error

For the digit classification error reported in Fig. 1.5, we used pretrained classifiers with network architectures in Table 1.C.2. Each classifier was trained for 15 epochs with the cross entropy loss and Adam optimizer with learning rate 10^{-3} and batch size 32. To evaluate a conditional accuracy, we computed an accuracy for each class and reported their average.

Network Architectures

The neural network architectures are summarized in Tables 1.C.3–1.C.5; here, f's are used in encoders, g's are in decoders, and h's are in discriminators.

Define $f^{\texttt{joint}}_{\mathsf{aggregate}}(\mathbf{x}, \mathbf{y}) := f^{\texttt{joint}}_{\mathsf{image}\to\mathsf{feature}}(\mathbf{x}, \mathbf{y})$.

Table 1.C.3. The neural network architectures in the MNIST–SVHN encoders. The output of the image feature network $f_{\text{image}\to\text{feature}}(\mathbf{x})$ has dimension (batch_size, 1024).

$f_{\text{image}\to\text{feature}}$
conv2d(32,5,2,2)-bn2d-LReLU(0.2)
conv2d(64,5,2,2)-bn2d-LReLU(0.2)
conv2d(128,5,2,2)-bn2d-LReLU(0.2)
conv2d(256,5,2,2)-bn2d-LReLU(0.2)
flatten

$f_{\text{feature}\to\mathbf{z}}$
fc(1024,dim_z)

$f_{\mathbf{z}\to\text{latent\_feature}}$
fc(dim_z,1024)-bn1d-LReLU(0.2)

$f_{\text{features}\to\mathbf{u}}$
fc(2048,dim_local)

Table 1.C.4. The neural network architectures in the MNIST–SVHN decoders.

$g_{(\mathbf{z},\mathbf{u})\to\text{feature}}$
fc(dim_z+dim_u,8192)-bn1d-LReLU(0.2)
reshape(batch_size,512,4,4)

$g_{\text{feature}\to\text{image}}$
deconv2d(256,5,2,2)-bn2d-LReLU(0.2)
deconv2d(128,5,2,2)-bn2d-LReLU(0.2)
deconv2d(c_out,5,2,2)-bn2d-LReLU(0.2)

Table 1.C.5. The neural network architectures in the MNIST–SVHN discriminators. The output of the image feature network $h_{\text{image}\to\text{feature}}(\mathbf{x})$ has dimension (batch_size, 2048).

$h^{\text{common}}_{\text{image}\to\text{feature}}$
conv2d(64,5,2,2)-LReLU(0.2)
conv2d(128,5,2,2)-bn2d-LReLU(0.2)
conv2d(256,5,2,2)-bn2d-LReLU(0.2)
conv2d(512,5,2,2)-bn2d-LReLU(0.2)
flatten

$h_{\text{latent}\to\text{feature}}$
fc(total_latent_dim, 512)-LReLU(0.2)

$h_{\text{feature}\to\text{ratio}}$
fc(1536, 512)-bn1d-LReLU(0.2)
fc(512,1)

Generator Models

- MNIST encoder / decoder
  - $\mathbf{z} \sim q(\mathbf{z}|\mathbf{x}) \equiv \mathbf{z} \leftarrow f^{\texttt{mnist}}_{\mathsf{feature}\rightarrow\mathbf{z}}(f^{\texttt{mnist}}_{\mathsf{image}\rightarrow\mathsf{feature}}(\mathbf{x}))$.
  - $\mathbf{u} \sim q(\mathbf{u}|\mathbf{z},\mathbf{x}) \equiv \mathbf{u} \leftarrow f^{\texttt{mnist}}_{\mathsf{features}\rightarrow\mathbf{u}}(f^{\texttt{mnist}}_{\mathbf{z}\rightarrow\mathsf{latent\_feature}}(\mathbf{z}),\mathbf{x})$.
  - $\mathbf{x} \sim p(\mathbf{x}|\mathbf{z},\mathbf{u}) \equiv \mathbf{x} \leftarrow g^{\texttt{mnist}}_{\mathsf{feature}\rightarrow\ \mathsf{image}}(g^{\texttt{mnist}}_{(\mathbf{z},\mathbf{u})\rightarrow\mathsf{feature}}(\mathbf{z},\mathbf{u}))$.
- SVHN encoder / decoder
  - $\mathbf{z} \sim q(\mathbf{z}|\mathbf{y}) \equiv \mathbf{z} \leftarrow f^{\texttt{svhn}}_{\mathsf{feature}\rightarrow\mathbf{z}}(f^{\texttt{svhn}}_{\mathsf{image}\rightarrow\mathsf{feature}}(\mathbf{y}))$.
  - $\mathbf{v} \sim q(\mathbf{v}|\mathbf{z},\mathbf{y}) \equiv \mathbf{v} \leftarrow f^{\texttt{svhn}}_{\mathsf{features}\rightarrow\mathbf{u}}(f^{\texttt{svhn}}_{\mathbf{z}\rightarrow\mathsf{latent\_feature}}(\mathbf{z}),\mathbf{y})$.
  - $\mathbf{y} \sim p(\mathbf{y}|\mathbf{z},\mathbf{v}) \equiv \mathbf{y} \leftarrow g^{\texttt{svhn}}_{\mathsf{feature}\rightarrow\ \mathsf{image}}(g^{\texttt{svhn}}_{(\mathbf{z},\mathbf{v})\rightarrow\mathsf{feature}}(\mathbf{z},\mathbf{v}))$.
- (MNIST, SVHN) encoder
  - $\mathbf{z} \sim q(\mathbf{z}|\mathbf{x},\mathbf{y}) \equiv \mathbf{z} \leftarrow f^{\texttt{joint}}_{\mathsf{feature}\rightarrow\mathbf{z}}(f^{\texttt{joint}}_{\mathsf{aggregate}}(\mathbf{x},\mathbf{y}))$.

Discriminator Models

As noted in Section 1.4, all the discriminators shared the same feature network, i.e., $h^{\texttt{common}}_{\mathsf{image}\rightarrow\mathsf{feature}}(\mathbf{x},\mathbf{y})$.

- Each discriminator to match distributions for $\texttt{model} \in \{\rightarrow \mathsf{xy}, \mathsf{x}\rightarrow\mathsf{y}, \mathsf{y}\rightarrow\mathsf{x}\}$ (see (1.11) and/or (1.17) for the model loss and e.g., (1.21) for the discriminator loss) has the following form:

$$r^{\texttt{model}}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u},\mathbf{v}) = h^{\texttt{model}}_{\mathsf{feature}\rightarrow\mathsf{ratio}}(h^{\texttt{common}}_{\mathsf{image}\rightarrow\mathsf{feature}}(\mathbf{x},\mathbf{y}), h^{\texttt{model}}_{\mathsf{latent}\rightarrow\mathsf{feature}}(\mathbf{z},\mathbf{u},\mathbf{v})).$$

- Each discriminator for common information of $\texttt{model} \in \{\rightarrow \mathsf{xy}, \mathsf{x}\rightarrow\mathsf{y}, \mathsf{y}\rightarrow\mathsf{x}\}$ (see (1.6) for the model loss and (1.23) for the discriminator loss) has the following form:

$$r^{\texttt{model,ci}}(\mathbf{x},\mathbf{y},\mathbf{z}) = h^{\texttt{model,ci}}_{\mathsf{feature}\rightarrow\mathsf{ratio}}(h^{\texttt{common}}_{\mathsf{image}\rightarrow\mathsf{feature}}(\mathbf{x},\mathbf{y}), h^{\texttt{model,ci}}_{\mathsf{latent}\rightarrow\mathsf{feature}}(\mathbf{z})).$$

- Each discriminator for the latent matching loss of $\texttt{model} \in \{\to \mathsf{xy}, \mathsf{x} \to \mathsf{y}, \mathsf{y} \to \mathsf{x}\}$ (see (1.16) for the model loss and (1.22) for the discriminator loss) has the following form:

$$r^{\texttt{model},\texttt{agg}}(\mathbf{z}) = h^{\texttt{model},\texttt{agg}}_{\mathsf{feature}\to\mathsf{ratio}}(h^{\texttt{model},\texttt{agg}}_{\mathsf{latent}\to\mathsf{feature}}(\mathbf{z})).$$

Training

We used the Adam optimizer (Kingma and Ba, 2014) with $(\beta_1, \beta_2) = (0.5, 0.999)$, learning rate 10^{-4} for the generators and 2×10^{-4} for the discriminators (Heusel et al., 2017). We trained for 25 epochs.

In the discriminators, we added a mean-zero gaussian noise of standard deviation 0.25 for each variable $\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v}$.

1.C.3 CUB Image–Caption

In this dataset, $\mathbf{x} =$ Image (pretrained ResNet features), $\mathbf{y} =$ Caption.

Evaluation Metrics

We followed the same evaluation procedure of (Massiceti et al., 2018) and (Shi et al., 2019, Section 4.3), which is to perform the canonical correlation analysis (CCA) (Hotelling, 1936) and report the correlation score with respect to feature vectors of image and caption. For the image dataset, recall that we already have 2048-dim. features from the pretrained ResNet-101, which were used in training. For each caption, we trained a FastText model (Bojanowski et al., 2017) using all sentences in the training dataset to convert each word to a 300-dim. vector; an embedding of a caption was obtained by taking the average embedding of each word in the sentence. After extracting features, we performed the CCA with projection dimension 40. For the jointly generated samples, we used 1000 samples. For the conditionally generated samples, we used the entire test set and reported the average score.

Network Architectures

The following architectures were adopted from (Shi et al., 2019) with some adjustments. When processing the caption, the maximum sentence length was 32 and the embedding dimension was 128. The neural network architectures are summarized in Tables 1.C.6–1.C.8; here, f's are used in encoders, g's are in decoders, and h's are in discriminators. Again, we note that a network without superscript indicates that the same network architecture is used in multiple places with different (i.e., not shared) realizations.

We made the embedding layer trainable as well.

Generator Models

- Image encoder / decoder
  - $\mathbf{z} \sim q(\mathbf{z}|\mathbf{x}) \equiv \mathbf{z} \leftarrow f^{\texttt{image}}_{\text{feature}\to\mathbf{z}}(f^{\texttt{image}}_{\text{image}\to\text{feature}}(\mathbf{x}))$.
  - $\mathbf{u} \sim q(\mathbf{u}|\mathbf{z},\mathbf{x}) \equiv \mathbf{u} \leftarrow f^{\texttt{image}}_{\text{features}\to\mathbf{u}}(f^{\texttt{image}}_{\mathbf{z}\to\text{latent\_feature}}(\mathbf{z}),\mathbf{x})$.
  - $\mathbf{x} \sim p(\mathbf{x}|\mathbf{z},\mathbf{u}) \equiv \mathbf{x} \leftarrow g^{\texttt{image}}_{\text{feature}\to\text{ image}}(g^{\texttt{image}}_{(\mathbf{z},\mathbf{u})\to\text{feature}}(\mathbf{z},\mathbf{u}))$.
- Caption encoder / decoder
  - $\mathbf{z} \sim q(\mathbf{z}|\mathbf{y}) \equiv \mathbf{z} \leftarrow f^{\texttt{sent}}_{\text{feature}\to\mathbf{z}}(f^{\texttt{sent}}_{\text{sent}\to\text{feature}}(\mathbf{y}))$.
  - $\mathbf{v} \sim q(\mathbf{v}|\mathbf{z},\mathbf{y}) \equiv \mathbf{v} \leftarrow f^{\texttt{sent}}_{(\mathbf{z},\text{feature})\to\mathbf{v}}(f^{\texttt{sent}}_{\mathbf{z}\to\text{latent\_feature}}(\mathbf{z}),\mathbf{y})$.
  - $\mathbf{y} \sim p(\mathbf{y}|\mathbf{z},\mathbf{v}) \equiv \mathbf{y} \leftarrow g^{\texttt{sent}}_{\text{feature}\to\text{ image}}(g^{\texttt{sent}}_{(\mathbf{z},\mathbf{v})\to\text{feature}}(\mathbf{z},\mathbf{v}))$.
- (Image, Caption) encoder
  - $\mathbf{z} \sim q(\mathbf{z}|\mathbf{x},\mathbf{y}) \equiv \mathbf{z} \leftarrow f^{\texttt{joint}}_{\text{feature}\to\mathbf{z}}(f^{\texttt{joint}}_{\text{aggregate}}(\mathbf{x},\mathbf{y}))$.

Discriminator Models

As noted in Section 1.4, all the discriminators shared the same feature network, i.e., $h^{\texttt{common}}_{(\text{image,sent})\to\text{feature}}(\mathbf{x},\mathbf{y}) := h^{\texttt{common}}_{\text{aggregate}}(h^{\texttt{common}}_{\text{image}\to\text{feature}}(\mathbf{x}), h^{\texttt{common}}_{\text{sent}\to\text{feature}}(\mathbf{y}))$. Note that the

following definitions are almost equivalent to the discriminators for the MNIST–SVHN model except the form of the shared joint feature map.

- Each discriminator to match distributions for $\texttt{model} \in \{\to \mathsf{xy}, \mathsf{x} \to \mathsf{y}, \mathsf{y} \to \mathsf{x}\}$ (see (1.11) and/or (1.17) for the model loss and e.g., (1.21) for the discriminator loss) has the following form:

$$r^{\texttt{model}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v}) = h^{\texttt{model}}_{\mathsf{feature} \to \mathsf{ratio}}(h^{\texttt{common}}_{(\mathsf{image},\mathsf{sent}) \to \mathsf{feature}}(\mathbf{x}, \mathbf{y}), h^{\texttt{model}}_{\mathsf{latent} \to \mathsf{feature}}(\mathbf{z}, \mathbf{u}, \mathbf{v})).$$

- Each discriminator for common information of $\texttt{model} \in \{\to \mathsf{xy}, \mathsf{x} \to \mathsf{y}, \mathsf{y} \to \mathsf{x}\}$ (see (1.6) for the model loss and (1.23) for the discriminator loss) has the following form:

$$r^{\texttt{model,ci}}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = h^{\texttt{model,ci}}_{\mathsf{feature} \to \mathsf{ratio}}(h^{\texttt{common}}_{(\mathsf{image},\mathsf{sent}) \to \mathsf{feature}}(\mathbf{x}, \mathbf{y}), h^{\texttt{model,ci}}_{\mathsf{latent} \to \mathsf{feature}}(\mathbf{z})).$$

- Each discriminator for the latent matching loss of $\texttt{model} \in \{\to \mathsf{xy}, \mathsf{x} \to \mathsf{y}, \mathsf{y} \to \mathsf{x}\}$ (see (1.16) for the model loss and (1.22) for the discriminator loss) has the following form:

$$r^{\texttt{model,agg}}(\mathbf{z}) = h^{\texttt{model,agg}}_{\mathsf{feature} \to \mathsf{ratio}}(h^{\texttt{model,agg}}_{\mathsf{latent} \to \mathsf{feature}}(\mathbf{z})).$$

Training

We used the Adam optimizer (Kingma and Ba, 2014) with $(\beta_1, \beta_2) = (0.5, 0.999)$, learning rate 10^{-4} for the generators and 2×10^{-4} for the discriminators (Heusel et al., 2017). We trained for 50 epochs.

In the discriminators, we added a zero-mean Gaussian noise of standard deviation 0.25 for each latent variable $\mathbf{z}, \mathbf{u}, \mathbf{v}$. For the perturbation in the data variables $\mathbf{x}, \mathbf{y}$, we did the following, adaptive noise injection based on the standard deviations of each

Table 1.C.6. The neural network architectures in the CUB encoders.

$f_{\text{image}\to\text{feature}}$
fc(2048,1024)-bn1d-LReLU(0.2)
fc(1024,512)-bn1d-LReLU(0.2)
fc(512,256)-bn1d-LReLU(0.2)

$f^{\text{image}}_{\text{feature}\to\mathbf{z}}$
fc(256,dim_z)

$f^{\text{image}}_{\mathbf{z}\to\text{latent\_feature}}$
fc(dim_z,256)-bn1d-LReLU(0.2)

$f^{\text{image}}_{\text{features}\to\mathbf{u}}$
fc(512,dim_u)

$f_{\text{sent}\to\text{feature}}$
embedding(1590,128)
reshape(batch_size,128,32)
conv2d(32,4,2,1)-bn2d-LReLU(0.2)
conv2d(64,4,2,1)-bn2d-LReLU(0.2)
conv2d(128,4,2,1)-bn2d-LReLU(0.2)
conv2d(256,(1,4),(1,2),(0,1))-bn2d-LReLU(0.2)
conv2d(512,(1,4),(1,2),(0,1))-bn2d-LReLU(0.2)

$f^{\text{sent}}_{\text{feature}\to\mathbf{z}}$
fc(8192,dim_z)

$f^{\text{sent}}_{\mathbf{z}\to\text{latent\_feature}}$
fc(dim_z,8192)-bn1d-LReLU(0.2)

$f^{\text{sent}}_{\text{features}\to\mathbf{v}}$
fc(16384,dim_v)

$f^{(\text{image},\text{sent})}_{\text{aggregate}}$
fc(8448,1024)-bn1d-LReLU(0.2)
fc(1024,1024)-bn1d-LReLU(0.2)
fc(1024,512)-bn1d-LReLU(0.2)

$f^{(\text{image},\text{sent})}_{\text{feature}\to\mathbf{z}}$
fc(512,dim_z)

Table 1.C.7. The neural network architectures in the CUB decoders.

$g^{\text{image}}_{(\mathbf{z},\mathbf{u})\to\text{feature}}$
fc(dim_z+dim_u,256)-LReLU(0.2)

$g^{\text{image}}_{\text{feature}\to\text{image}}$
fc(256,512)-bn1d-LReLU(0.2)
fc(512,1024)-bn1d-LReLU(0.2)
fc(1024,2048)

$g^{\text{sent}}_{(\mathbf{z},\mathbf{v})\to\text{feature}}$
fc(dim_z+dim_v,8192)
reshape(batch_size,512,4,4)
bn2d-LReLU(0.2)

$g^{\text{sent}}_{\text{feature}\to\text{sent}}$
reshape(batch_size,512,4,4)
deconv2d(256,(1,4),(1,2),(0,1))-bn2d-LReLU(0.2)
deconv2d(128,(1,4),(1,2),(0,1))-bn2d-LReLU(0.2)
deconv2d(64,4,2,1)-bn2d-LReLU(0.2)
deconv2d(32,4,2,1)-bn2d-LReLU(0.2)
deconv2d(1,4,2,1)
reshape(batch_size,32,128)
fc(128,1590)

Table 1.C.8. The neural network architectures in the CUB discriminators.

$h^{\text{common}}_{\text{image}\to\text{feature}}$
fc(2048,2048)-LReLU(0.2)
fc(2048,1024)-bn1d-LReLU(0.2)
fc(1024,512)-bn1d-LReLU(0.2)

$h^{\text{common}}_{\text{sent}\to\text{feature}}$
embedding(1590,128)
reshape(batch_size,128,32)
conv2d(64,4,2,1)-bn2d-LReLU(0.2)
conv2d(128,4,2,1)-bn2d-LReLU(0.2)
conv2d(256,4,2,1)-bn2d-LReLU(0.2)
conv2d(512,(1,4),(1,2),(0,1))-bn2d-LReLU(0.2)
conv2d(1024,(1,4),(1,2),(0,1))-bn2d-LReLU(0.2)

$h^{\text{common}}_{\text{aggregate}}$
fc(16896,2048)-bn1d-LReLU(0.2)
fc(2048,2048)-bn1d-LReLU(0.2)
fc(2048,1024)-bn1d-LReLU(0.2)

$h_{\text{latent}\to\text{feature}}$
fc(total_latent_dim, 512)-LReLU(0.2)

$h_{\text{feature}\to\text{ratio}}$
fc(1536, 512)-bn1d-LReLU(0.2)
fc(512,1)

feature dimension.

- For $\mathbf{x}$ (images), we injected noise to the ResNet feature of dimension 2048. We precomputed the standard deviation σ_i^{resnet} for each dimension i. For each evaluation of a discriminator that takes the ResNet feature as one of its arguments, we injected a zero-mean Gaussian noise of standard deviation $\alpha_{\text{resnet}} \times \sigma_i^{\text{resnet}}$ to the dimension i, where we used $\alpha_{\text{resnet}} = 2$.

- For $\mathbf{y}$ (sentences), we injected noise at an embedding level. We used an embedding layer that maps a word from the vocabulary of size 1590 to a 128-dimensional dense vectors. Whenever we computed a discriminator value, we computed the standard deviation σ_i^{embed} of the embedding layer for each embedding dimension $i \in \{1, \ldots, 128\}$; note here that the standard deviation changed along training as the embedding layer was set to be trainable. Then, with a scale of $\alpha_{\text{sent}} = 0.05$, we added a zero-mean Gaussian noise of standard deviation $\alpha_{\text{embed}} \times \sigma_i^{\text{embed}}$ to the embedding dimension i.

1.C.4 ZS-SBIR

In this dataset,

$$\mathbf{x} = \text{Sketch image (pretrained VGG features)},$$
$$\mathbf{y} = \text{Photo image (pretrained VGG features)}.$$

We followed the same experiment setting of (Hwang et al., 2020), including the network architectures.

Evaluation Metrics

In this experiment, we evaluated Precision@100 (P@100) and mean average precision (mAP), by translating the Tensorflow implementation of the codebase[5] of

[5]https://github.com/gr8joo/IIAE

(Hwang et al., 2020), which is in turn based on (Shen et al., 2018; Yelamarthi et al., 2018).

Network Architectures

The neural network architectures are summarized in Tables 1.3.9–1.3.11; here, f's are used in encoders, g's are in decoders, and h's are in discriminators. We note that a network without superscript indicates that the same network architecture is used in multiple places with different (i.e., not shared) realizations.

The generators and discriminators are defined in the same manner as for the MNIST–SVHN model (see Appendix 1.C.3), and thus omitted here.

Training

We used the Adam optimizer (Kingma and Ba, 2014) with $(\beta_1, \beta_2) = (0.5, 0.999)$, learning rate 5×10^{-4} for the generators and 10^{-3} for the discriminators (Heusel et al., 2017). We trained for 50 epochs.

The noise injection for discriminators was done similarly to the CUB model. Namely, we added a zero-mean Gaussian noise of standard deviation 0.5 for each latent variable $\mathbf{z}, \mathbf{u}, \mathbf{v}$, and we added a zero-mean Gaussian noise with adaptive standard deviation to based on the standard deviations of the features as for the CUB-image data, with the multiplicative scale $\alpha_{\mathbf{x}} = \alpha_{\mathbf{y}} = 0.5$.

Acknowledgement

Chapter 1, in part, is a reprint of the material in the paper: J. Jon Ryu, Yoojin Choi, Young-Han Kim, Mostafa El-Khamy, and Jungwon Lee, "Learning with Succinct Common Representation Based on Wyner's Common Information," arXiv:1905.10945v2, July 2022, submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence*. The dissertation author was the primary investigator and author of this paper. This work was supported in part by the National Science Foundation under Grant CCF-1911238 and SoC R&D, Samsung Semiconductor, Inc.

Table 1.3.9. The neural network architectures in the ZS-SBIR encoders.

$f_{\text{image}\to\text{feature}}$
`fc(512,512)-LReLU(0.2)`

$f_{\text{feature}\to\mathbf{z}}$
`fc(512,dim_z)`

$f^{\text{image}}_{\mathbf{z}\to\text{latent\_feature}}$
`fc(dim_z,512)-LReLU(0.2)`

$f_{\text{features}\to\mathbf{u}}$
`fc(1024,dim_local)`

$f^{\text{joint}}_{\text{aggregate}}$
`fc(1024,512)-LReLU(0.2)`

$f^{\text{joint}}_{\text{feature}\to\mathbf{z}}$
`fc(512,dim_z)`

Table 1.3.10. The neural network architectures in the ZS-SBIR decoders.

$g_{(\mathbf{z},\mathbf{u})\to\text{feature}}$
`fc(dim_z+dim_u,128)-ReLU`

$g_{\text{feature}\to\text{image}}$
`fc(128,512)`

Table 1.3.11. The neural network architectures in the ZS-SBIR discriminators.

$h^{\text{common}}_{\text{image}\to\text{feature}}$
`fc(1024,1024)-LReLU(0.2)`

$h_{\text{latent}\to\text{feature}}$
`fc(total_latent_dim, 512)-LReLU(0.2)`

$h_{\text{feature}\to\text{ratio}}$
`fc(1536, 512)-LReLU(0.2)` `fc(512,1)`

Bibliography

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *Proc. Int. Conf. Learn. Repr.*, 2017.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, 2013.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.*, 5:135–146, 2017.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv:2108.07258*, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proc. Int. Conf. Mach. Learn.*, pages 1597–1607. PMLR, 2020.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, New York, second edition, 2006.

Paul Cuff. Distributed channel synthesis. *IEEE Trans. Inf. Theory*, 59(11):7071–7096, Nov. 2013.

Andreas Damianou, Carl Ek, Michalis Titsias, and Neil Lawrence. Manifold relevance determination. In *Proc. Int. Conf. Mach. Learn.*, 2012.

Vincent Dutordoir, Hugh Salimbeni, James Hensman, and Marc Deisenroth. Gaussian process conditional density estimation. In *Proc. Adv. Neural Info. Proc. Syst.*, pages 2385–2395, 2018.

Anjan Dutta and Zeynep Akata. Semantically tied paired cycle consistency for zero-shot sketch-based image retrieval. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 5089–5098, 2019.

Carl Henrik Ek, Jon Rihan, Philip HS Torr, Grégory Rogez, and Neil D Lawrence. Ambiguity modeling in latent spaces. In *Int. Workshop Mach. Learn. Multimodal Interaction*, pages 62–73. Springer, 2008.

Abbas El Gamal and Young-Han Kim. *Network Information Theory*. Cambridge University Press, Cambridge, 2011.

Peter Gács and János Körner. Common information is far less than mutual information. *Probl. Control Inf. Theory*, 2(2):149–162, 1973.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proc. Int. Conf. Mach. Learn.*, pages 1180–1189. PMLR, 2015.

Shuyang Gao, Rob Brekelmans, Greg Ver Steeg, and Aram Galstyan. Auto-encoding total correlation explanation. In *Proc. Int. Conf. Artif. Int. Stat.*, pages 1157–1166, 2019.

Hans Gebelein. Das statistische Problem der Korrelation als Variations-und Eigenwertproblem und sein Zusammenhang mit der Ausgleichsrechnung. *ZAMM Z. fur Angew. Math. Mech.*, 21(6):364–379, 1941.

Abel Gonzalez-Garcia, Joost Van De Weijer, and Yoshua Bengio. Image-to-image translation for cross-domain disentanglement. In *Proc. Adv. Neural Info. Proc. Syst.*, volume 31, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. Adv. Neural Info. Proc. Syst.*, volume 27, 2014.

RM Gray and AD Wyner. Source coding for a simple network. *Bell Syst. Tech. J.*, 53(9): 1681–1721, 1974.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 770–778, 2016.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Proc. Adv. Neural Info. Proc. Syst.*, volume 30, 2017.

Hermann O Hirschfeld. A connection between correlation and contingency. In *Math. Proc. Camb. Philos. Soc.*, volume 31, pages 520–524. Cambridge University Press, 1935.

Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

Shao-Lun Huang, Anuran Makur, Gregory W Wornell, and Lizhong Zheng. On universal features for high-dimensional learning and inference. *arXiv:1911.09105*, 2019.

Shao-Lun Huang, Xiangxiang Xu, Lizhong Zheng, and Gregory W Wornell. A local characterization for Wyner common information. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 2252–2257. IEEE, 2020.

Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proc. Eur. Conf. Comput. Vis.*, pages 172–189, 2018.

HyeongJoo Hwang, Geon-Hyeong Kim, Seunghoon Hong, and Kee-Eung Kim. Variational interaction information maximization for cross-domain disentanglement. In *Proc. Adv. Neural Info. Proc. Syst.*, volume 33, 2020.

Harold Jeffreys. *The Theory of Probability*. OUP Oxford, 1998.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *Proc. Int. Conf. Learn. Repr.*, 2014.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Yann LeCun. The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/, 1998.

Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Proc. Eur. Conf. Comput. Vis.*, pages 35–51, 2018.

EL Lehmann and Henry Scheffé. Completeness, similar regions, and unbiased estimation: Part I. *Sankhyā: The Indian Journal of Statistics*, pages 305–340, 1950.

Kaiyi Lin, Xing Xu, Lianli Gao, Zheng Wang, and Heng Tao Shen. Learning cross-aligned latent embeddings for zero-shot cross-modal retrieval. In *Proc. AAAI Conf. Artif. Int.*, volume 34, pages 11515–11522, 2020.

Alexander H Liu, Yen-Cheng Liu, Yu-Ying Yeh, and Yu-Chiang Frank Wang. A unified feature disentangler for multi-domain image translation and manipulation. In *Proc. Adv. Neural Info. Proc. Syst.*, volume 31, 2018.

Li Liu, Fumin Shen, Yuming Shen, Xianglong Liu, and Ling Shao. Deep sketch hashing:

Fast free-hand sketch-based image retrieval. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 2862–2871, 2017.

Daniela Massiceti, Puneet K Dokania, N Siddharth, and Philip HS Torr. Visual dialogue without vision or dialogue. In *Proc. Adv. Neural Info. Proc. Syst.*, volume 31, 2018.

William McGill. Multivariate information transmission. *IRE Trans. Inf. Theory*, 4(4): 93–111, 1954.

Tomer Michaeli, Weiran Wang, and Karen Livescu. Nonparametric canonical correlation analysis. In *Proc. Int. Conf. Mach. Learn.*, pages 1967–1976. PMLR, 2016.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *neurips Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Trans. Inf. Theory*, 56(11):5847–5861, 2010.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. In *Proc. Adv. Neural Info. Proc. Syst.*, volume 29, pages 271–279, 2016.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS 2017 Workshop Autodiff*, 2017.

Ori Press, Tomer Galanti, Sagie Benaim, and Lior Wolf. Emerging disentanglement in auto-encoder based unsupervised image content transfer. In *Proc. Int. Conf. Learn. Repr.*, 2019.

Yuchen Pu, Weiyao Wang, Ricardo Henao, Liqun Chen, Zhe Gan, Chunyuan Li, and Lawrence Carin. Adversarial symmetric variational autoencoder. In *Proc. Adv. Neural Info. Proc. Syst.*, pages 4330–4339, 2017.

Alfréd Rényi. On measures of dependence. *Acta Math. Hungarica*, 10(3-4):441–451, 1959.

Mathieu Salzmann, Carl Henrik Ek, Raquel Urtasun, and Trevor Darrell. Factorized orthogonal latent spaces. In *Proc. Int. Conf. Artif. Int. Stat.*, pages 701–708, 2010.

San Diego Supercomputer Center. Triton shared computing cluster, 2022. University of California San Diego. Service. doi:https://doi.org/10.57873/T34W2R.

Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: Learning to retrieve badly drawn bunnies. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 2016.

Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, Aug. 2020. Version 0.2.1.

Ohad Shamir, Sivan Sabato, and Naftali Tishby. Learning and generalization with the information bottleneck. *Theor. Comput. Sci.*, 411(29-30):2696–2711, 2010.

Yuming Shen, Li Liu, Fumin Shen, and Ling Shao. Zero-shot sketch-image hashing. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 3598–3607, 2018. doi: 10.1109/CVPR.2018.00379.

Yuge Shi, Narayanaswamy Siddharth, Brooks Paige, and Philip HS Torr. Variational mixture-of-experts autoencoders for multi-modal deep generative models. In *Proc. Adv. Neural Info. Proc. Syst.*, volume 32, 2019.

Aaron Shon, Keith Grochow, Aaron Hertzmann, and Rajesh P Rao. Learning shared latent structure for image synthesis and robotic imitation. In *Proc. Adv. Neural Info. Proc. Syst.*, pages 1233–1240, 2006.

Rui Shu, Hung H. Bui, and Mohammad Ghavamzadeh. Bottleneck conditional density estimation. In *Proc. Int. Conf. Mach. Learn.*, pages 3164–3172, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. Int. Conf. Learn. Repr.*, 2015.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Proc. Adv. Neural Info. Proc. Syst.*, volume 28, pages 3483–3491, 2015.

Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. Amortised map inference for image super-resolution. In *Proc. Int. Conf. Learn. Repr.*, 2017.

Jiaming Song, Pratyusha Kalluri, Aditya Grover, Shengjia Zhao, and Stefano Ermon. Learning controllable fair representations. In *Proc. Int. Conf. Artif. Int. Stat.*, pages 2164–2173. PMLR, 2019.

Erixhen Sula and Michael Gastpar. Common information components analysis. *Entropy*, 23(2):151, 2021.

Masahiro Suzuki, Kotaro Nakayama, and Yutaka Matsuo. Joint multimodal learning

with deep generative models. *arXiv:1611.01891*, 2016.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. In *Proc. 37th Ann. Allerton Conf. Comm. Control Comput.*, pages 368–377, 1999.

Ramakrishna Vedantam, Ian Fischer, Jonathan Huang, and Kevin Murphy. Generative models of visually grounded imagination. In *Proc. Int. Conf. Learn. Repr.*, 2018.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD birds-200-2011 dataset. Technical report, California Institute of Technology and University of California, San Diego, 2011.

Weiran Wang, Xinchen Yan, Honglak Lee, and Karen Livescu. Deep variational canonical correlation analysis. *arXiv:1610.03454*, 2016.

Hans S Witsenhausen. On sequences of pairs of dependent random variables. *SIAM J. Appl. Math.*, 28(1):100–113, 1975.

Hans S Witsenhausen. Values and bounds for the common information of two discrete random variables. *SIAM J. Appl. Math.*, 31(2):313–333, 1976.

Aaron Wyner. The common information of two dependent random variables. *IEEE Trans. Inf. Theory*, 21(2):163–179, 1975.

Aolin Xu and Maxim Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. In *Proc. Adv. Neural Info. Proc. Syst.*, volume 30, pages 2524–2533, 2017.

Ge Xu, Wei Liu, and Biao Chen. A lossy source coding interpretation of Wyner's common information. *IEEE Trans. Inf. Theory*, 62(2):754–768, Feb. 2016.

Sasi Kiran Yelamarthi, Shiva Krishna Reddy, Ashish Mishra, and Anurag Mittal. A zero-shot framework for sketch based image retrieval. In *Proc. Eur. Conf. Comput. Vis.*, pages 300–317, 2018.

Lei Yu, Vincent YF Tan, et al. Common information, noise stability, and their extensions. *Found. Trends Commun. Inf. Theory*, 19(2):107–389, 2022.

Xiaoming Yu, Yuanqi Chen, Thomas Li, Shan Liu, and Ge Li. Multi-mapping image-to-image translation via learning disentanglement. In *Proc. Adv. Neural Info. Proc. Syst.*, volume 32, 2019.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. The information autoencoding family: A Lagrangian perspective on latent variable generative models. In *Proc. Uncertain.*

Artif. Intell., pages 1031–1041, 2018.

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Proc. Adv. Neural Info. Proc. Syst.*, volume 30, 2017.

Chapter 2

Kernel Embedding without Eigendecomposition of a Matrix

2.1 Introduction

Finding a good embedding of data for discovering *meaningful structures* is one of the fundamental problems in machine learning and data science, with important applications such as clustering, dimensionality reduction, and data visualization. Among a myriad of algorithms which have been proposed in the last few decades, we particularly focus on a class of kernel-based spectral embedding algorithms, which find embedding of data based on eigenvectors of data-dependent similarity kernel matrices (Bengio et al., 2004; Ham et al., 2004)—this class subsumes kernel principal component analysis (PCA) (Schölkopf et al., 1998), Laplacian eigenmaps (Belkin and Niyogi, 2003), spectral clustering (Ng et al., 2001; Shi and Malik, 2000), multidimensional scaling (Cox and Cox, 2008), locally linear embedding (Roweis and Saul, 2000), and Isomap (Tenenbaum et al., 2000). Proven to be extremely powerful in various applications, the common disadvantage of such methods is the computational complexity of eigendecomposition of a kernel matrix, which could be prohibitively large in big data analysis.

As an attempt to resolve the computational bottleneck, in this paper, we propose a new kernel embedding framework, which suggests a sample based embedding algorithm without eigendecomposition of a matrix for special choices of kernels. To

motivate our approach, we first review kernel PCA and introduce Laplacian eigenmaps as a special case of kernel PCA framework with a kernel with density regularization in Section 2.2. In Section 2.3, we then propose and study a new density-regularized kernel, which separates the underlying density and spectral decomposition of the kernel operator. We describe the resulting sample based algorithm, which simply combines density estimates given sample and known eigenfunctions of a kernel operator. In Section 2.4, dot-product kernels over hypersphere are discussed as a concrete example to which the proposed embedding framework may apply. We briefly discuss relevant literature in Section 2.6.

Notation Throughout the paper, we assume that a random vector $\mathbf{X}$ is drawn from density p over a closed subset $\mathcal{X} \subset \mathbb{R}^d$, and data points $\mathbf{x}_{1:N} := \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ are independently and identically distributed (i.i.d.) random variables drawn from p. Given $\mathcal{X} \subset \mathbb{R}^d$ and a density μ on $\mathcal{X}$, we consider a Hilbert space $L^2_\mu(\mathcal{X}) := \{f \colon \mathcal{X} \to \mathbb{C} | \int |f(\mathbf{x})|^2 \, \mathrm{d}\mu(\mathbf{x}) < \infty\}$ with inner product $\langle f, g \rangle_\mu := \int f(\mathbf{x}) \overline{g(\mathbf{x})} \, \mathrm{d}\mu(\mathbf{x})$. For a kernel function $k \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, we denote the *associated Hilbert–Schmidt integral operator* in boldface $\mathbf{K} \colon L^2_\mu(\mathcal{X}) \to L^2_\mu(\mathcal{X})$, which is defined as $(\mathbf{K}f)(\mathbf{x}) := \int_\mathcal{X} k(\mathbf{x}, \mathbf{t}) f(\mathbf{t}) \, \mathrm{d}\mu(\mathbf{t})$. In what follows, we always assume that a kernel is symmetric, i.e., $k(\mathbf{x}, \mathbf{t}) = k(\mathbf{t}, \mathbf{x})$, and satisfies $\int\int_{\mathcal{X}\times\mathcal{X}} k^2(\mathbf{x}, \mathbf{t}) \, \mathrm{d}\mu(\mathbf{x}) \, \mathrm{d}\mu(\mathbf{t}) < \infty$, so that the operator $\mathbf{K}$ is self-adjoint and compact.

2.2 Review of Kerenl PCA and Laplacian Eigenmaps

2.2.1 Kernel PCA

Feature space formulation

Kernel PCA (Schölkopf et al., 1998) was proposed as an efficient method to perform PCA over transformed samples with a given nonlinear mapping. Let $|\phi(\cdot)\rangle \colon \mathcal{X} \to \mathcal{F}$ be a feature map that maps a data point $\mathbf{x}$ to a point in a feature space $|\phi(\mathbf{x})\rangle \in \mathcal{F}$,

where $\mathcal{F}$ is a vector space with inner product $\langle \cdot | \cdot \rangle$.[1] For simplicity, assume for now that $\mathbb{E}[|\phi(\mathbf{X})\rangle] = |0\rangle$. Kernel PCA aims to perform PCA over the lifted random vector $|\phi(\mathbf{X})\rangle$, that is, to solve

$$\begin{aligned} &\underset{|u_\ell\rangle \in \mathcal{F}}{\text{maximize}} \ \sum_{\ell=1}^{L} \langle u_\ell | \mathbf{C}_\phi | u_\ell \rangle \\ &\text{subject to } \langle u_\ell | u_{\ell'} \rangle = \delta_{\ell\ell'} \end{aligned} \tag{2.1}$$

Here, $\mathbf{C}_\phi := \mathbb{E}[|\phi(\mathbf{X})\rangle\langle\phi(\mathbf{X})|]$ denotes the covariance operator of $|\phi(\mathbf{X})\rangle$. We call this the *(population) feature space problem* of kernel PCA.[2] When $\mathcal{F}$ is high- or infinite-dimensional, it is often not feasible to directly solve this problem.

Function space formulation

To avoid the issue with high-dimensionality of the feature space $\mathcal{F}$, we can convert the feature space problem (2.1) into an equivalent optimization problem over a *function space* by the so-called *kernel trick* as follows. Define a symmetric kernel function $k(\mathbf{x}, \mathbf{t}) := \langle \phi(\mathbf{x}) | \phi(\mathbf{t}) \rangle$. Consider the following optimization problem

$$\begin{aligned} &\underset{f_\ell \in L_p^2(\mathcal{X})}{\text{maximize}} \ \sum_{\ell=1}^{L} \langle f_\ell, \mathbf{K} f_\ell \rangle_p \\ &\text{subject to } \langle f_\ell, f_{\ell'} \rangle_p = \delta_{\ell\ell'} \end{aligned} \tag{2.2}$$

Since $\mathbf{K}$ is self-adjoint and compact, the solution is characterized by the top-L eigenfunctions and eigenvalues of $\mathbf{K}$; see, e.g., (Bolla, 2013, Proposition A.2.10). The following proposition establishes the equivalence between (2.1) and (2.2); the proof is easy and thus omitted.

Proposition 2.2.1. *Let $\lambda_1, \ldots, \lambda_L$ and $|u_1^\star\rangle, \ldots, |u_L^\star\rangle$ be the top-L eigenvalues and orthonormal eigenvectors of the operator $\mathbf{C}_\phi$, respectively. Let $\mu_1, \ldots, \mu_L$ and $f_1^\star, \ldots, f_L^\star$ be the top-L*

[1] We use the bra-ket notation to note that $\mathcal{F}$ may be infinite dimensional.
[2] If $|\phi(\mathbf{x})\rangle = \mathbf{x}$, it boils down to the original PCA.

eigenvalues and orthonormal eigenfunctions of the operator $\mathbf{K}$*, respectively. Then,* $\lambda_\ell = \mu_\ell$*,*

$$f_\ell^\star(\mathbf{x}) = \frac{1}{\sqrt{\lambda_\ell}} \langle \phi(\mathbf{x}) | u_\ell^\star \rangle, \text{ and} \tag{2.3}$$

$$|u_\ell^\star\rangle = \frac{1}{\sqrt{\lambda_\ell}} \int f_\ell^\star(\mathbf{x}) |\phi(\mathbf{x})\rangle p(\mathbf{x}) \, \mathrm{d}\mathbf{x}. \tag{2.4}$$

for each $\ell \in [L]$*.*

Hence, we call this problem (2.6) as the *(population) function space problem* of kernel PCA. If the top-L eigenfunctions $f_1^\star, \ldots, f_L^\star$ of the operator $\mathbf{K}$ are given, then the embedding of a query point $\mathbf{x}$ by kernel PCA is the projection of the lifted data $|\phi(\mathbf{x})\rangle$ onto the principal directions $|u_1^\star\rangle, \ldots, |u_L^\star\rangle$, or equivalently in view of (2.3),

$$\psi_{\mathsf{KPCA}}(\mathbf{x}) := [\sqrt{\lambda_1} f_1^\star(\mathbf{x}), \ldots, \sqrt{\lambda_L} f_L^\star(\mathbf{x})]^T. \tag{2.5}$$

Sample solution

The spectral decomposition of $\mathbf{K}$ in $L_p^2(\mathcal{X})$ cannot be performed directly in general even if the density p is known. Given sample $\mathbf{x}_{1:N}$, we can approximately solve (2.2) in practice. Let $\mathsf{K} \in \mathbb{R}^{N \times N}$ denote the *sample kernel matrix* whose (m, n)-th entry is $(\mathsf{K})_{mn} = k(\mathbf{x}_m, \mathbf{x}_n)$. Then, we can solve

$$\begin{aligned} &\underset{\mathbf{f}_\ell \in \mathbb{R}^N}{\text{maximize}} \sum_{\ell=1}^{L} \frac{\mathbf{f}_\ell^T}{\sqrt{N}} \frac{\mathsf{K}}{N} \frac{\mathbf{f}_\ell}{\sqrt{N}} \\ &\text{subject to } \frac{\mathbf{f}_\ell^T}{\sqrt{N}} \frac{\mathbf{f}_{\ell'}}{\sqrt{N}} = \delta_{\ell\ell'} \end{aligned} \tag{2.6}$$

as a proxy to (2.2), which is equivalent to the eigendecomposition of K. The optimal solution is characterized by the top-L eigenvectors $\mathbf{f}_1^\star, \ldots, \mathbf{f}_L^\star \in \mathbb{R}^N$ of the normalized sample kernel matrix K/N with eigenvalues $\lambda_1, \ldots, \lambda_L$ with norm $\|\mathbf{f}_\ell^\star\|_2 = \sqrt{N}$. The

L-dimensional embedding of a point $\mathbf{x}$ is then

$$\hat{\psi}_{\mathsf{KPCA}}(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} k(\mathbf{x}, \mathbf{x}_i)\Big[\frac{(\mathbf{f}_1^\star)_i}{\sqrt{\lambda_1}}, \ldots, \frac{(\mathbf{f}_L^\star)_i}{\sqrt{\lambda_L}}\Big]^T. \tag{2.7}$$

This is often referred to the *Nyström formula*; see, e.g., (Bengio et al., 2004). In particular, for a sample point $\mathbf{x}_n$, the embedding is simply

$$\hat{\psi}_{\mathsf{KPCA}}(\mathbf{x}_n) := [\sqrt{\lambda_1}(\mathbf{f}_1^\star)_n, \ldots, \sqrt{\lambda_L}(\mathbf{f}_L^\star)_n]^T.$$

We refer to kernel PCA as the procedure consisting of the eigendecomposition of the kernel matrix K and the embedding (2.7).

Remark 2.2.2 (Centering)**.** In (2.1), (2.2), (2.6), and (2.7), we assume $\mathbb{E}[|\phi(\mathbf{X})\rangle] = |0\rangle$. Hence, given sample $\mathbf{x}_{1:N}$, we need to center the sample kernel matrix K as $\mathsf{K}_c = (I_N - 1_N)\mathsf{K}(I_N - 1_N) \in \mathbb{R}^{N\times N}$, where $1_N := \frac{1}{N}\, 1_N\, 1_N^T \in \mathbb{R}^{N\times N}$.

Remark 2.2.3. In practice, any choice of symmetric kernel function k can be deployed in kernel PCA. Note, however, that the feature space formulation and PCA interpretation via Proposition 2.2.1 remain valid if and only if the kernel is in the form $k(\mathbf{x}, \mathbf{t}) = \langle\phi(\mathbf{x})|\phi(\mathbf{t})\rangle$ for some inner product space $\mathcal{F}$ and function $\phi\colon \mathcal{X} \to \mathcal{F}$. Mercer's theorem (Mercer, 1909) establishes positive definiteness of a kernel as an equivalent condition for the existence of such a mapping.

2.2.2 Laplacian eigenmaps

Laplacian eigenmaps (Belkin and Niyogi, 2003) is one of the most popular embedding algorithm, which can be justified as an approximation of the Laplacian–Beltrami operator or a relaxed solution to the graph min-cut problem (Shi and Malik, 2000). Here, we introduce Laplacian eigenmaps as a special instance of kernel PCA. Given a base symmetric kernel function k, we first define the *kernelized density* $p_k(\mathbf{x}) := \int k(\mathbf{x}, \mathbf{t})p(\mathbf{t})\,\mathrm{d}\mathbf{t}$

and define a new kernel function as

$$\overline{k}_p(\mathbf{x}, \mathbf{t}) := \frac{k(\mathbf{x}, \mathbf{t})}{\sqrt{p_k(\mathbf{x}) p_k(\mathbf{t})}}.$$

The (kernelized) Laplacian eigenmaps with the base kernel k is characterized by the population function space optimization problem (2.2) of kernel PCA with the kernel $\overline{k}_p$ which is the function space optimization problem (2.6) of kernel PCA with the kernel $\overline{k}_p(\mathbf{x}, \mathbf{t})$. Let $f_1^\star, \ldots, f_L^\star$ denote the top-L orthonormal eigenfunctions of the operator $\overline{\mathbf{K}}_p$. Then, the Laplacian eigenmaps of a point $\mathbf{x}$ is defined as the evaluations of the eigenfunctions:

$$\psi_{\mathsf{LE}}(\mathbf{x}) := [f_1^\star(\mathbf{x}), \ldots, f_L^\star(\mathbf{x})]^T. \tag{2.8}$$

As in kernel PCA, given samples $\mathbf{x}_{1:N}$, we perform eigendecomposition of the sample kernel matrix $\overline{\mathsf{K}}_p$ defined as

$$(\overline{\mathsf{K}}_p)_{ij} := \frac{k(\mathbf{x}_i, \mathbf{x}_j)}{\sqrt{\hat{p}_k(\mathbf{x}_i) \hat{p}_k(\mathbf{x}_j)}},$$

where $\hat{p}_k(\mathbf{x}) := \frac{1}{N} \sum_{i=1}^N k(\mathbf{x}, \mathbf{x}_i)$ denotes the empirical estimate of the kernelized density $p_k(\mathbf{x})$. The embedding by Laplacian eigenmaps of a sample point $\mathbf{x}_n$ is then

$$\hat{\psi}_{\mathsf{LE}}(\mathbf{x}_n) := [(\mathbf{f}_1^\star)_n, \ldots, (\mathbf{f}_L^\star)_n]^T.$$

Remark 2.2.4. Despite the apparent mathematical equivalence, kernel PCA embedding with the kernel $\overline{k}_p$ may differ from Laplacian eigenmaps embedding significantly due to centering in kernel PCA (see Remark 2.2.2) and the different definitions of embeddings (see (2.5) and (2.8)).

Remark 2.2.5. Laplacian eigenmaps may also use the k-th neighborhood adjacency matrix instead of kernel-based weight matrix given samples—however, it does not fit

to the current population optimization framework, and thus studying this version is beyond the scope of the paper.

Remark 2.2.6. Eigendecomposition of $\overline{\mathsf{K}}_p$ is approximately equivalent to the that of the *symmetric normalized graph Laplacian*, which is typically used in spectral embedding. Define a weight matrix $\mathsf{W} \in \mathbb{R}^{N\times N}$ as $(\mathsf{W})_{ij} = (1-\delta_{ij})k(\mathbf{x}_i, \mathbf{x}_j)$ and define a degree matrix D as the diagonal matrix with entry $(\mathsf{D})_{ii} = \sum_{j=1}^{N}(\mathsf{W})_{ij}$. The symmetric normalized graph Laplacian is then defined as $\mathsf{L}_{\mathsf{sym}} := \mathsf{D}^{-1/2}\mathsf{W}\mathsf{D}^{-1/2}$. Since the difference

$$\overline{\mathsf{K}}_p - \mathsf{L}_{\mathsf{sym}} = \frac{1}{N}\operatorname{diag}\Big(\frac{k(\mathbf{x}_1,\mathbf{x}_1)}{\hat{p}_k(\mathbf{x}_1)}, \ldots, \frac{k(\mathbf{x}_N,\mathbf{x}_N)}{\hat{p}_k(\mathbf{x}_N)}\Big)$$

vanishes in the operator norm as $N \to \infty$, eigendecomposition of $\overline{\mathsf{K}}_p$ becomes equivalent to that of $\mathsf{L}_{\mathsf{sym}}$ in the sample limit.

2.3 Kernel Embedding Without Eigendecomposition

2.3.1 A new density-regularized kernel

So far, we reviewed the two important kernel-based embedding frameworks, kernel PCA and Laplacian eigenmaps: Laplacian eigenmaps fits into the framework of kernel PCA with a specific form of density-regularized kernel $\overline{k}_p$; see Table 2.3.1. The population problems cannot be solved directly, but given sample, we can approximately solve them via eigendecomposition of a matrix of possibly large size.

In this section, motivated by the form of the kernel $\overline{k}_p$ of Laplacian eigenmaps, we introduce a new kernel function

$$k_p(\mathbf{x}, \mathbf{t}) := \frac{k(\mathbf{x}, \mathbf{t})}{\sqrt{p(\mathbf{x})p(\mathbf{t})}} \tag{2.9}$$

and propose the population function space optimization problem (2.2) of kernel PCA

Table 2.3.1. Overview of population and sample problems of kernel PCA, Laplacian eigenmaps, and the proposed kernel embedding.

	Kernel PCA		**Laplacian eigenmaps**	**Proposed**
	Feature space	Function space		
Population	$\underset{\lvert u_\ell\rangle \in \mathcal{F}}{\text{maximize}} \sum_{\ell=1}^{L} \langle u_\ell \rvert \mathbf{C}_\phi \rvert u_\ell \rangle$ (2.1) subjecto to $\langle u_\ell \rvert u_{\ell'} \rangle = \delta_{\ell\ell'}$	$\underset{f_\ell \in L_p^2(\mathcal{X})}{\text{maximize}} \sum_{\ell=1}^{L} \langle f_\ell, \mathbf{K} f_\ell \rangle_p$ (2.2) subjecto to $\langle f_\ell, f_{\ell'} \rangle_p = \delta_{\ell\ell'}$	$\mathbf{K} \leftarrow \overline{\mathbf{K}}_p$	$\mathbf{K} \leftarrow \mathbf{K}_p$ (2.10)
Sample	$\underset{\lvert u_\ell\rangle \in \mathcal{F}}{\text{maximize}} \sum_{\ell=1}^{L} \langle u_\ell \rvert \hat{\mathbf{C}}_\phi \rvert u_\ell \rangle$ subjecto to $\langle u_\ell \rvert u_{\ell'} \rangle = \delta_{\ell\ell'}$	$\underset{\mathbf{f}_\ell \in \mathbb{R}^N}{\text{maximize}} \sum_{\ell=1}^{L} \frac{\mathbf{f}_\ell^T}{\sqrt{N}} \frac{\mathsf{K}}{N} \frac{\mathbf{f}_\ell}{\sqrt{N}}$ (2.6) subjecto to $\frac{\mathbf{f}_\ell^T}{\sqrt{N}} \frac{\mathbf{f}_{\ell'}}{\sqrt{N}} = \delta_{\ell\ell'}$	$\mathsf{K} \leftarrow \overline{\mathsf{K}}_p$	–

with k_p, that is,

$$\begin{aligned} &\underset{f_\ell \in L^2_p(\mathcal{X})}{\text{maximize}} \ \sum_{\ell=1}^{L} \langle f_\ell, \mathbf{K}_p f_\ell \rangle_p \\ &\text{subject to } \langle f_\ell, f_{\ell'} \rangle_p = \delta_{\ell\ell'} \end{aligned} \tag{2.10}$$

as a new criterion for kernel embedding. Compared to the kernel $\overline{k}_p$ of Laplacian eigenmaps, the base kernel function k is now regularized by the true underlying density p instead of the kernelized density p_k.

With the new kernel k_p, we can reshape the population optimization problem (2.10) into a much simpler form. For a weighting function $w\colon \mathcal{X} \to \mathbb{R}_+$ whose support subsumes the support of p, we define the density-scaled function

$$g_\ell(\mathbf{x}) := \sqrt{\frac{p(\mathbf{x})}{w(\mathbf{x})}} f_\ell(\mathbf{x}). \tag{2.11}$$

Note that if $f_\ell \in L^2_p(\mathcal{X})$, then $g_\ell \in L^2_w(\mathcal{X})$. If we define $k_w(\mathbf{x}, \mathbf{t}) := k(\mathbf{x}, \mathbf{t})/\sqrt{w(\mathbf{x})w(\mathbf{t})}$, we have

$$\langle f_\ell, \mathbf{K}_p f_\ell \rangle_p = \langle g_\ell, \mathbf{K}_w g_\ell \rangle_w \quad \text{and} \quad \langle f_\ell, f_{\ell'} \rangle_p = \langle g_\ell, g_{\ell'} \rangle_w,$$

which imply that the new problem (2.10) can be recast as

$$\begin{aligned} &\underset{g_\ell \in L^2_w(\mathcal{X})}{\text{maximize}} \ \sum_{\ell=1}^{L} \langle g_\ell, \mathbf{K}_w g_\ell \rangle_w \\ &\text{subject to } \langle g_\ell, g_{\ell'} \rangle_w = \delta_{\ell\ell'}. \end{aligned} \tag{2.12}$$

We remark that (2.12) solely depends on the choice of kernel k and the weighting function w. Provided that $\mathbf{K}_w$ is compact, the solution of this optimization problem

is characterized by the top-L eigenfunctions $g_1^\star, \ldots, g_L^\star$ of the operator $\mathbf{K}_w$. Somewhat surprisingly, for a few special cases, the eigenexpansion of $\mathbf{K}_w$ is given in an analytical form; see Section 2.4. The eigenfunctions of $\mathbf{K}_p$ are then given as the functions $f_1^\star, \ldots, f_L^\star$, where $f_\ell^\star(\mathbf{x}) := \sqrt{w(\mathbf{x})/p(\mathbf{x})}g_\ell^\star(\mathbf{x})$. Provided that the density $p(\mathbf{x})$ can be evaluated, the L-dimensional embedding of a query point $\mathbf{x}$ is

$$\psi_{\mathsf{KE}}(\mathbf{x}) := \sqrt{\frac{w(\mathbf{x})}{p(\mathbf{x})}}[g_1^\star(\mathbf{x}), \ldots, g_L^\star(\mathbf{x})]^T. \tag{2.13}$$

2.3.2 A new sample based kernel embedding

Provided that spectral decomposition of $\mathbf{K}_w$ is known for a choice of k and w, the only unknown in the embedding (2.13) is the density p. Hence, given sample $\mathbf{x}_{1:N}$, we only need to estimate the density, without any spectral decomposition of a matrix. This yields the following kernel embedding algorithm.

Algorithm 1. Kernel embedding without spectral decomposition

Input a base kernel k, a weighting function w, a density estimator $\hat{p}(\cdot)$, sample $\{\mathbf{x}_n\}_{n=1}^N$, a target dimension $L \in \mathbb{N}$.

1: Find the top-L orthonormal eigenfunctions $g_1^\star, \ldots, g_L^\star$ of the integral operator $\mathbf{K}_w \colon L_w^2(\mathcal{X}) \to L_w^2(\mathcal{X})$.

2: Given a query point $\mathbf{x} \in \mathcal{X}$, output the L-dimensional embedding of $\mathbf{x}$ as

$$\hat{\psi}_{\mathsf{KE}}(\mathbf{x}) := \sqrt{\frac{w(\mathbf{x})}{\hat{p}(\mathbf{x})}}[g_1^\star(\mathbf{x}), \ldots, g_L^\star(\mathbf{x})]^T.$$

2.4 Dot-Product Kernels Over Hypersphere

In this section, we focus on a special class of kernel functions of the form of $k_w(\mathbf{x}, \mathbf{t}) = f(\mathbf{x}^T\mathbf{t})$ for some function $f \colon \mathbb{R} \to \mathbb{R}$, which are called *dot-product kernels*. This class contains many interesting kernels including homogeneous polynomial $f(u) = u^p$ ($p > 0$), inhomogeneous polynomial $f(u) = (1 + u)^p$ ($p > 0$), Vovk's real polynomial

$f(u) = (1 - u^p)/(1 - u)$ ($p > 0$), Vovk's infnite polynomial $f(u) = 1/(1 - u)$, and hyperbolic tangent $f(u) = \tanh(a + u)$ ($a \in \mathbb{R}$) kernels (Smola et al., 2001).

Further, we consider a special domain, the unit hypersphere $\mathbb{S}^{d-1} := \{\mathbf{x} \in \mathbb{R}^d \colon \|\mathbf{x}\|_2 = 1\}$ in $\mathbb{R}^d$. On $\mathbb{S}^{d-1}$, the class of dot-product kernels include additional popular kernels such as Gaussian kernels $f(u) = e^{-(1+u)/\sigma^2}$ ($\sigma > 0$) and arccosine kernel $f(u) = 1 - (2/\pi)\cos^{-1}(u)$. Note that some real-world data such as images approximately lie on a hypersphere (Minh et al., 2006; Smola et al., 2001) and dot-product kernels may work best on $\mathbb{S}^{d-1}$ by nature. The key property of $\mathbb{S}^{d-1}$ is that with uniform weighting function w, the eigensystem of $\mathbf{K}_w$ is characterized by *spherical harmonics*.

Definition 2.4.1 (Spherical harmonics)**.** Let $\Delta = -\sum_{i=1}^{d} \frac{\partial^2}{\partial x_i^2}$ denote the Laplacian operator on $\mathbb{R}^d$. A homogeneous polynomial of degree n in $\mathbb{R}^d$ whose Laplacian vanishes is called a *homogeneous harmonic* of order n. Let $\mathcal{P}_n$ denote the space of $\mathbb{C}$-valued homogeneous polynomials of degree n in d real variables. Let $\mathcal{Y}_n(d)$ denote the subspace of all homogeneous harmonics of order n, that is, $\mathcal{Y}_n(d) := \{p \in \mathcal{P}_n \colon \Delta p = 0\}$. The *spherical harmonics* of order n and dimension d are defined as the functions in $\mathcal{Y}_n(d)$ restricted over $\mathbb{S}^{d-1}$.

Remark 2.4.2. The dimension of the subspace $\mathcal{Y}_n(d)$ is

$$N(d, n) := \dim \mathcal{Y}_n(d) = \frac{2n + d - 2}{n} \binom{n + d - 3}{n - 1}$$

for $n \geq 0$.

The following elegant theorem, which is often referred to as the Funk–Hecke formula, shows that spherical harmonics fully characterize the eigenfunctions of any dot-product kernel over $\mathbb{S}^{d-1}$. Let $P_\ell^m(t)$ denote the *associated Legendre polynomial* of degree ℓ and order m for integers $0 \leq m \leq \ell$. Let $|\mathbb{S}^{d-1}| := (2\pi^{d/2})/\Gamma(d/2)$ denote the surface area of $\mathbb{S}^{d-1}$.

Theorem 2.4.3 (Funk–Hecke (Müller, 2012)). *Let $f\colon [-1,1] \to \mathbb{R}$ be a continuous function. For $Y_n \in \mathcal{Y}_n(d)$ for $n \geq 0$, we have*

$$\int_{\mathbb{S}^{d-1}} f(\mathbf{x}^T\mathbf{t}) Y_n(\mathbf{t}) \, \mathrm{d}\mathbb{S}^{d-1}(\mathbf{t}) = \lambda_n Y_n(\mathbf{x}) \quad \forall \mathbf{x} \in \mathbb{S}^{d-1},$$

where $\lambda_n = |\mathbb{S}^{d-2}| \int_{-1}^{1} f(u) P_n^d(u) (1-u^2)^{\frac{d-3}{2}} \, \mathrm{d}u$.

Corollary 2.4.4. *Let $\mathcal{X} = \mathbb{S}^{d-1}$ for $d \geq 2$ and let w be the uniform density on $\mathbb{S}^{d-1}$. For any dot-product kernel of the form $k_w(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}^T\mathbf{y})$ for some continuous function $f\colon [-1,1] \to \mathbb{R}$, there is an orthonormal basis of $\mathcal{Y}_n(d)$ comprised by the eigenfuntions of $\mathbf{K}_w$ with eigenvalue λ_n defined in Theorem 2.4.3.*

Remark 2.4.5. Minh et al. (2006, Theorems 2 and 3) computed the nonzero eigenvalues of gaussian kernels $f(u) = \exp(-(1+u)/\sigma^2)$ ($\sigma > 0$) and polynomial kernels $f(u) = (1+u)^p$ ($p \in \mathbb{N}$) in terms of special hypergeometric functions. In particular, the eigenvalues $(\lambda_n)_{n=0}^{\infty}$ of gaussian kernels are decreasing in n if $\sigma^2 \geq 2/d$, and those of the polynomial kernel of degree p are always decreasing in n and $\lambda_n = 0$ for $n \geq p+1$.

Hence, if we choose Gaussian or polynomial kernels, we only need to evaluate the first L real spherical harmonics to compute the kernel embedding (2.13) on $\mathcal{X} = \mathbb{S}^{d-1}$. For practical implementation, here we present a version of real orthonormal basis $\{Y_{n,j}^d(\mathbf{x})\}_{j=1}^{N(d,n)}$ of spherical harmonics of order n and dimension d (Higuchi, 1987, Section 2). Given a point $\mathbf{x} = (x_1, \ldots, x_D) \in \mathbb{S}^{d-1}$, define a hyperspherical coordinate system $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{d-1}) \in [0, 2\pi) \times [0, \pi]^{d-2}$ as

$$\theta_1 := \begin{cases} \cos^{-1} \dfrac{x_2}{\sqrt{x_1^2 + x_2^2}} & \text{if } x_1 \geq 0, \\ 2\pi - \cos^{-1} \dfrac{x_2}{\sqrt{x_1^2 + x_2^2}} & \text{if } x_1 < 0, \end{cases}$$

$$\theta_i := \cos^{-1} \frac{x_{i+1}}{\sqrt{x_1^2 + \ldots + x_{i+1}^2}}, \quad 2 \leq i \leq d-1.$$

Here, θ_1 and $\theta_2, \ldots, \theta_{d-1}$ are called the *azimuthal angle* and the *polar angles*, respectively. For integers $|\ell_1| \leq \ell_2 \leq \ldots \leq \ell_{d-1} = n$, we define a *canonical spherical harmonics of degree* $\ell_{d-1} = n$ *and order* $(\ell_1, \ldots, \ell_{d-2})$ as

$$Y_{\ell_1,\ldots,\ell_{d-1}}(\boldsymbol{\theta}) := \frac{1}{\sqrt{2\pi}} e^{i\ell_1\theta_1} \prod_{j=2}^{d-1} {}_j\overline{P}_{\ell_j}^{\ell_{j-1}}(\theta_j), \quad \text{where}$$

$${}_j\overline{P}_{\ell'}^{\ell}(\theta) := \sqrt{\left(\ell + \frac{j-1}{2}\right) \frac{(\ell+\ell'+j-2)!}{(\ell'-\ell)!}} \frac{P_{\ell'+\frac{j-2}{2}}^{-(\ell+\frac{j-2}{2})}(\cos\theta)}{\sin^{\frac{j-2}{2}}(\theta)}$$

for $\ell \leq \ell'$ and $j \geq 2$. Here, $P_\lambda^\mu(z)$ denotes the *Legendre functions of the first kind* for $z \in \mathbb{C}$ such that $|1 - z| < 2$. Finally, we define a real-valued version, often called the *tesseral harmonics*, as

$$\tilde{Y}_{\ell_1,\ell_2,\ldots,\ell_{d-1}} := \begin{cases} \sqrt{2}(-1)^{\ell_1} \operatorname{Im}(Y_{|\ell_1|,\ell_2,\ldots,\ell_{d-1}}) & \text{if } \ell_1 < 0, \\ Y_{0,\ell_2,\ldots,\ell_{d-1}} & \text{if } \ell_1 = 0, \\ \sqrt{2}(-1)^{\ell_1} \operatorname{Re}(Y_{|\ell_1|,\ell_2,\ldots,\ell_{d-1}}) & \text{if } \ell_1 > 0. \end{cases}$$

Then, $\mathcal{B}_n(d) := \{\tilde{Y}_{\ell_1,\ldots,\ell_{d-1}} : |\ell_1| \leq \ell_2 \leq \ldots \leq \ell_{d-1} = n\}$ forms a real orthonormal eigenbasis of $\mathcal{Y}_n(d)$.

A few remarks on related results are in order.

Remark 2.4.6 (Multiplicative dot-product kernels over a torus)**.** Since Corollary 2.4.4 remains valid for $d = 2$, the eigenfunctions of any dot-product kernels over $\mathbb{S}^1$ are the Fourier basis $\{e^{i\ell\theta}/\sqrt{2\pi}\}_{\ell=0}^{\infty}$ for $\theta \in [0, 2\pi)$. Hence, any dot-product kernel of a multiplicative form such as Gaussian kernels over the d-dimensional torus $\mathbb{T}^d := \mathbb{S}^1 \times \cdots \times \mathbb{S}^1$ (with d products) has the product of 1-dimensional Fourier bases as eigenfunctions. This result may be of particular interest for real-world data naturally lying on the torus such as RNA structure data (Eltzner et al., 2018).

Remark 2.4.7 (Dot-product kernels over a ball)**.** Smola et al. (2001, Section 6) provided a version of the eigensystem of a dot-product kernel over the unit ball $\mathbb{B}^d := \{\mathbf{x} \in \mathbb{R}^d \colon \|\mathbf{x}\|_2 \leq 1\}$ by the separation of variables trick. Here we present the idea with a minor correction. Let $f \colon \mathbb{R} \to \mathbb{R}$ be an analytic function such that $f(t) = \sum_{m=0}^{\infty} f_m t^m$. Plugging in the expansion of monomial u^m ($u \in [-1, 1], m \geq 0$) with respect to the associated Legendre polynomials $(P_n^d(u))_{n \geq d}$ of dimension d, we can write

$$f(\mathbf{x}^T\mathbf{t}) = \sum_{n=d}^{\infty} \kappa_n(\|\mathbf{x}\|\|\mathbf{t}\|) P_n^d\Big(\frac{\mathbf{x}^T\mathbf{t}}{\|\mathbf{x}\|\|\mathbf{t}\|}\Big),$$

where we define $\kappa_n(u) := \sum_{m=0}^{\infty} f_m c_m(d, n) u^m$ and $c_m(d, n) := \frac{2n+1}{2} \frac{(n-d)!}{(n+d)!} \int_{-1}^{1} u^m P_n^d(u) \, \mathrm{d}u$ for $n \geq d$. Now, for each $n \geq d$, let $(\varphi_{nm} \in L^2_{r \mapsto r^{d-1}}([0, 1]))_{m=1}^{\infty}$ and $(\rho_{nm})_{m=1}^{\infty}$ be the eigensytem of the 1D kernel κ_n, i.e.,

$$\int_0^1 \kappa_n(r\tilde{r}) \varphi_{nm}(\tilde{r}) \tilde{r}^{d-1} \, \mathrm{d}\tilde{r} = \rho_{nm} \varphi_{nm}(r). \tag{2.14}$$

With the addition theorem (Müller, 2012, p. 18) on the expansion of P_n^d with $(Y_{n,j}^d)_{j=1}^{N(d,n)}$, it is then easy to check that $\{\varphi_{nm}(r) Y_{n,j}^d(\boldsymbol{\theta}) \colon m \geq 1, n \geq d, 1 \leq j \leq N(d, n)\}$ forms an orthonormal eigenbasis of $\mathbf{K}$ over $\mathbb{B}^d$ with eigenvalues $\frac{\mathbb{S}^{d-1}}{N(d,n)} \rho_{nm}$ of multiplicity $N(d, n)$. We note, in practice, that the integral equation (2.14) can be solved by eigendecomposition of a matrix with approximation of κ_n with finite terms; we illustrate how to perform the approximation in Appendix.

Remark 2.4.8 (Gaussian kernels with Gaussian weighting)**.** For $\mathcal{X} = \mathbb{R}^d$, when $\mathbf{K}_w$ is a Gaussian kernel with a Gaussian weighting function w, the eigensystem of $\mathbf{K}_w$ is characterized by Hermite polynomials (Fasshauer, 2011; Rasmussen, 2003). Note, however, that since $w(\mathbf{x})$ is non-uniform being Gaussian, the base kernel $k(\mathbf{x}, \mathbf{t}) = \sqrt{w(\mathbf{x})} k_w(\mathbf{x}, \mathbf{t}) \sqrt{w(\mathbf{t})}$ becomes a Gaussian kernel with an additional attenuation term.

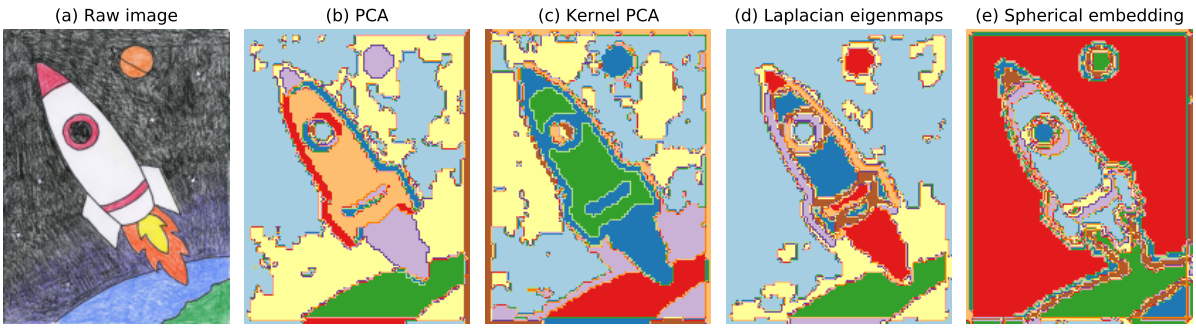


Figure 2.5.1. An illustrative example with image segmentation.

2.5 Experiments

To illustrate the applicability of the proposed framework, we consider the following simple image segmentation procedure. Suppose that we are given an (color) image $Y \in [0,1]^{H \times W \times 3}$. For each pixel $Y^{(i)} \in \mathbb{R}^3$, we consider the $P \times P \times 3$ patch centered at $Y^{(i)}$, denoted as $\mathbf{y}^{(i)} \in [0,1]^{P \times P \times 3} \cong [0,1]^{3P^2}$, as its representation. We apply a kernel embedding algorithm such as Laplacian eigenmaps or the proposed kernel embedding to the patches $\{\mathbf{y}^{(i)}\}_{i=1}^{HW}$, and apply the k-means algorithm (Hartigan and Wong, 1979; Lloyd, 1982) as in spectral clustering (Shi and Malik, 2000); the resulting labels can be viewed as a segmentation of the image.

We present a sample image segmentation result with $P = 2$ in Fig. 2.5.1. For kernel PCA and Laplacian eigenmaps, we used isotropic Gaussian kernels with bandwidth selected as median of all pairwise Euclidean distances. For the proposed kernel embedding, we applied the kernel embedding based on spherical harmonics in Section 2.4, by mapping the data onto a unit hypersphere and used the Gaussian kernel density estimator with the same bandwidth. The number of clusters used in the k-means algorithm was 8. We remark that the spherical embedding has orders-of-magnitude lesser time complexity ($\sim$2s) than the other kernel-based embeddings ($\sim$100s), while providing a comparable result.

2.6 Related Work

Spectral clustering (Ng et al., 2001; Shi and Malik, 2000; Weiss, 1999) has many versions depending on the form of graph Laplacian in the procedure, and Laplacian eigenmaps (Belkin and Niyogi, 2003) is equivalent to the spectral embedding used in the version of spectral clustering by Shi and Malik (2000). Schiebinger et al. (2015) analyzed the normalized kernel operator $\overline{\mathbf{K}}_p$ to establish the performance of spectral clustering.

The mathematical equivalence between Laplacian eigenmaps and kernel PCA established in Section 2.2 is not entirely new. For example, Ng et al. (2001) pointed out a link between spectral clustering and kernel PCA. More generally, Ham et al. (2004) and Bengio et al. (2004) interpreted Laplacian eigenmaps, multidimensional scaling, Isomap, and locally linear embedding as specific instantiations of kernel PCA. Note, however, that they only considered the sample based algorithms not the underlying population optimization problems, while this paper crucially relies on the population formulation.

Dot-product kernels have been studied in the context its regularization property for support vector machines (Smola et al., 2001) and their feature functions (Minh et al., 2006). For a more detailed account on spherical harmonics, we refer an interested reader to (Efthimiou and Frye, 2014; Müller, 2012).

2.7 Concluding Remarks

In this work, we proposed a rather unorthodox perspective on kernel-based spectral embedding. We introduced a new criterion for kernel embedding with a new density-regularized kernel, which results in a kernel embedding algorithm without spectral decomposition of a matrix. The advantage comes from the special structure of the kernel k_p in (2.9), which allows the separation of the density from the eigendecomposition of the kernel operator.

We emphasize that the proposed algorithm is not proposed to replace the existing

spectral methods; instead, it should be viewed as an extremely low-cost kernel-based embedding, which may be particularly advantageous when a dataset is large and computational resource is limited. A deeper investigation and more extensive experiments including its variations in Remarks 2.4.6, 2.4.7, and 2.4.8 will be reported elsewhere.

Appendix

2.A A Numerical Solution to the Eigenequation (2.14)

In this section, we elaborate how to solve (2.14) numerically given a function f that characterizes a kernel.

Given an analytic function $f(t) = \sum_{m=0}^{\infty} f_m t^m$, let $b_{nm} := f_m c_m(d, n)$ so that we can write $\kappa_n(u) = \sum_{m=0}^{\infty} b_{nm} u^m$.

2.A.1 Compute $c_m(d, n)$

To compute the coefficient $c_m(d, n)$, essentially $\int_{-1}^{1} u^m P_n^d(u)\,\mathrm{d}u$, we utilize the following closed form expression of the associated Legendre function: for $0 \le d \le n$,

$$P_n^d(x) = (-1)^d 2^n (1-x^2)^{d/2} \sum_{k=d}^{n} \frac{k!}{(k-d)!} x^{k-d} \binom{n}{k} \binom{\frac{n+k-1}{2}}{n}$$

for $x \in [-1, 1]$. Hence, for $n \ge d$, we have

$$\begin{aligned} c_m(d, n) &= \frac{2n+1}{2} \frac{(n-d)!}{(n+d)!} \int_{-1}^{1} u^m P_n^d(u)\,\mathrm{d}u \\ &= \frac{2n+1}{2} \frac{(n-d)!}{(n+d)!} (-1)^d 2^n \sum_{k=d}^{n} \frac{k!}{(k-d)!} \binom{n}{k} \binom{\frac{n+k-1}{2}}{n} \int_{-1}^{1} (1-u^2)^{d/2} u^{k+m-d}\,\mathrm{d}u. \end{aligned}$$

Now, it is enough to compute for $d \le k \le n$

$$a_{nmk} := \int_{-1}^{1} (1-u^2)^{d/2} u^{k+m-d}\,\mathrm{d}u.$$

By the change of variables with $u = \cos\theta$, we can write

$$a_{nmk} = \int_0^\pi \sin^{d+1}\theta \cos^{k+m-d}\theta \, \mathrm{d}\theta.$$

Note that it is easy to compute that for $m, n > 0$,

$$\int \sin^n\theta \cos^m\theta \, \mathrm{d}\theta = -\frac{\sin^{n-1}\theta \cos^{m+1}\theta}{n+m} + \frac{n-1}{n+m}\int \sin^{n-2}\theta \cos^m\theta \, \mathrm{d}\theta.$$

By recursively applying this relation, we obtain

$$\begin{aligned}
&\int \sin^n\theta \cos^m\theta \, \mathrm{d}\theta \\
&= -\cos^{m+1}\theta \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \frac{(n-1)(n-3)\cdots(n-2i-1)}{(n+m)(n+m-2)\cdots(n+m-2i)} \frac{\sin^{n-2i-1}\theta}{n+m-2i-1} \\
&\quad + \frac{(n-1)(n-3)\cdots(n-2\lfloor \frac{n}{2} \rfloor + 1)}{(n+m)(n+m-2)\cdots(n+m-2\lfloor \frac{n}{2} \rfloor + 2)} \int \sin^{n-2\lfloor \frac{n}{2} \rfloor}\theta \cos^m\theta \, \mathrm{d}\theta.
\end{aligned}$$

For the remaining integral: we can use either

$$\int \sin\theta \cos^m\theta \, \mathrm{d}\theta = -\frac{1}{m+1}\cos^{m+1}\theta + C$$

for $m \neq -1$, or

$$\int \cos^m\theta \, \mathrm{d}\theta = \frac{1}{m}\cos^{m-1}\theta \sin\theta + \frac{m-1}{m}\int \cos^{m-2}\theta \, \mathrm{d}\theta,$$

which yields

$$\begin{aligned}
\int \cos^m\theta \, \mathrm{d}\theta = \sin\theta \sum_{i=0}^{\lfloor \frac{m}{2} \rfloor} &\frac{(m-1)(m-3)\cdots(m-2i-1)}{m(m-2)\cdots(m-2i)} \frac{\cos^{m-2i-1}\theta}{m-2i-1} \\
&+ \frac{(m-1)(m-3)\cdots(m-2\lfloor \frac{m}{2} \rfloor + 1)}{m(m-2)\cdots(m-2\lfloor \frac{m}{2} \rfloor + 2)} \int \cos^{m-2\lfloor \frac{m}{2} \rfloor}\theta \, \mathrm{d}\theta.
\end{aligned}$$

For the integration over $[0, \pi]$,

$$\int_0^\pi \cos^m \theta \, \mathrm{d}\theta = \begin{cases} 0 & \text{if } m \text{ is odd} \\ \frac{(m-1)(m-3)\cdots 3\cdot 1}{m(m-2)\cdots 4\cdot 2}\pi & \text{if } m \text{ is even} \end{cases}$$

and

$$\int_0^\pi \sin\theta \cos^m \theta \, \mathrm{d}\theta = \begin{cases} 0 & \text{if } m \text{ is odd} \\ \frac{2}{m+1} & \text{if } m \text{ is even.} \end{cases}$$

After all, we have

$$\begin{aligned} &\int_0^\pi \sin^n \theta \cos^m \theta \, \mathrm{d}\theta \\ &= \begin{cases} \frac{(n-1)(n-3)\cdots 3\cdot 1}{(n+m)(n+m-2)\cdots(m+4)(m+2)} \frac{(m-1)(m-3)\cdots 3\cdot 1}{m(m-2)\cdots 4\cdot 2}\pi & \text{if } n \text{ is even and } m \text{ is even} \\ \frac{(n-1)(n-3)\cdots 3\cdot 1}{(n+m)(n+m-2)\cdots(m+4)(m+2)} \frac{2}{m+1} & \text{if } n \text{ is even and } m \text{ is even} \\ 0 & \text{if } m \text{ is odd.} \end{cases} \\ &= \begin{cases} \frac{(n-1)!(m-1)!}{(\frac{n}{2}-1)!(\frac{m}{2}-1)!(\frac{n+m}{2})!2^{n+m}}\pi & \text{if } n \text{ is even and } m \text{ is even} \\ \frac{(m-1)!(\frac{n+m-1}{2})!(\frac{n-1}{2})!}{(n+m)!(\frac{m}{2}-1)!}2^{n+1} & \text{if } n \text{ is even and } m \text{ is even} \\ 0 & \text{if } m \text{ is odd.} \end{cases} \end{aligned}$$

2.A.2 Compute $(\rho_{nm}, \varphi_{nm}(r))$

Now, we describe how to solve the eigenequation $(\boldsymbol{\kappa}_n \varphi_{nm})(r) = \rho_{nm}\varphi_{nm}(r)$. Since we have an infinite series $\kappa_n(r\tilde{r}) = \sum_{m=0}^\infty b_{nm} r^m \tilde{r}^m$ for each n, we first truncate this series with finite, say M, terms:

$$\tilde{\kappa}_n(r\tilde{r}) := \sum_{m=0}^M b_{nm} r^m \tilde{r}^m = \sum_{m=0}^M \sqrt{b_{nm}} r^m \sqrt{b_{nm}} \tilde{r}^m.$$

As we saw in the last lecture, the eigenspectrum of $\tilde{\boldsymbol{\kappa}}$ can be found by the eigendecomposition of the corresponding matrix $\tilde{\mathsf{k}}$ defined as

$$(\tilde{\mathsf{k}})_{ij} = \int_0^1 \sqrt{b_i}\tilde{r}^i \sqrt{b_j}\tilde{r}^j \tilde{r}^{d-1}\,\mathrm{d}\tilde{r} = \sqrt{b_i b_j}\int_0^1 \tilde{r}^{i+j+d-1}\,\mathrm{d}\tilde{r} = \frac{\sqrt{b_i b_j}}{i+j+d}.$$

Acknowledgement

Chapter 2, in part, is a reprint of the material in the paper with permission: © 2021 IEEE. J. Jon Ryu, Jiun-Ting Huang, and Young-Han Kim, "On the role of eigendecomposition in kernel embedding," In *Proceedings of IEEE International Symposium on Information Theory*, pp. 2030–2035, Melbourne, Australia, July 2021. The dissertation author was the primary investigator and author of this paper. This work was supported in part by the National Science Foundation under Grant CCF-1911238.

Bibliography

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396, 2003.

Yoshua Bengio, Olivier Delalleau, Nicolas Le Roux, Jean-François Paiement, Pascal Vincent, and Marie Ouimet. Learning eigenfunctions links spectral embedding and kernel pca. *Neural Comput.*, 16(10):2197–2219, 2004.

Marianna Bolla. *Spectral clustering and biclustering: Learning large graphs and contingency tables*. John Wiley & Sons, 2013.

Michael AA Cox and Trevor F Cox. Multidimensional scaling. In *Handbook of data visualization*, pages 315–347. Springer, 2008.

Costas Efthimiou and Christopher Frye. *Spherical harmonics in p dimensions*. World Scientific, 2014.

Benjamin Eltzner, Stephan Huckemann, and Kanti V Mardia. Torus principal component analysis with applications to RNA structure. *Ann. Appl. Statist.*, 12(2):1332–1359, 2018.

Gregory E Fasshauer. Positive definite kernels: past, present and future. *Dolomites Research Notes on Approximation*, 4:21–63, 2011.

Jihun Ham, Daniel D Lee, Sebastian Mika, and Bernhard Schölkopf. A kernel view of the dimensionality reduction of manifolds. In *Proc. Int. Conf. Mach. Learn.*, page 47, 2004.

John A Hartigan and Manchek A Wong. A k-means clustering algorithm. *J. R. Stat. Soc. Ser. C Appl. Stat*, 28(1):100–108, 1979.

Atsushi Higuchi. Symmetric tensor spherical harmonics on the n-sphere and their application to the de sitter group so (n, 1). *Journal of mathematical physics*, 28(7): 1553–1566, 1987.

Stuart Lloyd. Least squares quantization in PCM. *IEEE Trans. Inf. Theory*, 28(2):129–137,

1982.

James Mercer. Xvi. functions of positive and negative type, and their connection the theory of integral equations. *Philos. Trans. R. Soc. A*, 209(441-458):415–446, 1909.

Ha Quang Minh, Partha Niyogi, and Yuan Yao. Mercer's theorem, feature maps, and smoothing. In *Int. Conf. Comput. Learn. Theory*, pages 154–168. Springer, 2006.

Claus Müller. *Analysis of spherical symmetries in Euclidean spaces*, volume 129. Springer Science & Business Media, 2012.

Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Adv. Neural Inf. Proc. Syst.*, 14:849–856, 2001.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

Geoffrey Schiebinger, Martin J Wainwright, and Bin Yu. The geometry of kernelized spectral clustering. *Ann. Statist.*, 43(2):819–846, 2015.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10(5):1299–1319, 1998.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.

Alex J Smola, Zoltan L Ovari, and Robert C Williamson. Regularization with dot-product kernels. *Adv. Neural Inf. Proc. Syst.*, pages 308–314, 2001.

Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

Yair Weiss. Segmentation using eigenvectors: a unifying view. In *IEEE Int. Conf. Comp. Vis.*, volume 2, pages 975–982. IEEE, 1999.

Part II

Nearest-Neighbors Methods

Chapter 3

Classification and Regression with One-Nearest Neighbors

3.1 Introduction

Arguably being the most primitive, yet powerful nonparametric approaches for various statistical problems, the k-nearest-neighbor (k-NN) based algorithms have been one of the essential toolkits in data science since their inception. They have been extensively studied and analyzed over several decades for canonical statistical procedures including classification (Cover and Hart, 1967; Fix and Hodges, 1951), regression (Cover, 1968a,b), density estimation (Fukunaga and Hostetler, 1973; Loftsgaarden and Quesenberry, 1965; Mack and Rosenblatt, 1979), and density functional estimation (Kozachenko and Leonenko, 1987; Leonenko et al., 2008). They are attractive even in this modern age due to their simplicity, decent performance, and rich understanding of their statistical properties.

There exist, however, clear limitations that hinder their wider deployment in practice. First, and most importantly, standard k-NN based algorithms are often deemed to be inherently infeasible for large-scale data, as they need to store and process the entire data in a single machine for NN search. Second, though the number of neighbors k needs to grow to infinity in the sample size to achieve statistical consistency in general for such procedures (Biau and Devroye, 2015), small k is highly preferred in practice to

avoid possibly demanding time complexity of large-k-NN search; see Section 3.3.1 for an in-depth discussion.

Recently, specifically for regression and classification, a few ensemble based methods (Duan et al., 2020; Qiao et al., 2019; Xue and Kpotufe, 2018) have been proposed aiming to reduce the computational complexity while achieving the accuracy of the optimal standard k-NN regression and classification rules; however, theoretical guarantees of those solutions still require large-k-NN search. Xue and Kpotufe (2018) proposed an idea dubbed as *denoising*, which is to draw (multiple) subsample(s) and preprocess them with the standard large-k-NN rule *over the entire data* in the training phase, so that the k-NN information can be hashed effectively by 1-NN searches in the testing phase. Though the resulting algorithm is provably optimal with a small statistical overhead, the denoising step may still suffer prohibitively large complexity for large N and/or large k in principle. More recently, to address the computational and storage complexity of the standard k-NN classifier with large N, Qiao et al. (2019) proposed the *bigNN classifier*, which splits data into subsets, applies the standard k-NN classifier to each, and aggregates the labels by a majority vote. This ensemble method works without any coordination among data splits, and thus they naturally fit to large-scale data which may be inherently stored and processed in distributed machines. However, they showed its minimax optimality only when both the number of splits M and the base k increase as the sample size N increases but only a strictly suboptimal guarantee for fixed k. Only with the optimality for increasingly large k, they suggested to use the bigNN classifier in the preprocessing phase of the denoising framework. A more recent work (Duan et al., 2020) on optimally weighted version of the bigNN classifier still assumes increasingly large k.

In this paper, we complete the missing theory for small k and show that the bigNN classifier with $k = 1$ suffices for minimax rate-optimal classification. More generally, we analyze a variant of the bigNN classifier, called the *M-split k-NN classifier*,

which is defined as the majority vote over the total kM nearest-neighbor labels obtained after running k-NN search over the M data splits. Roughly put, we show that the M-split k-NN classification rule behaves almost equivalently to the standard $\Theta(M)$-NN rules, for any fixed $k \geq 1$. In particular, the M-split 1-NN rule, equivalent to the bigNN classifier with $k = 1$, is shown to attain a minimax optimal rate up to logarithmic factors under smooth measure conditions. We also provide a minimax-rate-optimal guarantee for regression task with an analogously defined M-split k-NN regression rule.

Albeit both the algorithm and analysis are simple in nature, the practical implication of theoretical guarantees provided herein together with the divide-and-conquer framework is significant: while running faster than the standard 1-NN rules by processing smaller data with small-k-NN search in parallel, the M-split k-NN rules can achieve the same statistical guarantee of the *optimal* standard k-NN rules run over the entire dataset. Moreover, when deploying the rules in practice, we only need to tune the number of splits M while fixing k, say, simply $k = 1$. We experimentally demonstrate that the split 1-NN rules indeed perform on par with the optimal standard k-NN rules as expected by theory, while running faster than the standard 1-NN rules.

The key technique in our analysis is to analyze intermediate rules that selectively aggregates the small-k-NN estimates from each data split based on the k-th-NN distances from a query point. The intuition is that these intermediate rules which average only neighbors close enough to a query point exactly behave like a standard $\Theta(M)$-NN rule for any fixed k. We establish the performance of the (M, k)-NN rules by showing that its performance is approximated by the intermediate rules, with a small (logarithmic) approximation overhead in rates. Indeed, these intermediate rules attain exact minimax optimal rates for respective problems at the cost of additional complexity for ordering the NN distances.

Organization

The rest of this chapter is organized as follows. Section 3.2 presents the main results with the formal definition of the split NN rules and their theoretical guarantees. In Section 3.3, we discuss computational complexity of the standard k-NN algorithms and the M-split k-NN rules, a refined aggregation scheme that removes the logarithmic factors in the previous guarantees, and a comparison to the bigNN classifier and its theoretical guarantee of (Qiao et al., 2019). We demonstrate the convergence rates of the split NN rules and their practicality over the standard k-NN rules with experimental results in Section 3.4. Due to the space limit, we discuss other related work and present all proofs in Appendix.

3.2 Main Results

Let $(\mathcal{X}, \rho)$ be a metric space and let $\mathcal{Y}$ be the outcome (or label) space, i.e., $\mathcal{Y} \subseteq \mathbb{R}$ for regression and $\mathcal{Y} = \{0, 1\}$ for binary classification. We denote by P a joint distribution over $\mathcal{X} \times \mathcal{Y}$, by μ the marginal distribution on $\mathcal{X}$, and by η the regression function $\eta(x) = \mathbb{E}[Y|X = x]$.

We denote an open ball of radius r centered at $x \in \mathcal{X}$ by $B^o(x, r) := \{x' \in \mathcal{X} \colon \rho(x, x') < r\}$ and the closed ball by $B(x, r) := \overline{B^o(x, r)}$. The support of a measure μ is denoted as $\text{supp}(\mu) := \{x \in \mathcal{X} \colon \mu(B^o(x, r)) > 0,\ \forall r > 0\}$.

Given sample $\mathcal{D} = (\mathbf{X}, \mathbf{Y}) = \{(X_i, Y_i)\}_{i=1}^N$ and a point $x \in \mathbb{R}^d$, we use $X_{(k)}(x; \mathbf{X})$ to denote the k-th-nearest neighbor of x from the sample instances $\mathbf{X} = X_{1:N}$ and use $Y_{(k)}(x; \mathcal{D})$ to denote the corresponding k-th-NN label among $\mathbf{Y} = Y_{1:N}$; any tie is broken arbitrarily. The k-th-NN distance of x is denoted as $r_k(x; \mathbf{X}) := \rho(x, X_{(k)}(x; \mathbf{X}))$ for $k \leq N$. We will omit the underlying data $\mathcal{D}$ or $\mathbf{X}$ whenever it is clear from the context.

Throughout in this chapter, we use N, M, and $n = N/M$ to denote the size of the entire data $\mathcal{D}$, the number of data splits, and the size of each data split, respectively,

assuming that M divides N.

3.2.1 Regression

Problem Setting

Given paired data $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^N$ drawn independently from P, the goal of regression is to design an estimator $\hat{\eta} = \hat{\eta}(\cdot; \mathcal{D})\colon \mathcal{X} \to \mathcal{Y}$ based on the data such that the estimate $\hat{\eta}(x)$ is *close* to the conditional expectation $\eta(x) = \mathbb{E}[Y|X = x]$, where the closeness between η and $\hat{\eta}$ is typically measured by the l_p-norm under μ, $\|\hat{\eta} - \eta\|_p := (\int |\hat{\eta}(x) - \eta(x)|^p \mu(\mathrm{d}x))^{1/p}$ for $p = 1, 2$, or the sup norm $\|\hat{\eta} - \eta\|_\infty := \sup_{x \in \mathcal{X}} |\hat{\eta}(x) - \eta(x)|$.

The Proposed Rule

Given a query $x \in \mathcal{X}$, we first recall that the *standard k-NN regression rule* outputs the average of the k-NN labels, i.e.,

$$\hat{\eta}^{(k)}(x; \mathcal{D}) := \frac{1}{k} \sum_{i=1}^{k} Y_{(i)}(x; \mathcal{D}).$$

Instead of running k-NN search over the entire data, given the number of splits $M \geq 1$, we first split the data $\mathcal{D}$ of size N into M subsets of equal size at random. Let $\mathcal{P} = \{\mathcal{D}_1, \ldots, \mathcal{D}_M\}$ denote the random subsets, where $\mathcal{D}_m$ corresponds to the m-th split. After finding k-NN labels for each data split, the *M-split k-NN (or (M, k)-NN in short) regression rule* is defined as the average of all kM of returned labels, i.e.,

$$\tilde{\eta}_M^{(k)}(x) := \tilde{\eta}^{(k)}(x; \mathcal{P}) := \frac{1}{M} \sum_{m=1}^{M} \hat{\eta}^{(k)}(x; \mathcal{D}_m). \tag{3.1}$$

Performance Guarantees

We claim that the proposed (M, k)-NN regression rule for any fixed $k \geq 1$ is nearly optimal in terms of error rate under a standard regularity condition. For a formal

statement, we borrow some standard assumptions on the metric measure space in the literature on analyzing k-NN algorithms (Dasgupta and Kpotufe, 2019).

Assumption 3.2.1 (Doubling and homogeneous measure). The measure μ on metric space $(\mathcal{X}, \rho)$ is *doubling with exponent* d, i.e., for any $x \in \mathrm{supp}(\mu)$ and $r > 0$,

$$\mu(B^o(x, r)) \leq 2^d \mu(B^o(x, r/2)).$$

The measure μ is (C_d, d)-*homogeneous*, i.e., for some $C_d > 0$ for any $x \in \mathrm{supp}(\mu)$ and $r > 0$,

$$\mu(B^o(x, r)) \geq C_d r^d \wedge 1.$$

Note that a measure μ is homogeneous if μ is doubling and $\mathrm{supp}(\mu)$ is bounded. The doubling exponent d can be interpreted as an intrinsic dimension of a measure space.

Assumption 3.2.2 (Hölder continuity). The conditional expectation function $\eta(x) = \mathbb{E}[Y|X = x]$ is (α_{H}, A)-*Hölder continuous* for some $0 < \alpha_{\mathrm{H}} \leq 1$ and $A > 0$ in metric space $(\mathcal{X}, \rho)$, i.e., for any $x, x' \in \mathcal{X}$,

$$|\eta(x) - \eta(x')| \leq A\rho^{\alpha_{\mathrm{H}}}(x, x').$$

Assumption 3.2.3 (Bounded conditional expectation and variance). The conditional expectation function $\eta(x) = \mathbb{E}[Y|X = x]$ and the conditional variance function $v(x) := \mathrm{Var}(Y|X = x)$ are bounded, i.e., $\sup_{x \in \mathcal{X}} |\eta(x)| < \infty$ and $\sup_{x \in \mathcal{X}} v(x) < \infty$.

The following condition is borrowed from (Xue and Kpotufe, 2018) to establish a high-probability bound.

Assumption 3.2.4. The collection of closed balls in $\mathcal{X}$ has finite VC dimension $\mathcal{V}$ and the outcome space $\mathcal{Y} \subset \mathbb{R}$ is contained in a bounded interval of length l_Y.

The main goal of this work is to demonstrate that the distributed (M, k)-NN rules can attain almost statistically equivalent performance to the optimal k-NN rules. Hence, our statements in what follows are written in parallel to the known results for the standard k-NN rules, to which we include the pointers after cf. for the interested reader. For example, the following statement is new and we refer to (Dasgupta and Kpotufe, 2019) for an analogous statement for the standard k-NN regression algorithm.

Theorem 3.2.1 (cf. (Dasgupta and Kpotufe, 2019, Theorem 1.3))**.** *Suppose that Assumptions 3.2.1 and 3.2.2 hold. Let $k \geq 1$ be fixed.*

(a) If Assumption 3.2.3 holds and the support of μ is bounded, for any $M \leq N$ such that $N/M \geq k$, we have

$$\mathbb{E}_{\mathcal{P}} \|\tilde{\eta}_M^{(k)} - \eta\|_2 \leq C_1 \Big(\Big(\frac{M}{N} \log \frac{M}{(\log M)^{1.01}} \Big)^{\frac{\alpha_{\mathrm{H}}}{d}} + \sqrt{\frac{(\log M)^{1.01}}{M}} \Big).$$

(b) If Assumption 3.2.4 holds, for any $0 < \delta < 1$, if $M \geq 16 \log \frac{1}{\delta}$, then with probability at least $1 - \delta$ over $\mathcal{P}$, we have

$$\|\tilde{\eta}_M^{(k)} - \eta\|_\infty \leq C_2 \Big(\Big(\frac{M}{N} \log N \Big)^{\frac{\alpha_{\mathrm{H}}}{d}} + \sqrt{\frac{1}{M} \log \frac{N}{\delta}} \Big).$$

In particular, C_1 and C_2 are constants and independent of the ambient dimension D.

Remark 3.2.2 (Minimax optimality)**.** If we set $M = \tilde{\Theta}(N^{2\alpha_{\mathrm{H}}/(2\alpha_{\mathrm{H}}+d)})$, Theorem 3.2.1 gives

$$\begin{aligned} \mathbb{E}_{\mathcal{P}} \|\tilde{\eta}_M^{(k)} - \eta\|_2 &= \tilde{O}(N^{-\alpha_{\mathrm{H}}/(2\alpha_{\mathrm{H}}+d)}) \text{ and} \\ \|\tilde{\eta}_M^{(k)} - \eta\|_\infty &= \tilde{O}(N^{-\alpha_{\mathrm{H}}/(2\alpha_{\mathrm{H}}+d)}) \text{ with high probability}, \end{aligned}$$

where $\tilde{O}(\cdot)$ hides any logarithmic multiplicative terms. This rate is known to be minimax optimal under the Hölder continuity of order α_{H}; for the standard k-NN regression algorithm, this rate optimality is attained for $k = \Theta(N^{2\alpha_{\mathrm{H}}/(2\alpha_{\mathrm{H}}+d)})$ (Dasgupta and Kpotufe,

2019; Xue and Kpotufe, 2018). In this view, the (M, k)-NN regression algorithm attains the performance of the standard $\Theta(M)$-NN regression algorithm for any fixed k.

3.2.2 Classification

Problem Setting

We consider the binary classification with $\mathcal{Y} = \{0, 1\}$. Given paired data $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^N$ drawn independently from P, the goal of binary classification is to design a (data-dependent) classifier $\hat{g}(\cdot; \mathcal{D})\colon \mathcal{X} \to \mathcal{Y}$ such that it minimizes the classification error $\mathsf{P}(\hat{g}(X; \mathcal{D}) \neq Y)$. For a classifier $\hat{g}\colon \mathcal{X} \to \mathcal{Y}$, we define its *pointwise risk* at $x \in \mathcal{X}$ as $R(\hat{g}; x) := \mathsf{P}(Y \neq \hat{g}(x) | X = x)$, and define the *(expected) risk* as $R(\hat{g}) := \mathsf{P}(Y \neq \hat{g}(X))$. Let $g(x)$ denote the *Bayes-optimal* classifier, i.e., $g(x) := 1\{\eta(x) \geq 1/2\}$ for all $x \in \mathcal{X}$, and let $R^*(x) := R(g; x) = \eta(x) \wedge (1 - \eta(x))$ and $R^* := R(g)$ denote the *pointwise-Bayes risk* and the *(expected) Bayes risk*, respectively. The canonical performance measure of a classifier $\hat{g}$ is its *excess risk* $R(\hat{g}) - R^*$.

Another important performance criterion is the *classification instability* proposed by (Sun et al., 2016), which quantifies a stablility of a classification procedure with respect to independent realizations of training data. Given $N \in \mathbb{N}$, with a slight abuse of notation, denote $\hat{g}$ as a classification procedure $\mathcal{D} \mapsto \hat{g}(\cdot; \mathcal{D})$ that maps a dataset $\mathcal{D}$ of size N to a classifier $\hat{g}(\cdot; \mathcal{D})$. The classification instability of the classification procedure is defined as

$$\mathrm{CIS}_N(\hat{g}) := \mathbb{E}[\mathsf{P}(\hat{g}(X; \mathcal{D}) \neq \hat{g}(X; \mathcal{D}') | \mathcal{D}, \mathcal{D}')],$$

where $\mathcal{D}$ and $\mathcal{D}'$ are independent, i.i.d. data of size N.

The Proposed Rule

The *standard k-NN classifier* is defined as the plug-in classifier of the standard k-NN regression estimate:

$$\hat{g}_k(x; \mathcal{D}) := 1\Big(\hat{\eta}^{(k)}(x; \mathcal{D}) \geq \frac{1}{2}\Big).$$

It can be equivalently viewed as the majority vote over the k-NN labels given a query.

Similarly, we define the *(M, k)-NN classification rule* as the plug-in classifier of the (M, k)-NN regression rule:

$$\tilde{g}_M^{(k)}(x) := \tilde{g}^{(k)}(x; \mathcal{P}) := 1\Big(\tilde{\eta}^{(k)}(x; \mathcal{P}) \geq \frac{1}{2}\Big).$$

Performance Guarantees

As shown in the previous section for regression, we can show that the proposed (M, k)-NN classifier behaves nearly identically to the standard $\Theta(M)$-NN rules for any fixed $k \geq 1$. Here, we focus on guarantees on rates of excess risk and classification instability, but the asymptotic Bayes consistency can be also established under a mild condition; see Theorem 3.B.13 in Appendix.

To establish rates of convergence for classification, we recall the following notion of smoothness for the conditional probability $\eta(x) = \mathsf{P}(Y = 1|X = x)$ defined in (Chaudhuri and Dasgupta, 2014) that takes into account the underlying measure μ to better capture the nature of classification than the standard Hölder continuity in Assumption 3.2.2.

Assumption 3.2.5 (Smoothness). For $\alpha \in (0, 1]$ and $A > 0$, $\eta(x)$ is *(α, A)-smooth in metric measure space* $(\mathcal{X}, \rho, \mu)$, i.e., for all $x \in \text{supp}(\mu)$ and $r > 0$,

$$|\eta(B(x, r)) - \eta(x)| \leq A\mu^{\alpha}(B^o(x, r)).$$

The following condition on the behavior of the measure μ around the decision boundary of η is a standard assumption to establish a fast rate of convergence (Audibert et al., 2007).

Assumption 3.2.6 (Margin condition). For $\beta \geq 0$, η satisfies the *β-margin condition* in $(\mathcal{X}, \rho, \mu)$, i.e., there exists a constant $C > 0$ such that

$$\mu(\partial\eta_\Delta) \leq C\Delta^\beta,$$

where $\partial\eta_\Delta := \{x \in \mathrm{supp}(\mu) \colon |\eta(x) - 1/2| \leq \Delta\}$ denotes the decision boundary with margin $\Delta \in (0, 1/2]$.

The following statement is new.

Theorem 3.2.3 (cf. (Chaudhuri and Dasgupta, 2014, Theorem 4))**.** *Under Assumptions 3.2.5 and 3.2.6, the following statements hold for any fixed $k \geq 1$, where M_o, C_o, and C_o' are constants depending on k, α, β, and C.*

(a) Pick any $\delta \in (0,1)$ and $M_o > 0$ such that $M = M_o N^{\frac{2\alpha}{2\alpha+1}} (\log \frac{1}{\delta})^{\frac{1}{2\alpha+1}} \leq N$. With probability at least $1 - \delta$ over $\mathcal{P}$,

$$\mathsf{P}(\tilde{g}_M^{(k)}(x) \neq g(X) | \mathcal{P}) \leq \delta + C_o \Big(\frac{1}{N} \log \frac{1}{\delta} \Big)^{\frac{\beta\alpha}{2\alpha+1}} \Big(\log \frac{N}{\log \frac{1}{\delta}} \Big)^{\beta\alpha}.$$

(b) Pick any $M_o \in (0, N^{\frac{1}{2\alpha+1}}]$ and set $M = M_o N^{\frac{2\alpha}{2\alpha+1}} \leq N$. Then

$$\mathbb{E}_{\mathcal{P}}[R(\tilde{g}_M^{(k)})] - R^* \leq C_o' \Big(\frac{(\log N)^{\alpha \vee \frac{1}{2}}}{N^{\frac{\alpha}{2\alpha+1}}} \Big)^{\beta+1} \quad \textit{and}$$
$$\mathrm{CIS}_N(\tilde{g}_M^{(k)}) \leq C_o'' \Big(\frac{(\log N)^{\alpha \vee \frac{1}{2}}}{N^{\frac{\alpha}{2\alpha+1}}} \Big)^{\beta}.$$

Remark 3.2.4 (Minimax optimality)**.** Suppose that η is (α_{H}, A)-Hölder continuous and μ has a density with respect to Lebesgue measure that is strictly bounded away from zero

on its support. Then, by (Chaudhuri and Dasgupta, 2014, Lemma 2), η is $(\frac{\alpha_{\mathrm{H}}}{d}, A)$-smooth. Hence, if we set $M = \tilde{\Theta}(N^{2\alpha_{\mathrm{H}}/(2\alpha_{\mathrm{H}}+d)})$ in Theorem 3.2.3(b) as in Remark 3.2.2, we have

$$\mathbb{E}_{\mathcal{P}}[R(\tilde{g}_M^{(k)})] - R^* = \tilde{O}(N^{-(\beta+1)\alpha_{\mathrm{H}}/(2\alpha_{\mathrm{H}}+d)}) \quad \text{and}$$
$$\mathrm{CIS}_N(\tilde{g}_M^{(k)}) = \tilde{O}(N^{-\beta\alpha_{\mathrm{H}}/(2\alpha_{\mathrm{H}}+d)}),$$

which are known to be minimax optimal under the Hölder continuity assumption (Chaudhuri and Dasgupta, 2014; Sun et al., 2016). In parallel to Remark 3.2.2, the standard k-NN classifier is known to achieve these rates for $k = \Theta(N^{2\alpha_{\mathrm{H}}/(2\alpha_{\mathrm{H}}+d)})$, and thus the (M, k)-NN classifier attains the performance of a standard $\Theta(M)$-NN classifier in this sense.

Remark 3.2.5 (Reduction to regression)**.** For a regression estimate $\hat{\eta}$, let $\hat{g}$ be the plug-in classifier with respect to $\hat{\eta}$. Then, via the inequality

$$R(\hat{g}) - R^* \leq 2\|\hat{\eta} - \eta\|_1,$$

the guarantees for the (M, k)-NN regression rule in Theorem 3.2.1 readily imply convergence rates of the excess risk (Dasgupta and Kpotufe, 2019) even for a multiclass classification scenario, by adapting the guarantee for a multivariate regression setting. The current statements, however, are more general results for binary classification that apply to beyond smooth distributions, following the analysis by Chaudhuri and Dasgupta (2014).

3.3 Discussion

3.3.1 Computational Complexity

The standard k-NN rules are known to be asymptotically consistent only if $k \to \infty$ as $N \to \infty$. Specifically to attain minimax rate-optimality, $k = \Theta(N^{2\alpha_{\mathrm{H}}/(2\alpha_{\mathrm{H}}+d)})$

is required under measures are Hölder continuity of order α_{H}; see Remarks 3.2.2 and 3.2.4. As alluded to earlier, this large-k requirement on the standard k-NN rules for statistical optimality may be problematic in practice. The main claim of this work is that the M-split 1-NN rules replace the large-k requirement of the standard k-NN rules with a large-M requirement without almost no loss in the statistical performance, while providing a natural, distributed solution to large-scale data with a possible speed-up via parallel computation.

To examine the complexity more carefully, consider Euclidean space $\mathbb{R}^d$ for a moment. Let $T_{\mathrm{NN}}(k, N)$ denote the test-time complexity of a k-NN search algorithm for data of size N. The simplest baseline NN search algorithm is the brute-force search, which has time complexity $T_{\mathrm{NN}}(k, N) = O(Nd)$ regardless of k.[1] For extremely large-scale data, however, even $O(N)$ may be unwieldy in practice. To reduce the complexity, several alternative data structures specialized for NN search such as KD-Trees (Bentley, 1975) for Euclidean data, and Metric Trees (Uhlmann, 1991) and Cover Trees (Beygelzimer et al., 2006) for non-Euclidean data have been developed; see (Dasgupta and Kpotufe, 2019; Kibriya and Frank, 2007) for an overview and comparison of empirical performance of these specialized data structures for k-NN search. These are preferred over the brute-force search for better test time complexity $O(\log N)$ in a moderate size of dimension, say $d \leq 10$, but for much higher-dimensional data, it is known that the brute-force search may be faster. In particular, the most popular choice of a KD-Tree based search algorithm has time complexity $T_{\mathrm{NN}}(1, N) = O(2^d \log N)$ for $k = 1$. The time complexity of exact k-NN search is $T_{\mathrm{NN}}(k, N) = O(k)T_{\mathrm{NN}}(1, N)$ for moderately small k,[2] but for a large k the time complexity could be worse than $O(k)T_{\mathrm{NN}}(1, N)$.

[1] Given a query point, (1) compute the distances from the dataset to the query ($O(Nd)$); (2) find the k-NN distance using introselect algorithm ($O(N)$), (3) pick the k-nearest neighbors; ($O(N)$).

[2] One possible implementation of exact k-NN search algorithm with KD-tree is to remove already found points and repeatedly find 1-NN points until k-NN points are found using KD-tree-based 1-NN search; after the search, the removed points may be reinserted into the KD-tree without affecting the overall complexity for a moderate size of k.

Thanks to the fully distributed nature, the (M, k)-NN classifier have computational advantage over the standard $\Theta(kM)$-NN classifier of nearly same statistical power run over the entire data. Suppose that we split data into M groups of equal size $\lceil \frac{N}{M} \rceil$ and they can be processed by S parallel processors, where each processor ideally manages $\lceil \frac{M}{S} \rceil$ data splits. Given the time complexity $T_{\mathrm{NN}}(k, N)$ of a base k-NN search algorithm, the (M, k)-NN algorithms have time complexity

$$T_{M;S}(k, N) = \left\lceil \frac{M}{S} \right\rceil T_{\mathrm{NN}}\Big(k, \left\lceil \frac{N}{M} \right\rceil\Big).$$

As stated in Section 3.2, the (M, k)-NN rules with $S \leq M$ parallel units may attain the performance of the standard $\Theta(kM)$-NN rules in a single machine with the relative speedup of

$$\frac{T_{M;S}(k, N)}{T_{\mathrm{NN}}(kM, N)} \sim \frac{1}{S}$$

with a brute-force search, and

$$\frac{T_{M;S}(k, N)}{T_{\mathrm{NN}}(kM, N)} \sim \frac{\frac{kM}{S} \log \frac{N}{M}}{kM \log N} = \frac{1}{S}\Big(1 - \frac{\log M}{\log N}\Big)$$

with a KD-Tree based search algorithm assuming $T_{\mathrm{NN}}(k, N) = O(k \log N)$ for simplicity. Hence, the most benefit of the proposed algorithms comes from their distributed nature which reduces both time and storage complexity.

3.3.2 A Refined Aggregation Scheme

As alluded to earlier, we can remove the logarithmic factors in the guarantees of Theorems 3.2.1 and 3.2.3 with a refined aggregation scheme which we call the *distance-selective aggregation*. With an additional hyperparameter $L \in \mathbb{N}$ such that $1 \leq L \leq M$, we take L estimates out of the total M values based on the k-th-NN distances from the query point to each data split instances. Formally, if $m_1, \ldots, m_L$ denote the L-smallest

values out of the $(k+1)$-th-NN distances $(r_{k+1}(x;\mathbf{X}_m))_{m=1}^M$, we take the partial average of the corresponding regression estimates:

$$\breve{\eta}_{M,L}^{(k)}(x) := \breve{\eta}_L^{(k)}(x;\mathcal{P}) := \frac{1}{L}\sum_{j=1}^{L}\hat{\eta}^{(k)}(x;\mathcal{D}_{m_j}). \tag{3.2}$$

We call the resulting rule the *M-split L-selective k-NN (or (M,L,k)-NN in short) regression rule* and analogously define the *(M,L,k)-NN classifier* $\breve{g}_{M,L}^{(k)}(x)$ as the plug-in classifier, i.e.,

$$\breve{g}_{M,L}^{(k)}(x) := 1\Big(\breve{\eta}_{M,L}^{(k)}(x) \geq \frac{1}{2}\Big). \tag{3.3}$$

Intuitively, it is designed to filter out some possible *outliers* based on the $(k+1)$-th-NN distances, since a larger $(k+1)$-th-NN distance to the query point likely indicates that the returned estimate from the corresponding group is more unreliable.[3]

The refined schemes are indeed minimax rate-optimal without the extra logarithmic factors, as stated in the following statements. We omit their proofs since they can be easily obtained from a straightforward modification of those of Theorems 3.2.1 and 3.2.3.

Proposition 3.3.1 (Regression)**.** *Under Assumptions 3.2.1, 3.2.2, and 3.2.3, for any fixed $k \geq 1$, any $L < M \leq N$ such that $N/M \geq k$,*

$$\mathbb{E}_{\mathcal{P}}\|\breve{\eta}_{M,L}^{(k)} - \eta\|_2 = O\Big(\Big(\frac{M}{N}\Big)^{\frac{\alpha_{\mathrm{H}}}{d}} + \sqrt{\frac{1}{M}}\Big).$$

Proposition 3.3.2 (Classification)**.** *Under Assumptions 3.2.5 and 3.2.6, if we set $M = M_o N^{\frac{2\alpha}{2\alpha+1}}$ and $L = \lceil (1-\kappa)M \rceil$ for any fixed $\kappa \in (0,1)$,*

$$\mathbb{E}_{\mathcal{P}}[R(\breve{g}_{M,L}^{(k)})] - R^* = O(N^{-\frac{\alpha(\beta+1)}{2\alpha+1}}) \quad \textit{and} \quad \mathrm{CIS}_N(\breve{g}_{M,L}^{(k)}) = O(N^{-\frac{\alpha\beta}{2\alpha+1}}).$$

[3]We use the $(k+1)$-th-NN distance instead of k-th-NN distance due to a technical reason for classification; see Lemma 3.B.8 in Appendix. For regression, our analysis remains valid for the k-th-NN distance.

3.3.3 Comparison to the bigNN classifier (Qiao et al., 2019)

The bigNN classifier proposed by Qiao et al. (2019) takes the majority vote over the M labels each of which is the output of the standard k-NN classifier from each data split. Formally, it is defined as $\hat{g}^{(k)}_{\mathsf{big}}(x;\mathcal{P}) := 1(\hat{\eta}^{(k)}_{\mathsf{big}}(x;\mathcal{P}) \geq 1/2)$, where $\hat{\eta}^{(k)}_{\mathsf{big}}(x;\mathcal{P}) := \frac{1}{M}\sum_{m=1}^{M} 1(\hat{\eta}^{(k)}(x;\mathcal{D}_m) \geq \frac{1}{2})$. Qiao et al. (2019) showed that the bigNN classifier is minimax rate-optimal, provided that k grows to infinity.

Theorem 3.3.3 ((Qiao et al., 2019, Theorems 1 and 2, rephrased)). *Assume Assumptions 3.2.5 and 3.2.6. Set $M = N^{\gamma}$ for some constant $\gamma \in (0, \frac{2\alpha}{2\alpha+1})$ and set $k = k_o N^{\frac{2\alpha}{2\alpha+1}-\gamma}$ for some constant $k_o \geq 1$ such that $k \leq N$. Then, we have*

$$\mathbb{E}_{\mathcal{P}}[R(\hat{g}^{(k)}_{\mathsf{big}})] - R^* = O(N^{-\frac{\alpha(\beta+1)}{2\alpha+1}}) \quad \textit{and} \quad \mathrm{CIS}_N(\hat{g}^{(k)}_{\mathsf{big}}) = O(N^{-\frac{\alpha\beta}{2\alpha+1}}).$$

Further, if $k \geq 1$ is fixed, then for $M = N^{\gamma}$ with $\gamma \in (0, \frac{2\alpha}{2\alpha+1})$, we have[4]

$$\mathbb{E}_{\mathcal{P}}[R(\hat{g}^{(k)}_{\mathsf{big}})] - R^* = O(N^{-\frac{\gamma(\beta+1)}{2}}) \quad \textit{and} \quad \mathrm{CIS}_N(\hat{g}^{(k)}_{\mathsf{big}}) = O(N^{-\frac{\gamma\beta}{2}}).$$

Note that the number of splits $M = N^{\gamma}$ is restricted to be strictly slower than $\Theta(N^{2\alpha/(2\alpha+1)})$, which is the optimal choice for our analysis. Further, in the first part of the statement, k is set to grow to infinity as $N \to \infty$; the second part only guarantees strictly suboptimal rates for fixed k. Their analysis is based on the intuition is that the k-NN classification results from each subset of data become consistent as k *grows to infinity*, and thus taking majority vote over the consistent guesses will likely result in a consistent guess; hence, it inherently results in a suboptimal performance guarantee when k is fixed. This technique is also not readily applicable for analyzing a regression algorithm.

In contrast, in the current work, the (M, k)-NN classifier takes the majority over

[4]This part of the statement is only informally alluded to in the experiments section of Qiao et al. (2019).

all kM returned labels and we establish the (near) rate-optimality *for any fixed $k \geq 1$*, as long as M grows properly. This implies that the M sets of k-NN labels over subsets are almost statistically equivalent to $\Theta(M)$-NN labels over the entire data. Our analysis is based on the refined aggregation scheme discussed in the previous section, which provides a careful control on the behavior of distributed nearest neighbors and is naturally compatible with the analysis of the regression rule. We remark, however, that the bigNN rule and the (M, k)-NN classifier become equivalent for the most practical case of $k = 1$, and both schemes also showed similar performance for small k's in our experiments (data not shown). Therefore, the key contribution is in our analysis rather in the algorithmic details.

3.4 Experiments

The goal of experiments in this section is twofold. First, we present simulated convergence rates of the (M, k)-NN rules for small k, say $k \in \{1, 3\}$, are polynomial as predicted by theory with synthetic dataset. Second, we demonstrate that their practical performance is competitive against that of the standard k-NN rules with real-world datasets, while generally reducing both validation complexity for model selection and test complexity. In both experiments, we also show the performance of the $(M, \frac{M}{2}, k)$-NN

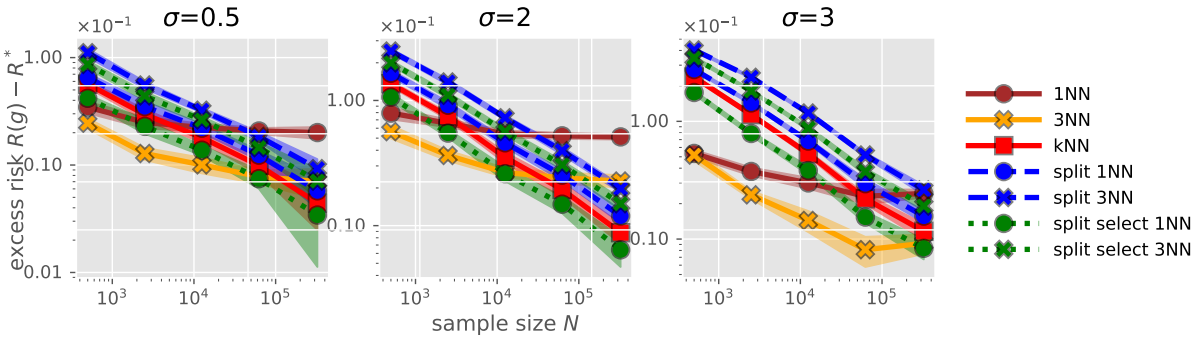


Figure 3.3.1. Summary of excess risks from the mixture of two Gaussians experiments.

rules[5] to examine the effect of the distance-selective aggregation.

Computing resources For each experiment, we used a single machine with one of the following CPUs: (1) Intel(R) Core(TM) i7-9750H CPU 2.60GHz with 12 (logical) cores or (2) Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz with 28 (logical) cores.

Implementation All implementations were based on Python 3.8 and we used the NN search algorithms implemented in scikit-learn package (Pedregosa et al., 2011) ver. 0.24.1 and utilized the multiprocessors using the python standard package `multiprocessing`. The code for experiments can be found in Supplementary Material.

3.4.1 Simulated Dataset

We first evaluated the performance of the proposed classifier with a synthetic data following Qiao et al. (2019). We consider a mixture of two isotropic Gaussians of equal weight $\frac{1}{2}\mathcal{N}(\mathbf{0}, I_d) + \frac{1}{2}\mathcal{N}(\mathbf{1}, \sigma^2 I_d)$, where $\mathbf{1} := [1, \dots, 1]^T \in \mathbb{R}^d$ and $I_d \in \mathbb{R}^{d\times d}$ denotes the identity matrix. With $d = 5$, we tested 3 different values of $\sigma \in \{0.5, 2, 3\}$ with 5 different sample sizes $N \in \{500, 2500, 12500, 62500, 312500\}$. We evaluated the (M, k)-NN rule and $(M, \frac{M}{2}, k)$-NN rule for $k \in \{1, 3\}$ with $M = 10N^{2\alpha_{\mathrm{H}}/(2\alpha_{\mathrm{H}}+d)} = 10N^{2/7}$

[5] As alluded to earlier, we used k-th-NN distance in experiments for the distance-selective classification rule instead of $(k+1)$-th-NN distance for simplicity.

Table 3.4.1. Summary of experiments with benchmark datasets. YearPredictionMSD in the last row is a regression dataset. Recall that $(M, 1)$-NN is a shorthand for the M-split 1-NN rules. The values in the parentheses correspond to the $(M, \frac{M}{2}, 1)$-NN rules. The best values are highlighted in bold.

Dataset	Error (% for classification)			Test time (s)			Valid. time (s)	
	1-NN	k-NN	$(M,1)$-NN	1-NN	k-NN	$(M,1)$-NN	k-NN	$(M,1)$-NN
GISETTE (Guyon et al., 2004)	7.26 ±1.65	**4.54** ±0.93	**5.11** ±1.01 (**4.86** ±0.86)	6.13	**5.75**	6.79 (6.18)	**52**	262 (270)
w/ brute-force	-	-	-	0.30	**0.26**	1.20 (2.06)	**38**	200 (207)
HTRU2 (Lyon et al., 2016)	2.91 ±0.40	**2.18** ±0.44	**2.08** ±0.28 (**2.28** ±0.37)	0.18	0.18	**0.04** (**0.04**)	18	**8** (10)
Credit (Dua and Graff, 2019)	26.73 ±0.99	**18.68** ±1.01	**18.65** ±1.05 (**18.93** ±0.95)	0.85	1.2	**0.2** (**0.2**)	122	**25** (29)
MiniBooNE (Dua and Graff, 2019)	13.72 ±1.57	**10.63** ±0.76	**10.69** ±0.86 (**10.62** ±0.64)	1.68	2.42	**0.98** (**0.94**)	264	**88** (92)
SUSY (Baldi et al., 2014)	28.27 ±1.50	**20.32** ±1.04	**20.55** ±1.35 (**20.52** ±1.31)	32	35	**14** (**13**)	3041	**1338** (1362)
BNG(letter,1000,1) (Vanschoren et al., 2013)	46.13 ±1.18	**40.88** ±1.12	**41.53** ±1.04 (**40.72** ±0.78)	379	350	17 (**14**)	2868	**619** (959)
YearPredictionMSD (Dua and Graff, 2019)	7.22 ±0.34	**6.72** ±0.25	**6.79** ±0.22 (**6.75** ±0.27)	33	**31**	40 (34)	1616	431 (**412**)
w/ brute-force	-	-	-	15	18	**3.5** (**3.6**)	1529	**300** (336)

based on $\alpha_{\mathrm{H}} = 1$ and $d = 5$. For comparison, we also ran the standard k-NN algorithm with $k \in \{1, 3, 10N^{2/7}\}$. We repeated experiments with 10 different random seeds and reported the averages and standard deviations.

The excess risks are plotted in Figure 3.3.1. We note that the $(M, 1)$-NN classifier performs similarly to the baseline k-NN classifier across different values of σ, and the performance can be improved by the $(M, \frac{M}{2}, 1)$-NN classifier. This implies that discarding possibly noisy information in the aggregation could actually improve the performance of the ensemble classifier. Note also that the convergence of the excess risks of the standard M-NN classifier and the $(M, \{1, 3\})$-NN classifiers is polynomial (indicated by the straight lines), as predicted by theory.

3.4.2 Real-world Datasets

We evaluated the proposed rules with publicly available benchmark datasets from the UCI machine learning repository (Dua and Graff, 2019) and the OpenML repository (Vanschoren et al., 2013), which were also used in (Xue and Kpotufe, 2018) and (Qiao et al., 2019); see Table 3.C.1 in Appendix for size, feature dimensions, and the number of classes of each dataset. All data were standardized to have zero mean and unit variances.

We tested four algorithms. The first two algorithms are (1) the standard 1-NN rule and (2) the standard k-NN rule with 10-fold cross-validation (CV) over an exponential grid $k \in \mathcal{K} := \{2^l - 1 : 2 \leq l \leq \log_2(\min\{2^{10}, 1 + N_{\mathsf{train}}/25\})\}$, where N_{train} denotes the size of training data. The rest are (3) the $(M, 1)$-NN rule and (4) the $(M, \frac{M}{2}, 1)$-NN rule both with 10-fold CV over $M \in \mathcal{K}$. We repeated with 10 different random (0.95,0.05) train-test splits and evaluated first $\min\{N_{\mathsf{test}}, 1000\}$ points from the test data to reduce the simulation time. Table 3.4.1 summarizes the test errors, test times, and validation times.[6] The optimal $(M, 1)$-NN rules consistently performed as well as the optimal

[6]Here, we used a KD-Tree based NN search by default. Since, however, a KD-Tree based algorithm

standard k-NN rules, even running faster than the standard 1-NN rules in the test phase. We remark that the optimally tuned $(M, \frac{M}{2}, 1)$-NN rules (i.e., with the distance-selective aggregation) performed almost identical to the $(M, 1)$-NN rules, except slight error improvements observed in high-dimensional datasets {GISETTE, YearPredictionMSD}. We additionally include Figure 3.C.1 in Appendix which summarizes the validation error profiles from the 10-fold CV procedures.

3.5 Concluding Remarks

In this chapter, we established the near statistical optimality of the (M, k)-NN rules when k is fixed, which makes the sample-splitting-based NN rules more appealing for practical scenarios with large-scale data. We also removed the logarithmic factors by the distance-selective aggregation and exhibited some level of performance boost in experimental results; it is an open question whether the logarithmic factor is fundamental for the vanilla (M, k)-NN rules or can be removed by a tighter analysis. As supported by both theoretical guarantees and empirical supports, we believe that the (M, k)-NN rules, especially for $k = 1$, can be widely deployed in practical systems and deserve further study including an optimally weighted version of the classifier as studied in (Duan et al., 2020). It would be also interesting if the current divide-and-conquer framework can be modified to be universally consistent for any general metric space, whenever such a consistent rule exists (Györfi and Weiss, 2021; Hanneke et al., 2020).

suffers a curse of dimensionality (recall Section 3.3.1), we ran additional trials with a brute-force search for high-dimensional datasets {GISETTE, YearPredictionMSD}, whose feature dimensions are 5000 and 90, respectively, and report the time complexities in the subsequent rows.

Appendix

In Appendix, we discuss other related work (Appendix 3.A), prove the technical statements (Appendix 3.B), and present some extra information on experiments (Appendix 3.C).

3.A Other Related Work

The asymptotic-Bayes consistency and convergence rates of the k-NN classifier have been studied extensively in the last century (Cover, 1968a,b; Cover and Hart, 1967; Devroye et al., 1994; Fix and Hodges, 1951; Fritz, 1975; Gyorfi, 1981; Kulkarni and Posner, 1995; Wagner, 1971). More recent theoretical breakthroughs include a strongly consistent margin regularized 1-NN classifier (Kontorovich and Weiss, 2015), a universally consistent sample-compression based 1-NN classifier over a general metric space (Györfi and Weiss, 2021; Hanneke et al., 2020; Kontorovich et al., 2017), nonasymptotic analysis over Euclidean space (Gadat et al., 2016) and over a doubling space (Dasgupta and Kpotufe, 2014), optimal weighted schemes (Samworth, 2012), stability (Sun et al., 2016), robustness against adversarial attacks (Bhattacharjee and Chaudhuri, 2020; Wang et al., 2018), and optimal classification with a query-dependent k (Balsubramani et al., 2019). For NN-based regression (Cover, 1968a,b; Dasgupta and Kpotufe, 2014, 2019), we mostly extend the analysis techniques of (Dasgupta and Kpotufe, 2019; Xue and Kpotufe, 2018); we refer the interested reader to a recent survey of Chen et al. (2018) for more refined analyses. For a more comprehensive treatment

on the k-NN based procedures, see (Biau and Devroye, 2015; Devroye et al., 1996) and references therein.

The most closely related work is (Qiao et al., 2019) as mentioned above. In a similar spirit, Duan et al. (2020) analyzed a distributed version of the optimally weighted NN classifier of Samworth (2012). More recently, Liu et al. (2021) studied a distributed version of an adaptive NN classification rule of Balsubramani et al. (2019).

The idea of an ensemble predictor for enhancing statistical power of a base classifier has been long known and extensively studied; see, e.g., (Hastie et al., 2009) for an overview. Among many ensemble techniques, bagging (Breiman, 1996) and pasting (Breiman, 1999) are closely related to this work. The goal of bagging is, however, mostly to improve accuracy by reducing variance when the sample size is small and the bootstrapping step is computationally demanding in general; see (Biau et al., 2010; Hall and Samworth, 2005) for the properties of bagged 1-NN rules. The motivation and idea of pasting is similar to the split NN rules, but pasting iteratively evolves an ensemble classifier based on an estimated prediction error based on random subsampling rather than splitting samples. The split NN rules analyzed in this paper are non-iterative and NN-based-rules-specific, and assume essentially no additional processing step beyond splitting and averaging.

Beyond ensemble methods, there are other attempts to make NN based rules scalable based on quantization (Gottlieb et al., 2018; Hanneke et al., 2020; Kontorovich et al., 2017; Kpotufe and Verma, 2017; Xue and Kpotufe, 2018) or regularization (Kontorovich and Weiss, 2015), where the common theme there is to carefully select subsample and/or preprocess the labels. We remark, however, that they typically involve onerous and rather complex preprocessing steps, which may not be suitable for large-scale data. Approximate NN (ANN) search algorithms (Har-Peled et al., 2012; Indyk and Motwani, 1998; Slaney and Casey, 2008) are yet another practical solution to reduce the query complexity, but ANN-search-based rules such as (Alabduljalil et al., 2013; Anastasiu and

Karypis, 2019) hardly have any statistical guarantee (Dasgupta and Kpotufe, 2019) with few exception (Efremenko et al., 2020; Gottlieb et al., 2014). Gottlieb et al. (2014) proposed an ANN-based classifier for general doubling spaces with generalization bounds. More recently, Efremenko et al. (2020) proposed a locality sensitive hashing (Datar et al., 2004) based classifier with Bayes consistency but a strictly suboptimal rate guarantee in $\mathbb{R}^d$. In contrast, this paper focuses on *exact*-NN-search based algorithms.

We conclude with remarks on a seeming connection between the proposed distance-selective aggregation and the k-NN based outlier detection methods. Ramaswamy et al. (2000) and Angiulli and Pizzuti (2002) proposed to use the k-NN distance, or some basic statistics such as mean or median of the k-NN distances to a query point, as an outlier score; a recent paper (Gu et al., 2019) analyzed these schemes. In view of this line of work, the split-and-select NN rules can be understood as a selective ensemble of *inliers* based on the k-NN distances. It would be an interesting direction to investigate a NN-based outlier detection method for large-scale dataset, extending the idea of the distance-selective aggregation.

3.B Deferred Proofs

In this section, we provide the full proofs of the statements in the main text. For both regression and classification problems, the key idea in our analysis of the (M, k)-NN rules is to consider the (M, L, k)-NN rules (3.2) and (3.3) as a proof device. It relies on the observations that (1) the (M, k)-NN rules can be closely approximated to the $(M, \lceil \kappa M \rceil, k)$-NN rules with $\kappa \approx 1$, and (2) $(M, \lceil \kappa M \rceil, k)$-NN rules attain minimax optimality for any fixed k and fixed $\kappa \in (0, 1)$, as long as M is chosen properly.

This section is organized as follows. In Appendix 3.B.1, we state and prove a key technical lemma for analyzing the distributed NN rules. As the regression rules are easier to analyze, we prove Theorem 3.2.1 in Appendix 3.B.2. The proof of Theorem 3.2.3

is presented in Appendix 3.B.3, including an additional statement on Bayes consistency.

3.B.1 A Key Technical Lemma

We first restate a simple yet important observation on the k-nearest-neighbors by Chaudhuri and Dasgupta (2014) that the k-nearest neighbors of x lies in a ball of probability mass of $O(\frac{k}{n})$ centered at x, with high probability. We define the *probability radius* of mass p centered at $x \in \mathcal{X}$ as the minimum possible radius of a closed ball containing probability mass at least p, that is,

$$r_p(x) := \inf\{r > 0 \colon \mu(B(x, r)) \geq p\}.$$

Lemma 3.B.1 (Chaudhuri and Dasgupta, 2014, Lemma 8)**.** *Pick any $x \in \mathcal{X}$, $0 \leq p \leq 1$, $0 \leq \xi \leq 1$, and any positive integers n and k such that $k \leq \xi np$. If $X_1, \ldots, X_n$ are drawn i.i.d. from μ, then*

$$\mathsf{P}(r_{k+1}(x; X_{1:n}) > r_p(x)) \leq e^{-np(1-\xi)^2/2} \leq e^{-k(1-\xi)^2/(2\xi)}.$$

We now state an analogous version of the above lemma for our analysis of the (M, k)-NN rules. The following lemma quantifies that, with high probability (exponentially in M) over the split instances $\mathcal{P}_X$, the the k-nearest neighbors of x from the selected data splits based on the $(k+1)$-th-NN distances will likely lie within a small probability ball of mass $O(\frac{kM}{N})$ around the query point.

Lemma 3.B.2. *Pick any positive integer $k \geq 1$ and $\tau \in (0, 1]$, and set $L = \lceil (1-\tau)^2 M \rceil$. If the data splits $\mathbf{X}_1, \ldots, \mathbf{X}_M$ are independent, we have*

$$\mathsf{P}\Big(\max_{j \in [L]} r_{k+1}(x; \mathbf{X}_{m_j}) > r_p(x)\Big) \leq e^{-\frac{(1-\tau)\tau^2}{2} M}$$

for $n \geq k + \ln\frac{1}{\tau} + \sqrt{2k\ln\frac{1}{\tau} + (\ln\frac{1}{\tau})^2}$ *and* $p = \frac{1}{n}(k + \ln\frac{1}{\tau} + \sqrt{2k\ln\frac{1}{\tau} + (\ln\frac{1}{\tau})^2}) \in (0, 1]$.

Proof. Define

$$\xi = \frac{k}{k + \ln\frac{1}{\tau} + \sqrt{2k\ln\frac{1}{\tau} + (\ln\frac{1}{\tau})^2}}$$

so that we can write $p = \frac{k}{\xi n} = \frac{kM}{\xi N}$. Note that $\xi \in (0, 1]$ for any $k \geq 1$ and $\tau \in (0, 1]$.

For each data split indexed by $m \in [M]$, we define a *bad* event

$$E_m = \{r_{k+1}(x; \mathbf{X}_m) > r_p(x)\}.$$

Observe that E_m occurs if any only if the closed ball of probability mass p contains less than k points from $\mathbf{X}_m$. By Lemma 3.B.1, the probability of the bad event E_m is upper bounded by $e^{-k(1-\xi)^2/(2\xi)}$, which is equal to τ by the choice of ξ. Now, since the data splits are independent, $(1(E_m))_{m=1}^M$ is a sequence of independent Bernoulli random variables with parameter $\mathsf{P}(E_1) \leq \tau$. Hence, we have

$$\begin{aligned}\mathsf{P}\Big(\max_{j\in[L]} r_{k+1}(x; \mathbf{X}_{m_j}) > r_p(x)\Big) &\leq \mathsf{P}\Big(\sum_{m=1}^{M} 1(E_m) > M - L\Big) \\ &\leq \mathsf{P}(B_{M,1-\tau} < (1-\tau)^2 M),\end{aligned}$$

where $B_{M,\tau} \sim \mathrm{Binom}(M, \tau)$ denotes a binomial random variable with parameters M and τ. Another application of the multiplicative Chernoff bound to the right-hand side concludes the desired bound. □

3.B.2 Regression: Proof of Theorem 3.2.1

Proof of Theorem 3.2.1(a)

This analysis extends the proof of (Dasgupta and Kpotufe, 2019, Theorem 1.3). Let $\mathcal{P}_X := \{\mathbf{X}_m\}_{m=1}^M$ denote the set of splits of $\mathbf{X}$. We let $V := \sup_{x\in\mathcal{X}} v(x) < \infty$ and $H := \sup_{x\in\mathcal{X}} |\eta(x)| < \infty$. Since the support of μ is bounded, we let $R := \mathrm{diam}(\mathrm{supp}(\mu)) < \infty$.

Step 1. Error decomposition

Recall that we wish to bound

$$\mathbb{E}_{\mathcal{P}}\|\tilde{\eta}_M^{(k)} - \eta\|_2 = \mathbb{E}_{\mathcal{P}}\sqrt{\mathbb{E}_X[(\tilde{\eta}^{(k)}(X;\mathcal{P}) - \eta(X))^2]} \\ \leq \sqrt{\mathbb{E}_{\mathcal{P}}\mathbb{E}_X[(\tilde{\eta}^{(k)}(X;\mathcal{P}) - \eta(X))^2]}.$$

Here, the inequality follows by Jensen's inequality. We will consider the (M, L, k)-NN regression rule with $L = \lceil (1-\tau)^2 M \rceil$ as a proof device, where τ is to be determined at the end of the proof. Pick any $x \in \mathcal{X}$. We denote the conditional expectation of the (M, L, k)-NN regression estimate $\breve{\eta}_L^{(k)}(x;\mathcal{P})$ by

$$\overline{\eta}_L^{(k)}(x;\mathcal{P}_X) := \mathbb{E}_{\mathbf{Y}|\mathcal{P}_X}[\breve{\eta}_L^{(k)}(x;\mathcal{P})] = \frac{1}{kL}\sum_{j=1}^{L}\sum_{i=1}^{k}\eta(X_{(i)}(x;\mathbf{X}_{m_j})),$$

where the expectation is over Y-values $\mathbf{Y}$ given the data splits $\mathcal{P}_X$. Note that with $L = M$, $\overline{\eta}_M^{(k)}(x;\mathcal{P}_X)$ becomes the conditional expectation of the (M, k)-NN regression estimate $\tilde{\eta}^{(k)}(x;\mathcal{P})$. We decompose the squared error $(\tilde{\eta}^{(k)}(x;\mathcal{P}) - \eta(x))^2$ as

$$\begin{aligned}
&(\tilde{\eta}^{(k)}(x;\mathcal{P}) - \eta(x))^2 \\
&= \Big(\tilde{\eta}^{(k)}(x;\mathcal{P}) - \overline{\eta}_M^{(k)}(x;\mathcal{P}_X) + \overline{\eta}_M^{(k)}(x;\mathcal{P}_X) - \overline{\eta}_L^{(k)}(x;\mathcal{P}_X) + \overline{\eta}_L^{(k)}(x;\mathcal{P}_X) - \eta(x)\Big)^2 \\
&\leq 3\Big\{(\tilde{\eta}^{(k)}(x;\mathcal{P}) - \overline{\eta}_M^{(k)}(x;\mathcal{P}_X))^2 + (\overline{\eta}_M^{(k)}(x;\mathcal{P}_X) - \overline{\eta}_L^{(k)}(x;\mathcal{P}_X))^2 + (\overline{\eta}_L^{(k)}(x;\mathcal{P}_X) - \eta(x))^2\Big\},
\end{aligned}$$

where we use the inequality $(a+b+c)^2 \leq 3(a^2+b^2+c^2)$. Taking expectation over the Y values given the data splits $\mathcal{P}_X$, we have

$$\mathbb{E}_{\mathbf{Y}|\mathcal{P}_X}[(\tilde{\eta}^{(k)}(x;\mathcal{P}) - \eta(x))^2] \leq 3\Big\{\underbrace{\mathrm{Var}_{\mathbf{Y}|\mathcal{P}_X}(\tilde{\eta}^{(k)}(x;\mathcal{P}))}_{(A)} \tag{3.4}$$

$$+ \underbrace{\mathbb{E}_{\mathbf{Y}|\mathcal{P}_X}[(\overline{\eta}_M^{(k)}(x;\mathcal{P}_X) - \overline{\eta}_L^{(k)}(x;\mathcal{P}_X))^2]}_{(B)}$$
$$+ \underbrace{(\overline{\eta}_L^{(k)}(x;\mathcal{P}_X) - \eta(x))^2}_{(C)}\Big\}.$$

We now bound the three terms separately in the next steps.

Step 2(A). Variance term

Consider

$$\begin{aligned}
\mathrm{Var}_{\mathbf{Y}|\mathcal{P}_X}(\tilde{\eta}^{(k)}(x;\mathcal{P})) &= \mathbb{E}_{\mathbf{Y}|\mathcal{P}_X}[(\tilde{\eta}^{(k)}(x;\mathcal{P}) - \overline{\eta}_M^{(k)}(x;\mathcal{P}))^2] \\
&= \mathbb{E}_{\mathbf{Y}|\mathcal{P}_X}\Big[\Big(\frac{1}{kM}\sum_{i=1}^{k}\sum_{m=1}^{M}(Y_{(i)}(x;\mathcal{D}_m) - \mathbb{E}[Y_{(i)}(x;\mathcal{D}_m)|\mathcal{P}_X])\Big)^2\Big] \\
&\overset{(a)}{=} \frac{1}{(kM)^2}\sum_{i=1}^{k}\sum_{m=1}^{M}\mathrm{Var}_{\mathbf{Y}|\mathcal{P}_X}(Y_{(i)}(x;\mathcal{D}_m)) \\
&= \frac{1}{(kM)^2}\sum_{i=1}^{k}\sum_{m=1}^{M}v(X_{(i)}(x;\mathbf{X}_m)) \overset{(b)}{\leq} \frac{V}{kM}. \qquad (3.5)
\end{aligned}$$

Here, (a) follows by the independence of Y_i's conditioned on the splits $\mathcal{P}_X$ and (b) follows from the assumption $v(x) \leq V$ for all $x \in \mathcal{X}$.

Step 2(B). Approximation term

We claim that the second term (B) is bounded as $O(\tau^2)$. We have

$$\begin{aligned}
|\overline{\eta}_M^{(k)}(x;\mathcal{P}) - \overline{\eta}_L^{(k)}(x;\mathcal{P})| &\leq \Big(1 - \frac{L}{M}\Big)|\overline{\eta}_L^{(k)}(x;\mathcal{P})| + \Big|\frac{1}{M}\sum_{j=L+1}^{M}\frac{1}{k}\sum_{i=1}^{k}\eta(X_{(i)}(x;\mathbf{X}_{m_j}))\Big| \\
&\overset{(a)}{\leq} \Big(1 - \frac{L}{M}\Big)H + \frac{M-L}{M}H \\
&= 2H\Big(1 - \frac{L}{M}\Big) \\
&\overset{(b)}{\leq} 4H\tau, \qquad (3.6)
\end{aligned}$$

where (a) follows by the assumption $|\eta(x)| \leq H$ for all $x \in \mathcal{X}$ and (b) follows since $L = \lceil (1-\tau)^2 M \rceil \geq (1-2\tau)M$.

Step 2(C). Bias term

It only remains to bound the term (C), which is the bias of the (M, L, k)-NN regression estimate $\breve{\eta}_L^{(k)}(x; \mathcal{P})$. Since η is (α_{H}, A)-Hölder continuous, it immediately follows that

$$\begin{aligned} |\overline{\eta}_L^{(k)}(x; \mathcal{P}_X) - \eta(x)| &\leq \frac{1}{kL} \sum_{i=1}^{k} \sum_{j=1}^{L} |\eta(X_{(i)}(x; \mathbf{X}_{m_j})) - \eta(x)| \\ &\leq A \max_{j \in [L]} r_{k+1}^{\alpha_{\mathrm{H}}}(x; \mathbf{X}_{m_j}). \end{aligned}$$

Now, for any $p \in (0,1)$, we observe that by the homogeneity of μ, we have

$$C_d \Big(\frac{r_p(x)}{2} \Big)^d \leq \mu \Big(B^o \Big(x, \frac{r_p(x)}{2} \Big) \Big) < p,$$

which implies that $r_p(x) < 2(\frac{p}{C_d})^{1/d}$. Now, if we set $p = \frac{1}{n}(k + \ln \frac{1}{\tau} + \sqrt{2k \ln \frac{1}{\tau} + (\ln \frac{1}{\tau})^2})$, then by Lemma 3.B.2 and the boundedness of the support, i.e., $\mathrm{diam}(\mathsf{supp}(\mu)) \leq R$, we have

$$\begin{aligned} \mathbb{E}_{\mathcal{P}_X}[(\overline{\eta}_L^{(k)}(x; \mathcal{P}_X) - \eta(x))^2] &\leq A^2 \mathbb{E}_{\mathcal{P}_X} \Big[\max_{j \in [L]} r_{k+1}^{2\alpha_{\mathrm{H}}}(x; \mathbf{X}_{m_j}) \Big] \\ &\leq A^2 \Big\{ r_p^{2\alpha_{\mathrm{H}}}(x) + R^{2\alpha_{\mathrm{H}}} \, \mathsf{P} \Big(\max_{j \in [L]} r_{k+1}(x; \mathbf{X}_{m_j}) > r_p(x) \Big) \Big\} \\ &\leq A^2 \Big(2^{2\alpha_{\mathrm{H}}} \Big(\frac{p}{C_d} \Big)^{\frac{2\alpha_{\mathrm{H}}}{d}} + R^{2\alpha_{\mathrm{H}}} e^{-\frac{(1-\tau)\tau^2}{2} M} \Big). \end{aligned} \tag{3.7}$$

Step 3.

Plugging in (3.5), (3.6), and (3.7) to the error decomposition (3.4) leads to

$$\mathbb{E}[(\tilde{\eta}^{(k)}(x;\mathcal{P})-\eta(x))^2] \le 3\Big\{\frac{V}{kM}+16H^2\tau^2+A^2 2^{2\alpha_{\mathrm{H}}}\Big(\frac{p}{C_d}\Big)^{\frac{2\alpha_{\mathrm{H}}}{d}}+A^2R^{2\alpha_{\mathrm{H}}}e^{-\frac{(1-\tau)\tau^2}{2}M}\Big\}.$$

If we set $\tau=\sqrt{(\ln M)^{1.01}/M}$, then we obtain the desired bound since $p=O(\frac{1}{n}(k+\ln\frac{1}{\tau}))=O(\frac{M}{N}\ln M)$ and $e^{-\frac{(1-\tau)\tau^2}{2}M}=e^{-\frac{1}{2}(\ln M)^{1.01}(1-\tau)}$ decays faster than any polynomial rate. □

Proof of Theorem 3.2.1(b)

This analysis adopts the proof technique of (Xue and Kpotufe, 2018, Proposition 1) and will invoke the following lemma therein.

Lemma 3.B.3 (Xue and Kpotufe, 2018, Lemma 1)**.** *Assume that μ is a (C_d,d)-homogeneous measure and the collection of all closed balls in $\mathcal{X}$ has finite VC dimension $\mathcal{V}$. Then, with probability at least $1-\tau$ over the sample $\mathbf{X}$ of size n, for any $k\in[n]$, we have*

$$\sup_{x\in\mathcal{X}} r_k(x;\mathbf{X}) \le \Big(\frac{3}{C_d n}\Big(k\vee\Big(\mathcal{V}\log 2n+\log\frac{8}{\tau}\Big)\Big)\Big)^{\frac{1}{d}}.$$

Step 1. Approximate with (M,L,k)-NN estimator

We consider the (M,L,k)-NN regression estimate $\breve{\eta}_L^{(k)}(x;\mathcal{P})$ with $L=\lceil(1-\tau)^2M\rceil$, where τ is to be determined. Similar to the proof of Theorem 3.2.1(a), we can decompose and upper-bound the error of $\tilde{\eta}_M^{(k)}$ as

$$\|\tilde{\eta}_M^{(k)}-\eta\|_\infty \le \|\tilde{\eta}_M^{(k)}-\breve{\eta}_L^{(k)}\|_\infty+\|\breve{\eta}_L^{(k)}-\eta\|_\infty \le 4H\tau+\|\breve{\eta}_L^{(k)}-\eta\|_\infty. \tag{3.8}$$

That is, with the approximation error $4H\tau$, it suffices to analyze the (M,L,k)-NN estimator $\breve{\eta}_L^{(k)}(x;\mathcal{P})$.

Step 2. Analyze (M, L, k)-NN estimator

To bound the sup-norm $\|\breve{\eta}_L^{(k)} - \eta\|_\infty = \sup_{x\in\mathcal{X}} |\breve{\eta}_L^{(k)}(x;\mathcal{P}) - \eta(x)|$, we consider the following bias-variance decomposition

$$|\breve{\eta}_L^{(k)}(x;\mathcal{P}) - \eta(x)| \le \underbrace{|\overline{\eta}_L^{(k)}(x;\mathcal{P}_X) - \eta(x)|}_{\text{bias}} + \underbrace{|\breve{\eta}_L^{(k)}(x;\mathcal{P}) - \overline{\eta}_L^{(k)}(x;\mathcal{P}_X)|}_{\text{variance}},$$

where we define the conditional expectation $\overline{\eta}_L^{(k)}(x;\mathcal{P}_X) := \mathbb{E}[\breve{\eta}_L^{(k)}(x;\mathcal{P})|\mathcal{P}_X]$ as in the proof of Theorem 3.2.1(a).

Step 2(a). Bias bound

The following lemma, which is a variant of Lemma 3.B.2, can be readily shown by invoking Lemma 3.B.3 with $n \leftarrow N/M$ and following the same line of the proof of Lemma 3.B.2.

Lemma 3.B.4. *Assume that μ is a (C_d, d)-homogeneous measure and the collection of all closed balls in $\mathcal{X}$ has finite VC dimension $\mathcal{V}$. Pick any $\delta \in (0,1)$. If the data splits $\mathcal{P}_X = \{\mathbf{X}_m\}_{m=1}^M$ are independent and of equal size N/M, for $L := \lceil (1-\tau)^2 M \rceil$, we have*

$$\mathsf{P}\Big(\max_{j\in[L]} r_k(x;\mathbf{X}_{m_j}) > \Big(\frac{3M}{C_d N}\Big(k \vee \Big(\mathcal{V}\log\frac{2N}{M} + \log\frac{8}{\tau}\Big)\Big)\Big)^{\frac{1}{d}}\Big) \le e^{-\frac{(1-\tau)\tau^2}{2}M}.$$

For $M \ge 16\log\frac{1}{\delta}$, we define $\tau = \sqrt{\frac{4}{M}\log\frac{1}{\delta}} \le \frac{1}{2}$ so that $\delta = e^{-\frac{\tau^2 M}{4}} \ge e^{-\frac{(1-\tau)\tau^2 M}{2}}$. Then, Lemma 3.B.4 and the Hölder continuity of η imply that with probability at least $1-\delta$ over the data splits $\mathcal{P}_X$, we have

$$\begin{aligned}\sup_{x\in\mathcal{X}} |\overline{\eta}_L^{(k)}(x;\mathcal{P}_X) - \eta(x)| &\le A \sup_{x\in\mathcal{X}} \max_{j\in[L]} r_k^{\alpha_{\mathrm{H}}}(x;\mathbf{X}_{m_j}) \\ &\le A\Big(\frac{3M}{C_d N}\Big(k \vee \Big(\mathcal{V}\log\frac{2N}{M} + \frac{1}{2}\log\frac{16M}{\log\frac{1}{\delta}}\Big)\Big)\Big)^{\frac{\alpha_{\mathrm{H}}}{d}}. \qquad (3.9)\end{aligned}$$

Step 2(b). Variance bound

For any fixed $x \in \mathcal{X}$ and split instances $\mathcal{P}_X = \{\mathbf{X}_m\}_{m=1}^M$, Hoeffding's inequality guarantees that with probability at least $1 - \delta_o$ over the labels $\{\mathbf{Y}_m\}_{m=1}^M$, we have

$$|\breve{\eta}_L^{(k)}(x; \mathcal{P}) - \overline{\eta}_L^{(k)}(x; \mathcal{P}_X)| \leq \sqrt{\frac{l_Y^2}{2kL} \log \frac{2}{\delta_o}}. \tag{3.10}$$

Now, observe that given $\mathcal{P}_X$, the left hand side is a function of x only via its nearest neighbors from $\mathbf{X}$, and thus only depends on a closed ball centered at x. The finite VC dimensionality assumption then implies that if we vary $x \in \mathcal{X}$, there are at most $(eN/\mathcal{V})^{\mathcal{V}}$ different such inequalities (3.10). Hence, letting $\delta = \delta_o (eN/\mathcal{V})^{\mathcal{V}}$ and applying union bound, we have, with probability at least $1 - \delta$ over $\{\mathbf{Y}_m\}_{m=1}^M$,

$$\sup_{x \in \mathcal{X}} |\breve{\eta}_L^{(k)}(x; \mathcal{P}) - \overline{\eta}_L^{(k)}(x; \mathcal{P}_X)| \leq \sqrt{\frac{\mathcal{V} l_Y^2}{kL} \log \frac{N}{\delta}}. \tag{3.11}$$

Since this inequality holds independent of $\mathcal{P}_X$, it also holds with probability at least $1 - \delta$ over the split data $\mathcal{P}$.

Step 3.

Continuing from (3.8) and combining the bias bound (3.9) and variance bound (3.11) by a union bound, we have with probability at least $1 - \delta$,

$$\begin{aligned} \|\tilde{\eta}_M^{(k)} - \eta\|_\infty \leq 4H \sqrt{\frac{4}{M} \ln \frac{1}{\delta}} + A\Big(\frac{3M}{C_d N}\Big(k \vee \Big(\mathcal{V} \log \frac{2N}{M} + \frac{1}{2} \log \frac{16M}{\log \frac{1}{\delta}}\Big)\Big)\Big)^{\frac{\alpha_{\mathrm{H}}}{d}} \\ + \sqrt{\frac{\mathcal{V} l_Y^2}{kL} \log \frac{2N}{\delta}}, \end{aligned}$$

which leads to the desired bound. □

3.B.3 Classification

All theoretical guarantees on classifiers in this paper are analogous to the results for the standard k-NN classifier established in the seminal paper by Chaudhuri and Dasgupta (2014).

Definitions

We first review some technical definitions introduced in (Chaudhuri and Dasgupta, 2014). For any $x \in \mathcal{X}$ and any $0 \leq p \leq 1$, define the *probability radius* of a ball centered at x as

$$r_p(x) = \inf\{r \colon \mu(B(x, r)) \geq p\}.$$

One can show that $\mu(B^o(x, r_p(x))) \geq p$, and $r_p(x)$ is the smallest radius for which this holds.

The support of the distribution μ is defined as

$$\text{supp}(\mu) := \{x \in \mathcal{X} \colon \mu(B(x, r)) > 0, \forall r > 0\}.$$

In separable metric spaces, it can be shown that $\mu(\text{supp}(\mu)) = 1$; see (Cover and Hart, 1967) or (Chaudhuri and Dasgupta, 2014, Lemma 24).

We define for any measurable set $A \subset \mathcal{X}$ with $\mu(A) > 0$,

$$\eta(A) := p(y = 1|A) = \frac{1}{\mu(A)} \int_A p(y = 1|x) \, \mathrm{d}\mu(x).$$

This is the conditional probability of Y being 1 given a point X chosen at random from the distribution μ restricted to the set A.

Based on the definitions above, we now define the effective interiors of the two classes, and the effective boundary. For $p \in [0, 1]$ and $\Delta > 0$, we define the *effective*

interiors for each class as

$$\mathcal{X}_{p,\Delta}^{+} := \mathrm{supp}(\mu) \cap \Big\{x \in \mathcal{X} \colon \eta(x) > \frac{1}{2}\Big\} \cap \Big\{x \in \mathcal{X} \colon \eta(B(x,r)) \geq \frac{1}{2} + \Delta, \forall r \leq r_p(x)\Big\}$$

and

$$\mathcal{X}_{p,\Delta}^{-} := \mathrm{supp}(\mu) \cap \Big\{x \in \mathcal{X} \colon \eta(x) < \frac{1}{2}\Big\} \cap \Big\{x \in \mathcal{X} \colon \eta(B(x,r)) \leq \frac{1}{2} - \Delta, \forall r \leq r_p(x)\Big\}.$$

For a measurable set $A \subseteq \mathcal{X}$, we define $\hat{Y}(A; \mathcal{D})$ as the mean of Y_i for points $X_i \in A$ given data $\mathcal{D} = (\mathbf{X}, \mathbf{Y})$. The quantity $\hat{Y}(A; \mathcal{D})$ is not defined if there exists no sample point X_i in A. We also define an average conditional distribution $\eta(A) := \frac{1}{\mu(A)} \int_A \eta \, \mathrm{d}\mu$ whenever $\mu(A) > 0$.

Let $g(x) := 1(\eta(x) \geq 1/2)$ denote the Bayes classifier. Let $\hat{g}^{(k)}(x; \mathcal{D})$ denote the k-NN classifier based on training data $\mathcal{D} = (\mathbf{X}, \mathbf{Y})$. Note that we can equivalently write

$$\hat{g}^{(k)}(x; \mathcal{D}) = 1\Big(\hat{Y}(B_k(x; \mathcal{D})) \geq \frac{1}{2}\Big).$$

For the sake of simplicity, we assume that there is no distance tie in what follows, but it can be handled by a similar argument in (Chaudhuri and Dasgupta, 2014).

A key technical lemma

The analysis of the standard k-NN classifier by Chaudhuri and Dasgupta (2014) relies on their key lemma (Chaudhuri and Dasgupta, 2014, Lemma 7), which proves a sufficient condition for the k-NN classifier to agree with the Bayes classifier. In this section, we provide an analogous lemma for the (M, k)-NN classifier. The key idea is to leverage the closeness of the (M, k)-NN classifier to a (M, L, k)-NN classifier for $L \leq M$ sufficiently large. We remark that the following lemma is the only place we use the (M, L, k)-NN rule in the rest of our analysis.

Lemma 3.B.5 (cf. (Chaudhuri and Dasgupta, 2014, Lemma 7)). *Pick any $x_o \in \mathcal{X}$, any $p \in (0,1)$, $\Delta \in (0, 1/2]$, and $\tau \in (0, \frac{\Delta}{8}]$. Let $L := \lceil (1-\tau)^2 M \rceil$. For each $m \in [M]$, define $B_m := B^o(x_o, r_{k+1}(x_o; \mathbf{X}_m))$. Then, we have*

$$\begin{aligned} 1(\tilde{g}^{(k)}(x_o; \mathcal{P}) \neq g(x_o)) \leq\ & 1(x_o \in \partial_{p,\Delta}) \\ & + 1\Big(\max_{j \in [L]} r_{k+1}(x_o; \mathbf{X}_{m_j}) > r_p(x_o)\Big) \\ & + 1\Big(\Big|\frac{1}{L}\sum_{j=1}^{L}(\hat{Y}(B_{m_j}; \mathcal{D}_{m_j}) - \eta(B_{m_j}))\Big| \geq \frac{\Delta}{2}\Big), \end{aligned} \tag{3.12}$$

where $m_1, \ldots, m_L$ are the indices that correspond to the L-smallest values among the $(k+1)$-th-NN distances $(r_{k+1}(x_o; \mathbf{X}_m))_{m=1}^M$.

Proof. Suppose $x_o \notin \partial_{p,\Delta}$. Without loss of generality, consider $x_o \in \mathcal{X}^+_{p,\Delta}$, whereupon $g(x_o) = 1$. By definition of the effective interior, $\eta(B(x_o, r)) \geq \frac{1}{2} + \Delta$ for all $r \leq r_p(x_o)$.

Now, suppose

$$\max_{j \in [L]} r_{k+1}(x_o; \mathbf{X}_{m_j}) \leq r_p(x_o).$$

Then, we have, for any $j \in [L]$, that

$$\eta(B_{m_j}) = \eta(B^o(x_o, r_{k+1}(x_o; \mathbf{X}_{m_j}))) \geq \frac{1}{2} + \Delta,$$

by Lemma 3.B.6 (stated below).

Further, if $|\frac{1}{L}\sum_{j=1}^{L}(\hat{Y}(B_{m_j}; \mathcal{D}_{m_j}) - \eta(B_{m_j}))| < \frac{\Delta}{2}$, then

$$\breve{\eta}_L^{(k)}(x_o; \mathcal{P}) = \frac{1}{L}\sum_{j=1}^{L}\hat{Y}(B_{m_j}; \mathcal{D}_{m_j}) \geq \frac{1}{2} + \frac{\Delta}{2},$$

where we recall that $\breve{\eta}_L^{(k)}(\cdot; \mathcal{P})$ denotes the (M, L, k)-NN regressor based on the training data splits $\mathcal{P} = \{(\mathbf{X}_m, \mathbf{Y}_m)\}_{m=1}^M$.

Finally, since $|\tilde{\eta}^{(k)}(x_o; \mathcal{P}) - \breve{\eta}_L^{(k)}(x_o; \mathcal{P})| \leq 2(1 - \frac{L}{M}) < 4\tau \leq \frac{\Delta}{2}$, we have $\tilde{\eta}^{(k)}(x_o; \mathcal{P}) >$

$\frac{1}{2}$, which concludes $\tilde{g}^{(k)}(x_o; \mathcal{P}) = 1 = g(x_o)$. □

Lemma 3.B.6 (Chaudhuri and Dasgupta, 2014, Lemma 26)**.** *Suppose that for some $x_o \in \mathrm{supp}(\mu)$ and $r_o > 0$ and $q > 0$, we have $[r \leq r_o \Rightarrow \eta(B(x_o, r)) \geq q]$. Then, we also have $[r \leq r_o \Rightarrow \eta(B^o(x_o, r)) \geq q]$.*

A general upper bound on the misclassification error

We first present a generalization of the main result in (Chaudhuri and Dasgupta, 2014), which is a general upper bound on the misclassification error rate. Theorem 3.2.3(a) and (b) will almost readily follow as corollaries of this theorem.

Theorem 3.B.7 (cf. (Chaudhuri and Dasgupta, 2014, Theorem 5))**.** *Let $k \geq 1$ be fixed and pick any $\delta \in (0, 1)$. Pick any integer $M \geq \frac{2^{14}}{15} \log \frac{2}{\delta}$, set*

$$\Delta := \sqrt{\frac{2^{12}}{15M} \log \frac{2}{\delta}} \in \Big(0, \frac{1}{2}\Big].$$

Pick any integer $n \geq k + \log \frac{8}{\Delta} + \sqrt{2k \log \frac{8}{\Delta} + (\log \frac{8}{\Delta})^2}$ and set

$$p := \frac{1}{n}\Big(k + \log \frac{8}{\Delta} + \sqrt{2k \log \frac{8}{\Delta} + (\log \frac{8}{\Delta})^2}\Big) \in (0, 1].$$

Then, for a set of data splits $\mathcal{P} = \{\mathcal{D}_1, \ldots, \mathcal{D}_M\}$, where every split $\mathcal{D}_m$ has n data points, with probability at least $1 - \delta$ over $\mathcal{P}$, we have

$$\mathsf{P}(\tilde{g}^{(k)}(X; \mathcal{P}) \neq g(X) | \mathcal{P}) \leq \delta + \mu(\partial_{p,\Delta}).$$

Proof. Given $k \geq 1$, $\delta \in (0, 1)$, and $\Delta \in (0, 1/2]$, we set $\tau = \frac{\Delta}{8}$ and define $L = \lceil (1-\tau)^2 M \rceil$ as stated in Lemma 3.B.2. Pick any $x_o \in \mathcal{X}$. Applying Lemma 3.B.5, we have

$$1(\tilde{g}^{(k)}(x_o; \mathcal{P}) \neq g(x_o)) \leq 1(x_o \in \partial_{p,\Delta}) + I_{\mathsf{bad}}(x_o; \mathcal{P}),$$

where we define the bad event indicator variable

$$I_{\mathsf{bad}}(x_o;\mathcal{P}) \triangleq 1\Big(\max_{j\in[L]} r_{k+1}(x_o;\mathbf{X}_{m_j}) > r_p(x_o)\Big) + 1\Big(\Big|\frac{1}{L}\sum_{j=1}^{L}(\hat{Y}(B_{m_j};\mathcal{D}_{m_j}) - \eta(B_{m_j}))\Big| \geq \frac{\Delta}{2}\Big), \tag{3.13}$$

where $m_1,\ldots,m_L$ are the indices for the L smallest distances among $\{r_{k+1}(x_o;\mathbf{X}_m)\}_{m=1}^{M}$. For any fixed point $x_o \in \mathcal{X}$, if we take the expectation over the training data splits $\mathcal{P}$, we have

$$\mathbb{E}[I_{\mathsf{bad}}(x_o;\mathcal{P})] \triangleq \mathsf{P}\Big(\max_{j\in[L]} r_{k+1}(x_o;\mathbf{X}_{m_j}) > r_p(x_o)\Big) + \mathsf{P}\Big(\Big|\frac{1}{L}\sum_{j=1}^{L}(\hat{Y}(B_{m_j};\mathcal{D}_{m_j}) - \eta(B_{m_j}))\Big| \geq \frac{\Delta}{2}\Big). \tag{3.14}$$

The first term can be bounded by Lemma 3.B.2. For the second term, we need the following lemma, whose proof is given at the end of this proof:

Lemma 3.B.8.

$$\mathsf{P}\Big(\Big|\frac{1}{L}\sum_{j=1}^{L}(\hat{Y}(B_{m_j};\mathcal{D}_{m_j}) - \eta(B_{m_j}))\Big| \geq \frac{\Delta}{2}\Big) \leq 2e^{-\frac{\Delta^2}{2}L} \leq 2e^{-\frac{\Delta^2}{8}M}. \tag{3.15}$$

By Lemmas 3.B.2 and 3.B.8, we have

$$\mathbb{E}[I_{\mathsf{bad}}(x_o,\mathcal{P})] \leq e^{-\frac{(1-\tau)\tau^2}{2}M} + 2e^{-\frac{\Delta^2}{8}M}.$$

Here, since $\Delta = \sqrt{\frac{2^{12}}{15M}\log\frac{2}{\delta}} \leq \frac{1}{2}$, we have $\tau = \frac{\Delta}{8} \leq \frac{1}{16}$, which implies that $\frac{(1-\tau)\tau^2 M}{2} \geq \frac{15\Delta^2 M}{2^{11}} = 2\log\frac{2}{\delta}$ and $\frac{\Delta^2 M}{8} = \frac{2^9}{15}\log\frac{2}{\delta} \geq 2\log\frac{2}{\delta}$. Therefore, we can further upper bound

the expectation as

$$\mathbb{E}[I_{\mathsf{bad}}(x_o, \mathcal{P})] \leq e^{-\frac{(1-\Delta/8)\Delta^2}{128}M} + 2e^{-\frac{\Delta^2}{8}M} \leq \delta^2.$$

Note that the expectation is over the training data $\mathcal{P}$. Taking expectation over the query point $X_o \sim \mu$, we have $\mathbb{E}[I_{\mathsf{bad}}(X_o, \mathcal{P})] \leq \delta^2$, which in turn implies

$$\mathsf{P}(\mathbb{E}[I_{\mathsf{bad}}(X_o, \mathcal{P})|\mathcal{P}] \geq \delta) \leq \delta.$$

The desired conclusion follows by noting that

$$\mathsf{P}(\tilde{g}_M^{(k)}(X_o; \mathcal{P}) \neq g(X_o)|\mathcal{P}) \leq \mu(\partial_{p,\Delta}) + \mathbb{E}[I_{\mathsf{bad}}(X_o, \mathcal{P})|\mathcal{P}]$$

from Lemma 3.B.5. □

Proof of Lemma 3.B.8. We note that this statement is a distributed version of (Chaudhuri and Dasgupta, 2014, Lemma 9). To prove it, first observe that we can draw the training data splits $\mathcal{D}_{1:M}$, $\mathcal{D}_m = \{(X_{mi}, Y_{mi})\}_{i=1}^n$, where $N = Mn$, by the following steps.

1. Draw M points $X_1^{(1)}, \ldots, X_1^{(M)} \in \mathcal{X}$ independently at random, according to the marginal distribution of the $(k+1)$-th nearest neighbor of the fixed point x_o with respect to n independent sample points.

2. Sort the M points $\{X_1^{(1)}, \ldots, X_1^{(M)}\}$ based on their distances to x_o. Let $\tilde{X}_1^{(1)}, \ldots, \tilde{X}_1^{(M)}$ denote the sorted points in the increasing order of the distances, where we break ties at random. Let $\tilde{B}_j := B^o(x_o, \rho(x_o, \tilde{X}_1^{(j)}))$.

3(a). For each $j \in [L]$, pick k points at random from the distribution μ restricted to $\tilde{B}_j$.

3(b). For each $j \in [L]$, pick $n - k - 1$ points at random from the distribution μ restricted to $\mathcal{X} \backslash \tilde{B}_j$.

4. For each $m \in [M] \backslash [L]$, repeat the same steps in 3a and 3b.

5. For each $m \in [M]$, randomly permute the n points obtained in this way.

6. For each $m \in [M]$ and for X_{mi} in the permuted order, draw a label Y_{mi} from the conditional distribution $\eta(X_{mi})$.

We now condition everything on Step 1 and Step 2, or equivalently, on $\tilde{X}_1^{(1)}, \ldots, \tilde{X}_1^{(M)}$. Recall that we denote by $m_1, \ldots, m_L$ the indices that correspond to the L-smallest values among $(r_{k+1}(x_o; \mathbf{X}_m))_{m=1}^M$. Since the corresponding sample points are $\tilde{X}_1^{(1)}, \ldots, \tilde{X}_1^{(L)}$, we can write $B_{m_j} = \tilde{B}_j$. Hence, in the desired inequality, $\hat{Y}(B_{m_j}; \mathcal{D}_{m_j})$ for each $j \in [L]$ is the average of the Y-values which correspond to the X's drawn from Step 3(a). Since the corresponding Y-values have expectation $\mathbb{E}[Y|\mathbf{X} \in \tilde{B}_j] = \eta(\tilde{B}_j)$ for each $j \in [L]$ and the total kL of Y-values are independent, we can apply Hoeffding's inequality and obtain

$$\mathsf{P}\Big(\Big|\frac{1}{L}\sum_{j=1}^{L}(\hat{Y}(B_{m_j}; \mathcal{D}_{m_j}) - \eta(B_{m_j}))\Big| \geq \frac{\Delta}{2}\Big|\tilde{X}_1^{(1)}, \ldots, \tilde{X}_1^{(M)}\Big) \leq 2e^{-\frac{\Delta^2}{2}L} \leq 2e^{-\frac{\Delta^2}{8}M}.$$

Taking expectations over $\tilde{X}_1^{(1)}, \ldots, \tilde{X}_1^{(M)}$, we prove the desired inequality. □

Proof of Theorem 3.2.3

Proof of Theorem 3.2.3(a)

Recall that we use $\partial\eta_\Delta := \{x \in \text{supp}(\mu) \colon |\eta(x) - 1/2| \leq \Delta\}$ to denote the decision boundary with margin $\Delta \geq 0$. Under the smoothness of the measure μ, the effective decision boundary $\partial_{p,\Delta}$ is a subset of the decision boundary with a certain margin as stated below:

Lemma 3.B.9 (Chaudhuri and Dasgupta, 2014, Lemma 18)**.** *If η is (α, A)-smooth in $(\mathcal{X}, \rho, \mu)$, then for any $p \in [0, 1]$ and $\Delta \in (0, 1/2]$, we have $\partial_{p,\Delta} \subset \partial\eta_{\Delta + Ap^\alpha}$.*

Set $\Delta = \sqrt{\frac{2^{12}}{15M} \log \frac{2}{\delta}}$. Under the margin condition, this lemma implies that

$$\mu(\partial_{p,\Delta}) = O\Big(\sqrt{\frac{1}{M} \log \frac{1}{\delta}} + \Big(\frac{M}{N} \log \frac{M}{\log \frac{1}{\delta}}\Big)^{\alpha}\Big).$$

Applying the general upper bound in Theorem 3.B.7 concludes the proof. □

Proof of Theorem 3.2.3(b) expected risk bound

This proof modifies that of (Chaudhuri and Dasgupta, 2014, Theorem 4) in accordance with Lemma 3.B.5 instead of (Chaudhuri and Dasgupta, 2014, Lemma 7). We pick and fix any $\tau \in (0, 1)$ for now, and will choose a specific choice at the end of the analysis. Set $L = \lceil (1-\tau)^2 M \rceil$, $p = \frac{M}{N}(k + \log \frac{1}{\tau} + \sqrt{2k \log \frac{1}{\tau} + (\log \frac{1}{\tau})^2})$ as in Lemma 3.B.5 and Theorem 3.B.7, respectively, and define $\Delta_o = Ap^{\alpha}$.

We first state and prove the following lemma.

Lemma 3.B.10 (cf. (Chaudhuri and Dasgupta, 2014, Lemma 20))**.** *For any $x_o \in \text{supp}(\mu)$ with $\Delta(x_o) \geq \Delta_o + 8\tau$. Under the (α, A)-smoothness condition, we have*

$$\mathbb{E}_{\mathcal{P}}[R(x_o; \tilde{g}_M^{(k)})] - R^*(x_o) \leq e^{-\frac{(1-\tau)\tau^2}{2} M} + 4\Delta(x_o) e^{-\frac{(\Delta(x_o)-\Delta_o)^2}{8} M}.$$

Proof of Lemma 3.B.10. Without loss of generality, assume that $\eta(x_o) > \frac{1}{2}$. By the smoothness condition, for any $0 \leq r \leq r_p(x_o)$, we have

$$\eta(B(x_o, r)) \geq \eta(x_o) - Ap^{\alpha} = \eta(x_o) - \Delta_o = \frac{1}{2} + (\Delta(x_o) - \Delta_o),$$

which implies $x_o \in \mathcal{X}^+_{p,\Delta(x_o)-\Delta_o}$ and thus $x_o \notin \partial_{p,\Delta(x_o)-\Delta_o}$.

Recall that for any classifier $\hat{g}$, we can write $R(x_o; \hat{g}) - R^*(x_o) = 2\Delta(x_o) 1(\hat{g}(x_o) \neq g(x_o))$, where $R^*(x_o)$ is the Bayes risk. Since we assume that $\tau \leq \frac{\Delta(x_o)-\Delta_o}{8}$, we can apply

Lemma 3.B.2 with $\Delta \leftarrow \Delta(x_o) - \Delta_o$ and have

$$R(x_o, \tilde{g}_M^{(k)}) - R^*(x_o) = 2\Delta(x_o) 1(\tilde{g}_M^{(k)}(x_o) \neq g(x_o)) \leq 2\Delta(x_o) I_{\mathsf{bad}}(x_o; \mathcal{P}), \tag{3.16}$$

where we define the bad-event indicator variable as

$$\begin{aligned} I_{\mathsf{bad}}(x_o; \mathcal{P}) \triangleq\ & 1\Big(\max_{j \in [L]} r_{k+1}(x_o; \mathbf{X}_{m_j}) > r_p(x_o)\Big) + \\ & 1\Big(\Big|\frac{1}{L}\sum_{j=1}^{L}(\hat{Y}(B_{m_j}; \mathcal{D}_{m_j}) - \eta(B_{m_j}))\Big| \geq \frac{\Delta(x_o) - \Delta_o}{2}\Big), \end{aligned}$$

as in (3.13) in the proof of Theorem 3.B.7 with $\Delta \leftarrow \Delta(x_o) - \Delta_o$. By taking the expectations over the random splits $\mathcal{P}$ in (3.16), we have

$$\mathbb{E}_{\mathcal{P}} R(x_o; \tilde{g}_M^{(k)}) - R^*(x_o) \leq 2\Delta(x_o) \mathbb{E}[I_{\mathsf{bad}}(x_o; \mathcal{P})].$$

Now, by applying Lemma 3.B.2 and Lemma 3.B.8 as in the proof of Theorem 3.B.7, we can bound the right hand side as

$$\begin{aligned} \mathbb{E}_{\mathcal{P}} R(x_o; \tilde{g}_M^{(k)}) - R^*(x_o) &\leq 2\Delta(x_o)(e^{-\frac{(1-\tau)\tau^2}{2}M} + 2e^{-\frac{(\Delta(x_o)-\Delta_o)^2}{8}M}) \\ &\leq e^{-\frac{(1-\tau)\tau^2}{2}M} + 4\Delta(x_o) e^{-\frac{(\Delta(x_o)-\Delta_o)^2}{8}M}, \end{aligned}$$

where the last inequality follows from the assumption that $\Delta(x_o) \leq 1/2$. □

We then prove the following statement under the margin condition.

Lemma 3.B.11 (cf. (Chaudhuri and Dasgupta, 2014, Lemma 21))**.** *Under the* (α, A)*-smoothness and the* (β, C)*-margin condition, we have*

$$\mathbb{E}_{\mathcal{P}} R(\tilde{g}_M^{(k)}) - R^* \leq e^{-\frac{(1-\tau)\tau^2}{2}M} + 6C\Big(\max\Big\{2A\Big(\frac{8M}{N}\log\frac{1}{\tau}\Big)^{\alpha}, \sqrt{\frac{\beta+2}{8M}}\Big\} + 8\tau\Big)^{\beta+1}. \tag{3.17}$$

Proof of Lemma 3.B.11. For each integer $i \geq 1$, define $\Delta_i = 2^i \Delta_o + 8\tau$. Fix any $i_o \geq 1$. To bound the expected risk, we apply Lemma 3.B.10 for any x_o with $\Delta(x_o) > \Delta_{i_o}$ and use $\mathbb{E}_{\mathcal{P}} R(x_o; \tilde{g}_M^{(k)}) - R^*(x_o) \leq 2\Delta_{i_o}$ for all remaining x_o. Taking expectations over X_o, we have

$$\begin{aligned}
&\mathbb{E}_{\mathcal{P}} R(\tilde{g}_M^{(k)}) - R^* \\
&\leq \mathbb{E}_{X_o}[2\Delta_{i_o} 1(\Delta(X_o) \leq \Delta_{i_o}) + e^{-\frac{(1-\tau)\tau^2}{2} M} + 4\Delta(X_o) e^{-\frac{(\Delta(X_o)-\Delta_o)^2}{8} M} 1(\Delta(X_o) > \Delta_{i_o})] \\
&\leq 2C\Delta_{i_o}^{\beta+1} + e^{-\frac{(1-\tau)\tau^2}{2} M} + 4\mathbb{E}_{X_o}[\Delta(X_o) e^{-\frac{(\Delta(X_o)-\Delta_o)^2}{8} M} 1(\Delta(X_o) > \Delta_{i_o})]. \qquad (3.18)
\end{aligned}$$

Here, we invoke the (β, C)-margin condition in the second inequality to bound the first term.

It only remains to bound the last term. First, by another application of the (β, C)-margin condition, we have

$$\begin{aligned}
\mathbb{E}_{X_o}[\Delta(X) e^{-\frac{(\Delta(X)-\Delta_o)^2}{8} M} 1(\Delta_i < \Delta(X) \leq \Delta_{i+1})] &\leq \mathbb{E}_{X_o}[\Delta_{i+1} e^{-\frac{(\Delta_i-\Delta_o)^2}{8} M} 1(\Delta(X) \leq \Delta_{i+1})] \\
&\leq C\Delta_{i+1}^{\beta+1} e^{-\frac{(\Delta_i-\Delta_o)^2}{8} M}. \qquad (3.19)
\end{aligned}$$

Now, we set

$$i_o = \max\Big(1, \Big\lceil \log_2 \sqrt{\frac{32(\beta+2)}{M\Delta_o^2}} \Big\rceil\Big),$$

so that the terms (3.19) are upper-bounded by a geometric series with ratio $1/2$. Indeed, for $i \geq i_o$, we have

$$\begin{aligned}
&\frac{\Delta_{i+1}^{\beta+1} \exp(-\frac{M}{8}(\Delta_i - \Delta_o)^2)}{\Delta_i^{\beta+1} \exp(-\frac{M}{8}(\Delta_{i-1} - \Delta_o)^2)} \\
&= \Big(\frac{2^{i+1}\Delta_o + 8\tau}{2^i \Delta_o + 8\tau}\Big)^{\beta+1} \exp\Big(-\frac{M}{8}\{(\Delta_i - \Delta_o)^2 - (\Delta_{i-1} - \Delta_o)^2\Big) \\
&\leq 2^{\beta+1} \exp\Big(-\frac{M}{8}\{((2^i - 1)\Delta_o + 8\tau)^2 - ((2^{i-1} - 1)\Delta_o + 8\tau)^2\Big)
\end{aligned}$$

$$
\begin{aligned}
&= 2^{\beta+1}\exp\Big(-\frac{M}{8}\{\Delta_o^2((2^i-1)^2-(2^{i-1}-1)^2)+16\Delta_o\tau(2^i-2^{i-1})\}\Big)\\
&\leq 2^{\beta+1}\exp(-M\Delta_o^2 2^{2i-5})\\
&\leq 2^{\beta+1}\exp(-(\beta+2)) \leq \frac{1}{2}.
\end{aligned}
$$

Therefore, we can bound the last term in (3.18) as

$$
\begin{aligned}
&\mathbb{E}_{X_o}[\Delta(X_o)e^{-\frac{(\Delta(X_o)-\Delta_o)^2}{8}M}1(\Delta(X_o)>\Delta_{i_o})]\\
&=\sum_{i=i_o}^{\infty}\mathbb{E}_{X_o}[\Delta(X_o)e^{-\frac{(\Delta(X_o)-\Delta_o)^2}{8}M}1(\Delta_i<\Delta(X_o)\leq\Delta_{i+1})]\\
&\leq C\sum_{i=i_o}^{\infty}\Delta_{i+1}^{\beta+1}e^{-\frac{(\Delta_i-\Delta_o)^2}{8}M} \leq C\Delta_{i_o}^{\beta+1}.
\end{aligned}
$$

Plugging this back into (3.18), we have $\mathbb{E}_{\mathcal{P}}R(\tilde{g}_M^{(k)}) - R^* \leq e^{-\frac{(1-\tau)\tau^2}{2}M} + 6C\Delta_{i_o}^{\beta+1}$. The desired inequality follows by substituting $\Delta_{i_o} = 2^{i_o}\Delta_o + 8\tau$. □

Now, back to the proof of Theorem 3.2.3(b), setting $\tau = \sqrt{\frac{(\log M)^{1.01}}{M}}$ and applying Lemma 3.B.11 lead to

$$
\mathbb{E}_{\mathcal{P}}R(\tilde{g}_M^{(k)}) - R^* = O\Big(\Big\{\Big(\frac{M}{N}\log\frac{M}{(\log M)^{1.01}}\Big)^{\alpha} + \sqrt{\frac{(\log M)^{1.01}}{M}}\Big\}^{\beta+1}\Big),
$$

since the first term $e^{-\frac{(1-\tau)\tau^2}{2}M}$ in (3.17) decays faster than any polynomial rate. Finally, setting $M \propto N^{2\alpha/(2\alpha+1)}$ concludes the proof. □

Proof of Theorem 3.2.3(b) CIS bound

Since the proof is an easy modification of the previous proof of the expected risk bound, we only outline the critical steps that differ from the proof of Theorem 3.2.3(b) regret bound. Observe that the classification instability is upper-bounded as

$$
\mathrm{CIS}_N(\hat{g}) \leq 2\mathbb{E}_{\mathcal{D}}[\mathsf{P}_X(\hat{g}(X;\mathcal{D}) \neq g(X))] \tag{3.20}
$$

for any classification procedure $\hat{g}(\cdot; \mathcal{D})$. Hence, following the exact same line of the proof of Lemma 3.B.10, we have

Lemma 3.B.12. *For any $x_o \in \text{supp}(\mu)$ with $\Delta(x_o) \geq \Delta_o + 8\tau$. Under the (α, A)-smoothness condition, we have*

$$\mathbb{E}_{\mathcal{D}}[1(\tilde{g}_M^{(k)}(x_o; \mathcal{D}) \neq g(x_o))] \leq e^{-\frac{(1-\tau)\tau^2}{2}M} + 4e^{-\frac{(\Delta(x_o)-\Delta_o)^2}{8}M}.$$

We then follow the same line of the proof of Lemma 3.B.11. For each integer $i \geq 1$, define $\Delta_i = 2^i\Delta_o + 8\tau$. Fix any $i_o \geq 1$. To bound the expected probability of the mismatch $\mathbb{E}_{\mathcal{D}}[\mathcal{P}_X(\tilde{g}_M^{(k)}(X_o; \mathcal{D}) \neq g(X_o))]$, we will apply Lemma 3.B.12 for any x_o with $\Delta(x_o) > \Delta_{i_o}$ and use a trivial bound $\mathbb{E}_{\mathcal{D}}[1(\tilde{g}_M^{(k)}(x_o; \mathcal{D}) \neq g(x_o))] \leq 1$ for all remaining x_o. Taking expectations over X_o and invoking the (β, C)-margin condition, we have

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}}[\mathcal{P}_{X_o}(\tilde{g}_M^{(k)}(X_o; \mathcal{D}) \neq g(X_o))] \\
&\leq \mathbb{E}_{X_o}[1(\Delta(X_o) \leq \Delta_{i_o}) + e^{-\frac{(1-\tau)\tau^2}{2}M} + 4e^{-\frac{(\Delta(X_o)-\Delta_o)^2}{8}M}1(\Delta(X_o) > \Delta_{i_o})] \\
&\leq e^{-\frac{(1-\tau)\tau^2}{2}M} + C\Delta_{i_o}^{\beta} + 4\mathbb{E}_{X_o}[e^{-\frac{(\Delta(X_o)-\Delta_o)^2}{8}M}1(\Delta(X_o) > \Delta_{i_o})]. \qquad (3.21)
\end{aligned}$$

By the same logic in the proof of Lemma 3.B.11, the last term can be bounded by $C\Delta_{i_o}^{\beta}$ with the same i_o. Plugging this back into (3.21), we have

$$\mathbb{E}_{\mathcal{D}}[\mathcal{P}_{X_o}(\tilde{g}_M^{(k)}(X_o; \mathcal{D}) \neq g(X_o))] \leq e^{-\frac{(1-\tau)\tau^2}{2}M} + 5C\Delta_{i_o}^{\beta}.$$

By substituting $\Delta_{i_o} = 2^{i_o}\Delta_o + 8\tau$, we have

$$\begin{aligned}
\text{CIS}_N(\tilde{g}_M^{(k)}) &\leq 2\mathbb{E}_{\mathcal{D}}[\mathsf{P}_X(\tilde{g}_M^{(k)}(X_o; \mathcal{D}) \neq g(X_o))] \\
&\leq 2e^{-\frac{(1-\tau)\tau^2}{2}M} + 10C\Big(\max\Big\{2A\Big(\frac{8M}{N}\log\frac{1}{\tau}\Big)^{\alpha}, \sqrt{\frac{\beta+2}{8M}}\Big\} + 8\tau\Big)^{\beta}
\end{aligned}$$

and setting $\tau = \sqrt{\frac{(\log M)^{1.01}}{M}}$ concludes the proof. □

Asymptotic Bayes consistency

As alluded to in the main text, we can establish the asymptotic Bayes consistency of the proposed rules under the *Lebesgue differentiation condition* on the metric measure space $(\mathcal{X}, \rho, \mu)$, i.e., for any bounded measurable function f,

$$\lim_{r \to 0} \frac{1}{\mu(B^o(x, r))} \int_{B^o(x,r)} f \, \mathrm{d}\mu = f(x)$$

for almost all (μ-a.e.) $x \in \mathcal{X}$. For example, any metric space with doubling measure satisfies this condition as a consequence of Vitali covering theorem; see, e.g., (Heinonen, 2012, Theorem 1.8).

Theorem 3.B.13 (cf. (Chaudhuri and Dasgupta, 2014, Theorem 1))**.** *Suppose that a metric measure space $(\mathcal{X}, \rho, \mu)$ satisfies the Lebesgue differentiation condition. Let $k \geq 1$ be fixed.*

(a) If $M \to \infty$ and $\frac{M}{N} \to 0$ as $N \to \infty$, for all $\epsilon > 0$, $\lim_{N \to \infty} \mathsf{P}(R(\tilde{g}_M^{(k)}) - R^* > \epsilon) = 0$.

(b) If $\frac{M}{\log N} \to \infty$ as $N \to \infty$, then $R(\tilde{g}_M^{(k)}) \to R^*$ almost surely.

Proof of Theorem 3.B.13. Observe that we have $R(x; \hat{g}) - R^*(x) = |1 - 2\eta(x)| 1(\hat{g}(x) \neq g(x))$ for any binary classifier $\hat{g}$, which implies

$$R(\tilde{g}_M^{(k)}) - R^* \leq \mathsf{P}(\eta(X) \neq 1/2, \tilde{g}_M^{(k)}(X) \neq g(X)).$$

Let $\partial_o := \{x \in \mathcal{X} \colon \eta(x) = 1/2\}$ denote the decision boundary. Then, we have the following corollary of Theorem 3.B.7.

Corollary 3.B.14 (cf. (Chaudhuri and Dasgupta, 2014, Corollary 13))**.** *Let $k \geq 1$ be fixed and let (δ_N) and (Δ_N) be any sequences of positive reals. For each N, set $M_N = \frac{2^{12}}{15 \Delta_N^2} \log \frac{2}{\delta_N}$.*

Then,

$$\mathsf{P}\{R(\hat{g}^{(k)}_{M_N}) - R^* > \delta_N + \mu(\partial_{p_N,\Delta_N} \setminus \partial_o)\} \leq \delta_N.$$

Note that the Lebesgue differentiation condition implies that $\mu(\partial_{p_N,\Delta_N} \setminus \partial_o) \to 0$:

Lemma 3.B.15 (Chaudhuri and Dasgupta, 2014, Lemma 15)**.** *Assume that $(\mathcal{X}, \rho, \mu)$ satisfies the Lebesgue differentiation condition. If $p_N, \Delta_N \downarrow 0$, then $\mu(\partial_{p_N,\Delta_N} \setminus \partial_o) \downarrow 0$ as $N \to \infty$.*

We are now ready to prove the consistency results.

Proof of Theorem 3.B.13(a)

Define $\delta_N = e^{-\sqrt{M_N}}$ and $\Delta_N = \sqrt{\frac{2^{12}}{15M_N} \log \frac{2}{\delta_N}}$. Then, as $N \to \infty$, $p_N \to 0$ and $\Delta_N \to 0$ by assumption.

Pick any $\epsilon > 0$. Choose a positive integer N' such that $\delta_N \leq \frac{\epsilon}{2}$ and $\mu(\partial_{p_N,\Delta_N} \setminus \partial_o) \leq \frac{\epsilon}{2}$ for all $N \geq N'$. Then by Corollary 3.B.14, for $N \geq N'$,

$$\mathsf{P}(R(\hat{g}^{(k)}_{M_N}) - R^* > \epsilon) \leq \delta_N.$$

Taking $N \to \infty$ concludes the proof.

Proof of Theorem 3.B.13(b)

We note that this proof is almost identical to that of (Chaudhuri and Dasgupta, 2014, Lemma 17). Choose $\delta_N = 1/N^2$ and for each N, set p_N and Δ_N as in Theorem 3.B.7. It is easy to see $p_N, \Delta_N \to 0$ as $N \to \infty$, provided that $M/\log N \to \infty$.

For any $\epsilon > 0$, there exists N' sufficiently large such that $\sum_{N \geq N'} \delta_N \leq \epsilon$. Letting ω denote the infinite training data, by Corollary 3.B.14, we have

$$\mathsf{P}\{\omega | \exists N \geq N' \colon R(\hat{g}^{(k)}_{M_N}(\omega) - R^* > \delta_N + \mu(\partial_{p_N,\Delta_N} \setminus \partial_o)\} \leq \sum_{N \geq N'} \delta_N \leq \epsilon.$$

Therefore, with probability at least $1 - \epsilon$ over ω, we have

$$\hat{g}_{M_N}^{(k)}(\omega) - R^* > \delta_N + \mu(\partial_{p_N,\Delta_N} \setminus \partial_o)$$

for all $N \geq N'$. Since $\mu(\partial_{p_N,\Delta_N} \setminus \partial_o) \to 0$ as $N \to \infty$ by Lemma 3.B.15, we conclude the proof. □

3.C Experiment details

Table 3.C.1. Summary of dimensions of the benchmark datasets.

Dataset	# training	# dim.	# class.
GISETTE (Guyon et al., 2004)	7k	5k	2
HTRU2 (Lyon et al., 2016)	18k	8	2
Credit (Dua and Graff, 2019)	30k	23	2
MiniBooNE (Dua and Graff, 2019)	130k	50	2
SUSY (Baldi et al., 2014)	5000k	18	2
BNG(letter,1000,1) (Vanschoren et al., 2013)	1000k	17	26
YearPredictionMSD (Dua and Graff, 2019)	463k	90	1

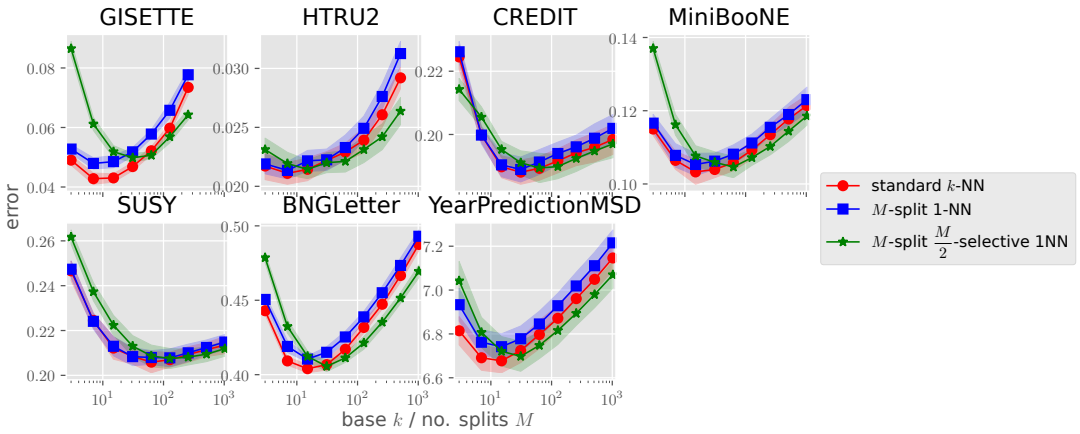


Figure 3.C.1. Validation error profiles from 10-fold cross validation. Here, as expected, the optimal M chosen for $(M, 1)$-NN rules is in the same order of the optimal k for the standard k-NN rules.

Acknowledgement

Chapter 3, in part, is a reprint of the material in the paper: J. Jon Ryu and Young-Han Kim, "One-Nearest-Neighbor Search Is All You Need for Minimax Regression and Classification," February 2022, arXiv:2202.02464, submitted to *The Thirty-Sixth Annual Conference on Neural Information Processing Systems*. The dissertation author was the primary investigator and author of this paper. This work was supported in part by the National Science Foundation under Grant CCF-1911238

Bibliography

Maha Ahmed Alabduljalil, Xun Tang, and Tao Yang. Optimizing parallel algorithms for all pairs similarity search. In *Proc. Int. Conf. Web Search Data Mining*, pages 203–212, 2013.

David C Anastasiu and George Karypis. Parallel cosine nearest neighbor graph construction. *J. Parallel. Distrib. Comput.*, 129:61–82, 2019.

Fabrizio Angiulli and Clara Pizzuti. Fast outlier detection in high dimensional spaces. In *Euro. Conf. Princ. Data Mining Knowledge Discov.*, pages 15–27. Springer, 2002.

Jean-Yves Audibert, Alexandre B Tsybakov, et al. Fast learning rates for plug-in classifiers. *Ann. Statist.*, 35(2):608–633, 2007.

Pierre Baldi, Peter Sadowski, and Daniel Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nat. Commun*, 5(1):1–9, 2014.

Akshay Balsubramani, Sanjoy Dasgupta, Yoav Freund, and Shay Moran. An adaptive nearest neighbor rule for classification. In *Adv. Neural Inf. Process. Syst.*, volume 32, pages 7579–7588, 2019.

Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, 1975.

Alina Beygelzimer, Sham Kakade, and John Langford. Cover trees for nearest neighbor. In *Proc. Int. Conf. Mach. Learn.*, pages 97–104, 2006.

Robi Bhattacharjee and Kamalika Chaudhuri. When are non-parametric methods robust? In *Proc. Int. Conf. Mach. Learn.*, pages 832–841, July 2020.

Gérard Biau and Luc Devroye. *Lectures on the Nearest Neighbor Method*. Springer International Publishing, 2015.

Gérard Biau, Frédéric Cérou, and Arnaud Guyader. On the rate of convergence of the bagged nearest neighbor estimate. *J. Mach. Learn. Res.*, 11(Feb):687–712, 2010.

Leo Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, 1996.

Leo Breiman. Pasting small votes for classification in large databases and on-line. *Mach. Learn.*, 36(1):85–103, 1999.

Kamalika Chaudhuri and Sanjoy Dasgupta. Rates of convergence for nearest neighbor classification. In *Adv. Neural Inf. Process. Syst.*, volume 27, pages 3437–3445. Curran Associates, Inc., 2014.

George H Chen, Devavrat Shah, et al. *Explaining the success of nearest neighbor methods in prediction*. Now Publishers, 2018.

Thomas M Cover. Estimation by the nearest neighbor rule. *IEEE Trans. Inf. Theory*, 14(1): 50–55, 1968a.

Thomas M Cover. Rates of convergence for nearest neighbor procedures. In *Proc. Hawaii Int. Conf. Sys. Sci.*, volume 415, 1968b.

Thomas M Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory*, 13(1):21–27, 1967.

Sanjoy Dasgupta and Samory Kpotufe. Optimal rates for k-NN density and mode estimation. In *Adv. Neural Inf. Process. Syst.*, volume 27, pages 2555–2563. Curran Associates, Inc., 2014.

Sanjoy Dasgupta and Samory Kpotufe. Nearest-neighbor classification and search. In Tim Roughgarden, editor, *Beyond Worst-Case Analysis*, chapter 1. Cambridge University Press, 2019.

Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proc. 20th Ann. Symp. Comput. Geom.*, pages 253–262, 2004.

Luc Devroye, Laszlo Gyorfi, Adam Krzyzak, and Gábor Lugosi. On the strong universal consistency of nearest neighbor regression function estimates. *Ann. Statist.*, pages 1371–1385, 1994.

Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 1996.

Dheeru Dua and Casey Graff. UCI Machine Learning Repository, 2019. URL http://archive.ics.uci.edu/ml.

Jiexin Duan, Xingye Qiao, and Guang Cheng. Statistical guarantees of distributed

nearest neighbor classification. In *Adv. Neural Inf. Process. Syst.*, volume 33. Curran Associates, Inc., 2020.

Klim Efremenko, Aryeh Kontorovich, and Moshe Noivirt. Fast and Bayes-consistent nearest neighbors. In *Proc. Int. Conf. Artif. Int. Statist.*, pages 1276–1286. PMLR, 2020.

Evelyn Fix and J.L. Hodges. Discriminatory analysis: Nonparametric discrimination, consistency properties. Technical Report 4; 21-49-004, USAF School of Aviation Medicine, 1951.

Jozsef Fritz. Distribution-free exponential error bound for nearest neighbor pattern classification. *IEEE Trans. Inf. Theory*, 21(5):552–557, 1975.

Keinosuke Fukunaga and L Hostetler. Optimization of k nearest neighbor density estimates. *IEEE Trans. Inf. Theory*, 19(3):320–326, 1973.

Sébastien Gadat, Thierry Klein, Clément Marteau, et al. Classification in general finite dimensional spaces with the k-nearest neighbor rule. *Ann. Statist.*, 44(3):982–1009, 2016.

Lee-Ad Gottlieb, Aryeh Kontorovich, and Robert Krauthgamer. Efficient classification for metric data (extended abstract COLT 2010). *IEEE Trans. Inf. Theory*, 60(9):5750–5759, 2014.

Lee-Ad Gottlieb, Aryeh Kontorovich, and Pinhas Nisnevitch. Near-optimal sample compression for nearest neighbors. *IEEE Trans. Inf. Theory*, 64(6):4120–4128, 2018.

Xiaoyi Gu, Leman Akoglu, and Alessandro Rinaldo. Statistical analysis of nearest neighbor methods for anomaly detection. In *Adv. Neural Inf. Process. Syst.*, volume 32. Curran Associates, Inc., 2019.

Isabelle Guyon, Steve R Gunn, Asa Ben-Hur, and Gideon Dror. Result Analysis of the NIPS 2003 Feature Selection Challenge. In *Adv. Neural Inf. Process. Syst.*, volume 4, pages 545–552. Curran Associates, Inc., 2004.

L Gyorfi. The rate of convergence of k_n-nn regression estimates and classification rules. *IEEE Trans. Inf. Theory*, 27(3):362–364, 1981.

László Györfi and Roi Weiss. Universal consistency and rates of convergence of multiclass prototype algorithms in metric spaces. *J. Mach. Learn. Res.*, 22(151):1–25, 2021.

Peter Hall and Richard J Samworth. Properties of bagged nearest neighbour classifiers. *J. R. Stat. Soc. B*, 67(3):363–379, 2005.

Steve Hanneke, Aryeh Kontorovich, Sivan Sabato, and Roi Weiss. Universal Bayes consistency in metric spaces. *Ann. Statist.*, page to appear, 2020.

Sariel Har-Peled, Piotr Indyk, and Rajeev Motwani. Approximate nearest neighbor: Towards removing the curse of dimensionality. *Theory Comput.*, 8(1):321–350, 2012.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

Juha Heinonen. *Lectures on analysis on metric spaces*. Springer Science & Business Media, 2012.

Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proc. Symp. Theory Comput.*, pages 604–613, 1998.

Ashraf M Kibriya and Eibe Frank. An empirical comparison of exact nearest neighbour algorithms. In *Euro. Conf. Princ. Data Mining Knowledge Discov.*, pages 140–151. Springer, 2007.

Aryeh Kontorovich and Roi Weiss. A Bayes consistent 1-NN classifier. In Guy Lebanon and S. V. N. Vishwanathan, editors, *Proc. Int. Conf. Artif. Int. Statist.*, volume 38 of *Proc. Mach. Learn. Res.*, pages 480–488, San Diego, California, USA, 09–12 May 2015. PMLR.

Aryeh Kontorovich, Sivan Sabato, and Roi Weiss. Nearest-neighbor sample compression: Efficiency, consistency, infinite dimensions. In *Adv. Neural Inf. Process. Syst.*, volume 30, pages 1572–1582. Curran Associates, Inc., 2017.

L F Kozachenko and Nikolai N Leonenko. Sample estimate of the entropy of a random vector. *Probl. Inf. Transm.*, 23(2):9–16, 1987. (Russian).

Samory Kpotufe and Nakul Verma. Time-accuracy tradeoffs in kernel prediction: controlling prediction quality. *J. Mach. Learn. Res.*, 18(1):1443–1471, 2017.

Sanjeev R Kulkarni and Steven E Posner. Rates of convergence of nearest neighbor estimation under arbitrary sampling. *IEEE Trans. Inf. Theory*, 41(4):1028–1039, 1995.

Nikolai Leonenko, Luc Pronzato, and Vippal Savani. A class of Rényi information estimators for multidimensional densities. *Ann. Statist.*, 36(5):2153–2182, October 2008.

Ruiqi Liu, Ganggang Xu, and Zuofeng Shang. Distributed adaptive nearest neighbor classifier: Algorithm and theory. *arXiv preprint arXiv:2105.09788*, 2021.

Don O Loftsgaarden and Charles P Quesenberry. A nonparametric estimate of a

multivariate density function. *Ann. Math. Statist.*, 36(3):1049–1051, 1965.

Robert J Lyon, BW Stappers, Sally Cooper, JM Brooke, and JD Knowles. Fifty years of pulsar candidate selection: From simple filters to a new principled real-time classification approach. *Mon. Not. R. Astron. Soc*, 459(1):1104–1123, 2016. Data doi: 10.6084/m9.figshare.3080389.v1.

YP Mack and Murray Rosenblatt. Multivariate k-nearest neighbor density estimates. *J. Multivar. Anal.*, 9(1):1–15, 1979.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.*, 12:2825–2830, 2011.

Xingye Qiao, Jiexin Duan, and Guang Cheng. Rates of convergence for large-scale nearest neighbor classification. In *Adv. Neural Inf. Process. Syst.*, volume 32, pages 10768–10779. Curran Associates, Inc., 2019.

Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proc. ACM Int. Conf. Manag. Data*, pages 427–438, 2000.

Richard J Samworth. Optimal weighted nearest neighbour classifiers. *Ann. Statist.*, 40 (5):2733–2763, 2012.

Malcolm Slaney and Michael Casey. Locality-sensitive hashing for finding nearest neighbors. *IEEE Signal Process. Mag.*, 25(2):128–131, 2008.

Will Wei Sun, Xingye Qiao, and Guang Cheng. Stabilized nearest neighbor classifier and its statistical properties. *J. Am. Statist. Assoc.*, 111(515):1254–1265, 2016.

Jeffrey K Uhlmann. Satisfying general proximity/similarity queries with metric trees. *Inf. Process. Lett.*, 40(4):175–179, 1991.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. OpenML: Networked Science in Machine Learning. *SIGKDD Explor.*, 15(2):49–60, 2013. doi: 10.1145/2641190.2641198.

T Wagner. Convergence of the nearest neighbor rule. *IEEE Trans. Inf. Theory*, 17(5): 566–571, 1971.

Yizhen Wang, Somesh Jha, and Kamalika Chaudhuri. Analyzing the robustness of nearest neighbors to adversarial examples. In *Proc. Int. Conf. Mach. Learn.*, pages

5133–5142, 2018.

Lirong Xue and Samory Kpotufe. Achieving the time of 1-NN, but the accuracy of k-NN. In *Proc. Int. Conf. Artif. Int. Statist.*, pages 1628–1636, 2018.

Chapter 4

Density Functional Estimation with Fixed-k-Nearest Neighbors

4.1 Introduction

In this chapter, we study the problem of estimating an entropy functional of the form

$$T_f(p) := \mathbb{E}_{\mathbf{X}\sim p}[f(p(\mathbf{X}))] = \int f(p(\mathbf{x}))p(\mathbf{x})\,\mathrm{d}\mathbf{x},$$

where $f\colon \mathbb{R}_+ \to \mathbb{R}$ is a given function and p is a probability density over $\mathbb{R}^d$. Table 4.1.1 lists examples of f and the corresponding functional T_f. The goal is to estimate $T_f(p)$ based on independent and identically distributed (i.i.d.) samples $\mathbf{X}_{1:m} = (\mathbf{X}_1, \ldots, \mathbf{X}_m)$ from p by forming an estimator $\hat{T}_f^m(\mathbf{X}_{1:m})$ that converges to $T_f(p)$ in L_2 as the sample size m grows to infinity, that is,

$$\lim_{m\to\infty} \mathbb{E}\big[\big(\hat{T}_f^m(\mathbf{X}_{1:m}) - T_f(p)\big)^2\big] = 0.$$

More generally, let $f\colon \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}$ and consider a divergence functional

$$T_f(p,q) := \mathbb{E}_{\mathbf{X}\sim p}[f(p(\mathbf{X}), q(\mathbf{X}))] = \int f(p(\mathbf{x}), q(\mathbf{x}))p(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

of a pair of probability densities p and q over $\mathbb{R}^d$. Table 4.1.2 lists examples of f and

the corresponding T_f. In this case, the main problem is to construct an estimator $\hat{T}_f^{m,n}(\mathbf{X}_{1:m}, \mathbf{Y}_{1:n})$ based on i.i.d. samples $\mathbf{X}_{1:m}$ from p and $\mathbf{Y}_{1:n}$ from q, independent of each other, such that

$$\lim_{m,n\to\infty} \mathbb{E}\big[\big(\hat{T}_f^{m,n}(\mathbf{X}_{1:m}, \mathbf{Y}_{1:n}) - T_f(p,q)\big)^2\big] = 0.$$

Consistent estimation of such quantities, such as Shannon's differential entropy ($f = \ln(1/p)$), (exponentiated) Rényi α-entropies ($f = p^{\alpha-1}$), Kullback–Leibler (KL) divergence ($f = \ln(p/q)$), Hellinger distance ($f = \sqrt{q/p}$), (exponentiated) Rényi α-divergences ($f = p^{\alpha-1}q^{-\alpha}$), and Jensen–Shannon divergence (see Table 4.1.2), is a problem of considerable practical interest, having wide-ranging applications in parameter estimation (Weidemann and Stear, 1969; Wolsztynski et al., 2005), goodness-of-fit testing (Crzcgorzewski and Wirczorkowski, 1999; Girardin and Lequesne, 2017; Goria et al., 2005), quantization (Marano et al., 2007), independent component analysis (Boukouvalas et al., 2016; Kraskov et al., 2004; Learned-Miller and Fisher III, 2003), texture classification (Hero et al., 2002; Susan and Hanmandlu, 2013), design of experiments (Lewi et al., 2007; Liepe et al., 2013), pattern recognition (Hero and Michel, 1999; Lajevardi and Hussain, 2009; Neemuchwala et al., 2005; Shan et al., 2005), clustering and feature selection (Aghagolzadeh et al., 2007; Lajevardi and Hussain, 2009; Peng et al., 2005; Sotoca and Pla, 2010), and statistical inference (Giet and Lubrano, 2008). In addition, divergence estimates can be used as measures of distance between two distributions and thus can generalize distance-based algorithms for metric spaces to the space of probability distributions; see, for example, (Henderson et al., 2015; Oliva et al., 2013) and the references therein.

One of the most basic and prominent nonparametric approaches is the k-nearest neighbor (k-NN) based method, which is appealing since its hyperparameter tuning is relatively simple and is computationally efficient, especially when k is held fixed,

Table 4.1.1. Examples of functionals of one density and their estimator functions $\phi_k(u)$. A reference is given whenever an estimator already exists in the literature. The last column presents a pair of exponents (a_k, b_k) of the polynomial envelope of the estimator function $\phi_k(u)$. The constant ϵ, if any, can be chosen as an arbitrarily small positive number. For the first three examples, $k > -a_k$ is required to guarantee the existence of the corresponding inverse Laplace transform. Here, $\Psi(\alpha)$ denotes the digamma function (Korn and Korn, 2000); see also Example 4.3.2.

Name	$T_f(p) = \mathbb{E}_p[f(p)]$	$\phi_k(u) = \frac{\Gamma(k)}{u^{k-1}} \mathcal{L}^{-1}\left\{\frac{f(p)}{p^k}\right\}(u)$	(a_k, b_k)
Differential entropy (Goria et al., 2005; Kozachenko and Leonenko, 1987; Singh et al., 2003) (Examples 4.3.2, 4.3.9, 4.3.24, 4.3.29, 4.5.5)	$\mathbb{E}\left[\ln \frac{1}{p}\right]$	$\ln u - \Psi(k)$	$(-\epsilon, \epsilon)$
α-entropy (Leonenko et al., 2008) ($\alpha \geq 0$) (Examples 4.3.3, 4.3.10, 4.3.25, 4.3.30, 4.5.6)	$\mathbb{E}[p^{\alpha-1}]$	$\frac{\Gamma(k)}{\Gamma(k-\alpha+1)} \left(\frac{1}{u}\right)^{\alpha-1}$	$(1-\alpha, 1-\alpha)$
Logarithmic α-entropy ($\alpha > 0$) (Example 4.3.4)	$\mathbb{E}\left[p^{\alpha-1} \ln \frac{1}{p}\right]$	$\frac{\Gamma(k)}{\Gamma(k-\alpha+1)} u^{-\alpha+1}(\ln u - \Psi(k-\alpha+1))$	$(1-\alpha-\epsilon, 1-\alpha+\epsilon)$
Exponential (α, β)-entropy ($\alpha > 0, \beta \geq 0$) (Example 4.3.5)	$\mathbb{E}[p^{\alpha-1} e^{-\beta p}]$	$\frac{\Gamma(k)}{\Gamma(k-\alpha+1)} \frac{(u-\beta)^{k-\alpha}}{u^{k-1}} 1_{[\beta,\infty)}(u)$	$(0, 1-\alpha)$ for $k \geq \alpha$

independent of the sample sizes m and n. In this paper, we propose a new, universal design principle of a L_2-consistent k-NN based estimator for a wide class of the density functionals $T_f(p)$ and $T_f(p, q)$ based on the inverse Laplace transform, which generalizes many existing estimators which have been developed and analyzed separately. Based on the proposed mathematical framework, we establish the consistency and the rate of convergence in MSE of the density functional estimator under fairly general regularity conditions, by extending and simplifying the existing analyses of the KL estimator by Bulinski and Dimitrov (2019a,b) and Gao et al. (2018).

4.1.1 The Proposed Single-Density Functional Estimators

Suppose that a metric $\rho\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ is associated with the d-dimensional space $\mathbb{R}^d$. Given samples $\mathbf{X}_{1:m}$ and a point $\mathbf{x} \in \mathbb{R}^d$, we denote the k-NN distance of $\mathbf{x}$ from the samples by $r_{km}(\mathbf{x}) := r_k(\mathbf{x}|\mathbf{X}_{1:m})$ for $k \leq m$. Here, $r_k(\mathbf{x}|A)$ denotes the k-NN distance of $\mathbf{x}$ from a set $A \subseteq \mathbb{R}^d$, where the distance tie is broken arbitrarily. The key statistic in this paper is a normalized volume

$$U_{km}(\mathbf{x}) := U_k(\mathbf{x}|\mathbf{X}_{1:m}) := m\,\lambda(\mathbf{B}(\mathbf{x}, r_k(\mathbf{x}|\mathbf{X}_{1:m}))) \tag{4.1}$$

of the k-NN ball centered at $\mathbf{x}$ with respect to $\mathbf{X}_{1:m}$. Here and henceforth, λ denotes the Lebesgue measure over $\mathbb{R}^d$, $\mathbf{B}(\mathbf{x}, r) := \{\mathbf{y} \in \mathbb{R}^d\colon \rho(\mathbf{x}, \mathbf{y}) < r\}$ denotes the open ball of radius $r > 0$ centered at $\mathbf{x} \in \mathbb{R}^d$, and $\overline{\mathbf{B}}(\mathbf{x}, r)$ denotes the closure of $\mathbf{B}(\mathbf{x}, r)$. When the k-NN distance r_k is evaluated at one of the samples $\mathbf{x} = \mathbf{X}_i$ $(1 \leq i \leq m)$, we define it as $r_k(\mathbf{X}_i|\mathbf{X}_{1:i-1}\mathbf{X}_{i+1:m})$ to exclude the trivial zero distance. Consequently, we use the convention

$$U_{km}(\mathbf{X}_i) := (m-1)\,\lambda(\mathbf{B}(\mathbf{x}, r_k(\mathbf{x}|\mathbf{X}_{1:i-1}\mathbf{X}_{i+1:m}))).$$

Note that under this convention, we have

$$U_{km}(\mathbf{X}_m) = U_{k,m-1}(\mathbf{X}_m). \tag{4.2}$$

Let $\mathsf{G}(\alpha, \beta)$ denote the Gamma distribution with shape parameter $\alpha > 0$ and rate parameter $\beta > 0$, whose density is

$$\frac{\beta^\alpha}{\Gamma(\alpha)} u^{\alpha-1} e^{-\beta u}, \quad u \geq 0.$$

Here $\Gamma(\alpha) := \int_0^\infty x^{\alpha-1} e^{-x} \, \mathrm{d}x$ denotes the Gamma function. The following fact on the asymptotic distribution of $U_{km}(\mathbf{x})$ is well known (Goria et al., 2005; Leonenko et al., 2008; Singh et al., 2003). The proof is presented in Appendix 4.B.2 for completeness.

Proposition 4.1.1. *Suppose that $k \geq 1$ is a fixed integer, and let $\mathbf{X}_{1:m}$ be i.i.d. samples drawn from p on $\mathbb{R}^d$. Then, for almost every $\mathbf{x}$, $U_{km}(\mathbf{x})$ converges to a $\mathsf{G}(k, p(\mathbf{x}))$ random variable in distribution as m goes to infinity.*

This general convergence result is the cornerstone of the design of our estimator. To be more specific, for functionals of one density p, consider an estimator of the form

$$\hat{T}_f^{(k)}(\mathbf{X}_{1:m}) = \frac{1}{m} \sum_{i=1}^{m} \phi_k(U_{km}(\mathbf{X}_i)) \tag{4.3}$$

that depends on the samples only through the k-NN distance evaluated at each of them. As a necessary condition for the L_2-consistency of this estimator, the function ϕ_k should be chosen such that

$$\lim_{m \to \infty} \mathbb{E}[\hat{T}_f^{(k)}] = T_f(p),$$

that is, the estimator is asymptotically unbiased. On the one hand, since $\mathbf{X}_{1:m}$ are identically distributed, we have, from (4.2) and (4.3), that $\mathbb{E}[\hat{T}_f^{(k)}] = \mathbb{E}[\phi_k(U_{k,m-1}(\mathbf{X}_m))]$, and thus the desired asymptotic unbiasedness for a *fixed* k can be expressed equivalently

Table 4.1.2. Examples of functionals of two densities and their estimator functions $\phi_{kl}(u, v)$. The absolute continuity $\mathsf{P} \ll \mathsf{Q}$ is assumed implicitly unless stated otherwise. A reference is given whenever an estimator already exists in the literature. The last column presents pairs of exponents (a_{kl}, b_{kl}) and $(\tilde{a}_{kl}, \tilde{b}_{kl})$ of the polynomial envelopes of the estimator function $\phi_{kl}(u, v)$ in u and v, respectively. The constant ϵ, if any, can be chosen as an arbitrarily small positive number. For each case, $k > -a_{kl}$ and $l > -\tilde{a}_{kl}$ is required to guarantee the existence of the corresponding inverse Laplace transform.

Name	$T_f(p, q) = \mathbb{E}_p[f(p, q)]$	$\phi_{kl}(u, v) = \frac{\Gamma(k)\Gamma(l)}{u^{k-1}v^{l-1}}\mathcal{L}^{-1}\Big\{\frac{f(p,q)}{p^k q^l}\Big\}(u, v)$	(a_{kl}, b_{kl}); $(\tilde{a}_{kl}, \tilde{b}_{kl})$
KL divergence (Wang et al., 2009) (Examples 4.4.2, 4.4.10, 4.4.16, 4.5.9, 4.E.2)	$\mathbb{E}\Big[\ln \frac{p}{q}\Big]$	$\ln \frac{v}{u} + \Psi(k) - \Psi(l)$	$(-\epsilon, \epsilon)$; $(-\epsilon, \epsilon)$
α-divergence (Póczos and Schneider, 2011) ($\alpha > 0$) (Examples 4.4.3, 4.4.11, 4.4.17, 4.5.10, 4.E.3)	$\mathbb{E}\Big[\Big(\frac{p}{q}\Big)^{\alpha-1}\Big]$	$\frac{\Gamma(k)\Gamma(l)}{\Gamma(k-\alpha+1)\Gamma(l+\alpha-1)}\Big(\frac{v}{u}\Big)^{\alpha-1}$	$(1-\alpha, 1-\alpha)$; $(\alpha-1, \alpha-1)$
Logarithmic α-divergence ($\alpha > 0$) (Examples 4.4.4, 4.E.4)	$\mathbb{E}\Big[\Big(\frac{p}{q}\Big)^{\alpha-1}\ln \frac{p}{q}\Big]$	$\frac{\Gamma(k)\Gamma(l)}{\Gamma(k-\alpha+1)\Gamma(l+\alpha-1)}\Big(\frac{v}{u}\Big)^{\alpha-1} \times \Big(\ln \frac{v}{u} + \Psi(k-\alpha+1) - \Psi(l+\alpha-1)\Big)$	$(1-\alpha-\epsilon, 1-\alpha+\epsilon)$; $(\alpha-1-\epsilon, \alpha-1+\epsilon)$
Le Cam distance (Examples 4.4.5, 4.4.20, 4.E.5)	$\mathbb{E}\Big[\frac{(p-q)^2}{2p(p+q)}\Big]$	$2\binom{k+l-2}{k-1}^{-1}\Big\{\sum_{j=0}^{l-1}\binom{k+l-2}{k-1+j}\Big(-\frac{u}{v}\Big)^j - \Big(-\frac{u}{v}\Big)^{l-1}\Big(1-\frac{v}{u}\Big)^{k+l-2}1_{[v,\infty)}(u)\Big\}$	$(-k+1, l-1)$; $(-l+1, k-1)$
Entropy difference ($\mathsf{Q} \ll \mathsf{P}$) (Example 4.E.6)	$\mathbb{E}\Big[\ln \frac{1}{p} - \frac{q}{p}\ln \frac{1}{q}\Big]$	$\frac{(l-1)}{k}\frac{u}{v}(\Psi(l-1) - \ln v) - (\Psi(k) - \ln u)$	$(-\epsilon, 1)$; $(-1-\epsilon, -1+\epsilon)$
Reverse KL divergence ($\mathsf{Q} \ll \mathsf{P}$) (Example 4.E.7)	$\mathbb{E}\Big[\frac{q}{p}\ln \frac{q}{p}\Big]$	$\frac{l-1}{k}\frac{u}{v}\Big(\ln \frac{u}{v} + \Psi(l-1) - \Psi(k+1)\Big)$	$(1-\epsilon, 1+\epsilon)$; $(-1-\epsilon, -1+\epsilon)$
Jensen–Shannon divergence ($\mathsf{Q} \ll \mathsf{P}$) (Examples 4.4.6, 4.4.21, 4.E.8)	$\mathbb{E}\Big[\frac{1}{2}\ln \frac{2p}{p+q} + \frac{q}{2p}\ln \frac{2q}{p+q}\Big]$	See Example 4.4.6.	$(-k+1, l-1)$; $(-l+1, k-1)$

as

$$\lim_{m\to\infty} \mathbb{E}[\phi_k(U_{k,m-1}(\mathbf{X}_m))] = T_f(p) = \int p(\mathbf{x}) f(p(\mathbf{x}))\,\mathrm{d}\mathbf{x}. \tag{4.4}$$

On the other hand, from Proposition 4.1.1, we expect that under certain regularity conditions,

$$\begin{aligned}\lim_{m\to\infty} \mathbb{E}[\phi_k(U_{k,m-1}(\mathbf{X}_m))] &= \mathbb{E}[\phi_k(U_{k\infty}(\mathbf{X}))] \\ &= \int p(\mathbf{x}) \mathbb{E}[\phi_k(U_{k\infty}(\mathbf{x}))]\,\mathrm{d}\mathbf{x},\end{aligned} \tag{4.5}$$

where $U_{k\infty}(\mathbf{x})$ is a $\mathsf{G}(k, p(\mathbf{x}))$ random variable, independent of $\mathbf{X} \sim p$ for every $\mathbf{x}$. We choose $\phi_k(u)$ so as to equate the integrands in (4.4) and (4.5), i.e., for every $p > 0$, if $U \sim \mathsf{G}(k, p)$, then

$$\begin{aligned} f(p) &= \mathbb{E}[\phi_k(U)] \\ &= \int_0^\infty \phi_k(u) \frac{p^k}{\Gamma(k)} u^{k-1} e^{-up}\,\mathrm{d}u \\ &= \frac{p^k}{\Gamma(k)} \mathcal{L}\{u^{k-1}\phi_k(u)\}(p), \end{aligned} \tag{4.6}$$

where $\mathcal{L}\{\cdot\}$ represents the *one-sided Laplace transform* (see, e.g., (Korn and Korn, 2000, Ch. 29)), defined as

$$\mathcal{L}\{g(u)\}(p) := \int_0^\infty g(\tilde{u}) e^{-p\tilde{u}}\,\mathrm{d}\tilde{u}.$$

Rearranging the terms in (4.6), we obtain the key equation of this paper via inverse Laplace transform

$$\phi_k(u) = \frac{\Gamma(k)}{u^{k-1}} \mathcal{L}^{-1}\Big\{\frac{f(p)}{p^k}\Big\}(u), \tag{4.7}$$

which we refer to as the *estimator function* ϕ_k for f with parameter k. In general, inverse

Laplace transform $\mathcal{L}^{-1}\{\cdot\}(\cdot)$ can be obtained by the *Bromwich integral*, which is the contour integral

$$\mathcal{L}^{-1}\{f(p)\}(u) = \frac{1}{2\pi i} \lim_{T\to\infty} \int_{\gamma-iT}^{\gamma+iT} e^{pu} f(p)\,\mathrm{d}p,$$

where γ is chosen so that all singularities of $f(p)$ lie to the left of the vertical line $\mathrm{Re}(p) = \gamma$ in the complex plane and that $f(p)$ is bounded on the line (see, e.g., (Cohen, 2007, Ch. 2)). For most cases of our interest (see Tables 4.1.1 and 4.1.2), however, inverse Laplace transforms can be computed using known transforms of elementary functions (Korn and Korn, 2000), along with several properties of Laplace transform, such as linearity, time-scaling, and convolution. The reader is referred to Table 4.E.1 in Appendix 4.E for a list of elementary Laplace transforms. Note, for example, that by the linearity of the inverse Laplace transform, if ϕ_k is the estimator function for f, then the estimator function for $af + b$ is $a\phi_k + b$ for any $a, b \in \mathbb{R}$. Concrete examples of estimator functions for different choices of f are presented in Table 4.1.1. See Appendix 4.E for detailed derivation of these examples.

The main contributions of this paper, for single-density functionals, are as follows: By establishing the asymptotic unbiasedness condition in (4.4) and (4.5) of the proposed estimator (4.3), the necessity of which was first observed in the Ph.D. thesis of one of the authors (Noh, 2011, Ch. 5), and by establishing that the variance of the estimator also vanishes asymptotically, we show that the proposed estimator is L_2-consistent under mild regularity conditions on densities. The general statement (Corollary 4.3.8) capture the hardness of estimating a given functional based on k-NN statistics as a polynomial tail behavior of its corresponding inverse Laplace transform. For smooth, bounded densities, we also establish the polynomial convergence rate in mean-squared error (MSE) by carefully bounding nonasymptotic error terms. Informally, under certain

regularity conditions, we establish that

$$\mathbb{E}[(\hat{T}_f^{(k)} - T_f(p))^2] = \tilde{O}(m^{-\lambda(\sigma_p,a,k)}) + O(m^{-1/2}),$$

where σ_p is the order of smoothness of the underlying distribution p, a quantifies how much the functional T_f is affected by *high* densities (see (4.21)), and $\lambda(\sigma, a, k)$ is the bias rate exponent defined in (4.25); see Section 4.3.2 and Corollary 4.3.20 for details. For example, when the densities are sufficiently smooth, i.e., $\sigma_p \geq 1$, the rate exponent becomes $\lambda \approx 1/d$ for k sufficiently large, implying the approximate MSE rate of $\tilde{O}(m^{-1/\max\{d,2\}})$.

4.1.2 The Proposed Double-Density Functional Estimators

For functionals of two densities, we naturally extend the same idea to the Laplace transform in two dimensional spaces. For $g : \mathbb{R}_+^2 \to \mathbb{R}$, we use (u, v) and (p, q) to denote "time domain" and "frequency domain" variables, respectively, and define

$$\mathcal{L}\{g(u,v)\}(p,q) := \int_0^\infty \int_0^\infty g(\tilde{u},\tilde{v}) e^{-p\tilde{u}} e^{-q\tilde{v}} \,\mathrm{d}\tilde{u}\,\mathrm{d}\tilde{v}.$$

Note we keep dummy variables such as u and v in $\mathcal{L}\{g(u, v)\}(p, q)$ explicit, so as to avoid any confusion on which function is being transformed. We define the *estimator function* ϕ_{kl} for f with parameters (k, l), computed through the two-dimensional inverse Laplace transform, as

$$\phi_{kl}(u,v) = \frac{\Gamma(k)\Gamma(l)}{u^{k-1}v^{l-1}} \mathcal{L}^{-1}\Big\{\frac{f(p,q)}{p^k q^l}\Big\}(u,v). \tag{4.8}$$

When $T_f(p, q)$ is in the form of divergence, i.e., $f(p, q)$ is a function of p/q, the corresponding estimator function $\phi_{kl}(u, v)$ is also a function of u/v; see Proposition 4.E.1 in Appendix 4.E. Concrete examples of estimator functions for different choices of f are

presented in Table 4.1.2. See Appendix 4.E for detailed derivations of these examples. Given two sets of samples $\mathbf{X}_{1:m}$ from p and $\mathbf{Y}_{1:n}$ from q, we further define

$$V_{ln}(\mathbf{x}) := V_l(\mathbf{x}|\mathbf{Y}_{1:n}) := n\,\lambda(\mathbf{B}(\mathbf{x}, r_l(\mathbf{x}|\mathbf{Y}_{1:n}))).$$

We then propose a (k,l)-NN estimator of the form

$$\hat{T}_f(\mathbf{X}_{1:m}, \mathbf{Y}_{1:n}) = \frac{1}{m}\sum_{i=1}^{m} \phi_{kl}(U_{km}(\mathbf{X}_i), V_{ln}(\mathbf{X}_i)). \tag{4.9}$$

As in the single-density case, we establish the L_2-consistency and MSE convergence rate of our estimator (4.9) under respective regularity conditions.

Throughout the paper, we assume the Euclidean distance, i.e., $\rho(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|$, but the results will continue to hold for the p-norm ($p \geq 1$) with minor modifications; see Section 4.7 for related remarks.

Notation. We use $\varrho_d(v) := (v/\upsilon_d)^{1/d}$ to denote the radius of a d-dimensional ball of a volume v and $\upsilon_d(r) := \varrho_d^{-1}(r) = \lambda(\mathbf{B}(0,r))$ to denote the volume of ball of radius r. We further use $\upsilon_d := \upsilon_d(1) = 2^d\Gamma(1+\frac{1}{2})^d\Gamma(1+\frac{d}{2})^{-1}$ to denote the volume of the unit ball $\mathbf{B}(0,1)$. We denote the density of a random variable U as $\rho_U(u)$. We use the calligraphic letters P and Q to denote the probability measures corresponding to the density p and q, respectively, and denote the support of a density p as

$$\mathrm{supp}(p) := \{\mathbf{x} \in \mathbb{R}^d \colon\ \mathsf{P}(\mathbf{B}(\mathbf{x},r)) > 0,\ \forall r > 0\}.$$

We use $\mathsf{P} \ll \mathsf{Q}$ to denote the absolute continuity of P with respect to Q. For nonnegative functions $A(x)$ and $B(x)$ of $x \in \mathcal{X}$, we write $A(x) \lesssim_\alpha B(x)$ if there exists $C(\alpha) > 0$, depending only on some parameter α, such that $A(x) \leq C(\alpha)B(x)$ for all $x \in \mathcal{X}$. We use the standard Bachmann–Landau notation O and Θ (see, e.g., (Cormen et al., 2009)) throughout the paper, and write $f(n) = \tilde{O}(g(n))$ to represent the polylogarithmic order

$f(n) = O(g(n)(\ln g(n))^k)$ for some $k \in \mathbb{R}$. We use the shorthand notation $a \wedge b = \min\{a, b\}$ and $a \vee b = \max\{a, b\}$. Finally, 1_A stands for the indicator function of a set A.

Organization

The rest of the paper is organized as follows. Section 4.2 discusses the relevant literature and positions our contributions in that context. We analyze the proposed estimator for functionals of one density (cf. (4.3) and (4.7)) in Section 4.3 and of two densities (cf. (4.9) and (4.8)) in Section 4.4. We discuss the convergence rate of the estimators with adaptive choices of k and l in Section 4.5. We present in Section 4.6 numerical results to demonstrate the proposed estimator for a few synthetic examples. Section 4.7 concludes the paper.

4.2 Related Work

One of the most straightforward estimators of the density functional $T_f(p) = \mathbb{E}_{\mathbf{X}\sim p}[f(p(\mathbf{X}))]$ is the "plug-in" estimator that first forms a density estimate $\mathbf{x} \mapsto \hat{p}(\mathbf{x})$ from the samples $\mathbf{X}_{1:m}$, such as the standard k-NN density estimate

$$\hat{p}_{km}(\mathbf{x}) = \hat{p}(\mathbf{x}) = \frac{k/m}{\lambda(\mathbf{B}(\mathbf{x}, r_{km}(\mathbf{x})))}, \tag{4.10}$$

then plugs it in as

$$\tilde{T}_f(\hat{p}) = \frac{1}{m}\sum_{i=1}^{m} f(\hat{p}(\mathbf{X}_i)). \tag{4.11}$$

Building on the consistency of the k-NN density estimate $\hat{p}_{km}$ when k increases sub-linearly with m (Biau and Devroye, 2015; Loftsgaarden and Quesenberry, 1965), one can establish the consistency and finite-sample analysis of the plug-in estimator when $k \to \infty$ (Moon and Hero, 2014a,b; Sricharan et al., 2012, 2013). For estimating the double-density functional $T_f(p, q) = \mathbb{E}_{\mathbf{X}\sim p}[f(p(\mathbf{X}), q(\mathbf{X}))]$, Berrett and Samworth (2019)

recently proposed a weighted version of the plug-in (k, l)-NN estimators of the form

$$\tilde{T}_f(\hat{p}, \hat{q}) = \frac{1}{m} \sum_{i=1}^{m} f(\hat{p}(\mathbf{X}_i), \hat{q}(\mathbf{X}_i)), \tag{4.12}$$

with the k-NN density estimate $\hat{p}_{km}$ and the l-NN density estimate $\hat{q}_{kn}$ based on the samples $\mathbf{X}_{1:m}$ from p and $\mathbf{Y}_{1:n}$ from q, respectively. They proved its efficiency by establishing a tight local asymptotic minimax lower bound and established a corresponding central limit theorem, given that k and l of the weighted-averaged plug-in estimators grow to infinity.

For a *fixed* k, however, an appropriate "bias correction" is necessary for the plug-in estimator in (4.11) to be asymptotically unbiased, since the fixed-k-NN density estimator in (4.10) is not consistent for a finite k. A fixed-k plug-in estimator with bias correction was first studied by Kozachenko and Leonenko (1987), who applied 1-NN distances to estimate differential entropies of densities on $\mathbb{R}^d$ based on an idea of Dobrushin (1958), and established the L_2-consistency of their estimator. Subsequently, Singh et al. (2003) and Goria et al. (2005) generalized the 1-NN Kozachenko–Leonenko estimator to $k \geq 1$ as

$$\begin{aligned} \hat{T}_{\mathsf{KL}}^{(k)}(\mathbf{X}_{1:m}) &= \tilde{T}_f(\hat{p}_{km}) + \ln k - \Psi(k) \\ &= \frac{1}{m} \sum_{i=1}^{m} \ln \frac{1}{\hat{p}_{km}(\mathbf{X}_i)} + \ln k - \Psi(k), \end{aligned} \tag{4.13}$$

where $\Psi(x) := \Gamma'(x)/\Gamma(x)$ denotes the digamma function (Korn and Korn, 2000). As the canonical fixed-k density functional estimator, the Kozachenko–Leonenko estimator has been investigated extensively in the literature. Beyond the L_2-consistency, Tsybakov and van der Meulen (1996) first established $\sqrt{m}$-consistency, i.e., the L_2-convergence rate of $O(m^{-1})$, of a truncated version of the 1-NN Kozachenko–Leonenko estimator in $\mathbb{R}$, which was extended by Gao et al. (2018) to $k \geq 1$ and $d \geq 1$. Some recent

developments include a central limit theorem (Delattre and Fournier, 2017), results on large-k behavior (Berrett et al., 2019), and minimax optimality (Han et al., 2020; Jiao et al., 2018).

Along the same line, L_2-consistent fixed-k or fixed-(k, l) plug-in estimators with proper additive or multiplicative bias correction were proposed[1] for KL divergence (Wang et al. (2009)), Rényi entropies (Leonenko et al. (2008)), Rényi divergences (Póczos and Schneider (2011)), and several other divergences of a specific polynomial form (Póczos et al. (2012)). These plug-in estimators can be expressed in general as

$$\tilde{T}_f^{\mathsf{aff}}(\hat{p}) = a_k \tilde{T}_f(\hat{p}) + b_k, \tag{4.14}$$

or

$$\tilde{T}_f^{\mathsf{aff}}(\hat{p}, \hat{q}) = a_{kl} \tilde{T}_f(\hat{p}, \hat{q}) + b_{kl}, \tag{4.15}$$

where $\hat{p}$ is the fixed-k-NN density estimator from $\mathbf{X}_{1:m}$ in (4.10), $\hat{q}$ is the fixed-l-NN density estimate similarly obtained from $\mathbf{Y}_{1:n}$, and (a_k, b_k) and (a_{kl}, b_{kl}) determine functional-specific bias correction, respectively. Many density functionals beyond the special examples mentioned earlier, however, do not allow such affine bias correction. For example, a plug-in estimator for the logarithmic α-entropy in Table 4.1.1 cannot be made unbiased, even asymptotically, by any affine bias correction.

A more general approach to correcting bias of the fixed-k plug-in estimator was

[1] As pointed out in (Pál et al., 2010), there are slight errors in the original analyses in (Goria et al., 2005; Kozachenko and Leonenko, 1987; Leonenko et al., 2008; Wang et al., 2009) when invoking asymptotic theory to establish L_2-consistency. Correct proofs were given later in (Bulinski and Dimitrov, 2019a,b; Leonenko and Pronzato, 2010).

proposed by Singh and Póczos (2016) as

$$\tilde{T}_{b\circ f}(\hat{p}) = \frac{1}{m}\sum_{i=1}^{m} b_{km}(f(\hat{p}_{km}(\mathbf{X}_i))), \tag{4.16}$$

which obviously subsumes affine bias correction. This estimator was shown to be L_2-consistent for a fixed k with definite convergence rate if there exists a bias-correcting function b_{km} that satisfies

$$\mathbb{E}[b_{km}(f(\hat{p}_{km}(\mathbf{x})))] = \mathbb{E}[f(\bar{p}_{km}(\mathbf{x}))] \tag{4.17}$$

for every m and any underlying density p, and for P-a.e. $\mathbf{x}$, where

$$\bar{p}_{km}(\mathbf{x}) = \frac{\mathsf{P}(\mathbf{B}(\mathbf{x}, r_{km}(\mathbf{x})))}{\lambda(\mathbf{B}(\mathbf{x}, r_{km}(\mathbf{x})))}$$

is the average density over the k-NN ball $\mathbf{B}(\mathbf{x}, r_{km}(\mathbf{x}))$. Despite the general form of this estimator, however, the existence of b_{km} satisfying the stringent condition of equality in (4.17) for every m could be established only for differential entropy (and only for KL divergence in case of functionals of two densities).

In contrast to the existing literature, our estimator

$$\begin{aligned}\hat{T}_f^{(k)}(\mathbf{X}_{1:m}) &= \frac{1}{m}\sum_{i=1}^{m} \phi_k(U_{km}(\mathbf{X}_i)) \\ &= \frac{1}{m}\sum_{i=1}^{m} \phi_k\Big(\frac{k}{\hat{p}_{km}(\mathbf{X}_i)}\Big)\end{aligned} \tag{4.18}$$

bypasses the whole bias correction issue of the plug-in approach by specifying the estimator function ϕ_k directly via the inverse Laplace transform (4.7). Here, we identified that $U_{km}(\mathbf{x}) = k/\hat{p}_{km}(\mathbf{x})$ by the respective definitions in (4.1) and (4.10). Our approach naturally unifies all existing estimators of the form (4.14) or (4.15), and finds new estima-

tors for logarithmic entropies and divergences that cannot be obtained even in the most general bias-corrected form (4.16) of the traditional plug-in estimator (4.11). For example, our estimator for the logarithmic α-entropy ($f(p) = p^{\alpha-1}\ln(1/p)$) is characterized by the estimator function

$$\begin{aligned}\phi_k(u) &= \phi_k\Big(\frac{k}{p}\Big) \\ &= \frac{\Gamma(k)}{\Gamma(k-\alpha+1)} k^{-\alpha+1} p^{\alpha-1}\Big(\ln\frac{k}{p} - \Psi(k-\alpha+1)\Big),\end{aligned} \tag{4.19}$$

which cannot be expressed as a function $b_{km}(f(p))$ for some b_{km}.

We comment on how analysis techniques of the proposed estimators are related to those in the literature. Through the design of our estimator functions (4.7) and (4.8) via inverse Laplace transform, we can naturally extend and simplify existing analyses for differential entropy and KL divergence by Bulinski and Dimitrov (2019a,b), and establish the asymptotic unbiasedness of our estimators (4.3) and (4.9) for a general functional. By adapting the nonasymptotic analysis for differential entropy in Gao et al. (2018), we can also establish the bias convergence rate of the estimator for a general functional, but without truncation. For variance analysis, we deviate from the aforementioned work (Bulinski and Dimitrov, 2019a,b; Gao et al., 2018) for simplicity and deploy a technique for the Euclidean space used by Singh and Póczos (2016); see also (Biau and Devroye, 2015, Ch. 7). Note, however, that the established variance results of our estimator continue to hold under the p-norm; see Remark 4.3.12. Our consistency analysis (unbiasedness and vanishing variance) strengthens and simplifies many existing ones including those for Rényi entropies (Leonenko et al., 2008), Rényi divergences (Póczos and Schneider, 2011), and divergences of polynomial form (Póczos et al., 2012). The convergence rates for the functionals in Tables 4.1.1 and 4.1.2 are established in this paper for the first time, except the Kozachenko–Leonenko estimator (Gao et al., 2018; Jiao et al., 2018; Singh and Póczos, 2016; Tsybakov and van der Meulen, 1996) and the

KL divergence estimator (Singh and Póczos, 2016).

In a different direction of investigation, kernel density estimator (KDE)-based approaches have been widely studied in the literature for estimation of smooth density functionals, which also include many of the examples presented in Sections IV and VI as special cases. Birge and Massart (1995) established a minimax optimal rate $O(m^{-\frac{8\sigma}{d+4\sigma}} + m^{-1})$ on convergence rates in MSE of estimators of certain integral functionals involving the density and its derivatives under Hölder smoothness of order σ (Definition 4.3.14) on the density and demonstrated that the parametric rate $O(1/m)$ is achievable if the density is sufficiently smooth, say, $\sigma \geq d/4$. For estimating polynomial divergence functionals, Krishnamurthy et al. (2014) proposed plug-in estimators corrected through estimating higher-order terms in the von Mises expansions, which may require computationally demanding numerical integration, and established a minimax lower bound $\Omega(m^{-\frac{8\sigma}{4\sigma+d}} + m^{-1})$ under Hölder smoothness of order $\sigma > 0$. Kandasamy et al. (2015) generalized this approach to more general functionals and mutual information and established similar rates. In another line of work, extending the boundary-corrected plug-in estimator for mutual information of (Liu et al., 2012), Singh and Póczos (2014a,b) established the MSE rate $O(m^{-\frac{2\sigma}{\sigma+d}} + m^{-1})$ for a kernel-based plug-in estimator of a class of density functionals under certain regularity conditions; we remark that this approach commonly requires a prior knowledge on the support.

Convergence of k-NN distance-based estimators of density functionals can be improved by using the so-called "ensemble method", where a convex combination of estimators with different k values is used. Moon et al. (2017) studied the ensemble method for estimation of the mutual information between two continuous random variables, and demonstrated that under certain broad regularity conditions on the density, the optimal convex combination, which can be computed by solving a convex optimization problem, yields the *parametric* MSE rate $O(1/m)$ provided that the density is sufficiently smooth. In a similar spirit, Moon et al. (2018), Noshad et al. (2017), and Wisler et al.

(2018) obtained the MSE rate $O(1/m)$ for estimating the KL divergence, f-divergences, and a wider class of density functionals including f-divergences, respectively, using the ensemble method. Analyzing the ensemble version of the proposed estimators is beyond the scope of this paper.

We finally remark that Nguyen et al. (2010) studied the estimation of f-divergences through minimization of empirical risk, by formulating the problem as a convex program. They established convergence rates when the likelihood ratio between the two distributions belongs to a reproducing kernel Hilbert space. It seems, however, quite nontrivial to compare these assumptions with those on smoothness used in the present work.

4.3 Functionals of One Density

Recall that we define the estimator function $\phi_k \colon \mathbb{R}_+ \to \mathbb{R}$ for a given $f \colon \mathbb{R}_+ \to \mathbb{R}$, with parameter $k \in \mathbb{N}$ as

$$\phi_k(u) = \frac{\Gamma(k)}{u^{k-1}} \mathcal{L}^{-1}\Big\{ \frac{f(p)}{p^k} \Big\}(u), \tag{4.7}$$

whenever the inverse Laplace transform exists, and then define the estimator as

$$\hat{T}_f^{(k)}(\mathbf{X}_{1:m}) = \frac{1}{m} \sum_{i=1}^{m} \phi_k(U_{km}(\mathbf{X}_i)). \tag{4.3}$$

Remark 4.3.1. One can check that, for all the examples in Table 4.1.1,

$$\lim_{k\to\infty} \phi_k\Big(\frac{k}{p}\Big) = f(p) \tag{4.20}$$

for each $p > 0$. In light of (4.18), this observation heuristically indicates that our estimator becomes closer to the plug-in estimator (4.11) as we use larger, fixed k. This

observation is consistent with our intuition that we do not need any bias correction for the plug-in estimator with very large k, since the plugged-in k-NN density estimate (4.10) becomes consistent as $k \to \infty$ in the sample limit (Loftsgaarden and Quesenberry, 1965).

To analyze the proposed estimator for general functionals $T_f(p)$ in a unified manner, we abstract polynomial tail behaviors of each estimator function $\phi_k(u)$ as $u \downarrow 0$ and $u \uparrow \infty$ by a pair of constants $(a_k, b_k) \in \mathbb{R}^2$ such that $|\phi_k(u)| \lesssim \psi_{a_k,b_k}(u)$, where we define a piecewise polynomial function $\psi_{a,b} \colon \mathbb{R}_+ \to \mathbb{R}$ for $a, b \in \mathbb{R}$ as

$$\psi_{a,b}(u) := \begin{cases} u^a & \text{if } 0 < u \leq 1, \\ u^b & \text{if } u > 1. \end{cases} \tag{4.21}$$

Note that as a gets larger and b gets smaller, the piecewise polynomial function $\psi_{a,b}(u)$ decays faster as $u \downarrow 0$ and as $u \uparrow \infty$, respectively. Therefore, a and b quantify the amount of contribution of low and high density values to the estimator function $\phi_k(u)$, respectively. Consistent with the observation that such extreme density values typically make the density functional estimation problem harder, we will establish stronger statements for functionals with larger a and smaller b. Below we present the estimator functions for a few representative functionals.

Example 4.3.2 (Differential entropy (Kozachenko and Leonenko, 1987))**.** *For $f(p) = \ln(1/p)$ and any $k \geq 1$, we can compute, as detailed in Example 4.E.2 in Appendix 4.E,*

$$\phi_k(u) = \ln u - \Psi(k).$$

Note that we can write $\Psi(k) = H_{k-1} - \gamma$ for $k \in \mathbb{N}$, where $H_k = \sum_{i=1}^{k}(1/i)$ denotes the k-th harmonic number and $\gamma := \lim_{k\to\infty}(H_k - \ln k)$ denotes the Euler–Mascheroni constant (Korn

and Korn, 2000). As a bound on the estimator function $\phi_k(u)$, we consider

$$|\phi_k(u)| \lesssim |\ln u| + 1 \lesssim \psi_{-\epsilon,\epsilon}(u)$$

for any arbitrarily small $\epsilon > 0$ throughout the paper. A finer analysis without relying on the polynomial bound $\psi_{-\epsilon,\epsilon}(u)$ may lead to a marginal improvement in the resulting performance guarantee (Bulinski and Dimitrov, 2019a,b; Gao et al., 2018), but we do not pursue that in this paper.

Example 4.3.3 (α-entropy (Leonenko et al., 2008))**.** *For $f(p) = p^{\alpha-1}$ ($\alpha \geq 0$), we refer to the density functional $T_f(p) = \int p^\alpha(\mathbf{x})\,\mathrm{d}\mathbf{x}$ as the* α-entropy*. In the literature, this functional appears in Rényi (1961) entropy $h_\alpha(p) = (\ln T_f(p))/(1-\alpha)$ and Havrda and Charvat (1967) or Tsallis (1988) entropy $\tilde{h}_\alpha(p) = (1 - T_f(p))/(\alpha - 1)$. For any $k \in \mathbb{N}$ such that $k > \alpha - 1$, we can compute, as verified in Example 4.E.3 in Appendix 4.E,*

$$\phi_k(u) = \frac{\Gamma(k)}{\Gamma(k-\alpha+1)}\Big(\frac{1}{u}\Big)^{\alpha-1},$$

which allows the tight polynomial bound

$$|\phi_k(u)| \lesssim \psi_{1-\alpha,1-\alpha}(u).$$

Example 4.3.4 (Logarithmic α-entropy)**.** *For $f(p) = p^{\alpha-1}\ln(1/p)$ ($\alpha > 0$), we refer to the density functional $T_f(p) = \int p^\alpha(\mathbf{x}) \ln(1/p(\mathbf{x}))\,\mathrm{d}\mathbf{x}$ as the* logarithmic α-entropy*. For any $k \in \mathbb{N}$ such that $k > \alpha - 1$, we can compute, as verified in Example 4.E.4 in Appendix 4.E,*

$$\phi_k(u) = \frac{\Gamma(k)}{\Gamma(k-\alpha+1)} u^{-\alpha+1}(\ln u - \Psi(k-\alpha+1)),$$

and we consider

$$|\phi_k(u)| \lesssim u^{-a+1}(|\ln u| + 1) \lesssim \psi_{1-\alpha-\epsilon,1-\alpha+\epsilon}$$

for any arbitrarily small $\epsilon > 0$ as its polynomial bound.

Example 4.3.5 (Exponential (α, β)-entropy)**.** *For $f(p) = p^{\alpha-1}e^{-\beta p}$ ($\alpha > 0$, $\beta \geq 0$), we refer to the density functional $T_f(p) = \int p^{\alpha}(\mathbf{x})e^{-\beta p(\mathbf{x})}\,\mathrm{d}\mathbf{x}$ as the* exponential (α, β)-entropy*. For any $k \in \mathbb{N}$ such that $k > \alpha - 1$, we can compute*

$$\phi_k(u) = \frac{\Gamma(k)}{\Gamma(k-\alpha+1)} \frac{(u-\beta)^{k-\alpha}}{u^{k-1}} 1_{[\beta,\infty)}(u)$$

using time shifting property of Laplace transform from the estimator function expression of the α-entropy. The estimator function ϕ_k can be bounded as

$$|\phi_k(u)| \lesssim \psi_{0,1-\alpha}(u)$$

for $k \geq \alpha$ and cannot be bounded by a piecewise polynomial function if $k < \alpha$.

In our subsequent analysis, regularity conditions for the consistency and convergence rate of the proposed estimator depend on k and f via the lower tail exponent a and the upper tail exponent b. By (4.18), extreme values of $\hat{p}_{km}$ are amplified more via ϕ_k as a decreases and and b increases. Hence, intuitively, when a is large and b is small, the regularity conditions are milder and the estimator converges faster.

4.3.1 Consistency

Focusing solely on the asymptotic behavior of our estimator, we can establish the L_2-consistency for general functionals under mild assumptions on densities. To state the results rigorously, we first define certain technical conditions. For future use in Section 4.4.1 for functionals of two densities, we state the conditions in terms of two densities p and $\tilde{p}$ such that $\mathsf{P} \ll \tilde{\mathsf{P}}$. Later, we identify $\tilde{p}$ as the density p for samples $\mathbf{X}_{1:m}$ or the density q for samples $\mathbf{Y}_{1:n}$.

For the sake of easy analysis of density functional estimators, the standard

simplifying assumptions are global upper- and lower-boundedness on the underlying density p, i.e., there exist $c > 0$ and $C > 0$ such that $c \leq p(\mathbf{x}) \leq C$ for any $\mathbf{x} \in \mathrm{supp}(p)$; note that the boundedness of the support follows from the lower boundedness of the density. In what follows, to establish the asymptotic consistency of the proposed estimators for a larger class of densities, we will consider weaker conditions than the boundedness assumptions, similar to those in (Bulinski and Dimitrov, 2019a,b).

For each $r > 0$, we define the local maximal operator M_r on $\mathbb{R}^d$ for a density p by

$$M_r p(\mathbf{x}) := \sup_{r' \in (0,r]} \frac{\mathsf{P}(\mathbf{B}(\mathbf{x}, r'))}{\lambda(\mathbf{B}(\mathbf{x}, r'))}.$$

Similarly, for each $r > 0$, we define the local minimal operator m_r on $\mathbb{R}^d$ for a density p by

$$m_r p(\mathbf{x}) := \inf_{r' \in (0,r]} \frac{\mathsf{P}(\mathbf{B}(\mathbf{x}, r'))}{\lambda(\mathbf{B}(\mathbf{x}, r'))}.$$

For each $r > 0$, $\mathbf{x} \mapsto M_r p(\mathbf{x})$ and $\mathbf{x} \mapsto m_r p(\mathbf{x})$ are lower- and upper-semicontinuous, respectively, and so are Borel measurable (Bulinski and Dimitrov, 2019a,b). In particular, $M_r p(\mathbf{x})$ and $m_r p(\mathbf{x})$ are pointwise upper and lower bounds, respectively, on the density p.

Given a non-decreasing function $\xi \colon \mathbb{R}_+ \to \mathbb{R}_+$, for densities p and $\tilde{p}$, we define the functionals

$$W(p, \tilde{p}; \vartheta, r) := \int p(\mathbf{x}) (M_r \tilde{p}(\mathbf{x}))^{\vartheta} \, \mathrm{d}\mathbf{x},$$
$$w(p, \tilde{p}; \xi, \vartheta, r) := \int p(\mathbf{x}) \xi((m_r \tilde{p}(\mathbf{x}))^{-\vartheta}) \, \mathrm{d}\mathbf{x},$$

and

$$R(p, \tilde{p}; \xi, \vartheta, r) := \iint_{\rho(\mathbf{x},\mathbf{y})>r} p(\mathbf{x})\tilde{p}(\mathbf{y})\xi(\upsilon^{\vartheta}(\rho(\mathbf{x}, \mathbf{y}))) \, \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{y}$$

for each $\vartheta > 0$ and $r > 0$. Here we define these quantities with possibly different densities p and $\tilde{p}$ for the future use with double-density functionals; for single-density functionals, the readers can simply assume $p = \tilde{p}$. In place of the upper- and lower-boundedness assumptions on the density $\tilde{p}$, we will impose the finiteness of the expected values $W(p, \tilde{p}; \vartheta, r)$ and $w(p, \tilde{p}; \xi, \vartheta, r)$, respectively. Further, $R(p, \tilde{p}; \xi, \vartheta, r)$ roughly quantifies how fast p and $\tilde{p}$ decay to zero in there tails. Observe that $R(p, \tilde{p}; \xi, \vartheta, r) \to 0$ as $r \to \infty$. Intuitively, as the tails of p and $\tilde{p}$ decay faster, the speed of convergence of $R(p, \tilde{p}; \xi, \vartheta, r)$ will be faster. In particular, if both p and $\tilde{p}$ have bounded support, then $R(p, \tilde{p}; \xi, \vartheta, r) = 0$ for r sufficiently large. Note further that W, w, and R become larger as ϑ increases.

Given $k \in \mathbb{N}$ and $(a, b) \in \mathbb{R}^2$, consider the following conditions.

($\mathbf{U}_{p\tilde{p}}$; k, a) Either $a \geq 0$, or if $a < 0$, then there exists $r > 0$ such that $W(p, \tilde{p}; k, r) < \infty$.

($\mathbf{L}_{p\tilde{p}}$; ξ, b) Either $b \leq 0$, or if $b > 0$, then there exists $r > 0$ such that $w(p, \tilde{p}; \xi, b, r) < \infty$ and

$$\limsup_{m\to\infty} \xi(m^b) R\big(p, \tilde{p}; \xi, b, \varrho\big(\frac{\kappa_m}{m}\big)\big) < \infty \tag{4.22}$$

for some κ_m such that $\kappa_m/m \to \infty$ and $(\ln \kappa_m)/m \to 0$ as $m \to \infty$.

Recall that the polynomial tail exponents a and b of the the k-NN estimator function (4.18) of a given density functional quantify the amount of contribution of high and low density values to the estimator, respectively. Hence, a is coupled with W that captures the upper boundedness of the density, while b is pertinent to w and R that quantify the lower boundedness. We note that as a gets larger, k gets smaller, and b gets

smaller, conditions **($\mathbf{L}_{p\tilde{p}}$; ξ, b)** and **($\mathbf{U}_{p\tilde{p}}$; k, a)** become weaker, thus encompassing a larger class of densities.

Let Ξ be the class of non-decreasing functions $\xi\colon \mathbb{R}_+ \to \mathbb{R}_+$ such that $\xi(t)/t \to \infty$ as $t \to \infty$, that $\xi(t_1 t_2) \leq \xi(t_1)\xi(t_2)$ for any $x, y > t_0$ for some $t_0 \in \mathbb{R}_+$, and that $\omega(\xi) := \inf\{\eta > 1\colon \xi(t)/t^\eta \to 0 \text{ as } t \to \infty\} < \infty$. For example, $\xi_1(t) = (t \ln t) \vee 0 \in \Xi$ with $t_0 = e$ and $\omega(\xi_1) = 1$, and $\xi_2(t) = t^\alpha \in \Xi$ for $\alpha > 1$ with $t_0 = 0$ and $\omega(\xi_2) = \alpha$.

We are now ready to state the L_2-consistency results. We show separately that the bias and variance converge to zero under certain regularity conditions. Note that all estimator functions presented in Table 4.1.1 are continuous. Throughout, we consider a fixed $(a, b) \in \mathbb{R}^2$ for a target functional $T_f(\cdot)$ that satisfies $|\phi_k(u)| \lesssim \psi_{a,b}(u)$, provided that the estimator function $\phi_k(u)$ exists for $k > -a$.

Theorem 4.3.6 (Vanishing bias)**.** *For a target functional $T_f(\cdot)$, if the estimator function ϕ_k is continuous and the underlying density p satisfies **($\mathbf{U}_{pp}$; k, a)** and **($\mathbf{L}_{pp}$; ξ, b)** with some function $\xi \in \Xi$, then the estimator* (4.3) *with fixed $k > -\omega(\xi)a$ is asymptotically unbiased.*

Theorem 4.3.7 (Vanishing variance)**.** *For a target functional $T_f(\cdot)$, if the underlying density p satisfies **($\mathbf{U}_{pp}$; k, a)** and **($\mathbf{L}_{pp}$; ξ, b)** with $\xi(t) = t^2$, the variance of the estimator* (4.3) *with fixed $k > -2a$ converges to zero as $m \to \infty$.*

Combining Theorems 4.3.6 and 4.3.7, the L_2-consistency readily follows as a corollary.

Corollary 4.3.8 (Consistency)**.** *For a target functional $T_f(\cdot)$, if the estimator function ϕ_k is continuous and the underlying density p satisfies **($\mathbf{U}_{pp}$; k, a)** and **($\mathbf{L}_{pp}$; ξ, b)** with $\xi(t) = t^2$, then the estimator* (4.3) *with fixed $k > -2a$ is L_2-consistent.*

In the following examples, we illustrate how Corollary 4.3.8 can be instantiated for a few representative functionals.

Example 4.3.9 (Differential entropy; Example 4.3.2 contd.)**.** *Recall that for any $k \in \mathbb{N}$, $|\phi_k(u)| \lesssim \psi_{-\epsilon,\epsilon}(u)$ for arbitrarily small $\epsilon > 0$. By Corollary 4.3.8, the estimator* (4.3) *is L_2-consistent if the underlying density p satisfies that* ***($\mathbf{U}_{pp}$; $k, -\epsilon$)*** *and* ***($\mathbf{L}_{pp}$; ξ, ϵ)*** *with $\xi(t) = t^2$ for some $\epsilon > 0$. We note that the condition* (4.22) *in* ***($\mathbf{L}_{pp}$; ξ, ϵ)*** *can be relaxed to a milder condition in which there exist some $\delta, R > 0$ such that*

$$\iint_{\rho(\mathbf{x},\mathbf{y})>R} p(\mathbf{x})p(\mathbf{y})|\ln \upsilon(\rho(\mathbf{x},\mathbf{y}))|^{\delta}\,\mathrm{d}\mathbf{x}\,\mathrm{d}\mathbf{y} < \infty$$

by performing a similar analysis based on the upper bound $|\phi_k(u)| \lesssim |\ln u| + 1$, i.e., without invoking the polynomial bound $\psi_{-\epsilon,\epsilon}(u)$ for an arbitrarily small $\epsilon > 0$. This recovers a similar result reported in (Bulinski and Dimitrov, 2019b).

Example 4.3.10 (α-entropy; Example 4.3.3 contd.)**.** *Recall that for any $k \in \mathbb{N}$, $|\phi_k(u)| \lesssim \psi_{1-\alpha,1-\alpha}(u)$. For $\alpha > 1$, since $b = 1 - \alpha < 0$, the estimator with fixed $k > 2(\alpha - 1)$ is L_2-consistent if p satisfies* ***($\mathbf{U}_{pp}$; k, a)****, which slightly generalizes the upper-boundedness condition and the requirement $k > 2\alpha - 1$ assumed in Leonenko et al. (2008). For $\alpha < 1$, since $a = 1 - \alpha > 0$, the estimator with fixed $k \geq 1$ is L_2-consistent if p satisfies* ***($\mathbf{L}_{pp}$; ξ, b)*** *with $\xi(t) = t^2$, for examples, if p is bounded away from zero and supported over a hyperrectangle. We remark that Leonenko and Pronzato (2010) reported the L_2-consistency of the estimator for densities satisfying alternate conditions when $\alpha < 1$.*

Proof of Theorem 4.3.6 (vanishing bias)

If the estimator function ϕ_k is continuous, by the continuous mapping theorem and Proposition 4.1.1, we have the convergence of the statistic $\phi_k(U_{k,m-1}(\mathbf{X}_m))$ to $\phi_k(U_{k\infty}(\mathbf{X}))$ in distribution as $m \to \infty$, where $U_{k\infty}(\mathbf{x})$ is a $\mathsf{G}(k, p(\mathbf{x}))$ random variable, independent of $\mathbf{X} \sim p$ for P-a.e. $\mathbf{x}$. Hence, if the sequence of random variables $(\phi_k(U_{k,m-1}(\mathbf{X}_m)))_{m\geq 1}$ is uniformly integrable, we readily establish the asymptotic unbi-

asedness:

$$\begin{aligned}\lim_{m\to\infty}\mathbb{E}[\hat{T}_f^{(k)}(\mathbf{X}_{1:m})] &= \lim_{m\to\infty}\mathbb{E}[\hat{p}_k(U_{k,m-1}(\mathbf{X}_m))]\\ &= \mathbb{E}[\phi_k(U_{k\infty}(\mathbf{X}))] = T_f(p).\end{aligned}$$

To show the uniform integrability of $(\phi_k(U_{k,m-1}(\mathbf{X}_m)))_{m\geq 1}$, we invoke the following lemma.

Lemma 4.3.11 (De la Vallée Poussin theorem (Borkar, 1995, Theorem 1.3.4))**.** *A collection of random variables $(X_i)_{i\in I}$ is uniformly integrable if and only if there exists a non-decreasing function $\xi\colon \mathbb{R}_+ \to \mathbb{R}_+$ such that $\sup_{i\in I}\mathbb{E}[\xi(|X_i|)] < \infty$ and $\xi(t)/t \to \infty$ as $t \to \infty$.*

Observe that we have

$$\begin{aligned}&\mathbb{E}[\xi(|\phi_k(U_{k,m-1}(\mathbf{X}_m))|)] = \int p(\mathbf{x})\mathbb{E}[\xi(|\phi_k(U_{k,m-1}(\mathbf{x}))|)]\,\mathrm{d}\mathbf{x}\\ &\lesssim \int p(\mathbf{x})\mathbb{E}[\xi(\psi_{a,b}(U_{k,m-1}(\mathbf{x})))]\,\mathrm{d}\mathbf{x}\\ &= \int p(\mathbf{x})\int_0^\infty \xi(\psi_{a,b}(u))\,\mathrm{d}F_{km}(u|\mathbf{x})\,\mathrm{d}\mathbf{x}.\end{aligned}$$

Since $\xi \in \Xi$, we have $-\int_0^1 u^k\,\mathrm{d}\xi(u^{a\wedge 0}) < \infty$ for $k > -\omega(\xi)a$ and $\int_0^\infty e^{-t}\xi(t^{b\vee 0})\,\mathrm{d}t < \infty$, and thus we can apply Lemma 4.B.19 in Appendix 4.B.4, which yields

$$\limsup_{m\to\infty}\mathbb{E}[\xi(|\phi_k(U_{k,m-1}(\mathbf{X}_m))|)] < \infty.$$

This ensures the uniform integrability of $(\phi_k(U_{k,m-1}(\mathbf{X}_m)))_{m\geq 1}$ by the de la Vallée Poussin theorem (Lemma 4.3.11), and thus concludes the proof. □

Proof of Theorem 4.3.7 (vanishing variance)

By Lemma 4.B.27 for the Euclidean space $(\mathbb{R}^d, \|\cdot\|)$, we have

$$\mathrm{Var}(\hat{T}_f^{(k)}) \le \frac{2(1+k\gamma_d)}{m}\{(2k+1)\mathbb{E}[\phi_k^2(U_{k,m-1}(\mathbf{X}_m))] + 2k\mathbb{E}[\phi_k^2(U_{k+1,m-1}(\mathbf{X}_m))]\},$$

where γ_d is a constant which only depends on d; see Lemma 4.B.27. Since $\xi(t) = t^2$ and $k > -2a$ imply that $-\int_0^1 u^k \,\mathrm{d}\xi(u^{a\wedge 0}) < \infty$ and $\int_0^\infty e^{-t}\xi(t^{b\vee 0})\,\mathrm{d}t < \infty$, we can apply Lemma 4.B.19, which ensures for $k' \in \{k, k+1\}$ that

$$\limsup_{m\to\infty} \mathbb{E}[\phi_k^2(U_{k',m-1}(\mathbf{X}_m))] < \infty.$$

It establishes $\mathrm{Var}(\hat{T}_f^{(k)}) = O(m^{-1})$ for m sufficiently large. □

Remark 4.3.12. The variance analysis relies on the Efron–Stein inequality (Lemma 4.B.28) and a covering lemma (Lemma 4.B.29) that only applies to the Euclidean space; see Appendix 4.B.6. An idea for the generic variance bound (Lemma 4.B.27) first appeared in Singh and Póczos (2016) as a generalization of a technique for analyzing the 1-NN Kozachenko–Leonenko estimator by Biau and Devroye (2015, Ch. 7), and has been employed in the literature to bound the variance of k-NN based estimators; see, e.g., Moon et al. (2017). We note that one can attain the same rate (up to polylogarithmic factors) under the p-norm, by instead adapting the analysis in Gao et al. (2018). As it demands a rather involved argument to bound a covariance term, however, we present a simpler approach in this paper.

4.3.2 Convergence Rates for Smooth, Bounded Densities

So far, we have established the L_2-consistency of the proposed estimator for general functionals under mild assumptions on densities. Under rather stronger assumptions such as smoothness and boundedness, we can actually establish the conver-

gence rate of the proposed estimator in MSE. Specifically, we consider certain regularity conditions adapted from (Gao et al., 2018).

First, we assume that

($\mathbf{U}_p$) there exists $0 < C_p < \infty$ such that $p(\mathbf{x}) \leq C_p$ almost everywhere (a.e.).

Further, we impose a few conditions related to lower-boundedness of the density, that is,

($\mathbf{L1}_p$) there exists $c_p > 0$ such that $p(\mathbf{x}) \geq c_p$ for $\mathbf{x} \in \mathrm{supp}(p)$,

($\mathbf{L2}_p$) the support of p is bounded, and

($\mathbf{L3}_p$) there exists $r > 0$ such that

$$\eta_p := \inf_{\mathbf{x} \in \mathrm{supp}(p)} \inf_{r' \in (0,r]} \frac{\lambda(\mathbf{B}(\mathbf{x}, r') \cap \mathrm{supp}(p))}{\lambda(\mathbf{B}(\mathbf{x}, r'))} > 0.$$

The last condition **($\mathbf{L3}_p$)** is called the (η_p, r)-regularity of $\mathrm{supp}(\mu)$ in the literature (Audibert et al., 2007).

Remark 4.3.13. The upper-boundedness condition **($\mathbf{U}_p$)** implies the condition **($\mathbf{U}_{pp}; k, a$)**, since $M_r p(\mathbf{x}) \leq C_p < \infty$ for every $\mathbf{x} \in \mathbb{R}^d$ and any $r > 0$. Also, the conditions **($\mathbf{L1}_p$)**, **($\mathbf{L2}_p$)**, and **($\mathbf{L3}_p$)** on lower-boundedness of p imply the condition **($\mathbf{L}_{pp}; \xi, b$)** for any nonnegative function ξ, since for $b > 0$ we have

$$\begin{aligned} w(p, p; \xi, b, r) &= \int p(\mathbf{x}) \xi((m_r p(\mathbf{x}))^{-b}) \, \mathrm{d}\mathbf{x} \\ &\leq \int p(\mathbf{x}) \xi((\eta_p c_p)^{-b}) \, \mathrm{d}\mathbf{x} = \xi((\eta_p c_p)^{-b}) < \infty \end{aligned}$$

for some $r > 0$ by **($\mathbf{L1}_p$)** and **($\mathbf{L3}_p$)**, and $R(p, p; \xi, b, \varrho(\kappa_m/m))) = 0$ for m sufficiently large by the boundedness of the support of p from **($\mathbf{L2}_p$)**.

We recall the following notion of Hölder continuity for smoothness of the density p, which is assumed commonly in nonparametric statistics; see, e.g., (Birge and Massart, 1995; Han et al., 2020; Jiao et al., 2018; Krishnamurthy et al., 2014; Singh and Póczos, 2016).

Definition 4.3.14. For $\sigma > 0$, a function $g\colon \mathbb{R}^d \to \mathbb{R}$ is said to be *σ-Hölder continuous over an open subset $\Omega \subseteq \mathbb{R}^d$* if g is continuously differentiable over Ω up to order $\kappa := \lceil \sigma \rceil - 1$ and

$$L(g;\Omega) := \sup_{\substack{\mathbf{r}\in\mathbb{Z}_+^d \\ |\mathbf{r}|=\kappa}} \sup_{\substack{\mathbf{y},\mathbf{z}\in\Omega \\ \mathbf{y}\neq\mathbf{z}}} \frac{|\partial^{\mathbf{r}} g(\mathbf{y}) - \partial^{\mathbf{r}} g(\mathbf{z})|}{\|\mathbf{y}-\mathbf{z}\|^{\beta}} < \infty, \tag{4.23}$$

where $\beta := \sigma - \kappa$. Here we use a multi-index notation (see, e.g., (Folland, 2013, Ch. 8)), that is, $|\mathbf{r}| := r_1 + \cdots + r_d$ for $\mathbf{r} \in \mathbb{Z}_+^d$ and $\partial^{\mathbf{r}} g(\mathbf{x}) := \partial^{\kappa} g(\mathbf{x}) / (\partial x_1^{r_1} \cdots \partial x_d^{r_d})$.

Since the density is not smooth on the boundary of the support due to the lower-boundedness condition **(L1$_p$)**, we assume a smoothness condition on the underlying density only over the interior of its support and impose a separate regularity condition on the boundary:

(S$_p$) The density p is σ_p-Hölder continuous over the interior of $\mathrm{supp}(p)$ for $\sigma_p \in (0, 2]$, and

(B$_p$) the boundary of $\mathrm{supp}(p)$ has finite $(d-1)$-dimensional Hausdorff measure (Folland, 2013).

Truncated versions of well-known distributions such as exponential, Gaussian, and Cauchy distributions, as well as distributions with bounded support, such as uniform distribution and beta distributions with parameters $\alpha, \beta \geq 1$, satisfy these conditions with $\sigma_p = 2$, and the truncated Laplace distribution satisfies the conditions with $\sigma_p = 1$; see Appendix 4.F for details on these examples. For densities of

unbounded support, we provide a separate treatment using a variant of our estimator; see Section 4.3.3.

Equipped with these regularity conditions, we upper bound the MSE of our estimator by considering its bias and variance separately.

Theorem 4.3.15 (Bias rate)**.** *For a target functional $T_f(\cdot)$, if the underlying density p satisfies the conditions* ***(U_p)****, **($L1_p$)**, **($L2_p$)**, **($L3_p$)**, **(S_p)**, and **(B_p)**, then the estimator* (4.3) *with fixed $k > -a$ satisfies*

$$\left|\mathbb{E}[\hat{T}_f^{(k)}] - T_f(p)\right| = \tilde{O}(m^{-\lambda(\sigma_p, a, k)}) \tag{4.24}$$

as $m \to \infty$, where

$$\lambda(\sigma, a, k) = \begin{cases} \frac{1}{d}(\sigma \wedge 1)(\frac{k+a}{k-1}) & \text{if } a \leq -\frac{\sigma}{d} - 1, \\ \frac{1}{d}(\sigma \wedge \frac{k+a}{k-1}) & \text{if } -\frac{\sigma}{d} - 1 < a \leq -1, \\ \frac{1}{d}(\sigma \wedge 1) & \text{if } a > -1. \end{cases} \tag{4.25}$$

Remark 4.3.16. Since $k > -a$ is required to apply Theorem 4.3.15, when $a \leq -1$ (for example, the 2-entropy), our estimator is well-defined and λ in (4.25) is positive only for $k > 1$. Conversely, our bias bound holds for 1-NN estimators of any functional $T_f(p)$ with estimator function $\phi_1(u)$ of lower tail exponent $a > 1$, the examples of which include differential entropy, the α-entropy with $\alpha < 2$, the logarithmic α-entropy with $\alpha < 2$, and exponential (α, β)-entropy with $\alpha \leq 1$ in Table 4.1.1.

Remark 4.3.17. The rate exponent λ increases as the lower-tail-polynomial exponent a increases, or equivalently, the estimator function $\phi_k(u)$ converges to 0 faster as $u \downarrow 0$. If a is independent of k, the rate exponent λ becomes larger with larger k. In Section 4.5, we show that a properly growing k in sample size can guarantee the largest rate exponent in (4.25). Note, however, that if a decreases as k increases, which is the case for some

exceptional cases (Examples 4.4.20 and 4.4.21), the rate exponent could become slower with larger k. This is in contrast to the large-k requirement for *plug-in* estimators, to guarantee the underlying k-NN density estimate to be consistent. We remind that our estimator is designed to be asymptotically unbiased for every fixed k, without appealing to the consistency of the k-NN density estimator, and it thus does not contradict the behavior of plug-in estimators.

Remark 4.3.18. The upper tail exponent b appears only in the exponent of polylogarithmic factors $O(\mathrm{poly}\ln(m))$ in the rate, and thus is hidden by $\tilde{O}$ in (4.25). At a finer scale, the rate increases as b decreases; see the proof of Theorem 4.B.25 and Lemmas 4.B.23 and 4.B.25 in Appendix 4.C.1.

The variance of the estimator can be bounded without the smoothness conditions.

Theorem 4.3.19 (Variance rate)**.** *For a target functional $T_f(\cdot)$, if the underlying density p satisfies **($\mathbf{U}_p$)**, **($\mathbf{L1}_p$)**, **($\mathbf{L2}_p$)**, and **($\mathbf{L3}_p$)**, then the estimator* (4.3) *with fixed $k > -2a$ satisfies*

$$\mathrm{Var}(\hat{T}_f^{(k)}) = O(m^{-1}). \tag{4.26}$$

Combining Theorem 4.3.15 on bias and Theorem 4.3.19 on variance, we can obtain the convergence rate in MSE and establish the L_2-consistency of the estimator.

Corollary 4.3.20 (Convergence rate)**.** *Under the same assumptions in Theorem 4.3.15, then the estimator* (4.3) *with fixed $k > -2a$ satisfies*

$$\mathbb{E}\big[\big(\hat{T}_f^{(k)} - T_f(p)\big)^2\big] = \tilde{O}(m^{-2\lambda(\sigma_p,a,k)} + m^{-1}). \tag{4.27}$$

Remark 4.3.21. For $d \geq 2$, the bias bound always dominates the variance bound so that the MSE is bounded as $\tilde{O}(m^{-2\lambda})$. For $d = 1$, the variance bound may dominate the bias bound, depending on σ_p and a.

Remark 4.3.22. We note that the bias rate of the proposed estimators under Hölder smoothness of order $\sigma > 0$ is at most $O(m^{-(\sigma \wedge 1)/d})$; it may be improved to $O(m^{-(\sigma \wedge 2)/d})$ if the *boundary bias* is ignored, as remarked in (Gao et al., 2018), but it still suffers the curse of dimensionality. As pointed out in Jiao et al. (2018), it is an inherent problem with any *positive*-kernel-based estimator that a higher smoothness $\sigma > 2$ cannot be exploited in density functional estimation (Tsybakov, 2009, Chapter 1). In particular, the key component in our analysis is Lemma 4.B.6 from (Jiao et al., 2018), which cannot be improved for $\sigma > 2$. See (Han et al., 2020) for an extensive deliberation on this issue and see (Delattre and Fournier, 2017; Moon et al., 2017, 2018; Noshad et al., 2017; Sricharan et al., 2012, 2013; Wisler et al., 2018) for a solution based on the jackknife idea for some density functionals. Providing a remedy to the limitation of the proposed estimators is left as an open problem.

Remark 4.3.23. An estimator of a given density functional is said to be *minimax* optimal if its MSE for the worst-case density is no larger than that of any other estimator. In general, the established convergence rates in MSE, including the rates for divergence functional estimators in Corollaries 4.4.14, are not minimax optimal (Kandasamy et al., 2015; Krishnamurthy et al., 2014; Singh and Póczos, 2014a,b) due to the suboptimal bias rates; see, e.g., Example 4.3.24. Since our main focus is on providing unified consistency and convergent rate analyses of the proposed generic estimators, we leave proving minimax optimality under proper regularity conditions with or without modifications of the proposed estimators as important future directions. For the special case of differential entropy, we note that Jiao et al. (2018) established an asymptotic minimax optimality of the Kozachenko–Leonenko estimator (Jiao et al., 2018) for for smooth densities of order $\sigma \in (0, 2]$ *over a torus* (no boundary condition), matching the lower bound of (Han et al., 2020) up to a polylogarithmic factor.

Example 4.3.24 (Differential entropy; Example 4.3.2 contd.)**.** *Recall from Example 4.3.2*

that $|\phi_k(u)| \lesssim \psi_{-\epsilon,\epsilon}(u)$ *for any arbitrarily small* $\epsilon > 0$*. Suppose that the underlying density* p *satisfies the conditions* ***(U**$_p$**), (L1**$_p$**), (L2**$_p$**), (L3**$_p$**), (S**$_p$**), and (B**$_p$**),*** *in Theorem 4.3.15 with some* $\sigma_p \in (0, 2]$*. Then we have the bias exponent* $\lambda = \sigma_p/d$ *as in the third case of* (4.25) *and the variance exponent of* 1 *from* (4.26)*. Consequently, by Corollary 4.3.20 the MSE of our estimator is bounded as* $\tilde{O}(m^{-2(\sigma_p \wedge 1)/d} + m^{-1})$*. This result recovers the same MSE rate of a truncated Kozachenko–Leonenko estimator in (Gao et al., 2018) for* $\sigma_p = 2$*. We remark that Gao et al. (2018) reported a lower bound* $\Omega(m^{-\frac{16}{d+8}} + m^{-1})$ *for estimating differential entropy under* $\sigma = 2$ *and hence, the convergence rate is not minimax optimal.*

Example 4.3.25 (α-entropy; Example 4.3.3 contd.)**.** *Recall from Example 4.3.3 that* $|\phi_k(u)| \lesssim \psi_{1-\alpha,1-\alpha}(u)$ *for any* $k \in \mathbb{N}$ *such that* $k > \alpha - 1$*. Hence, for densities satisfying the conditions* ***(U**$_p$**), (L1**$_p$**), (L2**$_p$**), (L3**$_p$**), (S**$_p$**), and (B**$_p$**),*** *the MSE of our estimator* (4.3) *with fixed* $k > 2(\alpha - 1)$ *is bounded as* (4.27) *with the bias rate exponent*

$$\lambda(\sigma_p, a, k) = \begin{cases} \frac{1}{d}(\sigma_p \wedge 1) & \text{if } \alpha < 2, \\ \frac{1}{d}(\sigma_p \wedge \frac{k+1-\alpha}{k-1}) & \text{if } 2 \le \alpha < 2 + \frac{\sigma_p}{d}, \\ \frac{1}{d}(\sigma_p \wedge 1)(\frac{k+1-\alpha}{k-1}) & \text{if } \alpha \ge 2 + \frac{\sigma_p}{d}. \end{cases} \tag{4.28}$$

Note that similar convergence rates can be established for the logarithmic α*-entropy and the exponential* (α, β)*-entropy.*

Proof of Theorem 4.3.15 (bias rate)

First note that $U_{km}(\mathbf{X}_1), \ldots, U_{km}(\mathbf{X}_m)$ are identically distributed, and $U_{km}(\mathbf{X}_m) = U_{k,m-1}(\mathbf{X}_m)$ by definition; see (4.2). Hence, we can write

$$\begin{aligned} \mathbb{E}[\hat{T}_f^{(k)}] &= \mathbb{E}[\phi_k(U_{k,m-1}(\mathbf{X}_m))] \\ &= \int \mathbb{E}[\phi_k(U_{k,m-1}(\mathbf{X}_m))|\mathbf{X}_m = \mathbf{x}]p(\mathbf{x})\,\mathrm{d}\mathbf{x} \\ &= \int \mathbb{E}[\phi_k(U_{k,m-1}(\mathbf{x}))]p(\mathbf{x})\,\mathrm{d}\mathbf{x}, \end{aligned} \tag{4.29}$$

where the last equality holds since $\mathbf{X}_m$ and $\mathbf{X}_{1:m-1}$ are independent. Recall from Proposition 4.1.1 that $U_{km}(\mathbf{x})$ converges to a $\mathsf{G}(k, p(\mathbf{x}))$ random variable $U_{k\infty}(\mathbf{x})$ for P-a.e. $\mathbf{x}$. Thus, by the construction (4.6) of the estimator function $\phi_k(u)$, we can express the density functional as

$$T_f(p) = \int f(p(\mathbf{x}))p(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int \mathbb{E}[\phi_k(U_{k\infty}(\mathbf{x}))]p(\mathbf{x})\,\mathrm{d}\mathbf{x}.$$

Applying the triangle inequality, we first have

$$\begin{aligned}\left|\mathbb{E}[\hat{T}_f^{(k)}] - T_f(p)\right| &\le \int p(\mathbf{x})\left|\mathbb{E}[\phi_k(U_{k,m-1}(\mathbf{x})) - \phi_k(U_{k\infty}(\mathbf{x}))]\right|\mathrm{d}\mathbf{x} \\ &= \int p(\mathbf{x})\left|\int_0^\infty \phi_k(u)(\rho_{U_{k,m-1}(\mathbf{x})}(u) - \rho_{U_{k\infty}(\mathbf{x})}(u))\,\mathrm{d}u\right|\mathrm{d}\mathbf{x}. \qquad (4.30)\end{aligned}$$

For some real numbers τ_m and ν_m such that $0 \le \tau_m \le 1 \le \nu_m < \infty$, which are to be determined later as functions of k, a, d, and σ_p, we break the inner integral and apply the polynomial bound $|\phi_k(u)| \lesssim \psi_{a,b}(u)$ with the triangle inequality to obtain

$$\left|\mathbb{E}[\hat{T}_f^{(k)}] - T_f(p)\right| \lesssim I_{\text{out},1} + I_{\text{in},1} + I_{\text{in},2} + I_{\text{out},2}, \qquad (4.31)$$

where

$$\begin{aligned}I_{\text{out},1} &:= \mathbb{E}_p[I_{\text{out},1}(\mathbf{X})] = \mathbb{E}_p\Big[\int_0^{\tau_m} \psi_{a,b}(u)(\rho_{U_{k,m-1}(\mathbf{X})}(u) + \rho_{U_{k\infty}(\mathbf{X})}(u))\,\mathrm{d}u\Big], \\ I_{\text{in},1} &:= \mathbb{E}_p[I_{\text{in},1}(\mathbf{X})] = \mathbb{E}_p\Big[\int_{\tau_m}^1 \psi_{a,b}(u)|\rho_{U_{k,m-1}(\mathbf{X})}(u) - \rho_{U_{k\infty}(\mathbf{X})}(u)|\,\mathrm{d}u\Big], \\ I_{\text{in},2} &:= \mathbb{E}_p[I_{\text{in},2}(\mathbf{X})] = \mathbb{E}_p\Big[\int_1^{\nu_m} \psi_{a,b}(u)|\rho_{U_{k,m-1}(\mathbf{X})}(u) - \rho_{U_{k\infty}(\mathbf{X})}(u)|\,\mathrm{d}u\Big],\end{aligned}$$

and

$$I_{\text{out},2} := \mathbb{E}_p[I_{\text{out},2}(\mathbf{X})] = \mathbb{E}_p\Big[\int_{\nu_m}^\infty \psi_{a,b}(u)(\rho_{U_{k,m-1}(\mathbf{X})}(u) + \rho_{U_{k\infty}(\mathbf{X})}(u))\,\mathrm{d}u\Big].$$

The *inner bias* terms $I_{\text{in},1}$ and $I_{\text{in},2}$ can be bounded by Lemma 4.B.23 under the conditions **($\mathbf{U}_p$)**, **($\mathbf{S}_p$)**, and **($\mathbf{B}_p$)**, and the *outer bias* terms $I_{\text{out},1}$ and $I_{\text{out},2}$ can be bounded by Lemma 4.B.25 under the conditions **($\mathbf{U}_p$)**, **($\mathbf{L1}_p$)**, **($\mathbf{L2}_p$)**, and **($\mathbf{L3}_p$)**. After putting the bounds from Lemmas 4.B.23 and 4.B.25 together, a proper choice of the break points (τ_m, ν_m) concludes the proof; see Appendix 4.C.1 for the details. □

Remark 4.3.26. The key step in this analysis is the decomposition in (4.31), which is based on the construction of the estimator (4.6) from its asymptotic unbiasedness. Moreover, by considering only the polynomial tail behavior of each estimator function and using (4.31), our analysis can deal with a general functional in a simple, unified manner. The rest of the bias analysis, that is, bounding the four bias terms, closely follows and naturally extends that of (Gao et al., 2018) for a truncated version of the Kozachenko–Leonenko estimator of differential entropy.

Proof of Theorem 4.3.19 (variance rate)

Since the boundedness conditions **($\mathbf{U}_p$)**, **($\mathbf{L1}_p$)**, **($\mathbf{L2}_p$)**, and **($\mathbf{L3}_p$)** imply **($\mathbf{U}_{pp}$; k, a)** and **($\mathbf{L}_{pp}$; ξ, b)** (see Remark 4.3.13), the variance rate directly follows from the proof of Theorem 4.3.7 in Section 4.3.1. □

4.3.3 Convergence Rates for Smooth Densities of Unbounded Support

Theorem 4.3.15 establishes the bias rate of the proposed estimator for smooth, bounded densities that inherently assume nonsmooth boundary. In this section, we establish convergence rate of a truncated version of the estimator for densities of unbounded support.

For functionals of one density, we define a truncated version of the estimator (4.3) as

$$\overline{T}_f^{(k)}(\mathbf{X}_{1:m}) := \frac{1}{m}\sum_{i=1}^{m} \bar{\phi}_k(U_{km}(\mathbf{X}_i); \tau_m, \nu_m), \tag{4.32}$$

where we define the truncated estimator function

$$\bar{\phi}_k(u;\tau,\nu) := \phi_k(u)\,1_{(\tau,\nu)}(u)$$

and the *lower and upper truncation points* $\tau_m, \nu_m \in \mathbb{R}_+$ are hyperparameters such that $0 \leq \tau_m \leq 1 \leq \nu_m < \infty$ that are to be determined based on the function f, the dimension d, the number of nearest neighbors k, and/or the smoothness order of the underlying density p.

We assume the following condition on the tail behavior of the underlying density, which is more general than **(L1$_p$)**:

(L1$'_p$) There exist $\theta > 0$ and $D_0 > 0$ such that $\int p(\mathbf{x})e^{-\beta p(\mathbf{x})}\,\mathrm{d}\mathbf{x} \leq D_0\beta^{-\theta}$ for all $\beta > 1$.

This tail condition with $\theta = 1$ was originally considered by Tsybakov and van der Meulen (1996) for their analysis in $\mathbb{R}$. As pointed out in (Tsybakov and van der Meulen, 1996), densities with strictly sub-exponential tails, such as Gaussian distributions, satisfy **(L1$'_p$)** with $\theta = 1$. It can also be shown that densities with polynomially decaying tails satisfy condition **(L1$'_p$)** for some $0 < \theta < 1$.

We additionally introduce the following functional-dependent condition on the behavior of the estimator function for small density values:

(L4$_p$) There exists $\delta > 0$ such that $\int p(\mathbf{x})(p(\mathbf{x}))^{-(1+\delta)b}\,\mathrm{d}\mathbf{x} < \infty$.

Finally, as we consider densities with unbounded support, we assume that

(S$'_p$) the density p is σ_p-Hölder continuous over $\mathbb{R}^d$ for $\sigma_p \in (0, 2]$,

in place of **(S$_p$)**.

Exclusively for the following proposition, we additionally assume that $\phi_k(u)$ satisfies $|\phi_k(u)| \lesssim \psi_{a,b}(u)$, $\phi_k(u)$ is differentiable at any $u > 0$, and $|\phi'_k(u)| \lesssim \psi_{a-1,b-1}(u)$, which hold for all the examples in Table 4.1.1.

Proposition 4.3.27 (Bias rate for smooth densities of unbounded support)**.** *For a target functional $T_f(\cdot)$, if the underlying density p satisfies the conditions* ***(U$_p$)***, ***(L1$'_p$)***, ***(L4$_p$)***, *and* ***(S$'_p$)***, *then the truncated estimator* (4.32) *with $-a < k < -b + \theta + 1$ and truncation points*

$$\tau_m = \begin{cases} \Theta(m^{-\frac{\sigma_p}{d}\frac{1}{k-\frac{\sigma_p}{d}-1}}) & \text{if } a \le -\frac{\sigma_p}{d} - 1, \\ O(m^{-\frac{\sigma_p}{d}\frac{1}{k+a}}) & \text{o.w.} \end{cases} \tag{4.33}$$

and

$$\nu_m = \begin{cases} \Theta(m^{(\frac{\sigma_p}{d}\wedge 1)\frac{1}{\theta-k-b+1}}) & \text{if } k \le -b-1, b \le -\frac{\sigma_p}{d} - 1, \\ \Theta\big(m^{\frac{\sigma_p}{d}\frac{1}{\theta-k+\frac{\sigma_p}{d}+2}}\big) & \text{if } k \le -b-1, b > -\frac{\sigma_p}{d} - 1, \\ \Theta(m^{\frac{1}{\theta+2}}) & \text{if } k > -b-1, b \le -\frac{\sigma_p}{d} - 1, \\ \Theta(m^{(\frac{\sigma_p}{d}\wedge 1)\frac{1}{\theta+2}}) & \text{if } k > -b-1, b > -\frac{\sigma_p}{d} - 1, \end{cases}$$

with $\nu_m = o(\sqrt{m})$ as $m \to \infty$ satisfies

$$\big|\mathbb{E}\big[\overline{T}_f^{(k)}\big] - T_f(p)\big| = O\big(m^{-\lambda_\tau \wedge \lambda_\nu}\big),$$

where

$$\lambda_\tau = \begin{cases} \frac{\sigma_p}{d}\frac{k+a}{k-\frac{\sigma_p}{d}-1} & \text{if } a \le -\frac{\sigma_p}{d} - 1, \\ \frac{\sigma_p}{d} & \text{o.w.}, \end{cases} \tag{4.34}$$

and

$$\lambda_\nu = \begin{cases} \frac{\sigma_p}{d} \wedge 1 & \text{if } k \leq -b-1, b \leq -\frac{\sigma_p}{d} - 1, \\ (\frac{\sigma_p}{d}(1 - \frac{b+\frac{\sigma_p}{d}+1}{\theta - k + \frac{\sigma_p}{d}+2})) \wedge 1 & \text{if } k \leq -b-1, b > -\frac{\sigma_p}{d} - 1, \\ \frac{\sigma_p}{d} \wedge (1 - \frac{k+b+1}{\theta+2}) & \text{if } k > -b-1, b \leq -\frac{\sigma_p}{d} - 1, \\ (\frac{\sigma_p}{d} \wedge 1)(1 - \frac{k+b+1}{\theta+2}) & \text{if } k > -b-1, b > -\frac{\sigma_p}{d} - 1. \end{cases} \tag{4.35}$$

We can establish the variance rate with truncation under only the upper-boundedness condition, without explicitly imposing the condition $k > -2a$ as required in Theorem 4.3.19.

Proposition 4.3.28 (Variance rate of truncated estimator)**.** *For a target functional $T_f(\cdot)$, if the underlying density p satisfies **($\boldsymbol{U}_p$)**, then the estimator* (4.32) *with $k > -a$ satisfies*

$$\mathrm{Var}\big(\overline{T}_f^{(k)}\big) = O\Big(\frac{k^2}{m}\big(k^{-k}\tau_m^{(k+2a)\wedge 0} + \nu_m^{2b\vee 0}\big)\Big). \tag{4.36}$$

Combining Propositions 4.3.27 and 4.3.28, we can obtain a corresponding consistency result as in Corollary 4.3.20, the formal statement of which is omitted.

At face value, Proposition 4.3.27 enlarges considerably the class of densities under the purview of our analyses. On the flip side, however, it requires the underlying density to be smooth over the whole of $\mathbb{R}^d$ and this rules out, for example, the uniform distribution, which is covered by **($\mathbf{L1}_p$)**. Thus, Proposition 4.3.27 and Theorem 4.3.15 complement each other.

The stringent requirement $k < -b + \theta + 1$ in Proposition 4.3.27 is due to a bias term $O(\nu_m^{b+k-1-\theta})$ that appears in the analysis; a smaller k, which is, of course, still larger than $-a$, gives a tighter bound on this term, whereas a larger k is desired to reduce the bias due to the lower truncation. Proposition 4.3.27 thus cannot guarantee the L_2-consistency of the estimator when k grows as $m \to \infty$, as the condition $k < \theta - b + 1$

is violated.

Example 4.3.29 (Differential entropy; Example 4.3.2 contd.)**.** *For estimating differential entropy, recall that* $|\phi_k(u)| \lesssim \psi_{-\epsilon,\epsilon}(u)$ *for arbitrarily small* $\epsilon > 0$*. Consider densities that satisfy the conditions* ***(U**$_p$**)***, ***(L1′**$_p$**)***, ***(L4**$_p$**)***, and* ***(S′**$_p$**)*** *for some* $0 < \theta \leq 1$*. Since Proposition 4.3.27 requires* $k < \theta + 1 - \epsilon$*, we need to choose* $k = 1$ *to guarantee the* L_2*-consistency of our estimator. We obtain a bias bound* $O(m^{-\frac{\theta-\epsilon}{\theta+2}(\frac{\sigma_p}{d}\wedge 1)})$*, a variance bound* $O(m^{-(1-\delta)})$ *for arbitrarily small* $\delta > 0$ *from Proposition 4.3.28, and thus the MSE rate* $O(m^{-\frac{2(\theta-\epsilon)}{\theta+2}(\frac{\sigma_p}{d}\wedge 1)})$*. In particular, for one-dimensional densities with* $\sigma_p \geq 1$ *and* $\theta = 1$*, we obtain the MSE rate* $O(m^{-\frac{2(1-\epsilon)}{3}})$*. Note that this rate is slightly worse than* $O(m^{-1})$*, as obtained by Tsybakov and van der Meulen (1996, Section 2, pp. 77–78) under different regularity conditions with a faster growing upper truncation point* $\nu_m = \Theta(\sqrt{m})$*.*

Example 4.3.30 (α-entropy; Example 4.3.3 contd.)**.** *Consider estimating the* α*-entropy* ($\alpha \neq 1$) *of densities that satisfy the conditions* ***(U**$_p$**)***, ***(L1′**$_p$**)***, ***(L4**$_p$**)***, and* ***(S′**$_p$**)*** *with some* $\theta > 0$*. Since* $|\phi_k(u)| \lesssim \psi_{1-\alpha,1-\alpha}(u)$*, we need to use* $k \in (\alpha - 1, \alpha + \theta)$ *for our estimator to apply Proposition 4.3.27. By setting the truncation points as*

$$(\tau_m, \nu_m) = \begin{cases} (O(m^{-\frac{\sigma_p}{d}\frac{1}{k-\alpha+1}}), \Theta(m^{(\frac{\sigma_p}{d}\wedge 1)\frac{1}{\theta+2}})), & \text{if } \alpha < \frac{\sigma_p}{d} + 2, \\ (\Theta(m^{-\frac{\sigma_p}{d}\frac{1}{k-\frac{\sigma_p}{d}-1}}), \Theta(m^{\frac{1}{\theta+2}})), & \text{if } \alpha \geq \frac{\sigma_p}{d} + 2, \end{cases}$$

our estimator achieves the bias rate $O(m^{-(\lambda_\tau \wedge \lambda_\nu)})$*, where*

$$(\lambda_\tau, \lambda_\nu) = \begin{cases} (\frac{\sigma_p}{d}, (\frac{\sigma_p}{d}\wedge 1)\frac{\theta+\alpha-k}{\theta+2}) & \text{if } \alpha < \frac{\sigma_p}{d} + 2, \\ (\frac{\sigma_p}{d}\frac{k-\alpha+1}{k-\frac{\sigma_p}{d}-1}, \frac{\sigma_p}{d} \wedge \frac{\theta+\alpha-k}{\theta+2}) & \text{if } \alpha \geq \frac{\sigma_p}{d} + 2. \end{cases}$$

From Proposition 4.3.28, we can bound the variance of our estimator as $O(m^{-\lambda_{\mathrm{v}}})$, where

$$\lambda_{\mathrm{v}} = \begin{cases} 1 - (\frac{\sigma_p}{d} \wedge 1)\frac{2(1-\alpha)\vee 0}{\theta+2} & \text{if } \alpha < \frac{\sigma_p}{d} + 2, \\ 1 - \frac{\sigma_p}{d}\frac{(2\alpha-k-2)\vee 0}{k-\frac{\sigma_p}{d}-1} & \text{if } \alpha \geq \frac{\sigma_p}{d} + 2, \end{cases}$$

and thus we establish the MSE rate $O(m^{-2(\lambda_\tau \wedge \lambda_\nu)} + m^{-\lambda_{\mathrm{v}}})$.

Remark 4.3.31. We remark in passing on the consistency of the truncated estimator (without convergence rate analysis). With lower truncation point τ_m such that $\tau_m^{k+2a} = o(m)$, the conditions $k > -2a$ can be relaxed to $k > -a$ in Corollary 4.3.8. Moreover, a very mild upper truncation of speed $\nu_m = e^{o(m)}$ can relax the condition **($\mathbf{L}_{pp}$)** assumed in the consistency results to a milder one, i.e.,

($\mathbf{L}'_{p\tilde{p}}$; ξ, b) Either $b \leq 0$, or if $b > 0$, then there exists $r > 0$ such that $w(p, \tilde{p}; \xi, b, r) < \infty$ with $\tilde{p} = p$.

4.4 Functionals of Two Densities

We now consider estimating a functional $T_f(p, q)$ of two densities p and q. Henceforth, we assume that $\mathsf{P} \ll \mathsf{Q}$. Recall that for fixed $k, l \in \mathbb{N}$ and a given $f\colon \mathbb{R}_+^2 \to \mathbb{R}$, we define the *estimator function* $\phi_{kl}\colon \mathbb{R}_+^2 \to \mathbb{R}$ of f with parameters k, l as

$$\phi_{kl}(u, v) = \frac{\Gamma(k)\Gamma(l)}{u^{k-1}v^{l-1}} \mathcal{L}^{-1}\Big\{\frac{f(p, q)}{p^k q^l}\Big\}(u, v), \tag{4.8}$$

whenever the inverse Laplace transform exists, and then define the estimator as

$$\hat{T}_f^{(k)}(\mathbf{X}_{1:m}, \mathbf{Y}_{1:n}) = \frac{1}{m}\sum_{i=1}^{m} \phi_{kl}(U_{km}(\mathbf{X}_i), V_{ln}(\mathbf{Y}_i)). \tag{4.9}$$

Here we define

$$V_{ln}(\mathbf{x}) := U_l(\mathbf{x}|\mathbf{Y}_{1:n}) = n\,\lambda(\mathbf{B}(\mathbf{x}, r_l(\mathbf{x}|\mathbf{Y}_{1:n}))).$$

Remark 4.4.1. Similar to the observation made in Remark 4.3.1, an analogous limiting behavior

$$\lim_{k,l\to\infty} \phi_{kl}\Big(\frac{k}{p}, \frac{l}{q}\Big) = f(p,q)$$

can be verified for all the examples in Table 4.1.2 except Le Cam distance and Jensen–Shannon divergence.

As for the single-density case, a polynomial tail behavior of the estimator function $\phi_{kl}(u,v)$ affects the convergence rate of each instantiated estimator. We describe a tail behavior of $\phi_{kl}(u,v)$ by a quadruple $(a_{kl}, b_{kl}, \tilde{a}_{kl}, \tilde{b}_{kl}) \in \mathbb{R}^4$ such that $|\phi_{kl}(u,v)| \lesssim \psi_{a_{kl},b_{kl}}(u)\psi_{\tilde{a}_{kl},\tilde{b}_{kl}}(v)$. This characterization allows us to handle the convergence of $U_{km}(\mathbf{x})$ and $V_{ln}(\mathbf{x})$ separately so that we can extend the analysis for the single-density case in a straightforward manner. Note that for all the examples presented in Table 4.1.2, $(a_{kl}, b_{kl}, \tilde{a}_{kl}, \tilde{b}_{kl})$ can be found as constants independent of k and l, except Le Cam distance and Jensen–Shannon divergence. Also note that all the estimator functions $\phi_{kl}(u,v)$ presented in Table 4.1.2 are continuous.

Example 4.4.2 (KL divergence (Wang et al., 2009))**.** *For $f(p,q) = \ln(p/q)$, we can compute, as shown in Example 4.E.2 in Appendix 4.E,*

$$\phi_{kl}(u,v) = \ln\frac{v}{u} + H_{k-1} - H_{l-1}.$$

As a bound on the estimator function $\phi_{kl}(u,v)$, we consider

$$\begin{aligned}|\phi_{kl}(u,v)| &\lesssim 1 + |\ln u| + |\ln v| \\ &\lesssim (1+|\ln u|)(1+|\ln v|) \lesssim \psi_{-\epsilon,\epsilon}(u)\psi_{-\epsilon,\epsilon}(v)\end{aligned}$$

for any arbitrarily small $\epsilon > 0$.

Example 4.4.3 (Polynomial functional (Póczos and Schneider, 2011; Póczos et al., 2012))**.** *For $f(p,q) = p^{\alpha-1}q^{\beta}$ $(\alpha > 0, \beta > 1-\alpha)$ and any $k, l \in \mathbb{N}$ such that $k > \alpha - 1$ and $l > \beta$, we can compute, as shown in Example 4.E.3 in Appendix 4.E,*

$$\phi_{kl}(u,v) = \frac{\Gamma(k)\Gamma(l)}{\Gamma(k-\alpha+1)\Gamma(l-\beta)} u^{1-\alpha} v^{-\beta},$$

which allows the tight polynomial bound

$$|\phi_k(u)| \lesssim \psi_{1-\alpha,1-\alpha}(u)\psi_{-\beta,-\beta}(v).$$

This class of polynomial functionals includes many important functionals. For the special instance of $\beta = 1-\alpha$, we refer to the density functional $T_f(p,q) = \int p^{\alpha}(\mathbf{x})q^{1-\alpha}(\mathbf{x})\,\mathrm{d}\mathbf{x}$ as the α-divergence*, which appears in the literature in a few different forms; see, e.g., Rényi (1961) and Cichocki et al. (2008).*

Example 4.4.4 (Logarithmic α-divergence)**.** *For $f(p,q) = (p/q)^{\alpha-1}\ln(p/q)$ $(\alpha > 0)$, we refer to the density functional $T_f(p,q) = \int p^{\alpha}(\mathbf{x})q^{1-\alpha}(\mathbf{x})\ln(p(\mathbf{x})/q(\mathbf{x}))\,\mathrm{d}\mathbf{x}$ as the* logarithmic α-divergence*. For any $k, l \in \mathbb{N}$ such that $k > \alpha - 1$ and $l > 1-\alpha$, we can compute, as shown in Example 4.E.4 in Appendix 4.E,*

$$\begin{aligned}\phi_{kl}(u,v) = {} & \frac{\Gamma(k)\Gamma(l)}{\Gamma(k-\alpha+1)\Gamma(l+\alpha-1)} \\ & \times u^{-\alpha+1}\Big(\ln\frac{v}{u} + \Psi(k-\alpha+1) - \Psi(l+\alpha-1)\Big).\end{aligned}$$

As a bound on the estimator function $\phi_{kl}(u,v)$, we consider

$$\begin{aligned}|\phi_{kl}(u,v)| &\lesssim u^{-\alpha+1}v^{\alpha-1}(1 + |\ln u| + |\ln v|) \\ &\lesssim u^{-\alpha+1}v^{\alpha-1}(1+|\ln u|)(1+|\ln v|) \\ &\lesssim \psi_{1-\alpha-\epsilon,1-\alpha+\epsilon}(u)\psi_{\alpha-1-\epsilon,\alpha-1+\epsilon}(v)\end{aligned}$$

for any arbitrarily small $\epsilon > 0$.

Example 4.4.5 (Le Cam distance)**.** *For $f(p,q) = (p-q)^2/(2p(p+q))$, the corresponding divergence functional*

$$\begin{aligned} D_{\mathsf{LC}}(p,q) &= \frac{1}{2}\int \frac{(p(\mathbf{x})-q(\mathbf{x}))^2}{p(\mathbf{x})+q(\mathbf{x})}\,\mathrm{d}\mathbf{x} \\ &= 1 - \int \frac{2p(\mathbf{x})q(\mathbf{x})}{p(\mathbf{x})+q(\mathbf{x})}\,\mathrm{d}\mathbf{x} \end{aligned}$$

is called Le Cam distance (Le Cam, 2012, p. 47) in the literature (Polyanskiy and Wu, 2019). We note in passing that this functional has a connection to the nearest neighborhood binary classification rule: it is well known that the asymptotic error of the nearest neighborhood binary classification for equiprobable classes is given as $\frac{1}{2}(1 - T_f(p,q))$ (Cover and Hart, 1967). For any $k, l \in \mathbb{N}$, we can compute, as shown in Example 4.E.5 in Appendix 4.E,

$$\begin{aligned} \phi_{kl}(u,v) = 2\binom{k+l-2}{k-1}^{-1}\Big(-\frac{u}{v}\Big)^{l-1} \times \\ \Big\{\sum_{i=0}^{l-1}\binom{k+l-2}{i}\Big(-\frac{v}{u}\Big)^{i} - \Big(1-\frac{v}{u}\Big)^{k+l-2}1_{[v,\infty)}(u)\Big\} - 1. \end{aligned}$$

As a bound on the estimator function $\phi_{kl}(u,v)$, we have

$$|\phi_{kl}(u,v)| \lesssim \psi_{-k+1,l-1}(u)\psi_{-l+1,k-1}(v).$$

Example 4.4.6 (Jensen–Shannon divergence)**.** *When $Q \ll P$, we can write Jensen–Shannon divergence as*

$$D_{\mathsf{JS}}(p,q) = \frac{1}{2}\Big(D\Big(p \,\Big\|\, \frac{p+q}{2}\Big) + D\Big(q \,\Big\|\, \frac{p+q}{2}\Big)\Big) = T_f(p,q)$$

for

$$f(p,q) = \frac{1}{2}\Big(\frac{q}{p}+1\Big)\ln\frac{2}{(q/p)+1} + \frac{q}{2p}\ln\frac{q}{p},$$

$$B_{kl}(u,v) = \begin{cases} \binom{k+l-2}{k-1}^{-1} \sum_{j=1}^{l-1} \binom{k+l-2}{k-1+j} \frac{(-u/v)^j}{j} & \text{if } \frac{u}{v} < 1, \\ -\ln \frac{u}{v} + \binom{k+l-2}{k-1}^{-1} \Big\{ - \sum_{j=-k+1}^{-1} \binom{k+l-2}{k-1+j} \frac{(-u/v)^j}{j} + \sum_{\substack{j=-k+1 \\ j\neq 0}}^{l-1} \binom{k+l-2}{k-1+j} \frac{(-1)^j}{j} \Big\} & \text{if } \frac{u}{v} \geq 1. \end{cases} \tag{4.37}$$

where $D(p \,\|\, q)$ denotes the KL divergence between p and q. For any $k \geq 1$ and $l \geq 2$, we can compute, as shown in Example 4.E.8 in Appendix 4.E,

$$\phi_{kl}(u,v) = \frac{1}{2}\Big\{ \ln 2 + \frac{l-1}{k}\frac{u}{v}\Big(\Psi(l-1) - \Psi(k+1) + \ln 2\frac{u}{v} \Big) + B_{kl}(u,v) + \frac{l-1}{k}\frac{u}{v} B_{k+1,l-1}(u,v) \Big\},$$

where $B_{kl}(u,v)$ is defined in (4.37)*. As a polynomial bound, we have*

$$|\phi_{kl}(u,v)| \lesssim \psi_{-k+1,l-1}(u)\psi_{-l+1,k-1}(v).$$

4.4.1 Consistency

As in Section 4.3.1, we can establish the L_2-consistency of the estimator of functionals of two densities under mild regularity conditions. Throughout, we consider a fixed $(a, b, \tilde{a}, \tilde{b}) \in \mathbb{R}^4$ for a target functional $T_f(\cdot,\cdot)$ whose estimator function ϕ_{kl} satisfies $|\phi_{kl}(u,v)| \lesssim \psi_{a,b}(u)\psi_{\tilde{a},\tilde{b}}(v)$, provided that the estimator function ϕ_{kl} exists for $k > -a$ and $l > -\tilde{a}$.

Theorem 4.4.7 (Vanishing bias)**.** *For a target functional $T_f(\cdot,\cdot)$, if the estimator function $\phi_{kl}(u,v)$ is continuous and the underlying densities p and q satisfy* ***($\boldsymbol{U}_{pp}$; k, a)****,* ***($\boldsymbol{L}_{pp}$; ξ^2, b)****,* ***($\boldsymbol{U}_{pq}$; $l, \tilde{a}$)****, and* ***($\boldsymbol{L}_{pq}$; $\xi^2, \tilde{b}$)*** *for some function $\xi \in \Xi$, then the estimator* (4.9) *with $k > -2\omega(\xi)a$ and $l > -2\omega(\xi)\tilde{a}$ is asymptotically unbiased as $m, n \to \infty$.*

Theorem 4.4.8 (Vanishing variance)**.** *For a target functional $T_f(\cdot,\cdot)$, if the underlying densities p and q satisfy* ***($\boldsymbol{U}_{pp}$; k, a)****,* ***($\boldsymbol{L}_{pp}$; ξ^2, b)****,* ***($\boldsymbol{U}_{pq}$; $l, \tilde{a}$)****, and* ***($\boldsymbol{L}_{pq}$; $\xi^2, \tilde{b}$)*** *with $\xi(t) = t^2$, then*

then the variance of the estimator (4.9) *with fixed* $k > -4a$ *and fixed* $l > -4\tilde{a}$ *converges to zero as* $m, n \to \infty$.

Corollary 4.4.9 (Consistency)**.** *For a target functional* $T_f(\cdot, \cdot)$*, if the estimator function* $\phi_{kl}(u, v)$ *is continuous and the underlying densities* p *and* q *satisfy* ***(U**$_{pp}$**;** k, a***)***, ***(L**$_{pp}$**;** ξ^2, b***)***, ***(U**$_{pq}$**;** $l, \tilde{a}$***)***, *and* ***(L**$_{pq}$**;** $\xi^2, \tilde{b}$***)*** *with* $\xi(t) = t^2$*, then the estimator* (4.9) *with fixed* $k > -4a$ *and fixed* $l > -4\tilde{a}$ *is* L_2*-consistent.*

In the following examples, we illustrate how Corollary 4.4.9 can be instantiated for a few representative functionals.

Example 4.4.10 (KL divergence; Example 4.4.2 contd.)**.** *Recall that for estimating differential entropy,* $|\phi_{kl}(u, v)| \lesssim \psi_{-\epsilon,\epsilon}(u)\psi_{-\epsilon,\epsilon}(v)$ *for arbitrarily small* $\epsilon > 0$ *and for any* $k, l \in \mathbb{N}$*. By Corollary 4.4.9, the estimator* (4.9) *with fixed* $k \geq 1$ *and* $l \geq 1$ *is* L_2*-consistent if the underlying densities* p *and* q *satisfy* ***(U**$_{pp}$**;** $k, -\epsilon$***)***, ***(L**$_{pp}$**;** ξ^2, ϵ***)***, ***(U**$_{pq}$**;** $l, -\epsilon$***)***, *and* ***(L**$_{pq}$**;** ξ^2, ϵ***)*** *with* $\xi(t) = t^2$*. As discussed in Example 4.3.9, a finer analysis recovers a similar consistency result established in (Bulinski and Dimitrov, 2019b).*

The proofs of the main results (Theorems 4.4.7, 4.4.8, 4.4.12, and 4.4.13) in this section follow with minor extensions to those of the single-density case, and are deferred to Appendix 4.C.

Example 4.4.11 (α-divergence; Example 4.4.3 contd.)**.** *Recall that for estimating the* α*-divergence (*$\alpha \neq 1$*), we have* $|\phi_{kl}(u, v)| \lesssim \psi_{1-\alpha,1-\alpha}(u)\psi_{\alpha-1,\alpha-1}(v)$ *for any* $k, l \in \mathbb{N}$ *such that* $k > \alpha - 1$ *and* $l > 1 - \alpha$*. For* $\alpha > 1$*, since* $b = 1 - \alpha < 0$ *and* $\tilde{a} = \alpha - 1 > 0$*, the estimator with fixed* $k > 4(\alpha - 1)$ *and* $l \geq 1$ *is* L_2*-consistent if the underlying densities* p *and* q *satisfy that* ***(U**$_{pp}$**;** $k, 1 - \alpha$***)*** *and* ***(L**$_{pq}$**;** $\xi^2, \alpha - 1$***)*** *with* $\xi(t) = t^2$*. For* $\alpha < 1$*, since* $a = 1 - \alpha > 0$ *and* $\tilde{b} = \alpha - 1 < 0$*, the estimator with* $k \geq 1$ *and* $l > 4(1 - \alpha)$ *is* L_2*-consistent if the underlying densities* p *and* q *satisfy that* ***(L**$_{pp}$**;** ξ^2, b***)*** *and* ***(U**$_{pq}$**;** $l, \tilde{a}$***)*** *with* $\xi(t) = t^2$*. This consistency result covers a strictly larger class of densities than an earlier result by Póczos and Schneider (2011), whereby the*

L_2-consistency of the estimator with $l = k$ is established under rather stronger assumptions such as boundedness and uniform continuity of densities. Moreover, Propositions 4.4.9 and 4.4.9 strengthen the L_2-consistency result established in Póczos et al. (2012) for a polynomial functional (see Example 4.4.3), which subsumes α-divergence.

4.4.2 Convergence Rates for Smooth, Bounded Densities

Theorem 4.4.12 (Bias rate)**.** *For a target functional $T_f(\cdot, \cdot)$, if the underlying density p satisfies **(U_p)**, **($L1_p$)**, **($L2_p$)**, **($L3_p$)**, **(S_p)**, and **(B_p)**, and q satisfies **(U_q)**, **($L1_q$)**, **($L2_q$)**, **($L3_q$)**, **(S_q)**, and **(B_q)**, then the estimator* (4.9) *with fixed $k > -a$ and $l > -\tilde{a}$ satisfies*

$$\left|\mathbb{E}[\hat{T}_f^{(k,l)}] - T_f(p,q)\right| = \tilde{O}\big(m^{-\lambda(\sigma_p,a,k)} + n^{-\lambda(\sigma_q,\tilde{a},l)}\big),$$

as $m, n \to \infty$, where the rate exponent function $\lambda(\sigma, a, k)$ is as defined in (4.25)*.*

Theorem 4.4.13 (Variance rate)**.** *For a target functional $T_f(\cdot, \cdot)$, if the underlying density p satisfies **(U_p)**, **($L1_p$)**, **($L2_p$)**, and **($L3_p$)**, and q satisfies **(U_q)**, **($L1_q$)**, **($L2_q$)**, and **($L3_q$)**, then the estimator* (4.9) *with fixed $k > -2a$ and fixed $l > -2\tilde{a}$ satisfies*

$$\mathrm{Var}\big(\hat{T}_f^{(k,l)}\big) = O(m^{-1}). \tag{4.38}$$

Combining Theorems 4.4.12 and Theorem 4.4.13, we obtain the convergence rate in MSE and conclude the L_2-consistency of the estimator.

Corollary 4.4.14 (Convergence rate)**.** *Under the same assumptions in Theorem 4.4.12, then the estimator* (4.9) *with fixed $k > -2a$ and fixed $l > -2\tilde{a}$ satisfies*

$$\begin{aligned}&\mathbb{E}\big[\big(\hat{T}_f^{(k,l)} - T_f(p,q)\big)^2\big] \\ &= \tilde{O}\big(m^{-2\lambda(\sigma_p,a,k)} + n^{-2\lambda(\sigma_q,\tilde{a},l)} + m^{-1}\big)\end{aligned} \tag{4.39}$$

and thus is L_2-consistent.

Remark 4.4.15. Similar to the single-density case, if $d \geq 2$, the bias bound dominates the variance bound.

Example 4.4.16 (KL divergence; Example 4.4.2 contd.)**.** *For estimating KL divergence, recall that* $|\phi_{kl}(u,v)| \lesssim \psi_{-\epsilon,\epsilon}(u)\psi_{-\epsilon,\epsilon}(v)$ *for any arbitrarily small* $\epsilon > 0$*. It can be shown, using Theorems 4.4.12 and 4.4.13, that for estimating the (forward) KL or reverse KL divergences between any two densities* p *and* q *such that* $P \ll Q$*, each of which is either the uniform distribution, or one of the truncated Gaussian, Cauchy, Laplace, or exponential distributions, we obtain a bias bound of* $\tilde{O}(m^{-1/d})$ *and a variance bound of* $O(m^{-1})$*, and therefore, the MSE rate of* $\tilde{O}(m^{-2/d} + n^{-2/d} + m^{-1})$ *as established in Corollary 4.4.14.*

Example 4.4.17 (α-divergence; Example 4.4.3 contd.)**.** *For estimating the* α*-divergence* ($\alpha > 0$)*, recall that* $|\phi_{kl}(u,v)| \lesssim \psi_{1-\alpha,1-\alpha}(u)\psi_{\alpha-1,\alpha-1}(v)$ *for any* $k, l \in \mathbb{N}$ *such that* $k > \alpha - 1$ *and* $l > 1 - \alpha$*. Hence, if* p *satisfies* ***(U**$_p$**)***, ***(L1**$_p$**)***, ***(L2**$_p$**)***, ***(L3**$_p$**)***, ***(S**$_p$**)***, and* ***(B**$_p$**)***, and* q *satisfies* ***(U**$_q$**)***, ***(L1**$_q$**)***, ***(L2**$_q$**)***, ***(L3**$_q$**)***, ***(S**$_q$**)***, and* ***(B**$_q$**)***, then the MSE of the estimator* (4.9) *with* $k > 2(\alpha - 1)$ *and* $l > 2(1 - \alpha)$ *is bounded as* (4.39) *with the bias rate exponents*

$$\lambda(\sigma_p, a, k) = \begin{cases} \frac{1}{d}(\sigma_p \wedge 1) & \text{if } \alpha < 2, \\ \frac{1}{d}(\sigma_p \wedge \frac{k+1-\alpha}{k-1}) & \text{if } 2 \leq \alpha < 2 + \frac{\sigma_p}{d}, \\ \frac{1}{d}(\sigma_p \wedge 1)(\frac{k+1-\alpha}{k-1}) & \text{if } \alpha \geq 2 + \frac{\sigma_p}{d}. \end{cases}$$

and

$$\lambda(\sigma_q, \tilde{a}, l) = \frac{1}{d}(\sigma_q \wedge 1).$$

This result also holds for the logarithmic α*-divergence.*

4.4.3 Le Cam Distance and Jensen–Shannon Divergence: Performance Guarantee with Truncation

The statements in the previous section do not apply to the estimators for Le Cam distance (Example 4.4.5) and Jensen–Shannon divergence (Example 4.4.6). The difficulty arises from the fact that the estimator function ϕ_{kl} for these divergences have lower-polynomial-tail exponents $(a, \tilde{a}) = (-k+1, -l+1)$ which become smaller with larger k and l. Therefore, while the bias guarantees (Theorems 4.4.7 and 4.4.12) are still applicable, we cannot control the variance of the estimator using Theorems 4.4.8 or 4.4.13, as $(a, \tilde{a}) = (-k+1, -l+1)$ does not meet the requirements $\{k > -4a, l > -4\tilde{a}\}$ or $\{k > -2a, l > -2\tilde{a}\}$.

To handle the variance of the estimator for these exceptional cases, we consider a truncated version of the estimator (4.9). For functionals of two densities, we define the truncated estimator as

$$\overline{T}_f^{(k,l)}(\mathbf{X}_{1:m}, \mathbf{Y}_{1:n}) := \frac{1}{m} \sum_{i=1}^{m} \bar{\phi}_{kl}(U_{km}(\mathbf{X}_i), V_{ln}(\mathbf{X}_i); \tau_m, \nu_m, \tilde{\tau}_n, \tilde{\nu}_n), \tag{4.40}$$

where we define the truncated estimator function

$$\bar{\phi}_{kl}(u, v; \tau, \nu, \tilde{\tau}, \tilde{\nu}) := \phi_{kl}(u, v)\, 1_{(\tau,\nu)}(u)\, 1_{(\tilde{\tau},\tilde{\nu})}(v)$$

and the *truncation points* $\tau_m, \nu_m, \tilde{\tau}_n, \tilde{\nu}_n$ are hyperparameters such that $0 \le \tau_m \le 1 \le \nu_m \le \infty$ and $0 \le \tilde{\tau}_n \le 1 \le \tilde{\nu}_n \le \infty$. As noted earlier, we do not require the upper-truncation points in contrast to Section 4.3.3 and thus only consider a *lower-truncated estimator* with $\nu_m = \infty$ and $\tilde{\nu}_n = \infty$ in this section.

We can first establish the consistency of the lower-truncated estimator.

Proposition 4.4.18 (Consistency)**.** *For a target functional $T_f(\cdot, \cdot)$, if the estimator function $\phi_{kl}(u, v)$ is continuous and the underlying densities p and q satisfy* ***($\boldsymbol{U}_{pp}$; k, a)****,* ***($\boldsymbol{L}'_{pp}$; ξ^2, b)****,*

*(**U_{pq}**; $l, \tilde{a}$), and (**L'_{pq}**; $\xi^2, \tilde{b}$) with $\xi(t) = t^2$, then the lower-truncated estimator* (4.40) *with fixed $k > -a$ and $l > -\tilde{a}$ and with lower-truncation points such that $\tau_m^{(k+4a)\wedge 0}\tilde{\tau}_n^{(l+4\tilde{a})\wedge 0} = o(m)$ is L_2-consistent.*

We can also establish convergence rate of the truncated estimator 4.40 for functionals of two densities. Define a lower truncation point function as

$$\tau(m, \sigma, a, k) = \begin{cases} \Theta\big(m^{-\frac{\sigma\wedge 1}{d(k-1)}}\big) & \text{if } a \le -\frac{\sigma}{d} - 1, \\ \Theta\big(m^{-\frac{1}{d(k-1)}}\big) & \text{if } -\frac{\sigma}{d} - 1 < a \le -1, \\ O\big(m^{-\frac{1}{d(a+1)}}\big) & \text{if } a > -1. \end{cases} \tag{4.41}$$

Proposition 4.4.19 (Convergence rate)**.** *For a target functional $T_f(\cdot, \cdot)$, if the underlying density p satisfies the conditions (**U_p**), (**$L1_p$**), (**S_p**), and (**B_p**), and q satisfies the conditions (**U_q**), (**$L1_q$**), (**S_q**), and (**B_q**), the truncated estimator* (4.40) *with fixed $k > -a$ and $l > -\tilde{a}$ satisfies*

$$\mathbb{E}\big[\big(\overline{T}_f^{(k,l)} - T_f(p,q)\big)^2\big] = \tilde{O}\big(m^{-2\lambda(\sigma_p,a,k)} + n^{-2\lambda(\sigma_q,\tilde{a},l)} + m^{-1}\tau_m^{(2a+k)\wedge 0}\tilde{\tau}_n^{(2\tilde{a}+l)\wedge 0}\big),$$

as $m, n \to \infty$, and thus is L_2-consistent.

Example 4.4.20 (Le Cam distance; Example 4.4.5 contd.)**.** *For estimating $T_f(p,q)$ with $f(p,q) = q/(p+q)$, recall that $|\phi_{kl}(u,v)| \lesssim \psi_{-k+1,l-1}(u)\psi_{-l+1,k-1}(v)$ for any $k \ge 1$ and $l \ge 1$. For densities p and q satisfying conditions in Proposition 4.4.18, the lower-truncated estimator* (4.40) *for Le Cam distance is L_2-consistent. In particular, the estimator with $k = l = 1$ is consistent even without lower truncation, since $\tau_m^{(k+4a)\wedge 0}\tilde{\tau}_n^{(l+4\tilde{a})\wedge 0} = \tau_m^0\tilde{\tau}_n^0 = 0$ with $\tau_m = \tilde{\tau}_n = 0$ and $k = l = 1$. If the underlying densities p and q satisfy the conditions in Proposition 4.4.19, then the lower-truncated estimator with fixed $k \ge 1$ and $l \ge 1$ and truncation points $\tau_m = \tau(m, \sigma_p, -k+1, k)$, and $\tilde{\tau}_n = \tau(n, \sigma_q, -l+1, l)$ satisfies*

$$\mathbb{E}\big[\big(\hat{T}_f^{(k,l)} - T_f(p,q)\big)^2\big] = \tilde{O}\big(m^{-2\lambda_k(\sigma_p)} + n^{-2\lambda_l(\sigma_q)} + m^{-1}\tau_m^{(-k+2)\wedge 0}\tilde{\tau}_n^{(-l+2)\wedge 0}\big), \tag{4.42}$$

as $m, n \to \infty$, where $\lambda_p = \lambda_k(\sigma_p)$ and $\lambda_q = \lambda_l(\sigma_q)$, where

$$\lambda_k(\sigma) := \lambda(\sigma, -k+1, k) = \begin{cases} \frac{1}{d}(\sigma \wedge 1) & \text{if } k = 1, \\ \frac{1}{d}(\sigma \wedge \frac{1}{k-1}) & \text{if } 2 \le k < 2 + \frac{\sigma}{d}, \\ \frac{1}{d}\frac{\sigma \wedge 1}{k-1}, & \text{if } k > 2 + \frac{\sigma}{d}. \end{cases}$$

Based on this rate-exponent expression and the additional factor of $\tau_m^{(2a+k)\wedge 0} \tilde{\tau}_n^{(2\tilde{a}+l)\wedge 0}$ in the variance rate which only worsens the rate with larger k and $l4$, one would expect that the convergence becomes only slower as k and/or l become large, and thus, the fastest rate achieved is $\tilde{O}(m^{-\frac{2}{d}(\sigma_p \wedge 1)} + n^{-\frac{2}{d}(\sigma_q \wedge 1)} + m^{-1})$, when $k = 1$ and $l = 1$ with lower truncation points $\tau_m = 0$ and $\tilde{\tau}_n = \Theta(n^{-\frac{1}{d}})$. This is in contrast with Remark 4.3.17, where we observed faster convergence with larger values of k when a does not decrease in k. We note that the experiments with synthetic data in Section 4.6 show that the estimator performs well even for large values of k and l, suggesting that the detrimental effect of the lower tail exponents might be removed with a tighter analysis.

Example 4.4.21 (Jensen–Shannon divergence; Example 4.4.6 contd.)**.** *For estimating Jensen–Shannon divergence, recall that $|\phi_{kl}(u, v)| \lesssim \psi_{-k+1,l-1}(u)\psi_{-l+1,k-1}(v)$ for any $k \ge 1$ and $l \ge 2$. For densities p and q satisfying conditions in Proposition 4.4.18, the lower-truncated estimator* (4.40) *for Jensen–Shannon divergence is L_2-consistent. Also, we do not require the lower-truncation τ_m for $k = 1$, by the same argument in the previous example. If the underlying densities p and q satisfy the conditions in Proposition 4.4.19 and additionally $\mathsf{Q} \ll \mathsf{P}$, then the estimator* (4.9) *with fixed $k \ge 1$ and $l \ge 2$ and the same truncation points in Example 4.4.20 satisfies* (4.42)*. The established rate seems to get only slower as k and/or l become large, and thus achieves its fastest rate $\tilde{O}(m^{-\frac{2}{d}(\sigma_p \wedge 1)} + n^{-\frac{2}{d}(\sigma_q \wedge 1)} + m^{-1})$ when $k = 1$ and $l = 2$ with lower truncation points $\tau_m = 0$ and $\tilde{\tau}_n = \Theta(n^{-\frac{1}{d}})$. Note, however, this conclusion might not hold in practice; see Example 4.4.20.*

4.5 Adaptive Choices of k and l

In Section 4.3, we established the convergence rate of the proposed estimator (4.3) for fixed k. Since $\mathbb{E}[\phi_k(U_{k\infty}(\mathbf{x}))] = f(p(\mathbf{x}))$ for each valid $k \in \mathbb{N}$ by design, we can choose any valid k without violating the asymptotic unbiasedness. In Remark 4.3.17, we observed that a larger *fixed* k in general leads to a larger rate exponent in (4.25), and thus, a faster convergence rate. This prompts the question of whether increasing $k \to \infty$ along with m improves the convergence rate upon fixed k. The following proposition answers this in the affirmative. The proof is deferred to Appendix 4.D.2.

Proposition 4.5.1 (Convergence rate and L_2-consistency with increasing k)**.** *For a target functional $T_f(\cdot)$, if the underlying density p satisfies* ***(U_p)***, ***($L1_p$)***, ***($L2_p$)***, ***($L3_p$)***, ***(S_p)***, *and* ***(B_p)***, *then the estimator* (4.3) *with $k = \Theta((\ln m)^{1.1})$ satisfies*

$$\left|\mathbb{E}\big[\hat{T}_f^{(k)}\big] - T_f(p)\right| = \tilde{O}\big(m^{-\frac{\sigma_p \wedge 1}{d}}\big) \tag{4.43}$$

as $m \to \infty$. Furthermore, the estimator (4.3) *satisfies*

$$\mathbb{E}\big[\big(\hat{T}_f^{(k)} - T_f(p)\big)^2\big] = \tilde{O}\big(m^{-\frac{2(\sigma_p \wedge 1)}{d}} + m^{-1}\big) \tag{4.44}$$

and thus is L_2-consistent.

Remark 4.5.2. As expected heuristically, the bias rate exponent $(\sigma_p \wedge 1)/d$ in (4.43) equals the limit of the finite-k rate exponent in (4.25) as $k \to \infty$.

Remark 4.5.3. There is no consensus on the optimal choice of k for functional estimation in the literature. For example, Singh and Póczos (2016) analyzed $k = O(1)$, whereas Berrett et al. (2019) suggested $k = O((\ln m)^5)$ for asymptotic efficiency of the estimator, a slightly faster choice than the previous theorem, for differential entropy. Pérez-Cruz (2009) discussed some relevant empirical results on the choice of k.

Remark 4.5.4. While our main focus in this paper is to establish consistency and convergence rates for the proposed estimators with fixed k (and l), we point out that a tighter analysis on the dependence on k may lead to a better asymptotic convergence rate. Note that the analysis of Kozachenko–Leonenko estimator by Berrett et al. (2019) allows polynomial growth of k in the sample size. The loose dependence on k in our analysis can be traced back to Lemma 4.B.4, which quantifies the gap between densities of the normalized volume of k-NN ball $U_{km}(\mathbf{x})$ and its limiting Poisson random variable $U_{k\infty}(\mathbf{x})$. To tighten the bound, one needs to sharpen Lemma 4.B.5 on the speed of convergence of a Poisson binomial random variable to a Poisson random variable.

Example 4.5.5 (Differential entropy; Example 4.3.24 contd.)**.** *Applying Proposition 4.5.1 on differential entropy with $k = \Theta((\ln m)^{1.05})$, we obtain the MSE rate* (4.44)*. This rate is the same as the fixed-k case in Example 4.3.24.*

Example 4.5.6 (α-entropy; Example 4.3.25 contd.)**.** *Applying Proposition 4.5.1 on α-entropy with $k = \Theta((\ln m)^{1.05})$, we obtain the bias rate exponent $(\sigma_p \wedge 1)/d$, which is greater than or equal to that in Example 4.3.25 with k fixed.*

Similarly to the single-density case, we can establish the convergence rate when k and l vary polylogarithmically with m and n, provided that m and n grow to infinity in the same speed, i.e., $m \asymp n$. The following proposition can be proved by extending the proof of Proposition 4.5.1 to the double-density case as in the proofs of Theorems 4.4.12 and 4.4.13, and thus is omitted.

Proposition 4.5.7 (Convergence rate and L_2-consistency with increasing k and l)**.** *For a target functional $T_f(\cdot,\cdot)$, if the underlying densities p and q satisfy the conditions* ***(U$_p$)***, ***(L1$_p$)***, ***(L2$_p$)***, ***(L3$_p$)***, ***(S$_p$)***, ***(B$_p$)***, ***(U$_q$)***, ***(L1$_q$)***, ***(L2$_q$)***, ***(L3$_q$)***, ***(S$_q$)***, *and* ***(B$_q$)****, then the estimator* (4.9) *with $k = \Theta((\ln m)^{1.1})$ and $l = \Theta((\ln n)^{1.1})$ satisfies*

$$\left|\mathbb{E}[\hat{T}_f^{(k,l)}] - T_f(p,q)\right| = \tilde{O}(m^{-\frac{\sigma_p \wedge 1}{d}} + n^{-\frac{\sigma_q \wedge 1}{d}}),$$

as $m, n \to \infty$ with $m \asymp n$. Furthermore, the estimator (4.9) *satisfies*

$$\mathbb{E}\big[(\hat{T}_f^{(k,l)} - T_f(p,q))^2\big] = \tilde{O}(m^{-\frac{2(\sigma_p \wedge 1)}{d}} + n^{-\frac{2(\sigma_q \wedge 1)}{d}} + m^{-1}), \tag{4.45}$$

and thus is L_2-consistent, provided that $m \asymp n$.

Remark 4.5.8. For $d \geq 2$, if k and l increase as in Proposition 4.5.7, the bias bound always dominates the variance bound so that the MSE is bounded as $O(m^{-1})$. For $d = 1$, the variance bound may dominate the bias bound depending on σ_p, σ_q, d, and/or the choices of k and l.

Example 4.5.9 (KL divergence; Example 4.4.16 contd.)**.** *Letting k and l increase as $k = \Theta((\ln m)^{1.05}))$ and $l = \Theta((\ln n)^{1.05}))$, we obtain the MSE rate* (4.45) *for estimating KL divergence. As a complementary asymptotic result, Wang et al. (2009) showed that the (k, l)-NN KL divergence estimator with $k = k_m$ and $l = l_n$ such that $k_m/m \to 0$ and $k_m/(\ln m) \to \infty$ as $m \to \infty$ and $l_n/n \to 0$ and $l_n/(\ln n) \to \infty$ as $n \to \infty$ converges to the true KL divergence almost surely for uniformly continuous densities bounded from below on their support.*

Example 4.5.10 (α-divergence; Example 4.4.17 contd.)**.** *Letting k and l increase as $k = \Theta((\ln m)^{1.05}))$ and $l = \Theta((\ln n)^{1.05}))$, the MSE of our estimator is bounded as* (4.45)*.*

4.6 Numerical Results

The performance of the proposed estimators (4.3) and (4.9) for several density functionals were simulated over 500 runs for sample sizes ranging from 100 till 25600.[2] For each dimension d from 1 through 5, we considered the uniform density $\mathrm{Unif}([0,1]^d)$, the Gaussian density $\mathsf{N}(0, I_d)$ restricted to $\|\mathbf{x}\| \leq 3$, and the Gaussian density $\mathsf{N}(0, I_d)$ as the density p. For double-density functionals, we considered $\mathrm{Unif}([0,2]^d)$, $\mathsf{N}(0, 4I_d)$

[2]The code is available at https://github.com/jongharyu/knn-functional-estimation.

restricted to $\mathbf{B}(0, 3)$, and $\mathsf{N}(0, 4I_d)$ as the density q.[3] Note that all the functionals considered in these simulations can be expressed in closed form up to incomplete gamma function, except the exponential entropies, Le Cam distance, and Jensen–Shannon divergences for Gaussian densities. We estimated the latter using Monte Carlo approximation. Polynomial rates of convergence were observed for all cases, and in each case, the exponent was calculated by ordinary least-squares linear regression between the logarithms of the sample sizes and the MSE. We considered $k \in \{1, 2, 3, 4, 5, 10, 15\}$ and, for double-density functional estimators, $l = k$ for simplicity.

Figure 4.7.1 presents the convergence of the estimator for differential entropy, α-entropies for $\alpha \in \{0.5, 1.5\}$, logarithmic 2-entropy, and exponential $(2.5, 1)$-entropy for 3-dimensional densities. The simulation results show that smaller k yields faster convergence while incurring larger variance, which suggests the use of a moderate size of k in practice. Figure 4.7.2 summarizes the empirical exponents of the estimator for each functional and density. A simple upper bound $(2/d)\wedge 1$ on the theoretical exponents established in Corollary 4.3.20 is also plotted for comparison; see also Examples 4.3.24 and 4.3.25. Empirical convergence rates are consistently better than theoretical bounds for the truncated densities.

Corresponding simulation results for a few representative double-density functionals (KL divergence, α-divergence, logarithmic α-divergence, Le Cam distance, and Jensen–Shannon divergence) are presented in Figures 4.7.3 and 4.7.4. These simulations indicate that the requirement $k > -4a$ and $l > -4\tilde{a}$ in Theorem 4.4.8 may be relaxed to the milder condition $k > -2a$ and $l > -2\tilde{a}$. For example, the estimator with $k = l = 4$

[3] As an exception for the experiment with the Jensen–Shannon divergence estimator, instead of $\mathrm{Unif}([0, 1]^d)$ and $\mathrm{Unif}([0, 2]^d)$, we used piecewise constant densities p and q supported on $[0, 1]^d$, which are defined as follows:

$$p(\mathbf{x}) = \begin{cases} 3/2 & \text{if } 0 \le x_1 \le 1/2, \\ 1/2 & \text{if } 1/2 < x \le 1, \end{cases} \quad \text{and} \quad q(\mathbf{x}) = \begin{cases} 1/2 & \text{if } 0 \le x_1 \le 1/2, \\ 3/2 & \text{if } 1/2 < x \le 1. \end{cases}$$

for logarithmic 2-divergence ($k = 3 \leq -4(1-2) = 4$ and $l = 3 \leq -4(1-2)$) still exhibit consistency in Figure 4.7.3. As presented in the last two rows in Figures 4.7.3 and 4.7.4, simulations also indicate that our estimator is consistent in practice for the exceptional examples of Le Cam distance and Jensen–Shannon divergence even without truncation. For estimating Le Cam distance, we observed that using too large values for k or l lead to bad convergence behavior for small dimensions; see, e.g., the case of $k = l = 15$ for $d = 1$ at the second column of the fourth row in Figure 4.7.4.

4.7 Concluding Remarks

In this paper, we developed a systematic approach to designing k-NN based consistent estimators for a variety of functionals, starting from the fundamental requirement of asymptotic unbiasedness and utilizing the limiting behavior of the k-NN statistics (Proposition 4.1.1). The proposed estimators rediscovered and unified several existing k-NN based estimators for Shannon entropy, KL divergence, α-entropies and α-divergences, and polynomial functionals, which have been sporadically studied and individually analyzed in the literature. It demystified the need of the known, but rather ad-hoc "bias corrections" for some functionals, providing an alternative, principled recipe to identify L_2-consistent estimators. Our list of examples is not exhaustive; other density functionals in the same form may exist or may be discovered in future, and our recipe will furnish consistent k-NN estimators for the same, with nonasymptotic performance predicted by our current analysis.

We remark that the established convergence rates are not minimax optimal; see Remark 4.3.23. As further noted in Remark 4.3.22, the proposed estimators cannot adapt to a higher order of smoothness $\sigma > 2$, due to the inherent limitation of positive-valued kernels. One possible solution to both problems is the ensemble approach (Moon and Hero, 2014a; Sricharan et al., 2013) that takes a weighted average of multiple estimators

based on the asymptotic bias expansion of each density functional estimator. Studying the ensemble version of the estimators is beyond our scope and left as a future direction; see (Berrett and Samworth, 2019) for a weighted version of the proposed divergence functional estimator with local minimax optimality.

Throughout the paper, we assumed the Euclidean distance $\rho(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$. We conclude the paper with specifying technical issues one needs to address in order to extend the results of this paper to a general metric measure space $(\mathcal{X}, \rho, \mu)$, where $(\mathcal{X}, \rho)$ is a complete separable metric space and μ is a locally finite measure on the Borel σ-algebra of $\mathcal{X}$ (see, e.g., Sturm (2006)). Consider a μ-absolutely continuous probability measure P with density p. In general, the weak convergence property in Proposition 4.1.1 for asymptotic unbiasedness (Theorems 4.3.6 and 4.4.7) requires the Lebesgue differentiation theorem to hold in the metric measure space $(\mathcal{X}, \rho, \mu)$, i.e., we need

$$\lim_{r \to 0} \frac{\mathsf{P}(\mathbf{B}(x, r))}{\mu(\mathbf{B}(x, r))} = p(x)$$

for μ-a.e. $x \in \mathcal{X}$. Further, for the bias rate analysis to work, we need to extend Lemma 4.B.6, which states that if p is locally σ-Hölder smooth on $\mathbf{B}(x, R)$, then for $r < R$,

$$\left| \frac{\mathsf{P}(\mathbf{B}(x, r))}{\mu(\mathbf{B}(x, r))} - p(x) \right| \lesssim r^{\sigma} \text{ and } \left| \frac{\mathrm{d}\,\mathsf{P}(\mathbf{B}(x, r))}{\mathrm{d}\mu(\mathbf{B}(x, r))} - p(x) \right| \lesssim r^{\sigma}.$$

If there exists a nonsmooth boundary, we then further need Lemma 4.B.24 to hold in the metric measure space. For the variance analysis to hold under p-norm and other norms, we can apply and extend the analysis in (Gao et al., 2018) as pointed out earlier in Remark 4.3.12.

Figure 4.7.1 ***(following page).*** Convergence of the single-density functional estimator for differential entropy, α-entropies $\alpha \in \{0.5, 1.5\}$, logarithmic 2-entropy, and exponential $(2.5, 1)$-entropy for 3-dimensional densities. The first, second, and third columns present simulation results with $\mathrm{Unif}([0,1]^3)$, $\mathsf{N}(0, I_3)$ restricted to $\|\mathsf{x}\| \leq 3$, and $\mathsf{N}(0, I_3)$, respectively. The true functional values are indicated as dashed lines and one sample standard deviations of the estimates are indicated as shaded area.

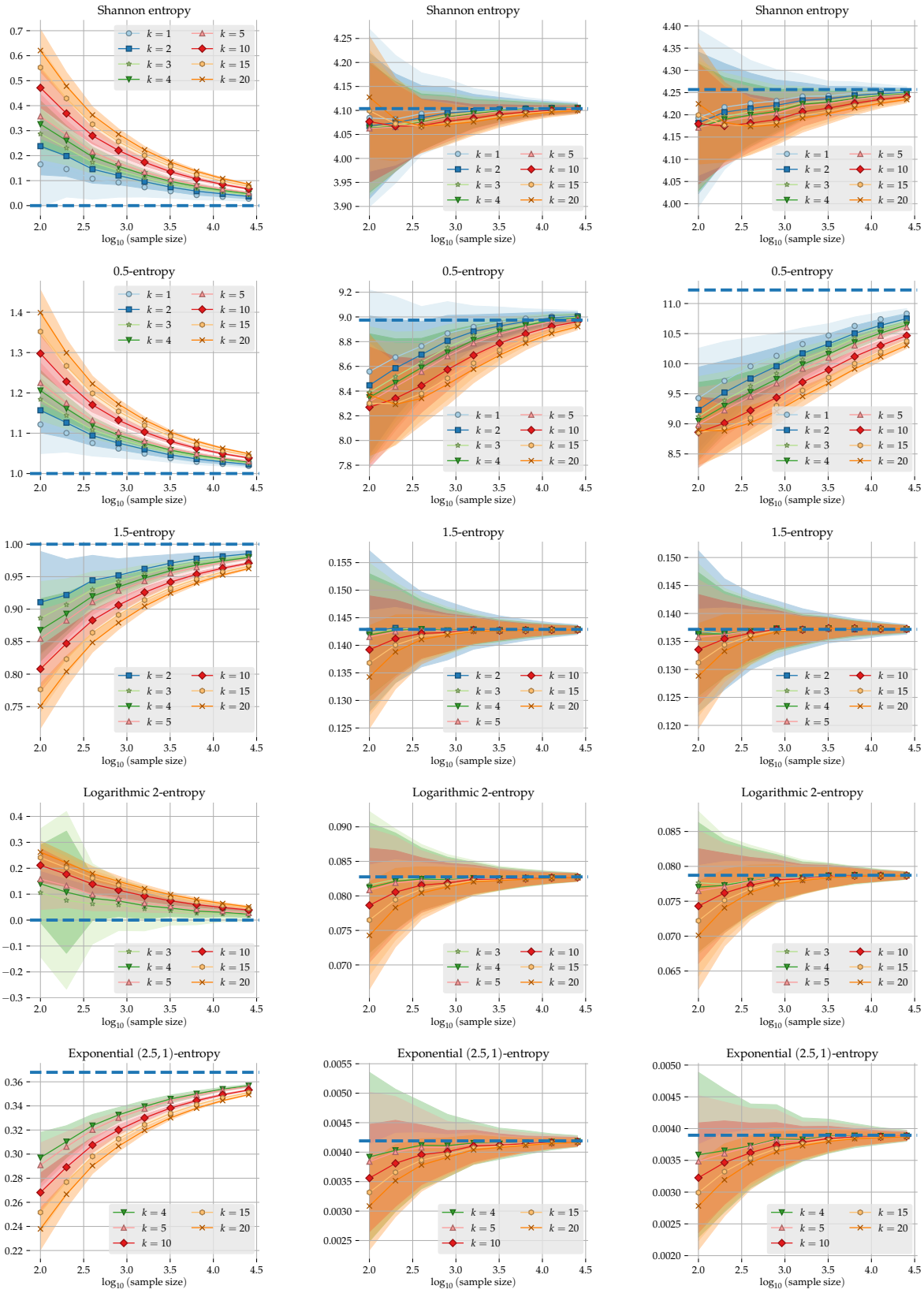

Shannon entropy
Shannon entropy
Shannon entropy
0.5-entropy
0.5-entropy
0.5-entropy
1.5-entropy
1.5-entropy
1.5-entropy
Logarithmic 2-entropy
Logarithmic 2-entropy
Logarithmic 2-entropy
Exponential (2.5, 1)-entropy
Exponential (2.5, 1)-entropy
Exponential (2.5, 1)-entropy
log10 (sample size)
k = 1
k = 2
k = 3
k = 4
k = 5
k = 10
k = 15
k = 20

Figure 4.7.2 ***(following page).*** Simulated MSE rate exponents of the single-density functional estimator for differential entropy, α-entropies for $\alpha \in \{0.5, 1.5\}$, logarithmic 2-entropy, and exponential $(2.5, 1)$-entropy. The first, second, and third columns present simulation results with $\mathrm{Unif}([0,1]^d)$, $\mathsf{N}(0, I_d)$ restricted to $\|\mathbf{x}\| \leq 3$, and $\mathsf{N}(0, I_d)$, respectively, for $d \in \{1, 2, 3, 4, 5\}$.

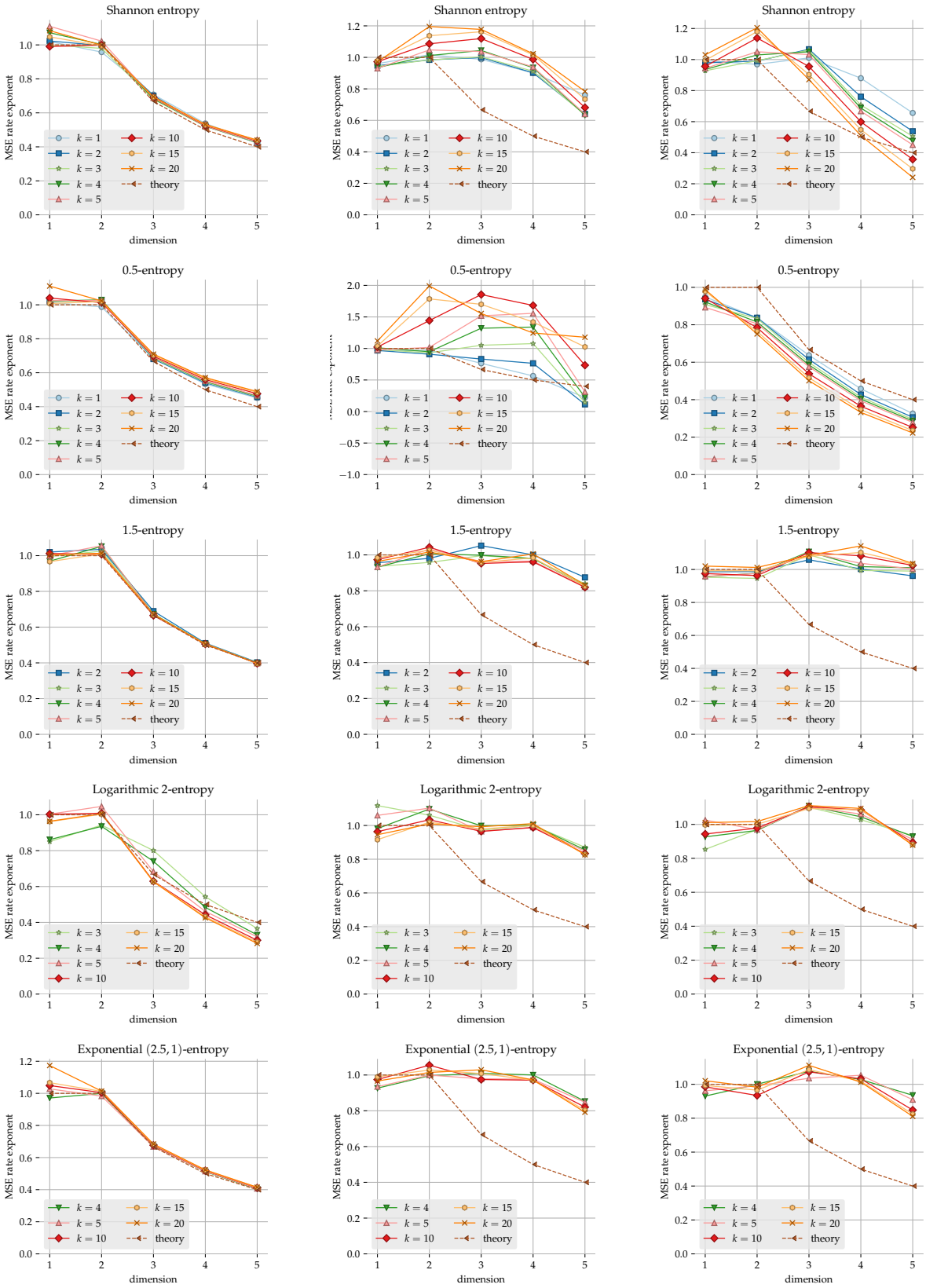

Shannon entropy
Shannon entropy
Shannon entropy
0.5-entropy
0.5-entropy
0.5-entropy
1.5-entropy
1.5-entropy
1.5-entropy
Logarithmic 2-entropy
Logarithmic 2-entropy
Logarithmic 2-entropy
Exponential (2.5, 1)-entropy
Exponential (2.5, 1)-entropy
Exponential (2.5, 1)-entropy
MSE rate exponent
dimension
k = 1
k = 2
k = 3
k = 4
k = 5
k = 10
k = 15
k = 20
theory

Figure 4.7.3 ***(following page).*** Convergence of the double-density functional estimator for KL divergence, 1.5-divergence, and logarithmic 2-divergence for 3-dimensional densities. The first, second, and third columns present simulation results for the densities p and q considered as $\mathrm{Unif}([0,1]^3)$ and $\mathrm{Unif}([0,2]^3)$, $\mathsf{N}(0, I_3)$ restricted to $\|\mathbf{x}\| \leq 3$ and $\mathsf{N}(0, 4I_3)$ restricted to $\|\mathbf{x}\| \leq 3$, and $\mathsf{N}(0, I_3)$ and $\mathsf{N}(0, 4I_3)$, respectively. The true functional values are indicated as dashed lines and one sample standard deviations of the estimates are indicated as shaded area. LCD and JSD are abbreviations for Le Cam distance and Jensen–Shannon divergence, respectively.

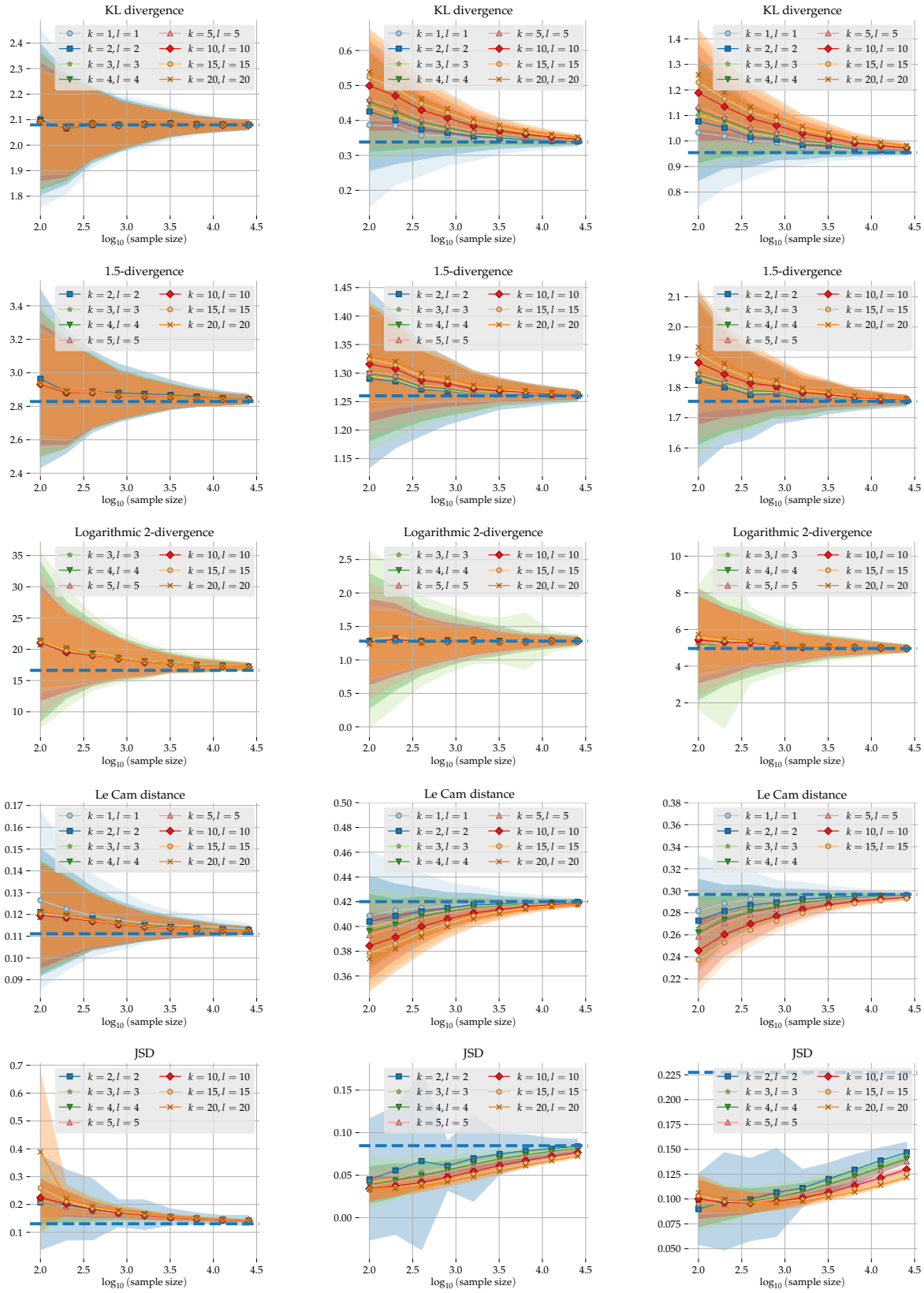
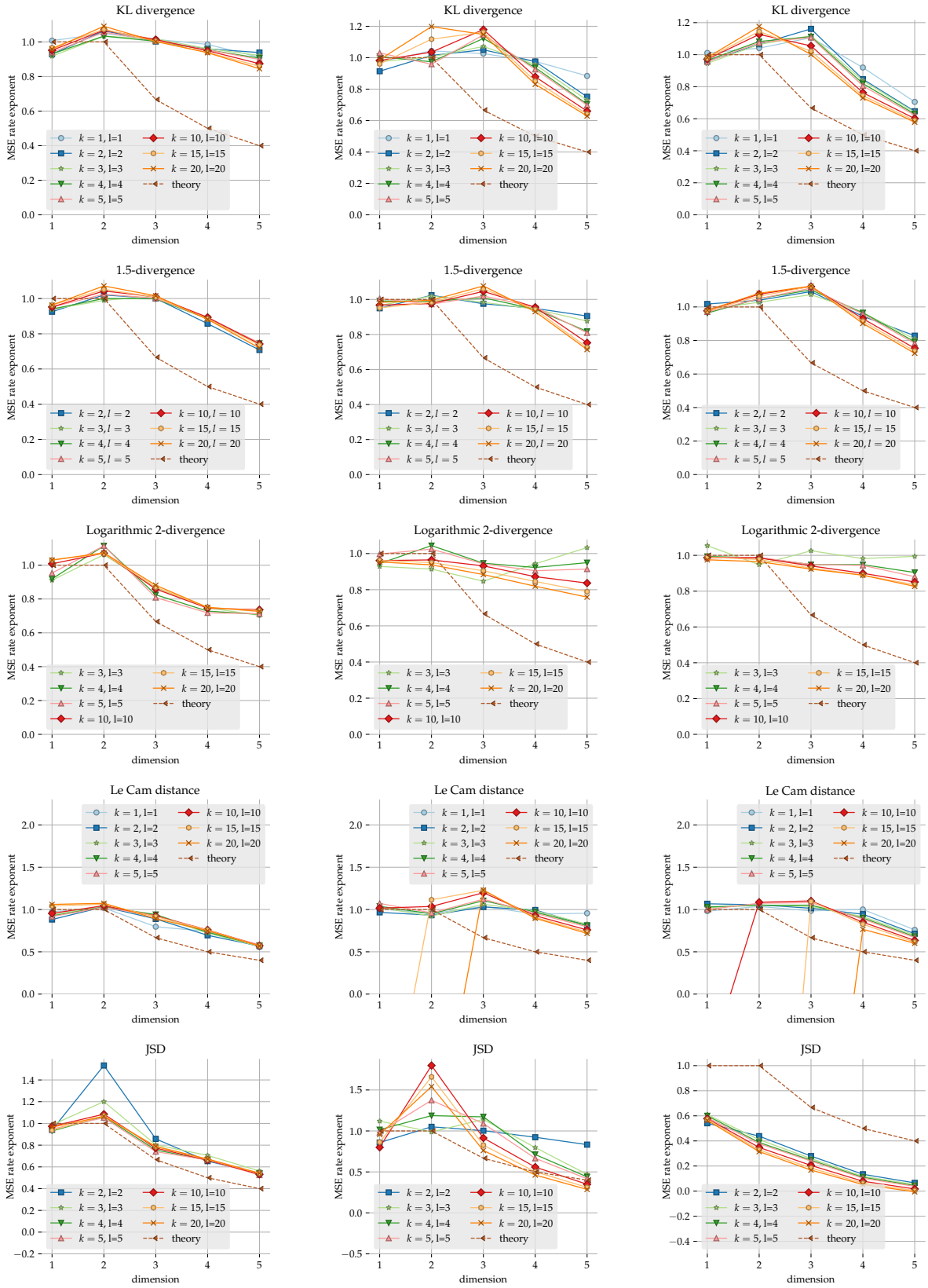

KL divergence
KL divergence
KL divergence
1.5-divergence
1.5-divergence
1.5-divergence
Logarithmic 2-divergence
Logarithmic 2-divergence
Logarithmic 2-divergence
Le Cam distance
Le Cam distance
Le Cam distance
JSD
JSD
JSD
$k = 1, l = 1$
$k = 2, l = 2$
$k = 3, l = 3$
$k = 4, l = 4$
$k = 5, l = 5$
$k = 10, l = 10$
$k = 15, l = 15$
$k = 20, l = 20$
$\log_{10}$ (sample size)

Figure 4.7.4 ***(following page).*** Simulated MSE rate exponents of the double-density functional estimator for KL divergence, 1.5-divergence, and logarithmic 2-divergence. The first, second, and third columns present simulation results for the densities p and q considered as $\mathrm{Unif}([0,1]^3)$ and $\mathrm{Unif}([0,2]^3)$, $\mathsf{N}(0, I_3)$ restricted to $\|\mathbf{x}\| \leq 3$ and $\mathsf{N}(0, 4I_3)$ restricted to $\|\mathbf{x}\| \leq 3$, and $\mathsf{N}(0, I_3)$ and $\mathsf{N}(0, 4I_3)$, respectively, for $d \in \{1, 2, 3, 4, 5\}$. LCD and JSD are abbreviations for Le Cam distance and Jensen–Shannon divergence, respectively.


KL divergence
1.5-divergence
Logarithmic 2-divergence
Le Cam distance
JSD
MSE rate exponent
dimension
k = 1, l=1
k = 2, l=2
k = 3, l=3
k = 4, l=4
k = 5, l=5
k = 10, l=10
k = 15, l=15
k = 20, l=20
theory
k = 2, l = 2
k = 3, l = 3
k = 4, l = 4
k = 5, l = 5
k = 10, l = 10
k = 15, l = 15
k = 20, l = 20

Appendix

4.A Notation

In what follows, let $\mathrm{P}_U(u) = \mathsf{P}\{U \leq u\}$ and $\rho_U(u) = \mathrm{d}\mathrm{P}_U(u)/\,\mathrm{d}u$ denote the cumulative distribution function (cdf) and the density of a random variable U, respectively. We use $B_{n,P}$ to denote a binomial random variable with parameters n and P. We use P_q to denote a Poisson random variable with rate $q > 0$. We use $X_{\alpha,\beta}$ to denote a beta random variable with parameters $\alpha, \beta > 0$ for $\alpha, \beta > 0$, whose density is

$$\frac{t^{\alpha-1}(1-t)^{\beta-1}}{\mathsf{B}(\alpha,\beta)}, \quad 0 \leq t \leq 1.$$

Here $\mathsf{B}(\alpha,\beta) := \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}\,\mathrm{d}t$ denotes the beta function. Finally, we use H^{d-1} to denote the $(d-1)$-dimensional Hausdorff measure.

4.B Technical Lemmas

4.B.1 Auxiliary Lemmas

Lemma 4.B.1. *Assume that P and $\tilde{\mathsf{P}}$ have densities p and $\tilde{p}$, respectively, with respect to the Lebesgue measure λ. If $\mathsf{P} \ll \tilde{\mathsf{P}}$, then $\mathsf{P}(\{\mathbf{x}\colon m_r\tilde{p}(\mathbf{x}) > 0\}) = 1$ for any $r > 0$.*

Proof. Let $r > 0$ be fixed. We first observe that $\mathsf{P}(\mathrm{supp}(\tilde{p})) = 1$, since

$$1 - \mathsf{P}(\mathrm{supp}(\tilde{p})) = \int p(\mathbf{x})(1 - 1_{\mathrm{supp}(\tilde{p})}(\mathbf{x}))\,\mathrm{d}\mathbf{x}$$

$$
\begin{aligned}
&= \int p(\mathbf{x})\, 1_{\{\exists \delta > 0 \text{ s.t.} \tilde{\mathsf{P}}(\mathbf{B}(\mathbf{x},\delta))=0\}}\, \mathrm{d}\mathbf{x} \\
&\overset{(a)}{\leq} \int p(\mathbf{x})\, 1_{\{\exists \delta > 0 \text{ s.t. } \mathsf{P}(\mathbf{B}(\mathbf{x},\delta))=0\}}\, \mathrm{d}\mathbf{x}, \\
&\overset{(b)}{=} 0.
\end{aligned}
$$

Here, (a) follows from the absolute continuity $\mathsf{P} \ll \tilde{\mathsf{P}}$, and (b) follows since $p(\mathbf{x}) = 0$ for a.e. $\mathbf{x}$ over the set $\{\mathbf{x} \colon \exists \delta > 0 \text{ s.t. } \mathsf{P}(\mathbf{B}(\mathbf{x}, \delta)) = 0\}$, by the Lebesgue differentiation theorem.

Now, define $A_\delta \tilde{p}(\mathbf{x}) = \tilde{\mathsf{P}}(\mathbf{B}(\mathbf{x}, \delta))/\, \lambda(\mathbf{B}(\mathbf{x}, \delta))$ for each $\delta > 0$ and $\mathbf{x} \in \mathbb{R}^d$. On the one hand, we have

$$
\lim_{\delta \to 0} A_\delta \tilde{p}(\mathbf{x}) = \tilde{p}(\mathbf{x})
$$

for λ-a.e. $\mathbf{x}$ by the Lebesgue differentiation theorem. On the other hand, for $\mathbf{x} \in T \cap \operatorname{supp}(\tilde{p})$ where $T := \{\mathbf{x} \colon m_r \tilde{p}(\mathbf{x}) = 0\}$, we have $A_\delta \tilde{p}(\mathbf{x}) > 0$ for every $\delta > 0$ and

$$
0 = m_r \tilde{p}(\mathbf{x}) = \inf_{0 < \delta \leq r} A_\delta \tilde{p}(\mathbf{x})
$$

for any $r > 0$. Hence, we must have

$$
\tilde{p}(\mathbf{x}) = \lim_{\delta \to 0} A_\delta \tilde{p}(\mathbf{x}) = 0
$$

for λ-a.e. $\mathbf{x} \in T \cap \operatorname{supp}(\tilde{p})$, which implies that $\tilde{\mathsf{P}}(T \cap \operatorname{supp}(\tilde{p})) = 0$, and thus $\mathsf{P}(T \cap \operatorname{supp}(\tilde{p})) = 0$ since $\mathsf{P} \ll \tilde{\mathsf{P}}$. This, together with $\mathsf{P}(\operatorname{supp}(\tilde{p})) = 1$, establishes that $\mathsf{P}(T) = 0$. □

Lemma 4.B.2. *For the* lower incomplete gamma function $\gamma(s, x) := \int_0^x t^{s-1} e^{-t}\, \mathrm{d}t$ *and the* upper incomplete gamma function $\Gamma(s, x) := \int_x^\infty t^{s-1} e^{-t}\, \mathrm{d}t$*, we have*

$$
\gamma(s, x) \leq \Gamma(s) \wedge \frac{x^s}{s}, \qquad \forall s > 0, x > 0, \tag{4.46}
$$

$$\Gamma(s,x) \le \Gamma(s) x^{s-1} e^{-x+1}, \qquad \forall s \ge 1, x \ge 1. \tag{4.47}$$

Proof. As $\Gamma(s,x)/\Gamma(s)$ is decreasing in s for fixed $x \ge 1$, we have that for $s \ge 1$,

$$\begin{aligned}\frac{\Gamma(s,x)}{\Gamma(s)} \le \frac{\Gamma(\lfloor s \rfloor, x)}{\Gamma(\lfloor s \rfloor)} &= e^{-x} \sum_{k=0}^{\lfloor s \rfloor - 1} \frac{x^k}{k!} \\ &\le e^{-x} x^{\lfloor s \rfloor - 1} \sum_{k=0}^{\infty} \frac{1}{k!} \le e^{-x+1} x^{s-1}.\end{aligned}$$

The second inequality follows since, for any $x > 0$, letting $t = xe^{-u}$, we have

$$\begin{aligned}\gamma(s,x) = \int_0^x t^{s-1} e^{-t} \,\mathrm{d}t &= x^s \int_0^\infty e^{-(su + xe^{-u})} \,\mathrm{d}u \\ &\le x^s \int_0^\infty e^{-su} \,\mathrm{d}u = \frac{x^s}{s}.\end{aligned}$$

□

4.B.2 Convergence of Distribution of k-NN Statistics

We first state a basic statistical property of k-NN statistics.

Lemma 4.B.3 (Distribution of k-NN distance)**.** *The cdf of $r_{km}(\mathbf{x})$ is*

$$\mathrm{P}_{r_{km}(\mathbf{x})}(r) = \mathsf{P}\{B_{m,\boldsymbol{P}(\mathbf{B}(\mathbf{x},r))} \ge k\} = \mathrm{P}_{X_{k,m-k+1}}(\boldsymbol{P}(\mathbf{B}(\mathbf{x},r))).$$

Proof. Consider

$$\begin{aligned}\mathrm{P}_{r_{km}(\mathbf{x})}(r) &= \mathsf{P}\{r_{km}(\mathbf{x}) \le r\} \\ &= \mathsf{P}\{\rho(\mathbf{x}, \mathbf{X}_{(k)}(\mathbf{x})) \le r\} \\ &= \mathsf{P}\big\{|\{i \in [m] \colon \mathbf{X}_i \in \mathbf{B}(\mathbf{x},r)\}| \ge k\big\} \\ &= \mathsf{P}\{B_{m,\mathsf{P}(\mathbf{B}(\mathbf{x},r))} \ge k\} \\ &= \mathrm{P}_{X_{k,m-k+1}}(\mathsf{P}(\mathbf{B}(\mathbf{x},r))).\end{aligned}$$

The last equality follows from the identity

$$\mathsf{P}\{B_{m,P} \geq k\} = \mathrm{P}_{X_{k,m-k+1}}(P). \qquad \square$$

Using this fact, Proposition 4.1.1, which claims the weak convergence of the k-NN statistics $U_{km}(\mathbf{x})$ to a Gamma random variable, readily follows.

Proof of Proposition 4.1.1. Fix $\mathbf{x} \in \mathbb{R}^d$ and $u > 0$, and let $P_m := \mathsf{P}(\mathbf{B}(\mathbf{x}, \varrho(\frac{u}{m})))$. Since $\mathrm{P}_{U_{km}(\mathbf{x})}(u) = \mathrm{P}_{r_{km}(\mathbf{x})}(\varrho(\frac{u}{m}))$, we have

$$\mathrm{P}_{U_{km}(\mathbf{x})}(u) = \mathsf{P}\{B_{m,P_m} \geq k\}$$

from Lemma 4.B.3. By the Lebesgue differentiation theorem (see, e.g., (Rudin, 1987)), for λ-a.e. $\mathbf{x}$,

$$\lim_{m\to\infty} mP_m = \lim_{m\to\infty} u\frac{\mathsf{P}(\mathbf{B}(\mathbf{x}, \varrho(\frac{u}{m})))}{\lambda(\mathbf{B}(\mathbf{x}, \varrho(\frac{u}{m})))} = up(\mathbf{x}).$$

Therefore, for each $i = 0, \ldots, k-1$, we have

$$\begin{aligned}
&\lim_{m\to\infty} \binom{m}{i} P_m^i (1-P_m)^{m-i} \\
&= \lim_{m\to\infty} \frac{i!}{m^i} \binom{m}{i} (1-P_m)^{m-i} \frac{(mP_m)^i}{i!} \\
&= e^{-up(\mathbf{x})} \frac{(up(\mathbf{x}))^i}{i!},
\end{aligned}$$

since

$$\lim_{m\to\infty} \frac{i!}{m^i}\binom{m}{i} = 1 \text{ and } \lim_{m\to\infty} (1-P_m)^{m-i} = e^{-up(\mathbf{x})}.$$

This leads us to concludes that

$$\begin{aligned}\lim_{m\to\infty} \mathsf{P}\{U_{km}(\mathbf{x}) > u\} &= \sum_{i=0}^{k-1} e^{-up(\mathbf{x})} \frac{up(\mathbf{x})^i}{i!} \\ &= \mathsf{P}\{U_{k\infty}(\mathbf{x}) > u\},\end{aligned}$$

where $U_{k\infty}(\mathbf{x})$ is a $\mathsf{G}(k, p(\mathbf{x}))$ random variable. □

Moreover, if the density p is locally smooth, then one can establish a polynomial convergence rate of the density of $U_{km}(\mathbf{x})$ to $U_{k\infty}(\mathbf{x})$ as follows.

Lemma 4.B.4 (Generalization of (Gao et al., 2018, Lemma 2))**.** *Suppose that $\nu_m = o(\sqrt{m})$ and $k = k_m = o(\sqrt{m})$ as $m \to \infty$. For $\mathbf{x} \in \mathrm{supp}(p)$, if $p(\mathbf{x}) \le C_p < \infty$ and p is σ_p-Hölder continuous ($\sigma_p \in [0, 2]$) over $\mathbf{B}(\mathbf{x}, \varrho(\frac{u}{m}))$ with Hölder constant L, we have*

$$\begin{aligned}&\left|\rho_{U_{km}(\mathbf{x})}(u) - \rho_{U_{k\infty}(\mathbf{x})}(u)\right| \\ &\lesssim_{\sigma_p, L, C_p, d} (1+u)\Big(\frac{u}{m}\Big)^{\frac{\sigma_p}{d}} + k^{-k}\frac{(k^2+u^2)u^{k-1}e^{-up(\mathbf{x})}}{m}\end{aligned}$$

for $u \in [0, \nu_m]$ and m sufficiently large.

We first state two technical lemmas required to prove Lemma 4.B.4, whose proofs are omitted here; we refer the interested readers to (Gao et al., 2018). The first lemma in the following establishes a rate of convergence of a Poisson binomial random variable $B_{m,Q/m} \sim \mathrm{Binom}(m, Q/m)$ to a Poisson random variable $P_Q \sim \mathrm{Poisson}(Q)$ in distribution.

Lemma 4.B.5 (Generalization of (Gao et al., 2018, Lemma 5))**.** *For any $Q, k = o(\sqrt{m})$ as $m \to \infty$, there exists a constant $C_0 > 0$ such that for m sufficiently large*

$$\left|\mathsf{P}\{B_{m,\frac{Q}{m}} = k\} - \mathsf{P}\{P_Q = k\}\right| \le C_0 \frac{Q^k e^{-Q}}{k!}\frac{(k^2+Q^2)}{m}.$$

The second lemma establishes the speed of convergence of $\mathsf{P}(\mathbf{B}(\mathbf{x}, r))/\lambda(\mathbf{B}(\mathbf{x}, r))$ and $\mathrm{d}\,\mathsf{P}(\mathbf{B}(\mathbf{x}, r))/\mathrm{d}\,\lambda(\mathbf{B}(\mathbf{x}, r))$ to $p(\mathbf{x})$ as $r \to 0$, when p is locally smooth at $\mathbf{x}$.

Lemma 4.B.6 (Generalization of (Gao et al., 2018, Lemma 4))**.** *If a density p is σ_p-Hölder continuous with constant $L > 0$ over $\mathbf{B}(\mathbf{x}, R)$ for $\mathbf{x} \in \mathbb{R}^d$ and some $\sigma_p \in [0, 2]$, we have for any $0 < r < R$,*

$$\left|\frac{\mathsf{P}(\mathbf{B}(\mathbf{x}, r))}{\lambda(\mathbf{B}(\mathbf{x}, r))} - p(\mathbf{x})\right| \leq \frac{d}{\sigma_p + d} L r^{\sigma_p},$$
$$\left|\frac{\mathrm{d}\,\mathsf{P}(\mathbf{B}(\mathbf{x}, r))}{\mathrm{d}\lambda(\mathbf{B}(\mathbf{x}, r))} - p(\mathbf{x})\right| \leq L r^{\sigma_p}.$$

The proof of the first inequality can be found in (Jiao et al., 2018) and the second inequality can be proved by a similar argument.

Remark 4.B.7. If g is bounded above over $\mathbf{B}(\mathbf{x}, R)$, then g is σ_p-Hölder continuous over $\mathbf{B}(\mathbf{x}, R)$ with $\sigma_p = 0$. The convergence of $U_{km}(\mathbf{x})$ to a $\mathsf{G}(k, p(\mathbf{x}))$ random variable as $m \to \infty$ can be quantified in terms of a gap between the densities using this lemma and the order of smoothness σ_p of the underlying density p; however, the bounds in Lemma 4.B.6 cannot be improved further beyond $O(r^2)$. It is consistent with the observation that the higher-order smoothness beyond 2 cannot be exploited with k-NN methods (Han et al., 2020; Tsybakov and van der Meulen, 1996).

Now we are ready to present the proof of Lemma 4.B.4.

Proof of Lemma 4.B.4. First note that the density of the k-th NN statistics $r_{km}(\mathbf{x})$ is

$$\begin{aligned}\rho_{r_{km}(\mathbf{x})}(r) &= m\mathsf{P}\{B_{m-1,\mathsf{P}(\mathbf{B}(\mathbf{x},r))} = k-1\}\frac{\mathrm{d}\,\mathsf{P}(\mathbf{B}(\mathbf{x}, r))}{\mathrm{d}r} \\ &= g_{km}(\mathsf{P}(\mathbf{B}(\mathbf{x}, r)))\frac{\mathrm{d}\,\mathsf{P}(\mathbf{B}(\mathbf{x}, r))}{\mathrm{d}r}\end{aligned}$$

from Lemma 4.B.3 in Appendix 4.B.2. Here we define

$$g_{km}(P) := m\mathsf{P}\{B_{m-1,P} = k-1\}$$

for $p \in [0,1]$, which is the density of the k-th order statistic from among m random samples drawn from the uniform distribution over $[0,1]$. It is easy to check that $g_{km}(P) \le m$ and $g'_{km}(P) \le 2m(m-1) \le 2m^2$ for any $P \in [0,1]$. Recall that $P_m(u|\mathbf{x}) := \mathsf{P}(\mathbf{B}(\mathbf{x}, \varrho(\frac{u}{m})))$. The density of $U_{km}(\mathbf{x})$ can then be written as

$$\begin{aligned}\rho_{U_{km}(\mathbf{x})}(u) &= \rho_{r_{km}(\mathbf{x})}\big(\varrho\big(\frac{u}{m}\big)\big)\frac{\mathrm{d}\,\varrho(\frac{u}{m})}{\mathrm{d}u} \\ &= g_{km}(P_m(u|\mathbf{x}))\frac{\mathrm{d}P_m(u|\mathbf{x})}{\mathrm{d}u}.\end{aligned}$$

We define an intermediate density approximation

$$\rho_{km}(u) := g_{km}\Big(\frac{up(\mathbf{x})}{m}\Big)\frac{p(\mathbf{x})}{m}$$

for $u \le m/C_p$, and bound the density gap by

$$|\rho_{U_{km}(\mathbf{x})}(u) - \rho_{U_{k\infty}(\mathbf{x})}(u)| \le |\rho_{U_{km}(\mathbf{x})}(u) - \rho_{km}(u)| + |\rho_{km}(u) - \rho_{U_{k\infty}(\mathbf{x})}(u)|.$$

We bound each term on the right hand side.

For the first term, consider

$$\begin{aligned}|\rho_{U_{km}(\mathbf{x})}(u) - \rho_{km}(u)| &\le g_{km}(P_m(u|\mathbf{x}))\Big|\frac{\mathrm{d}P_m(u|\mathbf{x})}{\mathrm{d}u} - \frac{p(\mathbf{x})}{m}\Big| \\ &\quad + \Big|g_{km}(P_m(u|\mathbf{x})) - g_{km}\Big(\frac{up(\mathbf{x})}{m}\Big)\Big|\frac{p(\mathbf{x})}{m} \\ &\le g_{km}(P_m(u|\mathbf{x}))\Big|\frac{\mathrm{d}P_m(u|\mathbf{x})}{\mathrm{d}u} - \frac{p(\mathbf{x})}{m}\Big| \\ &\quad + \max_{p\in(0,1)}|g'_{km}(p)|\Big|P_m(u|\mathbf{x}) - \frac{up(\mathbf{x})}{m}\Big|\frac{p(\mathbf{x})}{m}\end{aligned}$$

$$
\begin{aligned}
&\leq m\Big|\frac{dP_m(u|\mathbf{x})}{\mathrm{d}u} - \frac{p(\mathbf{x})}{m}\Big| + 2m^2\Big|P_m(u|\mathbf{x}) - \frac{up(\mathbf{x})}{m}\Big|\frac{p(\mathbf{x})}{m} \\
&= \Big|\frac{\mathrm{d}\,\mathsf{P}(\mathbf{B}(\mathbf{x}, \varrho(\frac{u}{m})))}{\mathrm{d}\lambda(\mathbf{B}(\mathbf{x}, \varrho(\frac{u}{m})))} - p(\mathbf{x})\Big| + 2up(\mathbf{x})\Big|\frac{\mathsf{P}(\mathbf{B}(\mathbf{x}, \varrho(\frac{u}{m})))}{\lambda(\mathbf{B}(\mathbf{x}, \varrho(\frac{u}{m})))} - p(\mathbf{x})\Big| \\
&\leq \Big(1 + 2C_p\frac{d}{\sigma_p + d}u\Big) L\, \varrho^{\sigma_p}\big(\frac{u}{m}\big) \\
&\lesssim_{\sigma_p, L, C_p, d} (1+u)\Big(\frac{u}{m}\Big)^{\frac{\sigma_p}{d}}.
\end{aligned}
$$

The second last inequality follows from Lemma 4.B.6. Note that this term is independent of k.

The second term can be bounded using Lemma 4.B.5. For m sufficiently large, we have

$$
\begin{aligned}
|\rho_{km}(u) - \rho_{U_{k\infty}(\mathbf{x})}(u)| &= \frac{k}{u}|\mathsf{P}\big\{B_{m,up(\mathbf{x})/m} = k\big\} - \mathsf{P}\{P_{up(\mathbf{x})} = k\}| \\
&\leq \frac{k}{u}C_0\frac{(up(\mathbf{x}))^k e^{-up(\mathbf{x})}}{k!}\frac{k^2 + u^2p^2(\mathbf{x})}{m} \\
&= \frac{C_0}{\Gamma(k)}\frac{(k^2 + (up(\mathbf{x}))^2)(up(\mathbf{x}))^k e^{-up(\mathbf{x})}}{mu} \\
&\lesssim_{C_0, C_p} k^{-k}\frac{(k^2+u^2)u^{k-1}e^{-up(\mathbf{x})}}{m},
\end{aligned}
$$

which holds uniformly for all $u, k = o(\sqrt{m})$ as $m \to \infty$. Here we use the Stirling approximation $C_p^k/k! \sim (eC_p)^k/k^{k+\frac{1}{2}}$. □

Remark 4.B.8. This proof closely follows the one in (Gao et al., 2018), while keeping track of the explicit dependence on the constants C_0, C_p and k.

The following lemma quantifies the convergence of the cdf of $U_{km}(\mathbf{x})$ to the cdf of $U_{k\infty}(\mathbf{x})$ when the underlying density p is smooth.

Lemma 4.B.9 (Generalization of (Gao et al., 2018, Lemma 3))**.** *Suppose that $\nu_m = o(\sqrt{m})$ and $k = k_m = o(\sqrt{m})$ as $m \to \infty$. For $\mathbf{x} \in \mathrm{supp}(p)$, if $p(\mathbf{x}) \leq C_p < \infty$ and p is σ_p-Hölder*

continuous ($\sigma_p \in [0,2]$) over $\mathbf{B}(\mathbf{x}, \varrho(u/m))$ with Hölder constant L, we have

$$|\mathrm{P}_{U_{km}(\mathbf{x})}(u) - \mathrm{P}_{U_{k\infty}(\mathbf{x})}(u)| \lesssim_{\sigma_p, L, C_p, d} ku\Big(\frac{u}{m}\Big)^{\frac{\sigma_p}{d}} + \frac{(k^2+u^2)u^{k-1}e^{-up(\mathbf{x})}}{m}, \tag{4.48}$$

for $u \in [1/(p(\mathbf{x})), \nu_m]$ for m sufficiently large.

Proof. First, note that

$$\mathrm{P}_{U_{k\infty}(\mathbf{x})}(u) = 1 - \sum_{j=0}^{k-1} \mathsf{P}\{P_{up(\mathbf{x})} = j\}$$

and

$$\mathrm{P}_{U_{km}(\mathbf{x})}(u) = 1 - \sum_{j=0}^{k-1} \mathsf{P}\{B_{m,P_m(u|\mathbf{x})} = j\},$$

from Lemma 4.B.3 in Appendix 4.B.2. By triangle inequality, we have

$$\begin{aligned}
&\big|\mathrm{P}_{U_{km}(\mathbf{x})}(u) - \mathrm{P}_{U_{k\infty}(\mathbf{x})}(u)\big| \\
&\le \sum_{j=0}^{k-1} \big|\mathsf{P}\{P_{up(\mathbf{x})} = j\} - \mathsf{P}\{B_{m,P_m(u|\mathbf{x})} = j\}\big| \\
&\le \sum_{j=0}^{k-1} \Big\{ \big|\mathsf{P}\{P_{up(\mathbf{x})} = j\} - \mathsf{P}\{B_{m,\frac{up(\mathbf{x})}{m}} = j\}\big| + \big|\mathsf{P}\{B_{m,\frac{up(\mathbf{x})}{m}} = j\} - \mathsf{P}\{B_{m,P_m(u|\mathbf{x})} = j\}\big| \Big\}.
\end{aligned}$$

For the first term, using Lemma 4.B.5, we obtain

$$\big|\mathsf{P}\{P_{up(\mathbf{x})} = j\} - \mathsf{P}\big\{B_{m,\frac{up(\mathbf{x})}{m}} = j\big\}\big| \le C_0 \frac{(up(\mathbf{x}))^j e^{-up(\mathbf{x})}}{j!} \frac{j^2 + (up(\mathbf{x}))^2}{m},$$

for each $j = 0, \ldots, k-1$, which implies that

$$\sum_{j=0}^{k-1} \big|\mathsf{P}\{P_{up(\mathbf{x})} = j\} - \mathsf{P}\big\{B_{m,\frac{up(\mathbf{x})}{m}} = j\big\}\big| \le C_0 \frac{k^2 + (up(\mathbf{x}))^2}{m} e^{-up(\mathbf{x})} \sum_{j=0}^{k-1} \frac{(up(\mathbf{x}))^j}{j!}$$

$$
\begin{aligned}
&= C_0 \frac{k^2 + (up(\mathbf{x}))^2}{m} \frac{\Gamma(k, up(\mathbf{x}))}{\Gamma(k)} \\
&\leq C_0 \frac{k^2 + (up(\mathbf{x}))^2}{m} (up(\mathbf{x}))^{k-1} e^{-up(\mathbf{x})+1},
\end{aligned}
$$

where the last inequality follows from Lemma 4.B.2.

For the second term, we have

$$
\begin{aligned}
\left|\mathsf{P}\{B_{m,\frac{up(\mathbf{x})}{m}} = j\} - \mathsf{P}\{B_{m,P_m(u|\mathbf{x})} = j\}\right| &\leq 2m \left| P_m(u|\mathbf{x}) - \frac{up(\mathbf{x})}{m} \right| \\
&= 2u \left| \frac{\mathsf{P}(\mathbf{B}(\mathbf{x}, \varrho(\frac{u}{m})))}{\lambda(\mathbf{B}(\mathbf{x}, \varrho(\frac{u}{m})))} - p(\mathbf{x}) \right| \\
&\leq 2u \frac{d}{\sigma_p + d} L\, \varrho^{\sigma_p}(\frac{u}{m}),
\end{aligned}
$$

for each $j = 0, \ldots, k-1$, from Lemma 4.B.6.

Putting the bounds together and using the triangle inequality, we have that for $k, u = o(\sqrt{m})$

$$
\begin{aligned}
\left|\mathrm{P}_{U_{km}(\mathbf{x})}(u) - \mathrm{P}_{U_{k\infty}(\mathbf{x})}(u)\right| &\leq \frac{2kud}{\sigma_p + d} L\, \varrho^{\sigma_p}(\frac{u}{m}) + C_0 \frac{k^2 + (up(\mathbf{x}))^2}{m} (up(\mathbf{x}))^{k-1} e^{-up(\mathbf{x})+1} \\
&\lesssim_{\sigma_p, d, L, C_0, C_p} ku \Big(\frac{u}{m}\Big)^{\frac{\sigma_p}{d}} + \frac{(k^2 + u^2) u^{k-1} e^{-up(\mathbf{x})}}{m},
\end{aligned}
$$

which concludes the proof. □

4.B.3 Bounds on Distribution of k-NN Statistics

We now present several bounds on

$$
\begin{aligned}
F_{km}(u|\mathbf{x}) &:= \mathsf{P}\{U_{km}(\mathbf{x}) \leq u\} \\
&= \mathsf{P}\big\{r_{km}(\mathbf{x}) \leq \varrho(\frac{u}{m})\big\} \\
&= \mathsf{P}\{B_{m,P_m(u|\mathbf{x})} \geq k\},
\end{aligned}
$$

which is the cdf of $U_{km}(\mathbf{x})$. Here and henceforth, for $\mathbf{x} \in \mathbb{R}^d$ and $u \geq 0$, we define

$$P_m(u|\mathbf{x}) := \mathsf{P}\big(\mathbf{B}\big(\mathbf{x}, \varrho\big(\frac{u}{m}\big)\big)\big) = \frac{u}{m} \frac{\mathsf{P}(\mathbf{B}(\mathbf{x}, \varrho(\frac{u}{m})))}{\lambda(\mathbf{B}(\mathbf{x}, \varrho(\frac{u}{m})))}.$$

Note that by the definitions of $m_r p(\mathbf{x})$ and $M_r p(\mathbf{x})$, we have

$$u' m_r p(\mathbf{x}) \leq m P_m(u'|\mathbf{x}) \leq m \wedge (u' M_r p(\mathbf{x}))$$

for $r = \varrho(\frac{u}{m})$ and for any $0 < u' \leq u$.

The following lemma presents an upper bound on the cdf $F_{km}(u|\mathbf{x})$.

Lemma 4.B.10 (Generalization of (Bulinski and Dimitrov, 2019b, Eq. (3.19)))**.** *For any* $\mathbf{x} \in \mathbb{R}^d$ *and* $u > 0$*, we have*

$$F_{km}(u|\mathbf{x}) \leq \frac{(m P_m(u|\mathbf{x}))^k}{k!}. \tag{4.49}$$

Proof. Since $F_{km}(u|\mathbf{x}) = \mathrm{P}_{T_{k,m-k+1}}(P_m(u|\mathbf{x}))$ from Lemma 4.B.3, we have

$$\begin{aligned} F_{km}(u|\mathbf{x}) &= \int_0^{P_m(u|\mathbf{x})} \frac{t^{k-1}(1-t)^{m-k}}{\mathsf{B}(k, m-k+1)} \, \mathrm{d}t \\ &\leq \frac{P_m^k(u|\mathbf{x})}{k \, \mathsf{B}(k, m-k+1)} \\ &= \binom{m}{k} P_m^k(u|\mathbf{x}) \\ &\leq \frac{(m P_m(u|\mathbf{x}))^k}{k!}, \end{aligned}$$

which concludes the proof. □

We present two upper bounds on the complementary cdf $1 - F_{km}(u|\mathbf{x})$.

Lemma 4.B.11 ((Bulinski and Dimitrov, 2019b, Eq. (3.23)))**.** *For any* $\mathbf{x} \in \mathbb{R}^d$*,* $0 < D < 1$*,*

and $u \geq 0$, we have

$$1 - F_{km}(u|\mathbf{x}) \leq (1-D)^{-k+1} e^{-DmP_m(u|\mathbf{x})}. \tag{4.50}$$

In particular, if $mP_m(u|\mathbf{x}) > k$, we have

$$1 - F_{km}(u|\mathbf{x}) \leq \Big(\frac{emP_m(u|\mathbf{x})}{k}\Big)^k e^{-mP_m(u|\mathbf{x})}. \tag{4.51}$$

Proof. Since we can write $1 - F_{km}(u|\mathbf{x}) = \mathsf{P}\{B_{m,P_m(u|\mathbf{x})} < k\}$ from Lemma 4.B.3, the bound follows immediately from a Chernoff bound on a binomial random variable. For any $\lambda > 0$,

$$\begin{aligned}
\mathsf{P}\{B_{m,P} < k\} &\leq e^{\lambda k}\mathbb{E}[e^{-\lambda B_{m,P}}] \\
&= e^{\lambda k}(1 - P + Pe^{-\lambda})^m \\
&\leq e^{\lambda k} e^{-mP(1-e^{-\lambda})},
\end{aligned}$$

and this proves (4.50) if we set $D := 1 - e^{-\lambda} \in (0,1)$. If $mP > k$, we then can minimize the right hand side by plugging in $\lambda = \ln \frac{mp}{k}$, which obtains

$$\mathsf{P}\{B_{m,P} < k\} \leq \Big(\frac{emP}{k}\Big)^k e^{-mP}. \qquad \square$$

Lemma 4.B.12 ((Bulinski and Dimitrov, 2019b, Eq. (3.32))). *For any $\mathbf{x} \in \mathbb{R}^d$, $\delta > 0$, $m \geq (1 + 1/\delta)(k-1)$, and $u \geq 0$, we have*

$$1 - F_{km}(u|\mathbf{x}) \leq (1+\delta)(1 - P_m(u|\mathbf{x})). \tag{4.52}$$

Proof. Consider

$$1 - F_{km}(u|\mathbf{x}) = \sum_{j=0}^{k-1} \binom{m}{j} P_m^j(u|\mathbf{x})(1 - P_m(u|\mathbf{x}))^{m-j}$$
$$= (1 - P_m(u|\mathbf{x})) \sum_{j=0}^{k-1} \frac{m}{m-j} \binom{m-1}{j} P_m^j(u|\mathbf{x})(1 - P_m(u|\mathbf{x}))^{m-j-1}.$$

For any fixed $\delta > 0$, if $m \geq (1 + \delta^{-1})(k-1)$, then

$$\frac{m}{m-j} \leq \frac{m}{m-k+1} \leq 1 + \delta$$

for $j = 0, \ldots, k-1$. Therefore, we have

$$1 - F_{km}(u|\mathbf{x}) \leq (1 + \delta)(1 - P_m(u|\mathbf{x})). \qquad \square$$

Lemma 4.B.13. *If $p(\mathbf{z}) \leq C_p$ for $\mathbf{z} \in \overline{\mathbf{B}}(\mathbf{x}, r)$, we have*

$$\rho_{U_{km}(\mathbf{x})}(u) \leq \frac{C_p^k u^{k-1}}{\Gamma(k)}.$$

We first prove the following lemma. Let us denote the sphere centered at $\mathbf{x} \in \mathbb{R}^d$ of radius $r > 0$ by $\mathbb{S}(\mathbf{x}, r) := \{\mathbf{y} \colon \rho(\mathbf{x}, \mathbf{y}) = r\}$. Note that the the Hausdorff measure $H^{d-1}(\mathbb{S}(\mathbf{x}, r))$ of the sphere is $d\upsilon_d r^{d-1}$.

Lemma 4.B.14. *If $p(\mathbf{z}) \leq C_p$ for $\mathbf{z} \in \mathbb{S}(\mathbf{x}, r)$, we have*

$$\frac{\mathrm{d}\, \boldsymbol{P}(\mathbf{B}(\mathbf{x}, r))}{\mathrm{d}\, \lambda(\mathbf{B}(\mathbf{x}, r))} \leq C_p.$$

Proof of Lemma 4.B.14. It is easy to see that $p(\mathbf{x}) \leq M_r p(\mathbf{x})$ for any $r > 0$ by contradiction.

From the coarea formula (Evans and Gariepy, 2015), we have

$$\begin{aligned}\frac{\mathrm{d}\,\mathsf{P}(\mathbf{B}(\mathbf{x},r))}{\mathrm{d}r} &= \frac{\mathrm{d}}{\mathrm{d}r}\int_{\mathbf{B}(\mathbf{x},r)} p(\mathbf{y})\,\mathrm{d}\mathbf{y} \\ &= \int_{\mathbb{S}(\mathbf{x},r)} p(\mathbf{y}) H^{d-1}(\mathrm{d}\mathbf{y}) \\ &\leq C_p(d\upsilon_d r^{d-1})\end{aligned}$$

since $p(\mathbf{x}) \leq C_p$ for $\mathbf{x} \in \mathbb{S}(\mathbf{x}, r)$. Therefore, we have

$$\frac{\mathrm{d}\,\mathsf{P}(\mathbf{B}(\mathbf{x},r))}{\mathrm{d}\,\lambda(\mathbf{B}(\mathbf{x},r))} = \frac{\frac{\mathrm{d}}{\mathrm{d}r}\mathsf{P}(\mathbf{B}(\mathbf{x},r))}{\frac{\mathrm{d}}{\mathrm{d}r}\lambda(\mathbf{B}(\mathbf{x},r))} \leq C_p. \qquad \square$$

Proof of Lemma 4.B.13. Now, from Lemma 4.B.3 and Lemma 4.B.14, if $p(\mathbf{y}) \leq C_p$ for $\mathbf{y} \in \mathbf{B}(\mathbf{x}, r)$, then

$$\begin{aligned}\rho_{r_{km}(\mathbf{x})}(r) &= \rho_{X_{k,m-k+1}}(\mathsf{P}(\mathbf{B}(\mathbf{x},r)))\frac{\mathrm{d}\,\mathsf{P}(\mathbf{B}(\mathbf{x},r))}{\mathrm{d}r} \\ &\leq \frac{m^k}{\Gamma(k)}\,\mathsf{P}^{k-1}(\mathbf{B}(\mathbf{x},r))\frac{\mathrm{d}\,\mathsf{P}(\mathbf{B}(\mathbf{x},r))}{\mathrm{d}r} \\ &\leq \frac{(C_p m)^k}{\Gamma(k)}\frac{d}{r}\,\lambda^k(\mathbf{B}(\mathbf{x},r)).\end{aligned}$$

We then bound the density of $U_{km}(\mathbf{x})$ as

$$\begin{aligned}\rho_{U_{km}(\mathbf{x})}(u) &= \rho_{r_{km}(\mathbf{x})}\big(\varrho\big(\frac{u}{m}\big)\big)\frac{\mathrm{d}\,\varrho(\frac{u}{m})}{\mathrm{d}u} \\ &\leq \frac{(C_p m)^k}{\Gamma(k)}\frac{d}{\varrho(\frac{u}{m})}\big(\frac{u}{m}\big)^k\frac{\varrho(\frac{u}{m})}{du} = \frac{C_p^k}{\Gamma(k)}u^{k-1},\end{aligned}$$

which concludes the proof. $\square$

4.B.4 Bounds on Expected Values of k-NN Statistics

Let $\tilde{f}_{km}(u|\mathbf{x}) := \rho_{\tilde{U}_{km}(\mathbf{x})}(v)$ denote the density of the normalized volume $\tilde{U}_{km}(\mathbf{x}) = \lambda(\mathbf{B}(\mathbf{x}, r_k(\mathbf{x}|\tilde{\mathbf{X}}_{1:m})))$, where $\tilde{\mathbf{X}}_{1:m}$ is drawn i.i.d. from density $\tilde{p}$. Later, the density $\tilde{p}$ may

be identified as the density p for $\mathbf{X}_{1:m}$ or the density q for $\mathbf{Y}_{1:n}$. Pick any numbers $0 \le \tau_m \le 1 \le \nu_m \le \kappa_m < \infty$. Suppose that we are given a nondecreasing function $\xi \in \Xi$. For $(a, b) \in \mathbb{R}^2$ and $k \in \mathbb{N}$, we define, for each $\mathbf{x} \in \mathbb{R}^d$

$$A_{km}(\mathbf{x}; \tilde{p}; \xi) := \int_0^{\tau_m} \xi(u^a) \tilde{f}_{km}(u|\mathbf{x}) \, \mathrm{d}u, \tag{4.53}$$

$$B^{(1)}_{km}(\mathbf{x}; \tilde{p}; \xi) := \int_1^{\nu_m} \xi(u^b) \tilde{f}_{km}(u|\mathbf{x}) \, \mathrm{d}u, \tag{4.54}$$

$$B^{(2)}_{km}(\mathbf{x}; \tilde{p}; \xi) := \int_{\nu_m}^{\kappa_m} \xi(u^b) \tilde{f}_{km}(u|\mathbf{x}) \, \mathrm{d}u, \tag{4.55}$$

and

$$B^{(3)}_{km}(\mathbf{x}; \tilde{p}; \xi) := \int_{\kappa_m}^{\infty} \xi(u^b) \tilde{f}_{km}(u|\mathbf{x}) \, \mathrm{d}u. \tag{4.56}$$

Lemma 4.B.15. *For $r = \varrho(\frac{\tau_m}{m})$, we have*

$$A_{km}(\mathbf{x}; \tilde{p}; \xi) \le \frac{(M_r \tilde{p}(\mathbf{x}))^k}{k!} \Big(\tau_m^k \xi(\tau_m^a) - 1_{(-\infty,0)}(a) \int_0^{\tau_m} u^k \, \mathrm{d}\xi(u^a) \Big).$$

In particular, if $\tau_m = 1$ and $-\int_0^1 u^k \, \mathrm{d}\xi(u^a) < \infty$, we have for $r = \varrho(\frac{1}{m})$,

$$A_{km}(\mathbf{x}; \tilde{p}; \xi) \lesssim \frac{(M_r \tilde{p}(\mathbf{x}))^k}{k!}.$$

Proof. Integrating by parts and applying Lemma 4.B.10, we have

$$\begin{aligned} A_{km}(\mathbf{x}; \tilde{p}; \xi) &= \int_0^{\tau_m} \xi(u^a) \, \mathrm{d}\overline{F}_{km}(u|\mathbf{x}) \\ &\le \xi(\tau_m^a) \overline{F}_{km}(\tau_m|\mathbf{x}) - \int_0^{\tau_m} \overline{F}_{km}(u|\mathbf{x}) \, \mathrm{d}\xi(u^a) \\ &\le \frac{(M_{\varrho(\frac{\tau_m}{m})} \tilde{p}(\mathbf{x}))^k}{k!} \tau_m^k \xi(\tau_m^a) - \int_0^{\tau_m} \overline{F}_{km}(u|\mathbf{x}) \, \mathrm{d}\xi(u^a). \end{aligned}$$

If $a < 0$, we again apply Lemma 4.B.10 again to the remaining integral and obtain

$$A_{km}(\mathbf{x}; \tilde{p}; \xi) \leq \frac{(M_{\varrho(\frac{\tau_m}{m})}\tilde{p}(\mathbf{x}))^k}{k!}\Big(\tau_m^k \xi(\tau_m^a) - \int_0^{\tau_m} u^k \,\mathrm{d}\xi(u^a)\Big). \qquad \square$$

Lemma 4.B.16. *If $b \leq 0$, we have*

$$B_{km}^{(1)}(\mathbf{x}; \tilde{p}; \xi) \lesssim 1.$$

If $b > 0$ and $\int_0^\infty e^{-t}\xi(t^b)\,\mathrm{d}t < \infty$, then for any $0 < D < 1$ and $r = \varrho(\frac{\nu_m}{m})$, we have

$$B_{km}^{(1)}(\mathbf{x}; \tilde{p}; \xi) \lesssim_{k,D} \xi(\nu_m^b) e^{-D\nu_m(m_r p(\mathbf{x}))} + \xi((Dm_r\tilde{p}(\mathbf{x}))^{-b}).$$

Proof. By definition, if $b \leq 0$, we have

$$\begin{aligned} B_{km}^{(1)}(\mathbf{x}; \tilde{p}; \xi) &= \int_1^{\nu_m} \xi(u^b) \tilde{f}_{km}(u|\mathbf{x})\,\mathrm{d}u \\ &\leq \xi(1) \int_1^{\nu_m} \tilde{f}_{km}(u|\mathbf{x})\,\mathrm{d}u \\ &\leq \xi(1). \end{aligned}$$

We now assume $b > 0$. Integrating by parts, we have

$$\begin{aligned} B_{km}^{(1)}(\mathbf{x}; \tilde{p}; \xi) &= -\int_1^{\nu_m} \xi(u^b)\,\mathrm{d}(1 - \overline{F}_{km}(u|\mathbf{x})) \\ &\leq \xi(1)(1 - \overline{F}_{km}(u|\mathbf{x})) + \int_1^{\nu_m} (1 - \overline{F}_{km}(u|\mathbf{x}))\,\mathrm{d}\xi(u^b). \end{aligned}$$

Applying Lemma 4.B.11 yields, for any $0 < D < 1$, that

$$\begin{aligned} B_{km}^{(1)}(\mathbf{x}; \tilde{p}; \xi) &\leq \xi(1) + (1-D)^{-k+1} \int_1^{\nu_m} e^{-Dm\tilde{\mathbf{P}}_m(u|\mathbf{x})}\,\mathrm{d}\xi(u^b) \\ &\leq \xi(1) + (1-D)^{-k+1} \int_1^{\nu_m} e^{-Du(m_r\tilde{p}(\mathbf{x}))}\,\mathrm{d}\xi(u^b) \end{aligned} \tag{4.57}$$

for $r = \varrho(\frac{\nu_m}{m})$. Integrating by parts again, we thus obtain

$$\begin{aligned}
&\int_1^{\nu_m} e^{-Du(m_r\tilde{p}(\mathbf{x}))}\,\mathrm{d}\xi(u^b) \\
&\le \xi(\nu_m^b)e^{-D\nu_m(m_r\tilde{p}(\mathbf{x}))} + D(m_r\tilde{p}(\mathbf{x}))\int_1^{\nu_m} e^{-Du(m_r\tilde{p}(\mathbf{x}))}\xi(u^b)\,\mathrm{d}u \\
&\le \xi(\nu_m^b)e^{-D\nu_m(m_r\tilde{p}(\mathbf{x}))} + \int_{D(m_r\tilde{p}(\mathbf{x}))}^{D\nu_m(m_r\tilde{p}(\mathbf{x}))} e^{-t}\xi(t^b(Dm_r\tilde{p}(\mathbf{x}))^{-b})\,\mathrm{d}t. \qquad (4.58)
\end{aligned}$$

Here, using the property that $\xi(xy) \le \xi(x)\xi(y)$ for any $x, y > t_0$ for some $t_0 \ge 0$, it is easy to show that

$$\begin{aligned}
&\int_{D(m_r\tilde{p}(\mathbf{x}))}^{D\nu_m(m_r\tilde{p}(\mathbf{x}))} e^{-t}\xi(t^b(Dm_r\tilde{p}(\mathbf{x}))^{-b})\,\mathrm{d}t \\
&\le \Big(t_0\xi(t_0^b) + \int_0^\infty e^{-t}\xi(t^b)\,\mathrm{d}t\Big)\xi((Dm_r\tilde{p}(\mathbf{x}))^{-b}) + \xi(t_0)\int_0^\infty e^{-t}\xi(t^b)\,\mathrm{d}\mathbf{x} \\
&\lesssim 1 + \xi((Dm_r\tilde{p}(\mathbf{x}))^{-b}). \qquad (4.59)
\end{aligned}$$

Putting (4.57), (4.58), and (4.59) together, we obtain the desired bound. □

Lemma 4.B.17. *For any $0 < D < 1$ and $r = \varrho(\frac{\nu_m}{m})$, we have*

$$B_{km}^{(2)}(\mathbf{x};\tilde{p};\xi) \lesssim_{k,D} \xi(\nu_m^b \vee \kappa_m^b)e^{-D\nu_m(m_r\tilde{p}(\mathbf{x}))}.$$

Proof. Integrating by parts, we have

$$\begin{aligned}
B_{km}^{(2)}(\mathbf{x};\tilde{p};\xi) &= -\int_{\nu_m}^{\kappa_m} \xi(u^b)\,\mathrm{d}(1-\overline{F}_{km}(u|\mathbf{x})) \\
&\le \xi(\nu_m^b)(1-\overline{F}_{km}(\nu_m|\mathbf{x})) + \int_{\nu_m}^{\kappa_m}(1-F_{km}(u|\mathbf{x}))\,\mathrm{d}\xi(u^b) \\
&\le 2\xi(\nu_m^b \vee \kappa_m^b)(1-\overline{F}_{km}(\nu_m|\mathbf{x})). \qquad (4.60)
\end{aligned}$$

Applying Lemma 4.B.11, we have that for any $0 < D < 1$ and $r = \varrho(\frac{\nu_m}{m})$

$$\begin{aligned} B_{km}^{(2)}(\mathbf{x};\tilde{p};\xi) &\leq 2(1-D)^{-k+1}\xi(\nu_m^b \vee \kappa_m^b)e^{-Dm\tilde{\mathsf{P}}_m(\nu_m|\mathbf{x})} \\ &\leq 2(1-D)^{-k+1}\xi((\nu_m^b \vee \kappa_m^b)e^{-D\nu_m(m_r p(\mathbf{x}))}. \end{aligned}$$ □

Lemma 4.B.18. *For any $\delta > 0$ and m sufficiently large, we have*

$$B_{km}^{(3)}(\mathbf{x};\tilde{p};\xi) \lesssim_\delta \xi(m^b) \int p(\mathbf{y})\xi(\upsilon^b(\rho(\mathbf{x},\mathbf{y}))) \, 1_{\{\rho(\mathbf{x},\mathbf{y})>\varrho(\frac{\kappa_m}{m})\}} \, \mathrm{d}\mathbf{y}.$$

Proof. We recall the following bound (4.52) on the complementary cdf $1 - \overline{F}_{km}(u|\mathbf{x})$ from Lemma 4.B.10: for any $\delta > 0$ and $m \geq (1 + 1/\delta)(k-1)$, we have

$$\begin{aligned} 1 - \overline{F}_{km}(u|\mathbf{x}) &\leq (1+\delta)(1 - \tilde{\mathsf{P}}_m(u|\mathbf{x})) \\ &= (1+\delta) \int \tilde{p}(\mathbf{y}) \, 1_{\{\rho(\mathbf{x},\mathbf{y})>\varrho(\frac{u}{m})\}} \, \mathrm{d}\mathbf{y}. \end{aligned}$$

Integrating by parts, we first obtain

$$\begin{aligned} B_{km}^{(3)}(\mathbf{x};\tilde{p};\xi) &= -\int_{\kappa_m}^{\infty} \xi(u^b) \, \mathrm{d}(1 - \overline{F}_{km}(u|\mathbf{x})) \\ &\leq \xi(\kappa_m^b)(1 - \overline{F}_{km}(\kappa_m|\mathbf{x})) + \int_{\kappa_m}^{\infty} (1 - \overline{F}_m(u|\mathbf{x})) \, \mathrm{d}\xi(u^b) \\ &\leq \xi(\kappa_m^b)(1 - \overline{F}_{km}(\kappa_m|\mathbf{x})) + (1+\delta) \int_{\kappa_m}^{\infty} (1 - \tilde{\mathsf{P}}_m(u|\mathbf{x})) \, \mathrm{d}\xi(u^b). \end{aligned} \tag{4.61}$$

Integrating the second term by parts leads to

$$\int_{\kappa_m}^{\infty} (1 - \tilde{\mathsf{P}}_m(u|\mathbf{x})) \, \mathrm{d}\xi(u^b) \leq \lim_{u\to\infty} \xi(u^b)(1 - \tilde{\mathsf{P}}_m(u|\mathbf{x})) + \int_{\kappa_m}^{\infty} \xi(u^b) \, \mathrm{d}\tilde{\mathsf{P}}_m(u|\mathbf{x}). \tag{4.62}$$

For the first term in (4.62), since for m sufficiently large with $m^b > t_0$ and $(\kappa_m/m)^b > t_0$,

we have $\xi(u^b) \le \xi(m^b)\xi((u/m)^b)$ for $u \ge \kappa_m$, it follows that

$$\begin{aligned}
\xi(u^b)(1-\tilde{\mathsf{P}}_m(u|\mathbf{x})) &= \xi(u^b)\int \tilde{p}(\mathbf{y})\,1_{\{\rho(\mathbf{x},\mathbf{y})>\varrho(\frac{u}{m})\}}\,\mathrm{d}\mathbf{y} \\
&\le \xi(m^b)\int \tilde{p}(\mathbf{y})\xi\big(\big(\frac{u}{m}\big)^b\big)\,1_{\{\rho(\mathbf{x},\mathbf{y})>\varrho(\frac{u}{m})\}}\,\mathrm{d}\mathbf{y} \\
&\le \xi(m^b)\int \tilde{p}(\mathbf{y})\xi(\upsilon^b(\rho(\mathbf{x},\mathbf{y})))\,1_{\{\rho(\mathbf{x},\mathbf{y})>\varrho(\frac{u}{m})\}}\,\mathrm{d}\mathbf{y} \\
&\le \xi(m^b)\int \tilde{p}(\mathbf{y})\xi(\upsilon^b(\rho(\mathbf{x},\mathbf{y})))\,1_{\{\rho(\mathbf{x},\mathbf{y})>\varrho(\frac{\kappa_m}{m})\}}\,\mathrm{d}\mathbf{y}.
\end{aligned}$$

Therefore,

$$\lim_{u\to\infty}\xi(u^b)(1-\tilde{\mathsf{P}}_m(u|\mathbf{x})) \le \xi(m^b)\int \tilde{p}(\mathbf{y})\xi(\upsilon^b(\rho(\mathbf{x},\mathbf{y})))\,1_{\{\rho(\mathbf{x},\mathbf{y})>\varrho(\frac{\kappa_m}{m})\}}\,\mathrm{d}\mathbf{y}. \tag{4.63}$$

The second term in (4.62) can be bounded similarly as

$$\begin{aligned}
\int_{\kappa_m}^{\infty}\xi(u^b)\,\mathrm{d}\tilde{\mathsf{P}}_m(u|\mathbf{x}) &= \int \tilde{p}(\mathbf{y})\xi((m\upsilon(\rho(\mathbf{x},\mathbf{y})))^b)\,1_{\{\rho(\mathbf{x},\mathbf{y})>\varrho(\frac{\kappa_m}{m})\}}\,\mathrm{d}\mathbf{y} \\
&\le \xi(m^b)\int \tilde{p}(\mathbf{y})\xi(\upsilon^b(\rho(\mathbf{x},\mathbf{y})))\,1_{\{\rho(\mathbf{x},\mathbf{y})>\varrho(\frac{\kappa_m}{m})\}}\,\mathrm{d}\mathbf{y}.
\end{aligned} \tag{4.64}$$

Plugging (4.62), (4.63), and (4.64) into (4.61) establishes the desired bound. □

The following is the key lemma in establishing vanishing bias and vanishing variance for single- and double-density cases.

Lemma 4.B.19. *Assume that* $-\int_0^1 u^k\,\mathrm{d}\xi(u^{a\wedge 0}) < \infty$ *and* $\int_0^\infty e^{-t}\xi(t^{b\vee 0})\,\mathrm{d}t < \infty$. *If the densities* p *and* $\tilde{p}$ *satisfy* $\tilde{\mathsf{P}} \ll \mathsf{P}$, **($\boldsymbol{U}_{p\tilde{p}}$;** k, a**)**, *and* **($\boldsymbol{L}_{p\tilde{p}}$;** ξ, b**)**, *we have*

$$\limsup_{m\to\infty}\int p(\mathbf{x})\int_0^\infty \xi(\psi_{a,b}(u))\,\mathrm{d}\overline{F}_{km}(u|\mathbf{x})\,\mathrm{d}\mathbf{x} < \infty.$$

Proof. Let $\tau_m = 1$ and $\kappa_m = e^{o(m)}$. Then, there exists ν_m such that $\nu_m \to \infty$, $\nu_m/m \to 0$,

and for any $c > 0$, $e^{-c\nu_m}\xi(\kappa_m^b) \to 0$, as $m \to \infty$. Consider

$$\begin{aligned}\int_0^\infty \xi(\psi_{a,b}(u))\,\mathrm{d}F_{km}(u|\mathbf{x}) &= \int_0^1 \xi(u^a)\,\mathrm{d}\overline{F}_{km}(u|\mathbf{x}) + \int_1^\infty \xi(u^b)\,\mathrm{d}\overline{F}_{km}(u|\mathbf{x}) \\ &= A_{km}(\mathbf{x};\tilde{p};\xi) + B_{km}(\mathbf{x};\tilde{p};\xi),\end{aligned}$$

where

$$B_{km}(\mathbf{x};\tilde{p};\xi) := B_{km}^{(1)}(\mathbf{x};\tilde{p};\xi) + B_{km}^{(2)}(\mathbf{x};\tilde{p};\xi) + B_{km}^{(3)}(\mathbf{x};\tilde{p};\xi).$$

Recall the definitions of $A_{km}(\mathbf{x};\tilde{p};\xi)$, $B_{km}^{(1)}(\mathbf{x};\tilde{p};\xi)$, $B_{km}^{(2)}(\mathbf{x};\tilde{p};\xi)$, and $B_{km}^{(3)}(\mathbf{x};\tilde{p};\xi)$ in (4.53), (4.54), (4.55), and (4.56), respectively. Letting

$$A_{km}(p,\tilde{p};\xi) := \int p(\mathbf{x})A_{km}(\mathbf{x};\tilde{p};\xi)\,\mathrm{d}\mathbf{x}$$

and

$$B_{km}(p,\tilde{p};\xi) := \int p(\mathbf{x})B_{km}(\mathbf{x};\tilde{p};\xi)\,\mathrm{d}\mathbf{x},$$

we show separately that $\limsup_{m\to\infty} A_{km}(p,\tilde{p};\xi) < \infty$ and $\limsup_{m\to\infty} B_{km}(p,\tilde{p};\xi) < \infty$.

Step 1. Bounding $A_{km}(p,\tilde{p};\xi)$. If $a \geq 0$, we trivially have $A_{km}(p,\tilde{p};\xi) \leq \xi(1)$. If $a < 0$, by Lemma 4.B.15, we have

$$\begin{aligned}A_{km}(p,\tilde{p};\xi) &\leq \frac{W(p,\tilde{p};k,\varrho(\frac{1}{m}))}{k!}\Big(\xi(1) - \int_0^1 u^k\,\mathrm{d}\xi(u^a)\Big) \\ &\lesssim_k W\big(p,\tilde{p};k,\varrho(\frac{1}{m})\big).\end{aligned}$$

Hence, since there exists $r' > 0$ such that $W(p,\tilde{p};k,r') < \infty$ by the the condition **($\mathbf{U}_{p\tilde{p}}$; k,a)** and $W(p,\tilde{p};k,r)$ is nonincreasing as $r \to 0$, we conclude that $A_{km}(p,\tilde{p};\xi) < \infty$ for m sufficiently large such that $\varrho(1/m) < r'$.

Step 2. Bounding $B_{km}(p,\tilde{p};\xi)$. If $b \leq 0$, then we trivially have $B_{km}(p,\tilde{p};\xi) \leq \xi(1)$. If $b > 0$, by applying Lemmas 4.B.16, 4.B.17, and 4.B.18, we have that for any $0 < D < 1$

and m sufficiently large

$$\begin{aligned} B_{km}(p, \tilde{p}; \xi) \lesssim \xi(\kappa_m^b) \int e^{-D\nu_m(m_{r_1}\tilde{p}(\mathbf{x}))} p(\mathbf{x}) \, \mathrm{d}\mathbf{x} \\ + w(p, \tilde{p}; \xi, b, r_1) \\ + \xi(m^b) R(p, \tilde{p}; \xi, b, r_2), \end{aligned}$$

where $r_1 = \varrho(\nu_m/m)$ and $r_2 = \varrho(\kappa_m/m)$.

- For the first term, since, by Lemma 4.B.1, $\mathsf{P} \ll \tilde{\mathsf{P}}$ implies that $\mathsf{P}(\{\mathbf{x} \colon m_{r_1}\tilde{p}(\mathbf{x}) > 0\}) = 1$, we have $\xi(\kappa_m^b) e^{-\nu_m(m_{r_1}\tilde{p}(\mathbf{x}))} \to 0$ as $m \to \infty$ for P-a.e. $\mathbf{x}$ by definition of ν_m and κ_m. Therefore, by the dominated convergence theorem,

$$\lim_{m\to\infty} \int \xi(\kappa_m^b) e^{-\nu_m(m_{r_1}\tilde{p}(\mathbf{x}))} p(\mathbf{x}) \, \mathrm{d}\mathbf{x} = 0.$$

- Since there exists $r'' > 0$ such that $w(p, \tilde{p}; \xi, b, r'') < \infty$ by the condition **($\mathbf{L}_{p\tilde{p}}$; ξ, b)** and $w(p, \tilde{p}; \xi, b, r)$ is nonincreasing as $r \to 0$, the second term is bounded for m sufficiently large such that $\varrho(\frac{\kappa_m}{m}) < r''$.

- The limit superior of the last term $\xi(m^b) R(p, \tilde{p}; \xi, b, r_2)$ as $m \to \infty$ is bounded by the condition **($\mathbf{L}_{p\tilde{p}}$; ξ, b)**.

Overall, we conclude that

$$\limsup_{m\to\infty} B_{km}(p, \tilde{p}; \xi) < \infty. \qquad \square$$

Following the proof of Lemma 4.B.19 with the stronger assumptions establishes the following bound.

Lemma 4.B.20. *Assume that* $-\int_0^1 u^k \, \mathrm{d}\xi(u^{a\wedge 0}) < \infty$ *and* $\int_0^\infty e^{-t} \xi(t^{b\vee 0}) \, \mathrm{d}t < \infty$. *If* $\tilde{p}$ *satisfies*

the conditions **(U_p)**, **($L1_p$)**, **($L2_p$)**, *and* **($L3_p$)**, *we have*

$$\int_0^\infty \xi(\psi_{a,b}(u))\,\mathrm{d}\overline{F}_{km}(u|\mathbf{x}) \lesssim 1$$

for P*-a.e.* $\mathbf{x}$.

Continuing from (4.60) and applying (4.51) in Lemma 4.B.11 yield the following bound, which is required for establishing performance guarantees with adaptive choices of k and l.

Lemma 4.B.21. *For* $r = \varrho(\nu_m/m)$, *we have*

$$B^{(2)}_{km}(\mathbf{x};\tilde{p};\xi) \le 2\xi(\nu_m^b \vee \kappa_m^b)\Big(\frac{e\nu_m M_r \tilde{p}(\mathbf{x})}{k}\Big)^k e^{-\nu_m m_r \tilde{p}(\mathbf{x})}.$$

Lemma 4.B.22. *If* $k + a > 0$, *for* $\mathbf{x} \in \mathrm{supp}(p)$, *we have*

$$\int_0^\infty \psi_{a,b}(u)\rho_{U_{k\infty}(\mathbf{x})}(u)\,\mathrm{d}u \le \frac{p^k(\mathbf{x})}{(k+a)\Gamma(k)} + \frac{\Gamma((k+b)\vee 1)}{\Gamma(k)}(p(\mathbf{x}))^{(k-1)\wedge(-b)}.$$

In particular, if $c_p \le p(\mathbf{x}) \le C_p$, *then*

$$\int_0^\infty \psi_{a,b}(u)\rho_{U_{k\infty}(\mathbf{x})}(u)\,\mathrm{d}u \lesssim 1.$$

Proof. First, consider

$$\begin{aligned}\int_0^1 u^a \rho_{U_{k\infty}(\mathbf{x})}(u)\,\mathrm{d}u &= \frac{p^k(\mathbf{x})}{\Gamma(k)}\int_0^1 u^{k+a-1}e^{-up(\mathbf{x})}\,\mathrm{d}u \\ &= \frac{(p(\mathbf{x}))^{-a}}{\Gamma(k)}\int_0^{p(\mathbf{x})} t^{k+a-1}e^{-t}\,\mathrm{d}t \\ &\le \frac{p^k(\mathbf{x})}{(k+a)\Gamma(k)},\end{aligned}$$

where the last inequality follows from the bound on the lower incomplete gamma in

Lemma 4.B.2. Similarly, we consider

$$\int_0^1 u^b \rho_{U_{k\infty}(\mathbf{x})}(u)\,\mathrm{d}u = \frac{(p(\mathbf{x}))^{-b}}{\Gamma(k)} \int_0^{p(\mathbf{x})} t^{k+b-1} e^{-t}\,\mathrm{d}t.$$

On the one hand, if $k+b>1$, by bounding the integral by $\Gamma(k+b)$, we have

$$\int_0^1 u^b \rho_{U_{k\infty}(\mathbf{x})}(u)\,\mathrm{d}u \le \frac{\Gamma(k+b)}{\Gamma(k)} (p(\mathbf{x}))^{-b}.$$

On the other hand, if $k+b\le 1$, we have

$$\begin{aligned}\int_0^1 u^b \rho_{U_{k\infty}(\mathbf{x})}(u)\,\mathrm{d}u &\le \frac{(p(\mathbf{x}))^{k-1}}{\Gamma(k)} \int_{p(\mathbf{x})}^{\infty} e^{-t}\,\mathrm{d}t \\ &\le \frac{(p(\mathbf{x}))^{k-1}}{\Gamma(k)}.\end{aligned}$$

Therefore, we obtain

$$\int_0^1 u^b \rho_{U_{k\infty}(\mathbf{x})}(u)\,\mathrm{d}u \le \frac{\Gamma((k+b)\vee 1)}{\Gamma(k)} (p(\mathbf{x}))^{(k-1)\wedge(-b)},$$

which completes the proof. □

4.B.5 Generic Bias Bounds

Lemma 4.B.23 (Generic inner bias bound)**.** *Suppose that the density p satisfies the conditions* ***(U_p)***, ***(S_p)***, *and* ***(B_p)***, *and let $k = o(\sqrt{m})$ as $m\to\infty$.*

1. *We have*

$$I_{in,1} = O\Big(\frac{\tau_m^{(a+\frac{\sigma_p}{d}+1)\wedge 0}}{m^{\frac{\sigma_p}{d}}} + \frac{k^{-k}}{m} + \Big(\frac{1}{m}\Big)^{\frac{1}{d}} \tau_m^{(a+1)\wedge 0}\Big). \tag{4.65}$$

2. *If $\nu_m = o(\sqrt{m})$ as $m \to \infty$, we have*

$$I_{in,2} = O\Big(\frac{\nu_m^{(b+\frac{\sigma_p}{d}+2)\vee 0}}{m^{\frac{\sigma_p}{d}}} + \frac{k^{-k}\nu_m^{(b+k+2)\vee 0}}{m} + \Big(\frac{\nu_m}{m}\Big)^{\frac{1}{d}} \nu_m^{(b+2)\vee 0} \Big). \tag{4.66}$$

Proof. We establish each bound separately.

Bounding the lower inner bias $I_{\text{in},1}$. For each $r > 0$, define a set

$$S_p(r) := \{\mathbf{x} \in \text{supp}(p) \colon p \text{ is } \sigma_p\text{-Hölder continuousover } \mathbf{B}(\mathbf{x}, r)\}.$$

By the smoothness assumption **($\mathbf{S}_p$)**, we can bound the inner bias incurred at the "smooth region", i.e.,

$$I_{\text{in},1,\text{smooth}} = \int_{S_p(\varrho(\frac{1}{m}))} I_{\text{in},1}(\mathbf{x}) p(\mathbf{x}) \, \mathrm{d}\mathbf{x},$$

by applying Lemma 4.B.4. Since $p(\mathbf{x}) \le C_p < \infty$ for P-a.e. $\mathbf{x}$, this lemma holds for m sufficiently large uniformly over P-a.e. $\mathbf{x}$. Applying Lemma 4.B.4 for $\mathbf{x} \in S_p(\varrho(\frac{1}{m}))$, we have

$$I_{\text{in},1}(\mathbf{x}) \lesssim_{\sigma_p, L, C_p, C_0, d} \int_{\tau_m}^1 u^a \Big\{ (1+u)\Big(\frac{u}{m}\Big)^{\frac{\sigma_p}{d}} + k^{-k} \frac{(k^2+u^2)u^{k-1}e^{-up(\mathbf{x})}}{m} \Big\} \, \mathrm{d}u. \tag{4.67}$$

It is easy to see that the first term is bounded by $O\big(\tau_m^{(a+\frac{\sigma_p}{d}+1)\wedge 0} m^{-\frac{\sigma_p}{d}}\big)$.[4] To bound the second term, we use the upper bound on the lower incomplete gamma function (Lemma 4.B.2). Since we always assume that $k + a > 0$, we have

$$\begin{aligned} &\int_{\tau_m}^1 \frac{k^{-k}}{m} (k^2+u^2) u^{k+a-1} e^{-up(\mathbf{x})} \, \mathrm{d}u \\ &\le \frac{k^{-k}}{m} \big\{ k^2 p(\mathbf{x})^{-(k+a)} \gamma(k+a, p(\mathbf{x})) + p(\mathbf{x})^{-(k+a+2)} \gamma(k+a+2, p(\mathbf{x})) \big\} = O\Big(\frac{k^{-k}}{m}\Big). \end{aligned}$$

[4]Here $a + \frac{\sigma_p}{d} + 1 \neq 0$ is implicitly assumed. If $a + \frac{\sigma_p}{d} + 1 = 0$, then the first term behaves as $O((\ln \tau_m) m^{-\frac{\sigma_p}{d}})$

Hence, we conclude that

$$I_{\text{in},1,\text{smooth}} = O(\tau_m^{(a+\frac{\sigma_p}{d}+1)\wedge 0} m^{-\frac{\sigma_p}{d}} + k^{-k}m^{-1}). \tag{4.68}$$

To control the inner bias incurred at $\mathbf{x} \in \text{supp}(p)\backslash S_p(\varrho(m^{-1}))$, i.e.,

$$I_{\text{in},1,\text{nonsmooth}} = \int_{\text{supp}(p)\backslash S_p(\varrho(\frac{1}{m}))} I_{\text{in},1}(\mathbf{x})p(\mathbf{x})\,\mathrm{d}\mathbf{x},$$

we first note that the bound (4.67) on $I_{\text{in},1}(\mathbf{x})$ holds with $\sigma_p = 0$ from the upper boundedness assumption **($\mathbf{U}_p$)**, which implies that

$$I_{\text{in},1,\text{nonsmooth}} = O(\lambda(\text{supp}(p)\backslash S_p(\varrho(m^{-1})))(\tau_m^{(a+1)\wedge 0} + k^{-k}m^{-1})). \tag{4.69}$$

We now only need to bound the Lebesgue measure of the set where $\text{supp}(p)\backslash S_p(\varrho(m^{-1}))$. Observe that for any $r > 0$

$$\text{supp}(p)\backslash S_p(r) \subseteq \{\mathbf{x} \in \mathbb{R}^d \colon \mathbf{B}(\mathbf{x}, r) \cap \partial(\text{supp}(p)) \neq \emptyset\},$$

where ∂A denotes the boundary of a set A. Using the following lemma with the condition **($\mathbf{B}_p$)** on the finiteness of the Hausdorff measure of the boundary of the support, we can bound the Lebesgue measure of $\mathbb{R}^d\backslash S_p(\varrho(m^{-1}))$ by $O(\varrho(1/m)) = O(m^{-\frac{1}{d}})$.

Lemma 4.B.24 ((Gao et al., 2018, Section A)). *For $S \subset \mathbb{R}^d$, suppose that $0 < H^{d-1}(S) < \infty$. Let $T(r) := \{\mathbf{x} \in \mathbb{R}^d \colon \mathbf{B}(\mathbf{x}, r) \cap S \neq \varnothing\}$ for $r > 0$. Then $\lambda(T(r)) = 2rH^{d-1}(S) + o(r)$ for r sufficiently small.*

Combining (4.68) and (4.69) establishes the desired bound (4.65).

Bounding the upper inner bias $I_{\text{in},2}$. The proof follows a similar line of argument

as that of (4.65). We first apply Lemma 4.B.4 for $\mathbf{x} \in S_p(\varrho(\frac{\nu_m}{m}))$ and obtain

$$I_{\text{in},2}(\mathbf{x}) \lesssim_{\sigma_p, L, C_p, C_0, d} \int_1^{\nu_m} u^b \Big\{ (1+u) \Big(\frac{u}{m} \Big)^{\frac{\sigma_p}{d}} + k^{-k} \frac{(k^2+u^2) u^{k-1} e^{-up(\mathbf{x})}}{m} \Big\} \, \mathrm{d}u.$$

The first term is bounded by $O(m^{-\frac{\sigma_p}{d}} \nu_m^{(b+\frac{\sigma_p}{d}+2) \vee 0})$. The second term is again bounded by the upper bound on the lower incomplete gamma function. If $b + k > 0$, we have

$$\begin{aligned}
&\int_1^{\nu_m} \frac{k^{-k}}{m} (k^2+u^2) u^{b+k-1} e^{-up(\mathbf{x})} \, \mathrm{d}u \\
&\leq \frac{k^{-k}}{m} (k^2 p^{-(b+k)}(\mathbf{x}) \gamma(b+k, \nu_m p(\mathbf{x})) + p^{-(b+k+2)}(\mathbf{x}) \gamma(b+k+2, \nu_m p(\mathbf{x}))) \\
&= O\Big(k^{-k} \frac{(k^2 \nu_m^{(b+k) \vee 0} + \nu_m^{(b+k+2) \vee 0})}{m} \Big) \\
&= O\Big(k^{-k} \frac{\nu_m^{(b+k+2) \vee 0}}{m} \Big).
\end{aligned}$$

One can easily show that the bound also holds when $b + k \leq 0$. Hence, we conclude that

$$\begin{aligned}
I_{\text{in},2,\text{smooth}} &= \int_{S_p(\varrho(\frac{\nu_m}{m}))} I_{\text{in},2}(\mathbf{x}) p(\mathbf{x}) \, \mathrm{d}\mathbf{x} \\
&= O(m^{-\frac{\sigma_p}{d}} \nu_m^{(b+\frac{\sigma_p}{d}+2) \vee 0} + m^{-1} \nu_m^{(b+k+2) \vee 0}).
\end{aligned} \tag{4.70}$$

Similar to (4.69), we have

$$\begin{aligned}
I_{\text{in},2,\text{nonsmooth}} &= \int_{\text{supp}(p) \setminus S_p(\varrho(\frac{\nu_m}{m}))} I_{\text{in},2}(\mathbf{x}) p(\mathbf{x}) \, \mathrm{d}\mathbf{x}, \\
&= O((\nu_m/m)^{\frac{1}{d}} (\nu_m^{(b+2) \vee 0} + m^{-1} \nu_m^{(b+k+2) \vee 0})),
\end{aligned} \tag{4.71}$$

since $\lambda(\text{supp}(p) \setminus S_p(\varrho(\nu_m/m))) = O(\varrho(\nu_m/m)) = O((\nu_m/m)^{\frac{1}{d}})$ by Lemma 4.B.24. Putting (4.70) and (4.71) together establishes the desired bound (4.66). □

Lemma 4.B.25 (Generic outer bias bound)**.** *Suppose that the density p satisfies **(U_p).***

1. *If $k > -a$, we have*

$$I_{out,1} = O\big(k^{-k}\tau_m^{k+a}\big). \tag{4.72}$$

2. *If p satisfies **(L1$_p$)**, **(L2$_p$)**, and **(L3$_p$)**, then, for m sufficiently large, we have*

$$I_{out,2} = O\big(k^b\nu_m^{b+k-1}e^{-c_p\nu_m} + (\nu_m^b \vee \kappa_m^b)\big(\frac{\nu_m}{k}\big)^k e^{-\eta_p c_p \nu_m}\big). \tag{4.73}$$

Proof. Recall that

$$\rho_{U_{k\infty}(\mathbf{x})}(u) = \frac{p^k(\mathbf{x})}{\Gamma(k)}u^{k-1}e^{-up(\mathbf{x})}.$$

Define

$$A_{k\infty}(\mathbf{x};p) := \int_0^{\tau_m} u^a \rho_{U_{k\infty}(\mathbf{x})}(u)\,\mathrm{d}u$$

and

$$B_{k\infty}(\mathbf{x};p) := \int_{\nu_m}^{\infty} u^b \rho_{U_{k\infty}(\mathbf{x})}(u)\,\mathrm{d}u.$$

For some $\kappa_m = \omega(m)$ such that $\kappa_m \geq \nu_m$, we also let

$$\begin{aligned} A_{km}(\mathbf{x};p) &:= A_{km}(\mathbf{x};p;\xi), \\ B_{km}^{(2)}(\mathbf{x};p) &:= B_{km}^{(2)}(\mathbf{x};p;\xi), \quad \text{and} \\ B_{km}^{(3)}(\mathbf{x};p) &:= B_{km}^{(3)}(\mathbf{x};p;\xi) \end{aligned}$$

for $\xi(t) = t$; recall the definitions in Appendix 4.B.4. Now we can write the lower outer bias as

$$I_{\text{out},1} = \int p(\mathbf{x})(A_{km}(\mathbf{x};p) + A_{k\infty}(\mathbf{x};p))\,\mathrm{d}\mathbf{x}$$

and the upper outer bias as

$$I_{\text{out},2} = \int p(\mathbf{x})(B_{km}^{(2)}(\mathbf{x};p) + B_{km}^{(3)}(\mathbf{x};p) + B_{k\infty}(\mathbf{x};p))\,\mathrm{d}\mathbf{x}$$

Bounding the lower outer bias $I_{\text{out},1}$. On the one hand, by invoking the lower incomplete gamma function in Lemma 4.B.2, we obtain

$$\begin{aligned} A_{k\infty}(\mathbf{x};p) &= \frac{p^k(\mathbf{x})}{\Gamma(k)} \int_0^{\tau_m} u^{k+a-1} e^{-up(\mathbf{x})}\,\mathrm{d}u \\ &= \frac{p^{-a}(\mathbf{x})}{\Gamma(k)} \gamma(k+a, \tau_m p(\mathbf{x})) \\ &\le \frac{p^k(\mathbf{x})\tau_m^{k+a}}{\Gamma(k)(k+a)} \\ &\le \frac{C_p^k \tau_m^{k+a}}{\Gamma(k)(k+a)} = O(k^{-k}\tau_m^{k+a}). \end{aligned}$$

On the other hand, by applying Lemma 4.B.15 with the upper boundedness condition **($\mathbf{U}_p$)**, we obtain

$$\begin{aligned} \int p(\mathbf{x}) A_{km}(\mathbf{x};p)\,\mathrm{d}\mathbf{x} &\le \frac{C_p^k \tau_m^{k+a}}{k!}\Big(1 \vee \frac{k}{k+a}\Big) \\ &= O(k^{-k}\tau_m^{k+a}). \end{aligned}$$

Combining the two bounds, we conclude that $I_{\text{out},1} = O(k^{-k}\tau_m^{k+a})$.

Bounding the upper outer bias $I_{\text{out},2}$. For the $B_{k\infty}(\mathbf{x};p)$ term in the upper outer bias $I_{\text{out},2}$, we apply the bound (4.47) on the upper incomplete gamma function in Lemma 4.B.2. Consider

$$\begin{aligned} B_{k\infty}(\mathbf{x};p) &= \frac{p^k(\mathbf{x})}{\Gamma(k)} \int_{\nu_m}^{\infty} u^{k+b-1} e^{-up(\mathbf{x})}\,\mathrm{d}u \\ &= \frac{p^{-b}(\mathbf{x})}{\Gamma(k)} \int_{\nu_m p(\mathbf{x})}^{\infty} t^{k+b-1} e^{-t}\,\mathrm{d}t. \end{aligned}$$

If $\nu_m p(\mathbf{x}) < 1$, we have

$$\begin{aligned} B_{k\infty}(\mathbf{x}; p) &\leq \frac{p^{-b}(\mathbf{x})}{\Gamma(k)} \int_0^\infty t^{k+b-1} e^{-t} \, \mathrm{d}t \\ &\leq \frac{\Gamma((k+b) \vee 1)}{\Gamma(k)} p^{-b}(\mathbf{x}). \end{aligned}$$

We now assume that $\nu_m p(\mathbf{x}) \geq 1$. If $k + b \geq 1$, we have

$$\begin{aligned} B_{k\infty}(\mathbf{x}; p) &= \frac{p^{-b}(\mathbf{x})}{\Gamma(k)} \Gamma(k+b, \nu_m p(\mathbf{x})) \\ &\leq \frac{p^{-b}(\mathbf{x})}{\Gamma(k)} \Gamma(k+b) (\nu_m p(\mathbf{x}))^{k+b-1} e^{-\nu_m p(\mathbf{x})+1} \\ &= \frac{\Gamma(k+b)}{\Gamma(k)} \nu_m^{k+b-1} p^{k-1}(\mathbf{x}) e^{-\nu_m p(\mathbf{x})+1}, \end{aligned}$$

where the inequality follows from Lemma 4.B.2. For $k + b < 1$, a similar bound can be derived:

$$\begin{aligned} B_{k\infty}(\mathbf{x}; p) &= \frac{p^{-b}(\mathbf{x})}{\Gamma(k)} (\nu_m p(\mathbf{x}))^{k+b-1} \int_{\nu_m p(\mathbf{x})}^\infty e^{-t} \, \mathrm{d}t \\ &= \frac{1}{\Gamma(k)} \nu_m^{k+b-1} p^{k-1}(\mathbf{x}) e^{-\nu_m p(\mathbf{x})}. \end{aligned}$$

To sum up, we can bound $B_{k\infty}(\mathbf{x}; p)$ as

$$\begin{aligned} B_{k\infty}(\mathbf{x}; p) &\leq \frac{\Gamma((k+b) \vee 1)}{\Gamma(k)} (p^{-b}(\mathbf{x}) \, 1_{\{\nu_m p(\mathbf{x}) < 1\}} + \nu_m^{k+b-1} p^{k-1}(\mathbf{x}) e^{-\nu_m p(\mathbf{x})+1}) \\ &\overset{(a)}{\leq} \frac{\Gamma((k+b) \vee 1)}{\Gamma(k)} (p^{-b}(\mathbf{x}) + \nu_m^{k+b-1} p^{k-1}(\mathbf{x})) e^{-\nu_m p(\mathbf{x})+1} \\ &\overset{(b)}{\leq} \frac{\Gamma((k+b) \vee 1)}{\Gamma(k)} ((C_p^{-b} \vee c_p^{-b}) + \nu_m^{k+b-1} C_p^{k-1}) e^{-\nu_m c_p+1} \\ &= O(k^b \nu_m^{k+b-1} e^{-c_p \nu_m}). \end{aligned}$$

Here, (a) follows from the inequality $1_{\{t \leq 1\}} \leq e^{-t+1}$, and (b) follows from the bounded-

ness conditions **($\mathbf{U}_p$)** and **($\mathbf{L1}_p$)**. Therefore, we conclude that

$$\int p(\mathbf{x}) B_{k\infty}(\mathbf{x}; p)\, \mathrm{d}\mathbf{x} = O(k^b \nu_m^{k+b-1} e^{-c_p \nu_m}). \tag{4.74}$$

Next, we bound $\int p(\mathbf{x})(B_{km}^{(2)}(\mathbf{x}; p) + B_{km}^{(3)}(\mathbf{x}; p))\, \mathrm{d}\mathbf{x}$. On the one hand, applying Lemma 4.B.21 with the upper boundedness condition **($\mathbf{U}_p$)**, we first have

$$\int p(\mathbf{x}) B_{km}^{(2)}(\mathbf{x}; p)\, \mathrm{d}\mathbf{x} \leq 2(\nu_m^b \vee \kappa_m^b)\Big(\frac{eC_p \nu_m}{k}\Big)^k \int p(\mathbf{x}) e^{-\nu_m m_r p(\mathbf{x})}\, \mathrm{d}\mathbf{x}$$

for $r = \varrho(\frac{\nu_m}{m})$. Further, since we have

$$\eta_p = \inf_{\mathbf{x} \in \mathrm{supp}(p)} \inf_{r' \in (0, r]} \frac{\lambda(\mathbf{B}(\mathbf{x}, r) \cap \mathrm{supp}(p))}{\lambda(\mathbf{B}(\mathbf{x}, r))} > 0$$

from condition **($\mathbf{L3}_p$)**, it follows that $m_r p(\mathbf{x}) \geq c_p \eta_p$ for $\mathbf{x} \in \mathrm{supp}(p)$, leading to

$$\int p(\mathbf{x}) B_{km}^{(2)}(\mathbf{x}; p)\, \mathrm{d}\mathbf{x} \leq 2(\nu_m^b \vee \kappa_m^b)\Big(\frac{e\nu_m C_p}{k}\Big)^k e^{-\eta_p c_p \nu_m}.$$

On the other hand, since the support of the density p is bounded by the condition **($\mathbf{L2}_p$)**, $R(p, p; \xi, b, \varrho(\kappa_m/m))$ becomes 0 for m sufficiently large, since $\kappa_m/m \to \infty$ as $m \to \infty$. Hence, by applying Lemma 4.B.18 for a fixed $\delta > 0$, we have

$$\int p(\mathbf{x}) B_{km}^{(3)}(\mathbf{x}; p)\, \mathrm{d}\mathbf{x} \leq 3(1+\delta) m^b R\big(p, p; \xi, b, \varrho\big(\frac{\kappa_m}{m}\big)\big) = 0$$

for m sufficiently large. Therefore, we conclude that

$$\int p(\mathbf{x})(B_{km}^{(2)}(\mathbf{x}; p) + B_{km}^{(3)}(\mathbf{x}; p))\, \mathrm{d}\mathbf{x} = O(\nu_m^b \vee \kappa_m^b)\big(\frac{\nu_m}{k}\big)^k e^{-\eta_p c_p \nu_m}\big). \tag{4.75}$$

Combining the bounds (4.74) and (4.75) establishes the desired bound (4.73). □

Remark 4.B.26. A more general condition, namely, that

(B1$'_p$) there exists $E_0, E_1 > 0$ such that $\int p(\mathbf{x})e^{-\beta p(\mathbf{x})}\,\mathrm{d}\mathbf{x} \leq E_0 e^{-E_1\beta}$ for all $\beta > 1$,

was originally assumed in (Gao et al., 2018). Known examples of densities that satisfy the condition **(B1$'_p$)** satisfy the more intuitive condition **(L1$_p$)**. We remark, however, that it is nontrivial to adapt the proofs in this paper to work with **(B1$'_p$)** in place of **(L1$_p$)**, as the lower boundedness condition **(L1$_p$)** is explicitly utilized to remove the upper truncation of the estimator in the analysis of (Gao et al., 2018).

4.B.6 Generic Variance Bounds

Lemma 4.B.27. *For a given function $\phi\colon \mathbb{R}_+ \to \mathbb{R}$, let $\zeta_k(\mathbf{x}|\mathbf{x}_{1:m}) := \phi(r_k(\mathbf{x}|\mathbf{x}_{1:m}))$ for any points $\mathbf{x}, \mathbf{x}_{1:m}$ in the d-dimensional Euclidean space $(\mathbb{R}^d, \|\cdot\|)$. Let*

$$\Phi(\mathbf{x}_{1:m}) = \frac{1}{m}\sum_{i=1}^{m} \zeta_k(\mathbf{x}_i|\mathbf{x}_{1:m}^{\sim i}). \tag{4.76}$$

If the samples $\mathbf{X}_{1:m}$ are i.i.d., then

$$\mathrm{Var}(\Phi(\mathbf{X}_{1:m})) \leq \frac{2(1+k\gamma_d)}{m}\{(2k+1)\mathbb{E}[\zeta_k^2(\mathbf{X}_m|\mathbf{X}_{1:m-1})] + 2k\mathbb{E}[\zeta_{k+1}^2(\mathbf{X}_m|\mathbf{X}_{1:m-1})]\},$$

where $\gamma_d \in \mathbb{N}$ is a constant which depends only on d.

Before we prove Lemma 4.B.27, we introduce two technical lemmas.

Lemma 4.B.28 (Efron–Stein inequality (Efron and Stein, 1981; Steele, 1986))**.** *Let $X_1,\ldots,X_n$ be independent random variables, and let $g(X_{1:n}) = g(X_1,\ldots,X_n)$ be a square-integrable function of $X_1,\ldots,X_n$. Then if $X'_1,\ldots,X'_n$ are independent copies of $X_1,\ldots,X_n$, we have*

$$\mathrm{Var}(g(X_{1:n})) \leq \frac{1}{2}\sum_{i=1}^{n}\mathbb{E}\big[|g(X_{1:n}) - g(X_{1:i-1}X'_iX_{i+1:n})|^2\big].$$

The proof of this lemma can be found in (Steele, 1986).

We need another fact on k-nearest neighbors in the Euclidean space, stated below in Lemma 4.B.27. Informally speaking, given a finite collection S of points in $\mathbb{R}^d$, each fixed point in $\mathbb{R}^d$ can be one of the k nearest neighbors of at most γ_d points in S, where γ_d depends only on d. Henceforth, for a set of points A such that $\mathbf{x} \notin A$, we use $N_k(\mathbf{x}|A)$ to denote the k-nearest neighbors of $\mathbf{x}$ in A.

Lemma 4.B.29 ((Biau and Devroye, 2015, Lemma 20.6), (Devroye et al., 2013, Ch. 5.3))**.** *In the d-dimensional Euclidean space $(\mathbb{R}^d, \|\cdot\|)$ there exists a constant $\gamma_d > 0$ which depends only on d such that for any $m \in \mathbb{N}$ and for any distinct points $\mathbf{x}, \mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^d$,*

$$\sum_{i=1}^{m} 1_{\{\mathbf{x} \in N_k(\mathbf{x}_i | \mathbf{x}_{1:m}^{\sim i}, \mathbf{x})\}} \leq k\gamma_d.$$

Proof. We follow the proof of Stone's lemma in Devroye et al. (2013, Ch. 5.3). For $\mathbf{z} \in \mathbb{R}^d \backslash \{\mathbf{0}\}$ and $\theta \in (0, \pi/2]$, we define a cone $\mathcal{C}(\mathbf{z}, \theta) := \{\mathbf{y} \in \mathbb{R}^d \colon \mathbf{y} = \mathbf{0} \text{ or } \angle(\mathbf{z}, \mathbf{y}) \leq \theta\}$. It is well known (Biau and Devroye, 2015, Theorem 20.16) that there exists a constant $\gamma_d > 0$, which depends only on the dimension d, such that there exist γ_d cones $\mathcal{C}(\mathbf{z}_1, \pi/6), \ldots, \mathcal{C}(\mathbf{z}_{\gamma_d}, \pi/6)$ which cover the entire space $\mathbb{R}^d$. Furthermore, it is easy to see that ($\star$) if $\mathbf{y}_1, \mathbf{y}_2 \in \mathcal{C}(\mathbf{x}, \pi/6)$ and $\|\mathbf{y}_1\| < \|\mathbf{y}_2\|$, then $\|\mathbf{y}_1 - \mathbf{y}_2\| < \|\mathbf{y}_2\|$; see, e.g., (Biau and Devroye, 2015, Lemma 20.5).

Now, for each $j \in [\gamma_d]$, *mark* all $\mathbf{x}_i$'s (if any) among the k-nearest neighbors of $\mathbf{x}$ in $\mathbf{x} + \mathcal{C}(\mathbf{z}_j, \pi/6)$. If $\mathbf{x}_i \in \mathbf{x} + \mathcal{C}(\mathbf{z}_j, \pi/6)$ for some $j \in [\gamma_d]$ and $\mathbf{x}_i$ is not marked, then $\mathbf{x}$ is not among the k-nearest neighbors of $\mathbf{x}_i$ in $\mathbf{x}_{1:i-1}, \mathbf{x}_{i+1:m}, \mathbf{x}$, i.e., $\mathbf{x} \notin N_k(\mathbf{x}_i | \mathbf{x}_{1:m}^{\sim i}, \mathbf{x})$, by the property ($\star$). Therefore, we have

$$\sum_{i=1}^{n} 1_{\{\mathbf{x} \in N_k(\mathbf{x}_i | \mathbf{x}_{1:m}^{\sim i}, \mathbf{x})\}} \leq \sum_{i=1}^{n} 1_{\{\mathbf{x}_i \text{ is marked}\}} \leq k\gamma_d,$$

since there exist at most $k\gamma_d$ marked points. □

We are now ready to prove Lemma 4.B.27.

Proof of Lemma 4.B.27. Let $\mathbf{X}_1'$ be an independent copy of $\mathbf{X}_1$. Then, by applying the Efron–Stein inequality (Lemma 4.B.28), we have

$$\begin{aligned}
&\mathrm{Var}\big(\Phi(\mathbf{X}_{1:m})\big) \\
&\leq \frac{m}{2}\mathbb{E}\big[\big(\Phi(\mathbf{X}_{1:m}) - \Phi(\mathbf{X}_1'\mathbf{X}_{2:m})\big)^2\big] \\
&\overset{(a)}{\leq} m\mathbb{E}\big[\big(\Phi(\mathbf{X}_{1:m}) - \frac{m-1}{m}\Phi(\mathbf{X}_{2:m})\big)^2 + \big(\Phi(\mathbf{X}_1'\mathbf{X}_{2:m}) - \frac{m-1}{m}\Phi(\mathbf{X}_{2:m})\big)^2\big] \qquad (4.77) \\
&= 2m\mathbb{E}\big[\big(\Phi(\mathbf{X}_{1:m}) - \frac{m-1}{m}\Phi(\mathbf{X}_{2:m})\big)^2\big], \qquad (4.78)
\end{aligned}$$

where (a) follows from the elementary inequality $(a-b)^2 \leq 2((a-x)^2 + (b-x)^2)$.

Define

$$E_i := \{\mathbf{X}_1 \text{ is one of the } k\text{-NNs of } \mathbf{X}_i \text{ in } \mathbf{X}_{1:m}^{\sim i}\}$$

for $2 \leq i \leq m$. Applying Lemma 4.B.29, we obtain

$$\sum_{i=2}^{m} 1_{E_i} \leq k\gamma_d.$$

Further, note that if E_i^c occurs, i.e., $\mathbf{X}_1$ is not among the k nearest neighbors of $\mathbf{X}_i$ in $\mathbf{X}_{1:m}^{\sim i}$, then $\zeta_k(\mathbf{X}_i|\mathbf{X}_{1:m}^{\sim i}) = \zeta_k(\mathbf{X}_i|\mathbf{X}_{2:m}^{\sim i})$. We thus obtain (4.79), where (b) follows from Cauchy–Schwarz inequality. By taking expectations with respect to $\mathbf{X}_{1:m}$ on both sides and multiplying by $2/m$, we can continue from (4.78) to obtain

$$\begin{aligned}
&\mathrm{Var}\big(\Phi(\mathbf{X}_{1:m})\big) \qquad (4.80) \\
&\leq \frac{2(1+k\gamma_d)}{m}\Big\{\mathbb{E}\big[\zeta_k^2(\mathbf{X}_1|\mathbf{X}_{2:m})\big] + 2\mathbb{E}\Big[\sum_{i=2}^{m} 1_{E_i}(\zeta_k^2(\mathbf{X}_i|\mathbf{X}_{1:m}^{\sim i}) + \zeta_k^2(\mathbf{X}_i|\mathbf{X}_{2:m}^{\sim i}))\Big]\Big\}.
\end{aligned}$$

Note that if E_i occurs, i.e., $\mathbf{X}_1$ is among the k nearest neighbors of $\mathbf{X}_i$ in $\mathbf{X}_{1:m}^{\sim i}$, we have

$$
\begin{aligned}
&m^2\big(\Phi(\mathbf{X}_{1:m}) - \frac{m-1}{m}\Phi(\mathbf{X}_{2:m})\big)^2 \\
&= \Big(\zeta_k(\mathbf{X}_1|\mathbf{X}_{2:m}) + \sum_{i=2}^{m} 1_{E_i}\big(\zeta_k(\mathbf{X}_i|\mathbf{X}_{1:m}^{\sim i}) - \zeta_k(\mathbf{X}_i|\mathbf{X}_{2:m}^{\sim i})\big)\Big)^2 \\
&\overset{(b)}{\leq} \Big(1 + \sum_{i=2}^{m} 1_{E_i}\Big)\Big(\zeta_k^2(\mathbf{X}_1|\mathbf{X}_{2:m}) + \sum_{i=2}^{m} 1_{E_i}\big(\zeta_k(\mathbf{X}_i|\mathbf{X}_{1:m}^{\sim i}) - \zeta_k(\mathbf{X}_i|\mathbf{X}_{2:m}^{\sim i})\big)^2\Big) \\
&\leq (1 + k\gamma_d)\Big(\zeta_k^2(\mathbf{X}_1|\mathbf{X}_{2:m}) + 2\sum_{i=2}^{m} 1_{E_i}\big(\zeta_k^2(\mathbf{X}_i|\mathbf{X}_{1:m}^{\sim i}) + \zeta_k^2(\mathbf{X}_i|\mathbf{X}_{2:m}^{\sim i})\big)\Big). \qquad (4.79)
\end{aligned}
$$

$\zeta_k(\mathbf{X}_i|\mathbf{X}_{2:m}^{\sim i}) = \zeta_{k+1}(\mathbf{X}_i|\mathbf{X}_{1:m}^{\sim i})$. Therefore, it follows that

$$
\begin{aligned}
&\mathbb{E}\Big[\sum_{i=2}^{m} 1_{E_i}(\zeta_k^2(\mathbf{X}_i|\mathbf{X}_{1:m}^{\sim i}) + \zeta_k^2(\mathbf{X}_i|\mathbf{X}_{2:m}^{\sim i}))\Big] \\
&= \mathbb{E}\Big[\sum_{i=2}^{m} 1_{E_i}(\zeta_k^2(\mathbf{X}_i|\mathbf{X}_{1:m}^{\sim i}) + \zeta_{k+1}^2(\mathbf{X}_i|\mathbf{X}_{1:m}^{\sim i}))\Big] \\
&\overset{(c)}{=} \mathbb{E}\Big[\sum_{i=2}^{m} 1_{\{\mathbf{X}_i \text{ is among the } k\text{-NNs of } \mathbf{X}_1 \text{ in } \mathbf{X}_{2:m}\}}(\zeta_k^2(\mathbf{X}_1|\mathbf{X}_{2:m}) + \zeta_{k+1}^2(\mathbf{X}_1|\mathbf{X}_{2:m}))\Big] \\
&= k\mathbb{E}[\zeta_k^2(\mathbf{X}_1|\mathbf{X}_{2:m}) + \zeta_{k+1}^2(\mathbf{X}_1|\mathbf{X}_{2:m})], \qquad (4.81)
\end{aligned}
$$

where (c) follows by exchanging $\mathbf{X}_1$ and $\mathbf{X}_i$ in each summand $2 \leq i \leq m$. Therefore, plugging the equation in (4.81) into (4.80) proves the desired bound. □

For the double-density case, we can establish a similar variance bound.

Lemma 4.B.30. *For a given function $\phi\colon \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}$, let*

$$
\zeta_{kl}(\mathbf{x}|\mathbf{x}_{1:m}, \mathbf{y}_{1:n}) := \phi(r_k(\mathbf{x}|\mathbf{x}_{1:m}), r_l(\mathbf{x}|\mathbf{y}_{1:n}))
$$

for any points $\mathbf{x}, \mathbf{x}_{1:m}, \mathbf{y}_{1:n}$ in the d-dimensional Euclidean space $(\mathbb{R}^d, \|\cdot\|)$. Let

$$
\Phi(\mathbf{x}_{1:m}, \mathbf{y}_{1:n}) := \frac{1}{m}\sum_{i=1}^{m} \zeta_{kl}(\mathbf{x}_i|\mathbf{x}_{1:m}^{\sim i}, \mathbf{y}_{1:n}). \qquad (4.82)
$$

If $\mathbf{X}_{1:m}$ and $\mathbf{Y}_{1:n}$ are independent i.i.d. samples, we have

$$\mathrm{Var}(\Phi(\mathbf{X}_{1:m}, \mathbf{Y}_{1:n})) \leq \frac{2(1+k\gamma_d)}{m}\{(2k+1)\mathbb{E}[\zeta_{kl}^2(\mathbf{X}_m|\mathbf{X}_{1:m-1}, \mathbf{Y}_{1:n})] + 2k\mathbb{E}[\zeta_{k+1,l}^2(\mathbf{X}_m|\mathbf{X}_{1:m-1}, \mathbf{Y}_{1:n})]\}.$$

Proof. Given $\mathbf{Y}_{1:n} = \mathbf{y}_{1:n}$, we can show that

$$\begin{aligned}
&\mathrm{Var}\big(\Phi\big(\mathbf{X}_{1:m}, \mathbf{y}_{1:n}\big)\big) \\
&\leq 2m\mathbb{E}\big[\big(\Phi(\mathbf{X}_{1:m}, \mathbf{y}_{1:n}) - \frac{m-1}{m}\Phi(\mathbf{X}_{2:m}, \mathbf{y}_{1:n})\big)^2\big] \\
&\leq \frac{2(1+k\gamma_d)}{m}\{(2k+1)\mathbb{E}[\zeta_{kl}^2(\mathbf{X}_m|\mathbf{X}_{1:m-1}, \mathbf{y}_{1:n})] + 2k\mathbb{E}[\zeta_{k+1,l}^2(\mathbf{X}_m|\mathbf{X}_{1:m-1}, \mathbf{y}_{1:n})]\}
\end{aligned}$$

by following the same line of reasoning as in the proof of Lemma 4.B.27. Since $\mathbf{Y}_{1:n}$ is independent of $\mathbf{X}_{1:m}$, taking expectation on both sides with respect to $\mathbf{Y}_{1:n}$ establishes the desired bound. □

4.C Deferred Proofs of Main Results

4.C.1 Detailed Proof of Theorem 4.3.15

We continue the proof from (4.31).

$$\big|\mathbb{E}[\hat{T}_f^{(k)}] - T_f(p)\big| \lesssim I_{\text{out},1} + I_{\text{in},1} + I_{\text{in},2} + I_{\text{out},2}. \tag{4.31}$$

Applying the bounds in Lemmas 4.B.23 and 4.B.25, we obtain the following bias bound for an underlying density p satisfying the conditions **($\mathbf{U}_p$)**, **($\mathbf{L1}_p$)**, **($\mathbf{S}_p$)**, and **($\mathbf{B}_p$)**, provided that $\nu_m = o(\sqrt{m})$ as $m \to \infty$ and $k \in \mathbb{N}$ is fixed:

$$\begin{aligned}
|\mathbb{E}[\hat{T}_f^{(k)}] - T_f(p)| \lesssim_{\sigma_p, L, C_p, C_0, d, k} \quad & m^{-\frac{\sigma_p}{d}}\tau_m^{(a+\frac{\sigma_p}{d}+1)\wedge 0} + m^{-1} + m^{-\frac{1}{d}}\tau_m^{(a+1)\wedge 0} \\
& + m^{-\frac{\sigma_p}{d}}\nu_m^{(b+\frac{\sigma_p}{d}+2)\vee 0} + m^{-1}\nu_m^{(b+k+2)\vee 0} + m^{-\frac{1}{d}}\nu_m^{(b+2)\vee 0+\frac{1}{d}}
\end{aligned}$$

$$+ \tau_m^{k+a} + \nu_m^{b+k-1} e^{-c_p \nu_m}.$$

First, by choosing $\nu_m = \Theta((\ln m)^{1+\delta})$ for some $\delta > 0$, we make the last term $\nu_m^{b+k-1} e^{-c_p \nu_m}$ decay faster than any polynomial rate. With this choice, the bound can be simplified as

$$|\mathbb{E}[\hat{T}_f^{(k)}] - T_f(p)| = \tilde{O}_{\sigma_p, L, C_p, C_0, d, k}(\tau_m^{(a+\frac{\sigma_p}{d}+1)\wedge 0} m^{-\frac{\sigma_p}{d}} + \tau_m^{(a+1)\wedge 0} m^{-\frac{1}{d}} + m^{-\frac{\sigma_p \wedge 1}{d}} + \tau_m^{k+a}).$$

We consider three different ranges of the lower tail exponent a.

1. If $a \leq -\sigma_p/d - 1$, we have

   $$|\mathbb{E}[\hat{T}_f^{(k)}] - T_f(p)| = \tilde{O}(\tau_m^{a+1} m^{-\frac{\sigma_p \wedge 1}{d}} + \tau_m^{k+a})$$

   as a suboptimal bound. By equating the two terms, we obtain a rate $\tilde{O}(m^{-\frac{(\sigma_p \wedge 1)}{d}\frac{k+a}{k-1}})$ with $\tau_m = \Theta(m^{-\frac{(\sigma_p \wedge 1)}{d}\frac{1}{k-1}})$.

2. If $-\sigma_p/d - 1 < a \leq -1$, the rate becomes

   $$|\mathbb{E}[\hat{T}_f^{(k)}] - T_f(p)| = \tilde{O}(\tau_m^{a+1} m^{-\frac{1}{d}} + m^{-\frac{\sigma_p \wedge 1}{d}} + \tau_m^{k+a}).$$

   Equating $\tau_m^{a+1} m^{-\frac{1}{d}}$ and τ_m^{k+a} as a suboptimal choice, we obtain $\tau_m = \Theta(m^{-\frac{1}{d}\frac{1}{k-1}})$, which results in the final rate

   $$\begin{aligned} |\mathbb{E}[\hat{T}_f^{(k)}] - T_f(p)| &= \tilde{O}(m^{-\frac{1}{d}\frac{k+a}{k-1}} + m^{-\frac{\sigma_p \wedge 1}{d}}) \\ &= \tilde{O}(m^{-\frac{1}{d}(\sigma_p \wedge \frac{k+a}{k-1})}) \end{aligned}$$

3. If $a > -1$, we can attain the bias rate $\tilde{O}(m^{-\frac{\sigma_p \wedge 1}{d}})$ by using $\tau_m = O(m^{-\frac{1}{d(a+1)}})$.

To sum up, by choosing

$$\tau_m = \tau(m, d, \sigma_p, a, k) \tag{4.83}$$
$$= \begin{cases} \Theta\big(m^{-\frac{\sigma_p \wedge 1}{d(k-1)}}\big) & \text{if } a \leq -\frac{\sigma_p}{d} - 1, \\ \Theta\big(m^{-\frac{1}{d(k-1)}}\big) & \text{if } -\frac{\sigma_p}{d} - 1 < a \leq -1, \\ O\big(m^{-\frac{1}{d(a+1)}}\big) & \text{if } a > -1, \end{cases}$$

we establish the bias bound in Theorem 4.3.15. □

4.C.2 Proof of Theorem 4.4.7

Following a similar line of reasoning as in the proof of Proposition 4.1.1 and using the continuous mapping theorem, it is easy to show that $\phi_k(U_{k,m-1}(\mathbf{X}_m), V_{ln}(\mathbf{X}_m))$ converges to $\phi_{kl}(U_{k\infty}(\mathbf{X}), V_{l\infty}(\mathbf{X}))$ in distribution as $m, n \to \infty$, where $U_{k\infty}(\mathbf{x})$ and $V_{l\infty}(\mathbf{x})$ are a $\mathsf{G}(k, p(\mathbf{x}))$ random variable and a $\mathsf{G}(l, q(\mathbf{x}))$ random variable, respectively, which are independent of each other and of $\mathbf{X} \sim p$, for P-a.e. $\mathbf{x}$. Hence, if we can only show that the collection of random variables $(\phi_{kl}(U_{k,m-1}(\mathbf{X}_m), V_{ln}(\mathbf{X}_m)))_{m,n\geq 1}$ is uniformly integrable, we can readily establish the asymptotic unbiasedness as follows:

$$\begin{aligned} \lim_{m,n\to\infty} \mathbb{E}[\hat{T}_f^{(kl)}(\mathbf{X}_{1:m}, \mathbf{Y}_{1:n})] &= \lim_{m,n\to\infty} \mathbb{E}[\phi_{kl}(U_{k,m-1}(\mathbf{X}_m), V_{ln}(\mathbf{X}_m))] \\ &= \mathbb{E}[\phi_{kl}(U_{k\infty}(\mathbf{X}), V_{l\infty}(\mathbf{X}))] \\ &= T_f(p, q). \end{aligned}$$

Consider

$$\mathbb{E}[\xi(|\phi_{kl}(U_{k,m-1}(\mathbf{X}_m), V_{ln}(\mathbf{X}_m))|)] = \int p(\mathbf{x}) \mathbb{E}[\xi(|\phi_{kl}(U_{k,m-1}(\mathbf{x}), V_{ln}(\mathbf{x}))|)] \, \mathrm{d}\mathbf{x}.$$

By invoking the polynomial bound $|\phi_{kl}(u,v)| \lesssim \psi_{a,b}(u)\psi_{\tilde{a},\tilde{b}}(v)$ and using the independence of $U_{k,m-1}(\mathbf{x})$ and $V_{ln}(\mathbf{x})$, we have

$$\begin{aligned}
&\mathbb{E}[\xi(|\phi_{kl}(U_{k,m-1}(\mathbf{X}_m), V_{ln}(\mathbf{X}_m))|)] \\
&\lesssim_{\xi(t_0)} 1 + \mathbb{E}[\xi(\psi_{a,b}(U_{k,m-1}(\mathbf{X}_m)))] \\
&\qquad + \mathbb{E}[\xi(\psi_{\tilde{a},\tilde{b}}(V_{ln}(\mathbf{X}_m)))] \\
&\qquad + \{\mathbb{E}[(\mathbb{E}[\xi(\psi_{a,b}(U_{km}(\mathbf{X}_m)))|\mathbf{X}_m]\mathbb{E}[\xi(\psi_{\tilde{a},\tilde{b}}(V_{ln}(\mathbf{X}_m)))|\mathbf{X}_m])]\}^2,
\end{aligned} \tag{4.84}$$

since $\xi(xy) \leq \xi(x)\xi(y)$ for any $x, y > t_0$. We can bound the last term as

$$\begin{aligned}
&\{\mathbb{E}[(\mathbb{E}[\xi(\psi_{a,b}(U_{km}(\mathbf{X}_m)))|\mathbf{X}_m]\mathbb{E}[\xi(\psi_{\tilde{a},\tilde{b}}(V_{ln}(\mathbf{X}_m)))|\mathbf{X}_m])]\}^2 \\
&\overset{(a)}{\leq} \mathbb{E}[(\mathbb{E}[\xi^2(\psi_{a,b}(U_{km}(\mathbf{X}_m)))|\mathbf{X}_m])^2]\mathbb{E}[(\mathbb{E}[\xi^2(\psi_{\tilde{a},\tilde{b}}(V_{ln}(\mathbf{X}_m)))|\mathbf{X}_m])^2] \\
&\overset{(b)}{\leq} \mathbb{E}[\xi^2(\psi_{a,b}(U_{k,m-1}(\mathbf{X}_m)))]\mathbb{E}[\xi^2(\psi_{\tilde{a},\tilde{b}}(V_{ln}(\mathbf{X}_m)))],
\end{aligned}$$

where (a) and (b) follow from Cauchy–Schwarz inequality and Jensen's inequality. We thus only need to show that

$$\limsup_{m\to\infty} \mathbb{E}[\xi^2(\psi_{a,b}(U_{k,m-1}(\mathbf{X}_m)))] < \infty$$

and

$$\limsup_{n\to\infty} \mathbb{E}[\xi^2(\psi_{\tilde{a},\tilde{b}}(V_{ln}(\mathbf{X}_m)))] < \infty,$$

since they would imply that all the terms in (4.84) are bounded. Applying Lemma 4.B.19 to both integrals for $k > -2a\omega(\xi)$ and $l > -2\tilde{a}\omega(\xi)$, we conclude the proof by the de la Vallée Poussin theorem (Lemma 4.3.11). □

4.C.3 Proof of Theorem 4.4.8

Recall from the generic variance bound (Lemma 4.B.30) that we have

$$\operatorname{Var}(T_f^{(kl)}) \leq \frac{2(1+k\gamma_d)}{m}\{(2k+1)\mathbb{E}[\phi_{kl}^2(U_{k,m-1}(\mathbf{X}_m), V_{ln}(\mathbf{X}_m))] + 2k\mathbb{E}[\phi_{kl}^2(U_{k+1,m-1}(\mathbf{X}_m), V_{ln}(\mathbf{X}_m))]\}.$$

Hence, following the same logic as in Section 4.C.2, in order to ensure that $\operatorname{Var}(\hat{T}_f^{(kl)}) = O(m^{-1})$ for m and n sufficiently large, it is enough to show that

$$\limsup_{m\to\infty} \mathbb{E}[\xi^2(\psi_{a,b}(U_{k',m-1}(\mathbf{X}_m)))] < \infty$$

and

$$\limsup_{n\to\infty} \mathbb{E}[\xi^2(\psi_{\tilde{a},\tilde{b}}(V_{ln}(\mathbf{X}_m)))] < \infty$$

for $\xi(t) = t^2$ and for $k' \in \{k, k+1\}$. By applying Lemma 4.B.19 to both integrals for $k > -4a$ and $l > -4\tilde{a}$ with $\xi(t) = t^2$, we conclude the proof. □

4.C.4 Proof of Theorem 4.4.12

Let $k > -a$ and $l > -\tilde{a}$ be fixed. First, following similar steps as in (4.29), we can write the expected value of $\hat{T}_f^{(kl)}(\mathbf{X}_{1:m}, \mathbf{Y}_{1:n})$ as

$$\mathbb{E}\big[\hat{T}_f^{(kl)}(\mathbf{X}_{1:m}, \mathbf{Y}_{1:n})\big] = \int p(\mathbf{x})\mathbb{E}\big[\phi_{kl}(U_{k,m-1}(\mathbf{x}), V_{ln}(\mathbf{x}))\big]\,\mathrm{d}\mathbf{x},$$

since $U_{k,m-1}(\mathbf{x})$ and $V_{ln}(\mathbf{x})$ are independent of $\mathbf{X}_m = \mathbf{x}$ for P-a.e. $\mathbf{x}$. Moreover, similar to (4.30), we can write the target density functional as

$$T_f(p,q) = \int p(\mathbf{x})\mathbb{E}[\phi_{kl}(U_{k\infty}(\mathbf{x}), V_{l\infty}(\mathbf{x}))]\,\mathrm{d}\mathbf{x},$$

$$I_{\text{in}}(\mathbf{x}) := \int_{\square_{m,n}} \psi_{a,b}(u)\psi_{\tilde{a},\tilde{b}}(v)\left|\rho_{U_{k\infty}(\mathbf{x})}(u)\rho_{V_{l\infty}(\mathbf{x})}(v) - \rho_{U_{k,m-1}(\mathbf{x})}(u)\rho_{V_{ln}(\mathbf{x})}(v)\right| \mathrm{d}u\,\mathrm{d}v, \tag{4.86}$$

$$I_{\text{out}}(\mathbf{x}) := \int_{\mathbb{R}^2_+\setminus\square_{m,n}} \psi_{a,b}(u)\psi_{\tilde{a},\tilde{b}}(v)(\rho_{U_{k\infty}(\mathbf{x})}(u)\rho_{V_{l\infty}(\mathbf{x})}(v) + \rho_{U_{k,m-1}(\mathbf{x})}(u)\rho_{V_{ln}(\mathbf{x})}(v))\,\mathrm{d}u\,\mathrm{d}v. \tag{4.87}$$

---

where $U_{k\infty}(\mathbf{x}) \sim \mathsf{G}(k, p(\mathbf{x}))$ and $V_{l\infty}(\mathbf{x}) \sim \mathsf{G}(l, q(\mathbf{x}))$ are independent each other, and of $\mathbf{X} \sim p$ for P-a.e. $\mathbf{x}$. Consider real numbers τ_m, ν_m, $\tilde{\tau}_n$, and $\tilde{\nu}_n$, to be determined later, such that $0 \le \tau_m \le 1 \le \nu_m < \infty$ and $0 \le \tilde{\tau}_n \le 1 \le \tilde{\nu}_n < \infty$. Using the polynomial bound $|\phi_{kl}(u,v)| \lesssim \psi_{a,b}(u)\psi_{\tilde{a},\tilde{b}}(v)$ and the triangle inequality, we then have

$$\begin{aligned} |\mathbb{E}[\hat{T}_f^{(kl)}] - T_f(p,q)| &\lesssim \int (I_{\text{in}}(\mathbf{x}) + I_{\text{out}}(\mathbf{x}))p(\mathbf{x})\,\mathrm{d}\mathbf{x} \\ &= I_{\text{in}} + I_{\text{out}}, \end{aligned} \tag{4.85}$$

where $I_{\text{in}}(\mathbf{x})$ and $I_{\text{out}}(\mathbf{x})$ are defined in (4.86) and (4.87), where $\square_{m,n} := (\tau_m, \nu_m) \times (\tilde{\tau}_n, \tilde{\nu}_n)$. We bound the inner bias $I_{\text{in}} = \int I_{\text{in}}(\mathbf{x})p(\mathbf{x})\,\mathrm{d}\mathbf{x}$ and the outer bias $I_{\text{out}} = \int I_{\text{out}}(\mathbf{x})p(\mathbf{x})\,\mathrm{d}\mathbf{x}$ separately. Henceforth, we use the following shorthand notation:

$$\overline{\psi}_{a,b}(u;\tau,\nu) = \psi_{a,b}(u)\,1_{(\tau,\nu)}(u)$$

and

$$\overline{\overline{\psi}}_{a,b}(u;\tau,\nu) = \psi_{a,b}(u)(1 - 1_{(\tau,\nu)}(u)).$$

Step 1: Bounding the inner bias. For $\mathbf{x} \in \mathbb{R}^d$, let

$$\delta_{km}^{(p)}(u|\mathbf{x}) := \left|\rho_{U_{k,m-1}(\mathbf{x})}(u) - \rho_{U_{k\infty}(\mathbf{x})}(u)\right|$$

and

$$\delta_{ln}^{(q)}(v|\mathbf{x}) := \left|\rho_{V_{ln}(\mathbf{x})}(v) - \rho_{V_{l\infty}(\mathbf{x})}(v)\right|.$$

$$I_{\text{out}}(\mathbf{x}) \leq \int_{\mathbb{R}\setminus(\tau_m,\nu_m)} \int_{\tilde{\tau}_n}^{\tilde{\nu}_n} (\rho_{U_{k\infty}(\mathbf{x})}(u)\rho_{V_{l\infty}(\mathbf{x})}(v) + \rho_{U_{k,m-1}(\mathbf{x})}(u)\rho_{V_{ln}(\mathbf{x})}(v))\psi_{a,b}(u)\psi_{\tilde{a},\tilde{b}}(v)\,\mathrm{d}u\,\mathrm{d}v \quad (4.88)$$
$$+ \int_{\tau_m}^{\nu_m} \int_{\mathbb{R}\setminus(\tilde{\tau}_n,\tilde{\nu}_n)} (\rho_{U_{k\infty}(\mathbf{x})}(u)\rho_{V_{l\infty}(\mathbf{x})}(v) + \rho_{U_{k,m-1}(\mathbf{x})}(u)\rho_{V_{ln}(\mathbf{x})}(v))\psi_{a,b}(u)\psi_{\tilde{a},\tilde{b}}(v)\,\mathrm{d}u\,\mathrm{d}v.$$

By the triangle inequality, we have

$$\begin{aligned}\left|\rho_{U_{k,m-1}(\mathbf{x})}(u)\rho_{V_{ln}(\mathbf{x})}(v) - \rho_{U_{k\infty}(\mathbf{x})}(u)\rho_{V_{l\infty}(\mathbf{x})}(v)\right| &\leq \delta_{km}^{(p)}(u|\mathbf{x})\rho_{V_{ln}(\mathbf{x})}(v) + \delta_{ln}^{(q)}(v|\mathbf{x})\rho_{U_{k\infty}(\mathbf{x})}(v) \\ &\leq \delta_{km}^{(p)}(u|\mathbf{x})\delta_{ln}^{(q)}(v|\mathbf{x}) + \delta_{km}^{(p)}(u|\mathbf{x})\rho_{V_{l\infty}}(\mathbf{x}) \\ &\quad + \delta_{ln}^{(q)}(v|\mathbf{x})\rho_{U_{k\infty}}(\mathbf{x}).\end{aligned}$$

Therefore, for each $\mathbf{x} \in \text{supp}(p)$, we can bound $I_{\text{in}}(\mathbf{x})$ as

$$\begin{aligned}I_{\text{in}}(\mathbf{x}) &\leq \int_{\tau_m}^{\nu_m} \psi_{a,b}(u)\delta_{km}^{(p)}(u|\mathbf{x})\,\mathrm{d}u \int_{\tilde{\tau}_n}^{\tilde{\nu}_n} \psi_{\tilde{a},\tilde{b}}(v)\delta_{ln}^{(q)}(v|\mathbf{x})\,\mathrm{d}v \\ &\quad + \mathbb{E}[\overline{\psi}_{\tilde{a},\tilde{b}}(V_{l\infty}(\mathbf{x}); \tilde{\tau}_n, \tilde{\nu}_n)] \int_{\tau_m}^{\nu_m} \psi_{a,b}(u)\delta_{km}^{(p)}(u|\mathbf{x})\,\mathrm{d}u \\ &\quad + \mathbb{E}[\overline{\psi}_{a,b}(U_{k\infty}(\mathbf{x}); \tau_m, \nu_m)] \int_{\tilde{\tau}_n}^{\tilde{\nu}_n} \psi_{\tilde{a},\tilde{b}}(v)\delta_{ln}^{(q)}(v|\mathbf{x})\,\mathrm{d}v \\ &\overset{(a)}{\lesssim} \int_{\tau_m}^{\nu_m} \psi_{a,b}(u)\delta_{km}^{(p)}(u|\mathbf{x})\,\mathrm{d}u + \int_{\tilde{\tau}_n}^{\tilde{\nu}_n} \psi_{\tilde{a},\tilde{b}}(v)\delta_{ln}^{(q)}(v|\mathbf{x})\,\mathrm{d}v,\end{aligned}$$

where (a) follows by applying Lemma 4.B.22 with the assumptions **($\mathbf{U}_p$)** and **($\mathbf{L1}_p$)**. Therefore, we have

$$I_{\text{in}} \lesssim \int p(\mathbf{x}) \Big(\int_{\tau_m}^{\nu_m} \psi_{a,b}(u)\delta_{km}^{(p)}(u|\mathbf{x})\,\mathrm{d}u + \int_{\tilde{\tau}_n}^{\tilde{\nu}_n} \psi_{\tilde{a},\tilde{b}}(v)\delta_{ln}^{(q)}(v|\mathbf{x})\,\mathrm{d}v \Big)\,\mathrm{d}\mathbf{x},$$

and we can now apply the generic inner bias bounds in Lemma 4.B.23 to bound the inner bias.

Step 2: Bounding the outer bias. We first consider the upper bound of $I_{\text{out}}(\mathbf{x})$ in (4.88). For the first integral, we have

$$
\begin{aligned}
&\int_{\mathbb{R}\setminus(\tau_m,\nu_m)} \int_{\tilde{\tau}_n}^{\tilde{\nu}_n} \{\rho_{U_{k\infty}(\mathbf{x})}(u)\rho_{V_{l\infty}(\mathbf{x})}(v) + \rho_{U_{k,m-1}(\mathbf{x})}(u)\rho_{V_{ln}(\mathbf{x})}(v)\}\psi_{a,b}(u)\psi_{\tilde{a},\tilde{b}}(v)\,\mathrm{d}u\,\mathrm{d}v \\
&= \mathbb{E}[\overline{\overline{\psi}}(U_{k\infty}(\mathbf{x});\tau_m,\nu_m) + \overline{\overline{\psi}}(U_{k,m-1}(\mathbf{x});\tau_m,\nu_m)]\mathbb{E}[\overline{\psi}(V_{l\infty}(\mathbf{x});\tilde{\tau}_n,\tilde{\nu}_n) + \overline{\psi}(V_{ln}(\mathbf{x});\tilde{\tau}_n,\tilde{\nu}_n)] \\
&\overset{(b)}{\lesssim} \mathbb{E}[\overline{\overline{\psi}}(U_{k\infty}(\mathbf{x});\tau_m,\nu_m) + \overline{\overline{\psi}}(U_{k,m-1}(\mathbf{x});\tau_m,\nu_m)],
\end{aligned}
$$

where (b) follows from Lemmas 4.B.22 and 4.B.20. The second integral can be bounded similarly. Overall, we have

$$
\begin{aligned}
I_{\text{out}} \lesssim &\int p(\mathbf{x})\mathbb{E}[\overline{\overline{\psi}}(U_{k\infty}(\mathbf{x});\tau_m,\nu_m) + \overline{\overline{\psi}}(U_{k,m-1}(\mathbf{x});\tau_m,\nu_m)]\,\mathrm{d}\mathbf{x} \\
&+ \int p(\mathbf{x})\mathbb{E}[\overline{\overline{\psi}}(V_{l\infty}(\mathbf{x});\tilde{\tau}_n,\tilde{\nu}_n) + \overline{\overline{\psi}}(V_{ln}(\mathbf{x});\tilde{\tau}_n,\tilde{\nu}_n)]\,\mathrm{d}\mathbf{x},
\end{aligned}
$$

and we can now apply the generic outer bias bounds in Lemma 4.B.25.

Step 3: Choosing break points. Putting the bounds on the inner and outer bias together and choosing the break points $(\tau_m, \nu_m, \tilde{\tau}_n, \tilde{\nu}_n)$ as in the proof of Theorem 4.4.12, we obtain the desired bias rates. □

4.C.5 Proof of Theorem 4.4.13

By Lemma 4.B.30, we have

$$
\begin{aligned}
\mathrm{Var}(T_f^{(kl)}(\mathbf{X}_{1:m},\mathbf{Y}_{1:n})) \leq \frac{2(1+k\gamma_d)}{m}\{&(2k-2)\mathbb{E}[\phi_{kl}^2(U_{k-1,m-1}(\mathbf{X}_m),V_{ln}(\mathbf{X}_m))] \\
&+ (2k+1)\mathbb{E}[\phi_{kl}^2(U_{k,m-1}(\mathbf{X}_m),V_{ln}(\mathbf{X}_m))] \\
&+ \mathbb{E}[\phi_{kl}^2(U_{k+1,m-1}(\mathbf{X}_m),V_{ln}(\mathbf{X}_m))]\}.
\end{aligned}
$$

Using Lemma 4.B.20, we have

$$
\begin{aligned}
\mathbb{E}[\phi_{kl}^2(U_{k',m-1}(\mathbf{X}_m),V_{ln}(\mathbf{X}_m))] &= \int p(\mathbf{x})\mathbb{E}[\phi_{kl}^2(U_{k',m-1}(\mathbf{x}),V_{ln}(\mathbf{x}))]\,\mathrm{d}\mathbf{x} \\
&\lesssim \int p(\mathbf{x})\mathbb{E}[\psi_{a,b}^2(U_{k',m-1}(\mathbf{x}))]\mathbb{E}[\psi_{\tilde{a},\tilde{b}}^2(V_{ln}(\mathbf{x}))]\,\mathrm{d}\mathbf{x}
\end{aligned}
$$

$$\lesssim 1$$

for

$$k \in \begin{cases} \{1,2\} & \text{if } k = 1, \\ \{k-1, k, k+1\} & \text{if } k \geq 2, \end{cases}$$

and for m and n sufficiently large, which concludes the proof. □

4.D Deferred Proofs of Auxiliary Results

4.D.1 Proof of Proposition 4.3.27

Similar to Lemmas 4.B.23 and 4.B.25, we establish the following bounds.

Lemma 4.D.1 (Generic inner bias bound under **($\mathbf{S}'_p$)**). *Suppose that the density p satisfies the conditions **($\mathbf{U}_p$)** and **($\mathbf{S}'_p$)** , and let $k = o(\sqrt{m})$ as $m \to \infty$.*

1. *We have*

$$I_{in,1} = O\Big(\frac{\tau_m^{(a+\frac{\sigma_p}{d}+1)\wedge 0}}{m^{\frac{\sigma_p}{d}}} + \frac{k^{-k}}{m}\Big).$$

2. *Suppose that $\phi_k(u)$ is differentiable at every $u > 0$ and $|\phi_k'(u)| \lesssim \psi_{a-1,b-1}(u)$. If $\nu_m = o(\sqrt{m})$ as $m \to \infty$, then we have*

$$I_{in,2} = O\Big(k\frac{\nu_m^{(b+\frac{\sigma_p}{d}+1)\vee 0}}{m^{\frac{\sigma_p}{d}}} + \frac{\nu_m^{(b+k+1)\vee 0}}{m}\Big).$$

Proof. To establish the second bound, we invoke Lemma 4.B.9 instead of Lemma 4.B.4; this helps us obtain a tighter bias bound by reducing the exponent of ν_m by at most 1, which comes at the cost of additional factors in k. Let

$$\Delta_{km}(u) := \big|\mathrm{P}_{U_{km}(\mathbf{x})}(u) - \mathrm{P}_{U_{k\infty}(\mathbf{x})}(u)\big|.$$

Since we assume that $\phi_k(u)$ is differentiable at any $u > 0$ and $|\phi_k'(u)| \lesssim \psi_{a-1,b-1}(u)$, integration by parts leads to

$$\begin{aligned} I_{\text{in},2}(\mathbf{x}) &= \left| [\phi_k(u)\Delta_{km}(u)]_1^{\nu_m} + \int_1^{\nu_m} \phi_k'(u)\Delta_{km}(u)\,\mathrm{d}u \right| \\ &\le |\phi_k(\nu_m)| \cdot \Delta_{km}(\nu_m) + |\phi_k(1)| \cdot \Delta_{km}(1) + \int_1^{\nu_m} |\phi_k'(u)| \cdot \Delta_{km}(u)\,\mathrm{d}u \\ &= \tilde{O}_{\sigma_p,L,d}\Big(k \frac{\nu_m^{(b+\frac{\sigma_p}{d}+1)\vee 0}}{m^{\frac{\sigma_p}{d}}} + \frac{\nu_m^{(k+b+1)\vee 0}}{m} \Big) \end{aligned}$$

for $\mathbf{x} \in \mathrm{supp}(p)$, establishing the second bound. □

Assuming **(L1$'_p$)** in place of **(L1$_p$)**, we obtain a different generic bound on the upper outer bias $I_{\text{out},2}$ than that of Lemma 4.B.25; see also Remark 4.B.26.

Lemma 4.D.2 (Generic outer bias bound under **(L1$'_p$)** and **(L4$_p$)**). *Suppose that the density p satisfies the conditions **(U$_p$)**, **(L1$'_p$)**, and **(L4$_p$)**, we have*

$$I_{out,2} = O(\nu_m^{b+k-1-\theta}).$$

For any density p satisfying the conditions **(U$_p$)**, **(L1$'_p$)**, **(L4$_p$)**, and **(S$'_p$)**, if $\nu_m = o(\sqrt{m})$ and k is fixed, we have the bias bound from Lemmas 4.D.1 and 4.D.2:

$$\begin{aligned} &\left| \mathbb{E}\big[\hat{T}_f^{(k)}\big] - T_f(p) \right| \\ &\lesssim_{\sigma_p,L,C_p,C_0,d,k} \frac{\tau_m^{(a+\frac{\sigma_p}{d}+1)\wedge 0}}{m^{\frac{\sigma_p}{d}}} + \frac{\nu_m^{(b+\frac{\sigma_p}{d}+1)\vee 0}}{m^{\frac{\sigma_p}{d}}} + \frac{\nu_m^{(b+k+1)\vee 0}}{m} + \tau_m^{k+a} + \nu_m^{b+k-1-\theta}. \end{aligned}$$

Since $\nu_m \to \infty$ as $m \to \infty$, we require $b + k - 1 - \theta < 0$ to guarantee that the bias vanishes in our analysis, which forces us to choose a fixed k.

We first choose τ_m. If $a + \frac{\sigma_p}{d} + 1 > 0$, we can take $\tau_m = O(m^{-\frac{\sigma_p}{d}\frac{1}{k+a}})$. Otherwise, we take $\tau_m = \Theta(m^{-\frac{\sigma_p}{d}\frac{1}{k-1-\frac{\sigma_p}{d}}})$ to make the first and the fourth terms decay with the same

$$\nu_m = \begin{cases} \Theta(m^{(\frac{\sigma_p}{d}\wedge 1)\frac{1}{\theta-k-b+1}}) & \text{if } k \le -b-1, b \le -\frac{\sigma_p}{d}-1, \\ \Theta(m^{\frac{\sigma_p}{d}\frac{1}{\theta-k+\frac{\sigma_p}{d}+2}}) & \text{if } k \le -b-1, b > -\frac{\sigma_p}{d}-1, \\ \Theta(m^{\frac{1}{\theta+2}}) & \text{if } k > -b-1, b \le -\frac{\sigma_p}{d}-1, \\ \Theta(m^{(\frac{\sigma_p}{d}\wedge 1)\frac{1}{\theta+2}}) & \text{if } k > -b-1, b > -\frac{\sigma_p}{d}-1 \end{cases} \tag{4.90}$$

---

speed. To summarize, we choose

$$\tau_m = \begin{cases} \Theta(m^{-\frac{\sigma_p}{d}\frac{1}{k-\frac{\sigma_p}{d}-1}}) & \text{if } a \le -\frac{\sigma_p}{d}-1, \\ O(m^{-\frac{\sigma_p}{d}\frac{1}{k+a}}) & \text{o.w.} \end{cases} \tag{4.89}$$

to bound the first and the fourth terms as

$$\frac{\tau_m^{(a+\frac{\sigma_p}{d}+1)\wedge 0}}{m^{\frac{\sigma_p}{d}}} + \tau_m^{k+a} = \begin{cases} O(m^{-\frac{\sigma_p}{d}\frac{k+a}{k-\frac{\sigma_p}{d}-1}}) & \text{if } a \le -\frac{\sigma_p}{d}-1, \\ O(m^{-\frac{\sigma_p}{d}}) & \text{o.w.} \end{cases}$$

Similarly, by choosing ν_m as defined in (4.90) with $\nu_m = o(\sqrt{m})$ as $m \to \infty$, we bound the second, third, and last terms as

$$\frac{1}{m^{\frac{\sigma_p}{d}}} + \frac{\nu_m^{(b+k+2)\vee 0}}{m} + \nu_m^{b+k-\theta-1} = O(m^{-\lambda_\nu}),$$

where λ_ν is as defined in (4.35). □

4.D.2 Proof of Proposition 4.5.1

For any density p satisfying the conditions **($\mathbf{U}_p$)**, **($\mathbf{L1}_p$)**, **($\mathbf{S}_p$)**, and **($\mathbf{B}_p$)**, if $\nu_m = o(\sqrt{m})$ and $k \to \infty$ with $k = o(\sqrt{m})$ as $m \to \infty$, we have the bias bound from Lemma 4.B.23:

$$\left|\mathbb{E}[\hat{T}_f^{(k)}] - T_f(p)\right| \lesssim_{\sigma_p, L, C_p, C_0, d} \frac{\tau_m^{(a+\frac{\sigma_p}{d}+1)\wedge 0}}{m^{\frac{\sigma_p}{d}}} + \frac{k^{-k}}{m} + \frac{\tau_m^{(a+1)\wedge 0}}{m^{\frac{1}{d}}}$$

$$+\frac{\nu_m^{(b+\frac{\sigma_p}{d}+2)\vee 0}}{m^{\frac{\sigma_p}{d}}}+k^{-k}\frac{\nu_m^{(b+k+2)\vee 0}}{m}+\frac{\nu_m^{(b+2)\vee 0+\frac{1}{d}}}{m^{\frac{1}{d}}}$$
$$+k^{-k}\tau_m^{k+a}+k^{(b\vee 0)}\nu_m^{b+k-1}e^{-c_p\nu_m}.$$

Setting $\nu_m = \Theta((\ln m)^{1+\delta})$ and $k = \Theta((\ln m)^{1+\delta'})$ for some $0 < \delta' < \delta$, the last term $k^{(b\vee 0)}\nu_m^{b+k-1}e^{-c_p\nu_m}$ decays faster than any polynomial rate, that is, for any $C > 0$,

$$(b \vee 0)\ln k + (b+k-1)\ln \nu_m - c_p\nu_m < -C\ln m$$

for m sufficiently large. With these choices of ν_m and k, the bias bound then can be simplified as

$$\left|\mathbb{E}[\hat{T}_f^{(k)}] - T_f(p)\right| = \tilde{O}_{\sigma_p,L,C_p,C_0,d}\Big(\frac{\tau_m^{(a+\frac{\sigma_p}{d}+1)\wedge 0}}{m^{\frac{\sigma_p}{d}}}+\frac{\tau_m^{(a+1)\wedge 0}}{m^{\frac{1}{d}}}+\frac{1}{m^{\frac{\sigma_p\wedge 1}{d}}}\Big).$$

By choosing

$$\tau_m = \tau'(m, a_k) \tag{4.91}$$
$$= \begin{cases} O((\text{poly}\ln m)^{-1}) & \text{if } a_k \le -1 \\ 0 & \text{if } a_k > -1, \end{cases}$$

we obtain

$$\left|\mathbb{E}[\hat{T}_f^{(k)}] - T_f(p)\right| = \tilde{O}_{\sigma_p,L,C_p,C_0,d}\big(m^{-\frac{\sigma_p\wedge 1}{d}}\big).$$

Now, we show that $\text{Var}(\hat{T}_f^{(k)}) = \tilde{O}(m^{-1})$ if $k = \Theta((\ln m)^{1+\delta})$ as $m \to \infty$ for some $\delta > 0$. Using Lemmas 4.B.15, 4.B.13, 4.B.21, and 4.B.18, if we choose ν_m and κ_m such that

$\nu_m/m \to 0$ and $\kappa_m/m \to \infty$ as $m \to \infty$, we have

$$\mathrm{Var}(\hat{T}_f^{(k)}) = O\Big(\frac{k^2}{m}\Big\{\frac{C_p^k}{k!} + \nu_m^{2b\vee 0} + (\nu_m^{2b} \vee \kappa_m^{2b}) e^{-\nu_m \eta_p c_p}\Big(\frac{eC_p\nu_m}{k}\Big)^k\Big\}\Big)$$

for m sufficiently large. Letting $\nu_m = (2b/(\eta_p c_p))(\ln m)^{1+\delta/2}$ and $\kappa_m = e^{(\ln m)^{1+\delta/4}}$ ensures that the bound is $\tilde{O}(m^{-1})$. □

4.E Derivation of Estimator Functions

In this section, we present derivations of some selected examples of estimator functions $\phi_{kl}(u, v)$ for some functions $f(p, q)$ in Table 4.1.2. Estimator functions $\phi_k(u)$ for the single-density case can be computed in a similar manner. In particular, we present the examples of KL divergence (Example 4.E.2), logarithmic α-divergences (Example 4.E.4), entropy difference (Example 4.E.6), reverse KL divergence (Example 4.E.7), polynomial functionals (Example 4.E.3), Le Cam distance (Example 4.E.5), and Jensen–Shannon divergence (Example 4.E.8).

We remark that as alluded to in the main text, the estimator function $\phi_{kl}(u, v)$ is a function of u/v if $f(p, q)$ is a function of q/p.

Proposition 4.E.1. *If $f(p, q)$ is a function of q/p, then there exists a function $\varphi_{kl}\colon \mathbb{R}_+ \to \mathbb{R}$ such that $\phi_{kl}(u, v) = \varphi_{kl}(u/v)$.*

Proof. Suppose that we can write $f(p, q) = g(q/p)$ for some function $g\colon \mathbb{R}_+ \to \mathbb{R}$. Recall that we have

$$\begin{aligned}\mathcal{L}\{u^{k-1}v^{l-1}\phi_{kl}(u, v)\}(p, q) &= \iint_{\mathbb{R}_+^2} u^{k-1}v^{l-1}e^{-pu}e^{-qv}\phi_{kl}(u, v)\,\mathrm{d}u\,\mathrm{d}v \\ &= \frac{\Gamma(k)\Gamma(l)}{p^k q^l} g\Big(\frac{q}{p}\Big).\end{aligned}$$

Now, for any $c > 0$, we consider

$$\begin{aligned}\mathcal{L}\{u^{k-1}v^{l-1}\phi_{kl}(cu,cv)\}(p,q) &= \iint_{\mathbb{R}_+^2} u^{k-1}v^{l-1}e^{-pu}e^{-qv}\phi_{kl}(cu,cv)\,\mathrm{d}u\,\mathrm{d}v \\ &= \frac{1}{c^{k+l}}\iint_{\mathbb{R}_+^2} \tilde{u}^{k-1}\tilde{v}^{l-1}e^{-p\tilde{u}/c}e^{-q\tilde{v}/c}\phi_{kl}(\tilde{u},\tilde{v})\,\mathrm{d}\tilde{u}\,\mathrm{d}\tilde{v} \\ &= \frac{1}{c^{k+l}}\cdot\frac{\Gamma(k)\Gamma(l)}{(p/c)^k(q/c)^l}g\Big(\frac{q/c}{p/c}\Big) \\ &= \frac{\Gamma(k)\Gamma(l)}{p^kq^l}g\Big(\frac{q}{p}\Big).\end{aligned}$$

Thus, by the (a.e.) uniqueness of Laplace transform, we have $\phi_{kl}(cu,cv) = \phi_{kl}(u,v)$, whence $\phi_{kl}(u,v)$ can be written as $\phi_{kl}(u,v) = \varphi_{kl}(u/v)$ for some function $\varphi\colon \mathbb{R}_+ \to \mathbb{R}$. $\square$

In what follows, for the one-dimensional inverse Laplace transform of two-variable functions, we will specify the transformed variable by a subscript of the inverse Laplace operator. For example, $\mathcal{L}_p^{-1}\{G(p,q)\}(u)$ denotes the inverse Laplace transform of $G(p,q)$ along the p-axis with a corresponding time-domain variable u.

Example 4.E.2 (KL divergence; Example 4.4.2)**.** *For $f(p,q) = \ln(p/q)$, the corresponding functional $T_f(p,q) = D(p \parallel q)$ is the KL divergence. This is one of the simplest cases, as we only need to deal with one-dimensional inverse Laplace transforms by linearity:*

$$\mathcal{L}^{-1}\Big\{\frac{1}{p^kq^l}\ln\frac{p}{q}\Big\} = \mathcal{L}^{-1}\Big\{\frac{\ln p}{p^k}\Big\}\mathcal{L}^{-1}\Big\{\frac{1}{q^l}\Big\} - \mathcal{L}^{-1}\Big\{\frac{1}{p^k}\Big\}\mathcal{L}^{-1}\Big\{\frac{\ln q}{q^l}\Big\}.$$

Note that for any $\kappa > 0$,

$$\mathcal{L}^{-1}\Big\{\frac{\ln p}{p^\kappa}\Big\} = \frac{u^{\kappa-1}}{\Gamma(\kappa)}\big(\Psi(\kappa) - \ln u\big). \tag{4.92}$$

This can be verified by taking Laplace transform of the right-hand expression. From the definition

of the estimator function $\phi_{kl}(u,v)$ in (4.9)*, we obtain*

$$\phi_{kl}(u,v) = \ln \frac{v}{u} + \Psi(k) - \Psi(l). \tag{4.93}$$

Example 4.E.3 (Polynomial functionals; Example 4.4.3)**.** *Consider $f(p,q) = p^{\alpha-1}q^{\beta}$ for some $\alpha, \beta \in \mathbb{R}$, which corresponds to the functional*

$$T_f(p,q) = \mathbb{E}\big[p^{\alpha-1}(\mathbf{X})q^{\beta}(\mathbf{X})\big] = \int p^{\alpha}(\mathbf{x})q^{\beta}(\mathbf{x})\,\mathrm{d}\mathbf{x}.$$

This includes many special cases such as Rényi entropies, Rényi divergences, Hellinger distance, and χ^2-divergence. The estimator function is

$$\phi_{kl}(u,v) = \frac{\Gamma(k)\Gamma(l)}{\Gamma(k-\alpha+1)\Gamma(l-\beta)} u^{1-\alpha} v^{-\beta}$$

for $k > \alpha - 1$ and $l > \beta$. We remark that our estimator recovers the bias-corrected estimator presented in (Póczos et al., 2012).

Example 4.E.4 (Logarithmic α-divergence; Example 4.4.4)**.** *For $\alpha \in \mathbb{R}$, consider a function $f(p,q) = (p/q)^{\alpha-1} \ln \frac{p}{q}$, which corresponds to the functional*

$$\begin{aligned} T_f(p,q) &= \mathbb{E}\Big[\Big(\frac{p(\mathbf{X})}{q(\mathbf{X})}\Big)^{\alpha-1} \ln \frac{p(\mathbf{X})}{q(\mathbf{X})}\Big] \\ &= \int p^{\alpha}(\mathbf{x}) q^{1-\alpha}(\mathbf{x}) \ln \frac{p(\mathbf{x})}{q(\mathbf{x})}\,\mathrm{d}\mathbf{x}. \end{aligned}$$

Similar to KL divergence, the estimator function can be found immediately from (4.92)*, i.e.,*

$$\phi_{kl}(u,v) = \frac{\Gamma(k)\Gamma(l)}{\Gamma(k-\alpha+1)\Gamma(l+\alpha-1)} \Big(\frac{v}{u}\Big)^{\alpha-1} \Big(\ln \frac{v}{u} + \Psi(k-\alpha+1) - \Psi(l+\alpha-1)\Big),$$

for $k > \alpha - 1$ and $l > -\alpha + 1$. Note that $\alpha = 1$ recovers the estimator function for the KL divergence (4.93)*.*

Example 4.E.5 (Le Cam distance; Example 4.4.5). *For $f(p,q) = 1 - 2q/(p+q)$, we wish to compute the estimator function $\phi_{kl}(u,v)$, that is,*

$$\phi_{kl}(u,v) = 2\frac{\Gamma(k)\Gamma(l)}{u^{k-1}v^{l-1}}\mathcal{L}^{-1}\Big\{\frac{1}{p^k q^l}\frac{1}{1+\frac{q}{p}}\Big\} - 1.$$

The two-dimensional inverse Laplace transform can be peeled off dimension by dimension as follows:

$$\mathcal{L}_{p,q}^{-1}\Big\{\frac{1}{p^k q^l}\frac{1}{1+\frac{q}{p}}\Big\}(u,v) = \mathcal{L}_p^{-1}\Big\{\frac{1}{p^{k+l}}\mathcal{L}_q^{-1}\Big\{\frac{1}{(\frac{q}{p})^l(1+\frac{q}{p})}\Big\}(v)\Big\}(u). \tag{4.94}$$

Letting $\tilde{q} = q/p$, we first find the inverse Laplace transform of

$$\frac{1}{\tilde{q}^l(1+\tilde{q})} = (-1)^l\Big(\sum_{i=1}^{l}\frac{(-1)^i}{\tilde{q}^i} + \frac{1}{1+\tilde{q}}\Big), \tag{4.95}$$

which is

$$\mathcal{L}_{\tilde{q}}^{-1}\Big\{\frac{1}{\tilde{q}^l(1+\tilde{q})}\Big\}(v) = (-1)^l\Big(e^{-v} - \sum_{i=0}^{l-1}\frac{(-v)^i}{i!}\Big),$$

since we have

$$\mathcal{L}_p^{-1}\Big\{\frac{1}{p^{n+1}}\Big\}(u) = \frac{u^n}{n!}\,1_{[0,\infty)}(u)$$

for $n \in \mathbb{N} \cup \{0\}$ and

$$\mathcal{L}_p^{-1}\Big\{\frac{1}{s+a}\Big\}(u) = e^{-au}\,1_{[0,\infty)}(u).$$

$$
\begin{aligned}
&\mathcal{L}_{p,q}^{-1}\Big\{\frac{1}{p^k q^l}\frac{1}{1+\frac{q}{p}}\Big\}(u,v)\\
&= \mathcal{L}_p^{-1}\Big\{(-1)^l\Big(\frac{e^{-pv}}{p^{k+l-1}} - \sum_{i=0}^{l-1}\frac{(-v)^i}{i!}\frac{1}{p^{k+l-i-1}}\Big)\Big\}(u)\\
&= (-1)^l\Big(\frac{(u-v)^{k+l-2}}{(k+l-2)!}\,1_{[v,\infty)}(u) - \sum_{i=0}^{l-1}\frac{(-v)^i}{i!}\frac{u^{k+l-i-2}}{(k+l-i-2)!}\Big)\\
&= (-1)^l\frac{u^{k+l-2}}{(k+l-2)!}\Big(\Big(1-\frac{v}{u}\Big)^{k+l-2}1_{[v,\infty)}(u) - \sum_{i=0}^{l-1}\binom{k+l-2}{i}\Big(\frac{-v}{u}\Big)^i\Big). \qquad (4.96)
\end{aligned}
$$

$$
\begin{aligned}
\phi_{kl}(u,v) = 2\binom{k+l-2}{k-1}^{-1}\Big(-\frac{u}{v}\Big)^{l-1}\times\\
\Big(\sum_{i=0}^{l-1}\binom{k+l-2}{i}\Big(-\frac{v}{u}\Big)^i - \Big(1-\frac{v}{u}\Big)^{k+l-2}1_{[v,\infty)}(u)\Big) - 1. \qquad (4.97)
\end{aligned}
$$

Moreover, by the time-scaling property, we have

$$
\mathcal{L}_q^{-1}\Big\{\frac{1}{(\frac{q}{p})^l(1+\frac{q}{p})}\Big\}(v) = (-1)^l\Big(pe^{-pv} - \sum_{i=0}^{l-1}\frac{(-v)^i}{i!}p^{i+1}\Big).
$$

Now, continuing from (4.94)*, we have* (4.96)*, which leads to the estimator function* (4.97)*. As a bound on the estimator function $\phi_{kl}(u,v)$, we observe that*

$$
\begin{aligned}
|\phi_{kl}(u,v)| &\lesssim \Big(\frac{u}{v}\Big)^{l-1}\Big(\sum_{i=0}^{l-1}\Big(\frac{v}{u}\Big)^i + \sum_{j=0}^{k+l-2}\Big(\frac{v}{u}\Big)^j\Big)\\
&\lesssim \psi_{-k+1,l-1}(u)\psi_{-l+1,k-1}(v).
\end{aligned}
$$

For the remaining examples, we assume that $\mathsf{Q} \ll \mathsf{P}$.

Example 4.E.6 (Entropy difference)**.** *For $f(p,q) = \ln(1/p) - (q/p)\ln(1/q)$, the corresponding functional $T_f(p,q) = h(p) - h(q)$ becomes the difference of the differential entropies $h(p)$ and*

$h(q)$. It is easy to show that

$$\phi_{kl}(u,v) = \frac{(l-1)}{k}\frac{u}{v}(\Psi(l-1) - \ln v) - (\Psi(k) - \ln u).$$

As a bound on the estimator function $\phi_{kl}(u,v)$, we have

$$\begin{aligned}
|\phi_{kl}(u,v)| &\lesssim \frac{u}{v}(1 + |\ln v|) + (1 + |\ln u|) \\
&\lesssim \psi_{1,1}(u)\psi_{-1-\epsilon,-1+\epsilon}(v) + \psi_{-\epsilon,\epsilon}(u) \\
&\lesssim \psi_{-\epsilon,1}(u)\psi_{-1-\epsilon,-1+\epsilon}(v).
\end{aligned}$$

Example 4.E.7 (Reverse KL divergence)**.** *When $Q \ll P$, we can write the reverse KL divergence as*

$$\begin{aligned}
D(q \parallel p) &= \int q(\mathbf{x}) \ln \frac{q(\mathbf{x})}{p(\mathbf{x})} \, \mathrm{d}\mathbf{x} \\
&= \int p(\mathbf{x}) \frac{q(\mathbf{x})}{p(\mathbf{x})} \ln \frac{q(\mathbf{x})}{p(\mathbf{x})} \, \mathrm{d}\mathbf{x} = T_f(p,q)
\end{aligned}$$

for $f(p,q) = (q/p)\ln(q/p)$. Then, for $k \geq 1$ and $l \geq 2$, we have

$$\begin{aligned}
&\mathcal{L}^{-1}\Big\{\frac{f(p,q)}{p^k q^l}\Big\} \\
&= \mathcal{L}^{-1}\Big\{\frac{1}{p^{k+1}}\Big\}\mathcal{L}_q^{-1}\Big\{\frac{\ln q}{q^{l-1}}\Big\} - \mathcal{L}^{-1}\Big\{\frac{\ln p}{p^{k+1}}\Big\}\mathcal{L}_q^{-1}\Big\{\frac{1}{q^{l-1}}\Big\} \\
&= \frac{u^k}{\Gamma(k+1)}\frac{v^{l-2}}{\Gamma(l-1)}\big(\Psi(l-1) - \ln v\big) - \frac{u^k}{\Gamma(k+1)}\big(\Psi(k+1) - \ln u\big)\frac{v^{l-2}}{\Gamma(l-1)}.
\end{aligned}$$

Here, the case $l = 1$ is excluded, since $\mathcal{L}^{-1}\{\ln s\}$ is ill-defined. Finally, we have

$$\begin{aligned}
\phi_{kl}(u,v) &= \frac{\Gamma(k)\Gamma(l)}{u^{k-1}v^{l-1}}\frac{u^k}{\Gamma(k+1)}\frac{v^{l-2}}{\Gamma(l-1)}\big\{\big(\Psi(l-1) - \ln v\big) - \big(\Psi(k+1) - \ln u\big)\big\} \\
&= \frac{l-1}{k}\frac{u}{v}\big(\ln\frac{u}{v} + \Psi(l-1) - \Psi(k+1)\big).
\end{aligned}$$

Table 4.E.1. Inverse Laplace transforms of few elementary functions and basic operations.

Frequency domain $F(p) = \mathcal{L}\{f(u)\}$	Time domain $f(u) = \mathcal{L}^{-1}\{F(p)\}$
p^{-k} $(k > 0)$	$u^{k-1}/\Gamma(k)$
$\ln p/p$	$-(\ln u + \gamma)$
$1/(p+\alpha)$	$e^{-\alpha u}$
$F(ap)$	$f\big(u/a\big)/a$
$e^{-ap}F(p)$	$f(u-a)\,1_{[a,\infty)}(u)$
$F^{(n)}(p)$	$(-1)^n u^n f(u)$
$F(p)/p$	$\int_0^u f(t)\,\mathrm{d}t$
$F(p)G(p)$	$(f*g)(u) = \int_0^u f(\tilde{u})g(u-\tilde{u})\,\mathrm{d}\tilde{u}$
$pF(p)$	$f'(u) - f(0)$

As a bound on the estimator function $\phi_{kl}(u,v)$, we have

$$\begin{aligned}|\phi_{kl}(u,v)| &\lesssim \frac{u}{v}(1 + |\ln u| + |\ln v|)\\ &\lesssim \frac{u}{v}(1 + |\ln u|)(1 + |\ln v|)\\ &\lesssim \psi_{1-\epsilon,1+\epsilon}(u)\psi_{-1-\epsilon,-1+\epsilon}(v).\end{aligned}$$

Example 4.E.8 (Jensen–Shannon divergence; Example 4.4.6)**.** *We wish to compute the estimator function $\phi_{kl}(u,v)$ for*

$$f(p,q) = \frac{1}{2}\Big(\frac{q}{p} + 1\Big)\ln\frac{2}{(q/p)+1} + \frac{q}{2p}\ln\frac{q}{p}.$$

For $l \geq 2$, we have

$$\frac{2f(p,q)}{p^k q^l} = \Big(\frac{1}{p^{k+1}q^{l-1}} + \frac{1}{p^k q^l}\Big)\ln 2 + \frac{1}{p^{k+1}q^{l-1}}\ln\frac{q}{p} + \frac{G_{l-1}(\frac{q}{p}) + G_l(\frac{q}{p})}{p^{k+l}},$$

where we define $G_l(q) := -\ln(q+1)/q^l$. *Using the identity* (4.95)*, we can show that for* $l \in \mathbb{N}$

$$\begin{aligned} g_l(v) &= \mathcal{L}_q^{-1}\{G_l(q)\}(v) \\ &= (-1)^l\Big(\int_1^\infty \frac{e^{-vx}}{x^l}\,\mathrm{d}x - \sum_{j=0}^{l-2}\frac{(-v)^j}{(l-1-j)j!}\Big). \end{aligned}$$

Now the desired estimator function can be written as

$$\begin{aligned} 2\phi_{kl}(u,v) &= \frac{\Gamma(k)\Gamma(l)}{u^{k-1}v^{l-1}}\mathcal{L}^{-1}\Big\{\frac{f(p,q)}{p^k q^l}\Big\}(u,v) \\ &= \frac{l-1}{k}\frac{u}{v}\Big(\Psi(l-1) - \Psi(k+1) + \ln\frac{u}{v}\Big) + \Big(\frac{l-1}{k}\frac{u}{v}+1\Big)\ln 2 + A_{kl}(u,v), \end{aligned} \tag{4.98}$$

where we define

$$\begin{aligned} A_{kl}(u,v) &= \frac{\Gamma(k)\Gamma(l)}{u^{k-1}v^{l-1}}\mathcal{L}_p^{-1}\Big\{\frac{\mathcal{L}_q^{-1}\{G_{l-1}(\frac{q}{p}) + G_l(\frac{q}{p})\}(v)}{p^{k+l}}\Big\}(u) \\ &\overset{(a)}{=} \frac{\Gamma(k)\Gamma(l)}{u^{k-1}v^{l-1}}\mathcal{L}_p^{-1}\Big\{\frac{g_{l-1}(pv) + g_l(pv)}{p^{k+l-1}}\Big\}(u) \\ &= B_{kl}(u,v) + \frac{l-1}{k}\frac{u}{v}B_{k+1,l-1}(u,v), \end{aligned} \tag{4.99}$$

where

$$B_{kl}(u,v) = \frac{\Gamma(k)\Gamma(l)}{u^{k-1}v^{l-1}}\mathcal{L}_p^{-1}\Big\{\frac{g_l(pv)}{p^{k+l-1}}\Big\}(u).$$

Here, (a) follows by the time scaling property, that is, $\mathcal{L}_q^{-1}\{G_l(q/p)\}(v) = pg_l(pv)$. *Now, since we have* (4.100)*, it follows that*

$$\begin{aligned} &\binom{k+l-2}{k-1}B_{kl}(u,v) \\ &= -1_{[1,\infty)}(w)(-w)^{-k+1}\int_1^w \frac{(x-w)^{k+l-2}}{x^l}\,\mathrm{d}x + \sum_{j=0}^{l-2}\binom{k+l-2}{j}\frac{(-w)^{l-1-j}}{l-1-j}, \end{aligned}$$

$$
\begin{aligned}
&\mathcal{L}_p^{-1}\Big\{\frac{g_l(pv)}{p^{k+l-1}}\Big\} \\
&= \int_1^\infty \frac{1}{x^l}\mathcal{L}_p^{-1}\Big\{\frac{e^{-pvx}}{p^{k+l-1}}\Big\}\,\mathrm{d}x - \sum_{j=0}^{l-2}\frac{(-v)^j}{(l-1-j)j!}\mathcal{L}_p^{-1}\Big\{\frac{1}{p^{k+l-1-j}}\Big\} \\
&= \int_1^\infty \frac{1}{x^l}\,1_{[vx,\infty)}(u)\frac{(u-vx)^{k+l-2}}{(k+l-2)!}\,\mathrm{d}x - \sum_{j=0}^{l-2}\frac{(-v)^j}{(l-1-j)j!}\frac{u^{k+l-2-j}}{(k+l-2-j)!},
\end{aligned}
\tag{4.100}
$$

---

where $w := u/v$.

Rearranging the integral in the parenthesis as

$$
\begin{aligned}
&(-w)^{k+1}\int_1^w \frac{(x-w)^{k+l-2}}{x^l}\,\mathrm{d}x \\
&= \sum_{\substack{i=0\\ i\neq k-1}}^{k+l-2}\binom{k+l-2}{i}\frac{(-1)^{k-1-i}-(-w^{-1})^{k-1-i}}{k-1-i} + \binom{k+l-2}{k-1}\ln w,
\end{aligned}
$$

we finally obtain

$$
B_{kl}(u,v) = \binom{k+l-2}{k-1}^{-1}\sum_{j=0}^{l-2}\binom{k+l-2}{j}\frac{(-u/v)^{l-1-j}}{l-1-j} \tag{4.101}
$$

if $\frac{u}{v} < 1$, *and*

$$
\begin{aligned}
B_{kl}(u,v) = &-\ln\frac{u}{v} + \binom{k+l-2}{k-1}^{-1} \\
&\times\Big\{\sum_{i=0}^{k-2}\binom{k+l-2}{i}\frac{(-v/u)^{k-1-i}}{k-1-i} - \sum_{\substack{i=0\\ i\neq k-1}}^{k+l-2}\binom{k+l-2}{i}\frac{(-1)^{k-1-i}}{k-1-i}\Big\}
\end{aligned}
\tag{4.102}
$$

if $\frac{u}{v} \geq 1$. *Substituting the expressions for* $B_{kl}(u,v)$ *from* (4.101) *and* (4.102) *into* (4.99) *and then into* (4.98) *yields the final expression for the estimator function as*

$$
\phi_{kl}(u,v) = \frac{1}{2}\Big\{\ln 2 + \frac{l-1}{k}\frac{u}{v}\Big(\ln 2 + \Psi(l-1) - \Psi(k+1) + \ln\frac{u}{v}\Big)
$$

$$+ B_{kl}(u,v) + \frac{l-1}{k}\frac{u}{v}B_{k+1,l-1}(u,v)\Big\}.$$

As a bound on the estimator function $\phi_{kl}(u,v)$, we have

$$|\phi_{kl}(u,v)| \lesssim \psi_{-k+1,l-1}(u)\psi_{-l+1,k-1}(v).$$

4.F Examples of Smooth Densities

In this section, we show that the d-dimensional truncated Gaussian, Cauchy, and exponential distributions, as well as the uniform distribution and the d-dimensional product of identical beta distributions with parameters $\alpha \geq 3$ and $\beta \geq 3$ satisfy the conditions **(U$_p$)**, **(L1$_p$)**, **(S$_p$)**, and **(B$_p$)** with $\sigma_p = 2$, and the d-dimensional truncated Laplace distribution satisfies the conditions with $\sigma_p = 1$. We remark that the boundedness of the Hessian of the density p over a compact set implies 2-Hölder continuity, if the Hessian is integrable. Since we have considered that the Hessian is integrable, we only need to prove the boundedness of the Hessian in order to demonstrate the 2-Hölder continuity.

Example 4.F.1 (Truncated Gaussian)**.** *Consider the truncated d-dimensional Gaussian distribution defined by the density*

$$p(\mathbf{x}) := \frac{\Gamma(d/2+1)}{\pi^{d/2}K_d(R)}e^{-\|\mathbf{x}\|_2^2/2}\,1_{(-\infty,R]}(\|\mathbf{x}\|_2),$$

where $K_d(R) := \int_0^R dr^{d-1}e^{-r^2/2}\,\mathrm{d}r$. Then, $\mathrm{supp}(p) = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| \leq R\}$ *and*

$$\frac{\Gamma(d/2+1)}{\pi^{d/2}K_d(R)}e^{-R^2/2} \leq p(\mathbf{x}) \leq \frac{\Gamma(d/2+1)}{\pi^{d/2}K_d(R)}$$

for $\mathbf{x} \in \mathrm{supp}(p)$*. Moreover, on* $\mathrm{supp}(p)^{\mathrm{o}}$*,*

$$\nabla^2 p(\mathbf{x})_{ij} = \frac{\Gamma(d/2+1)}{\pi^{d/2}K_d(R)}(x_ix_j - \delta_{ij})e^{-\|\mathbf{x}\|_2^2/2},$$

whence,

$$\|\nabla^2 p(\mathbf{x})\| \leq \|\nabla^2 p(\mathbf{x})\|_F \leq \frac{\Gamma(d/2+1)}{\pi^{d/2} K_d(R)} \sqrt{R^4 + d}.$$

Finally, $\partial\mathrm{supp}(p) = \mathbb{S}(\mathbf{0}, R)$ *satisfies*

$$H^{d-1}(\mathbb{S}(\mathbf{0}, R)) = d\upsilon_d R^{d-1}.$$

Therefore, this density satisfies the conditions ***(U_p)***, ***($L1_p$)***, ***(S_p)***, *and* ***(B_p)*** *with* $\sigma_p = 2$ *and*

$$\begin{aligned}
\sup_{\mathbf{x}} p(\mathbf{x}) &= \frac{\Gamma(d/2+1)}{\pi^{d/2} K_d(R)}, \\
L(p; \mathrm{supp}(p)^{\mathrm{o}}) &= \frac{\Gamma(d/2+1)}{\pi^{d/2} K_d(R)} \sqrt{R^4 + d}, \\
H^{d-1}(\partial\mathrm{supp}(p)) &= d\upsilon_d R^{d-1}.
\end{aligned}$$

Example 4.F.2 (Truncated exponential)**.** *Let* $S_R := \{\mathbf{x} \in \mathbb{R}^d \colon x_1, \ldots, x_d \geq 0, x_1 + \ldots + x_d \leq R\}$*. The truncated* d*-dimensional exponential distribution defined by the density*

$$p(\mathbf{x}) := \frac{e^{-(x_1 + \cdots + x_d)}}{1 - \big(\sum_{i=0}^{d-1} \frac{R^i}{i!}\big) e^{-R}} \, 1_{S_R}(\mathbf{x})$$

is 2*-Hölder continuous over* $\mathrm{supp}(p)$ *and satisfies*

$$\begin{aligned}
\sup_{\mathbf{x}} p(\mathbf{x}) &= \Big(1 - \Big(\sum_{i=0}^{d-1} \frac{R^i}{i!}\Big) e^{-R}\Big)^{-1}, \\
L(p; \mathrm{supp}(p)^{\mathrm{o}}) &= d \sup_{\mathbf{x}} p(\mathbf{x}),
\end{aligned}$$

and

$$H^{d-1}(\partial\mathrm{supp}(p)) = \Big(\frac{\sqrt{d}}{(d-1)!} + d\Big) R^{d-1},$$

as can be seen by an analysis similar to that in the previous example.

Example 4.F.3 (Truncated Laplace)**.** *Consider the truncated d-dimensional Laplace distribution defined by the density*

$$p(\mathbf{x}) := \frac{e^{-(|x_1|+\cdots+|x_d|)}}{2^d\big(1-\big(\sum_{i=0}^{d-1}\frac{R^i}{i!}\big)e^{-R}\big)}\,1_{(-\infty,R]}(\|\mathbf{x}\|_1).$$

*Then, **($\mathbf{U}_p$)**, **($\mathbf{L1}_p$)**, and **($\mathbf{B}_p$)** can be demonstrated similarly to the previous examples. For **($\mathbf{S}_p$)**, note that for $x, y \in \mathbb{R}$,*

$$\big|e^{-|x|} - e^{-|y|}\big| \le |x-y|.$$

Generalizing this to d dimensions, we have

$$\big|e^{-(|x_1|+\cdots+|x_d|)} - e^{-(|y_1|+\cdots+|y_d|)}\big| \le \|\mathbf{x}-\mathbf{y}\|_1 \le \sqrt{d}\|\mathbf{x}-\mathbf{y}\|_2.$$

Therefore, the truncated d-dimensional Laplace distribution is 1-Hölder continuous over $\operatorname{supp}(p)$ and satisfies

$$\sup_{\mathbf{x}} p(\mathbf{x}) = \Big(2^d\Big(1-\Big(\sum_{i=0}^{d-1}\frac{R^i}{i!}\Big)e^{-R}\Big)\Big)^{-1},$$
$$L(p;\operatorname{supp}(p)^{\circ}) = \sqrt{d}\sup_{\mathbf{x}} p(\mathbf{x}),$$

and

$$H^{d-1}(\partial\operatorname{supp}(p)) = \frac{2^d\sqrt{d}}{(d-1)!}R^{d-1}.$$

Example 4.F.4 (Truncated Cauchy)**.** *Consider the truncated d-dimensional Cauchy distribution defined by the density*

$$p(\mathbf{x}) := \frac{\Gamma\big((d+1)/2\big)}{\pi^{(d+1)/2}L_d(R)\big(1+\|\mathbf{x}\|_2^2\big)^{(d+1)/2}}\,1_{(-\infty,R]}(\|\mathbf{x}\|_2),$$

where

$$L_d(R) := \frac{\int_0^{\arctan R} \sin^{d-1}\theta \,\mathrm{d}\theta}{\int_0^{\pi/2} \sin^{d-1}\theta \,\mathrm{d}\theta} \in [0,1].$$

Then, we have

$$\nabla^2 p(\mathbf{x})_{ij} = \frac{(d+1)\Gamma((d+1)/2)}{\pi^{(d+1)/2} L_d(R) \big(1+\|\mathbf{x}\|_2^2\big)^{(d+5)/2}} \big((d+3)x_i x_j - \big(1+\|\mathbf{x}\|_2^2\big)\delta_{ij}\big),$$

which leads to the bound

$$\big\|\nabla^2 p(\mathbf{x})\big\| \leq \frac{(d+1)\Gamma((d+1)/2)}{\pi^{(d+1)/2} L_d(R)} \sqrt{R^4(d+1)(d+3)+d}$$

on $\mathrm{supp}(p)^\circ$*. Therefore, the truncated* d*-dimensional Cauchy distribution is* 2*-Hölder continuous over* $\mathrm{supp}(p)$ *and satisfies*

$$\begin{aligned} \sup_{\mathbf{x}} p(\mathbf{x}) &= \frac{\Gamma\big((d+1)/2\big)}{\pi^{(d+1)/2} L_d(R)}, \\ L(p; \mathrm{supp}(p)^\circ) &= \frac{(d+1)\Gamma((d+1)/2)}{\pi^{(d+1)/2} L_d(R)} \sqrt{R^4(d+1)(d+3)+d}, \end{aligned}$$

and

$$H^{d-1}(\partial \mathrm{supp}(p)) = d\upsilon_d R^{d-1}.$$

Acknowledgement

Chapter 4, in part, is a reprint of the material in the paper with permission: © 2022 IEEE. J. Jon Ryu, Shouvik Ganguly, Young-Han Kim, Yung-Kyun Noh, Daniel Lee, "Nearest neighbor density functional estimation from inverse Laplace transform," *IEEE Transactions on Information Theory*, vol. 68, Issue 6, pp. 3511–3551, June 2022. The dissertation author was the primary investigator and author of this paper. This work was supported in part by the National Science Foundation under Grant CCF-1911238.

Bibliography

Mehdi Aghagolzadeh, Hamid Soltanian-Zadeh, Babak Araabi, and Ali Aghagolzadeh. A hierarchical clustering based on mutual information maximization. In *Proc. IEEE Int. Conf. Image Proc.*, volume 1, 2007.

Jean-Yves Audibert, Alexandre B Tsybakov, et al. Fast learning rates for plug-in classifiers. *Ann. Statist.*, 35(2):608–633, 2007.

Thomas B Berrett and Richard J Samworth. Efficient two-sample functional estimation and the super-oracle phenomenon. *arXiv preprint arXiv:1904.09347*, 2019.

Thomas B Berrett, Richard J Samworth, and Ming Yuan. Efficient multivariate entropy estimation via k-nearest neighbour distances. *Ann. Statist.*, 47(1):288–318, 2019.

Gérard Biau and Luc Devroye. *Lectures on the Nearest Neighbor Method*. Springer International Publishing, 2015.

Lucien Birge and Pascal Massart. Estimation of integrals functionals of a density. *Ann. Statist.*, 23(1):11–29, 1995.

Vivek S Borkar. *Probability theory: an advanced course*. Springer Science & Business Media, 1995.

Zois Boukouvalas, Rami Mowakeaa, Geng-Shen Fu, and Tulay Adali. Independent Component Analysis by Entropy Maximization with Kernels. *arXiv preprint arXiv:1610.07104*, 2016.

Alexander Bulinski and Denis Dimitrov. Statistical estimation of the shannon entropy. *Acta Mathematica Sinica, English Series*, 35(1):17–46, 2019a.

Alexander Bulinski and Denis Dimitrov. Statistical estimation of the Kullback–Leibler divergence. *arXiv preprint arXiv:1907.00196*, 2019b.

Andrzej Cichocki, Hyekyoung Lee, Yong-Deok Kim, and Seungjin Choi. Non-negative matrix factorization with α-divergence. *Pattern Recogni. Letters*, 29(9):1433–1440, 2008.

Alan M Cohen. *Numerical Methods for Laplace Transform Inversion*, volume 5. Springer Science & Business Media, 2007.

Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 2009.

Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory*, 13(1):21–27, 1967.

Przeniyslaw Crzcgorzewski and Robert Wirczorkowski. Entropy-based goodness-of-fit test for exponentiality. *Commun. Statist. Theory Methods*, 28(5):1183–1202, 1999.

Sylvain Delattre and Nicolas Fournier. On the Kozachenko–Leonenko entropy estimator. *J. Statist. Plan. Inference*, 185:69–93, 2017.

Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 2013.

Roland L'vovich Dobrushin. A simplified method of experimentally evaluating the entropy of a stationary sequence. *Theory of Probability & Its Applications*, 3(4):428–430, 1958.

B Efron and C Stein. The Jackknife Estimate of Variance. *Ann. Statist.*, 9(3):586–596, 1981.

Lawrence C Evans and Ronald F Gariepy. *Measure Theory and Fine Properties of Functions*. CRC Press, 2015.

Gerald B Folland. *Real Analysis: Modern Techniques and Their Applications*. John Wiley & Sons, 2013.

Weihao Gao, Sewoong Oh, and Pramod Viswanath. Demystifying fixed k-nearest neighbor information estimators. *IEEE Trans. Inf. Theory*, 64(8):5629–5661, August 2018.

Ludovic Giet and Michel Lubrano. A minimum Hellinger distance estimator for stochastic differential equations: An application to statistical inference for continuous time interest rate models. *Comput. Statist. Data Anal.*, 52(6):2945–2965, 2008.

Valérie Girardin and Justine Lequesne. Entropy-based goodness-of-fit tests – a unifying framework: Application to DNA replication. *Commun. Statist. Theory Methods*, pages 1–13, 2017.

M. N. Goria, N. N. Leonenko, V. V. Mergel, and P. L.Novi Inverardi. A new class of

random vector entropy estimators and its applications in testing statistical hypotheses. *J. Nonparametr. Statist.*, 17(3):277–297, 2005.

Yanjun Han, Jiantao Jiao, Tsachy Weissman, and Yihong Wu. Optimal rates of entropy estimation over Lipschitz balls. *Ann. Statist.*, 48(6):3228–3250, 2020.

J Harvda and F Charvat. Quantification method of classification processes. concept of structural α-entropy. *Kybernetika (Prague)*, 3:30–35, 1967.

Keith Henderson, Brian Gallagher, and Tina Eliassi-Rad. EP-MEANS: An efficient nonparametric clustering of empirical probability distributions. In *Proc. Symp. Appl. Comput.*, pages 893–900. ACM, 2015.

Alfred O Hero and Olivier J J Michel. Asymptotic theory of greedy approximations to minimal k-point random graphs. *IEEE Trans. Inf. Theory*, 45(6):1921–1938, 1999.

Alfred O Hero, Bing Ma, Olivier J J Michel, and John Gorman. Applications of entropic spanning graphs. *IEEE Signal Process. Mag.*, 19(5):85–95, 2002.

Jiantao Jiao, Weihao Gao, and Yanjun Han. The Nearest Neighbor Information Estimator is Adaptively Near Minimax Rate-Optimal. In *Adv. Neural Inf. Proc. Syst.*, volume 31, December 2018.

Kirthevasan Kandasamy, Akshay Krishnamurthy, Barnabas Poczos, Larry A Wasserman, and James M Robins. Nonparametric von Mises estimators for entropies, divergences and mutual informations. In *Adv. Neural Inf. Proc. Syst.*, volume 28, pages 397–405, 2015.

Granino Arthur Korn and Theresa M Korn. *Mathematical Handbook for Scientists and Engineers: Definitions, Theorems, and Formulas for Reference and Review*. Courier Corporation, 2000.

L F Kozachenko and Nikolai N Leonenko. Sample estimate of the entropy of a random vector. *Probl. Inf. Transm.*, 23(2):9–16, 1987. (Russian).

Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Phys. Rev. E. Statist. Phys. Plasmas Fluids Relat. Interdiscip. Topics*, 69(6): 066138, 2004.

Akshay Krishnamurthy, Kirthevasan Kandasamy, Barnabás Póczos, and Larry Wasserman. Nonparametric estimation of Rényi divergence and friends. In *Proc. Int. Conf. Mach. Learn.*, pages 919–927, 2014.

Seyed M Lajevardi and Zahir M Hussain. Feature extraction for facial expression

recognition based on hybrid face regions. *Adv. Electr. Comput. Eng.*, 9(3):63–67, 2009.

Lucien Le Cam. *Asymptotic methods in statistical decision theory*. Springer Science & Business Media, 2012.

Erik G Learned-Miller and John W Fisher III. ICA using spacings estimates of entropy. *J. Mach. Learn. Res.*, 4(December):1271–1295, 2003.

Nikolai Leonenko, Luc Pronzato, and Vippal Savani. A class of Rényi information estimators for multidimensional densities. *Ann. Statist.*, 36(5):2153–2182, October 2008. Corrected in LEONENKO, N. and PROZANTO, L. (2010). Correction: A class of Rényi information estimators for multidimensional densities. *Ann. Statist.* **38** 3837–3838.

Nikolai N Leonenko and Luc Pronzato. Correction: A class of Rényi information estimators for multidimensional densities. *Ann. Statist.*, 38(6):3837–3838, 2010.

Jeremy Lewi, Robert Butera, and Liam Paninski. Real-time adaptive information-theoretic optimization of neurophysiology experiments. In *Adv. Neural Inf. Proc. Syst.*, volume 20, pages 857–864, 2007.

Juliane Liepe, Sarah Filippi, Komorowski Michał, and Michael P H Stumpf. Maximizing the information content of experiments in systems biology. *PLoS Comput. Biol.*, 9(1): e1002888, 2013.

H Liu, J Lafferty, and L Wasserman. Exponential concentration inequality for mutual information estimation. In *Adv. Neural Inf. Proc. Syst.*, volume 25, 2012.

Don O Loftsgaarden and Charles P Quesenberry. A nonparametric estimate of a multivariate density function. *Ann. Math. Statist.*, 36(3):1049–1051, 1965.

Stefano Marano, Vincenzo Matta, and Peter Willett. Asymptotic design of quantizers for decentralized MMSE estimation. *IEEE Trans. Signal Process.*, 55(11):5485–5496, 2007.

Kevin R Moon and Alfred O Hero. Ensemble estimation of multivariate f-divergence. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 356–360. IEEE, 2014a.

Kevin R Moon and Alfred O Hero. Multivariate f-divergence estimation with confidence. In *Adv. Neural Inf. Proc. Syst.*, volume 27, pages 2420–2428, 2014b.

Kevin R. Moon, Kumar Sricharan, and Alfred O. Hero. Ensemble estimation of mutual information. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 3030–3034. IEEE, June 2017.

Kevin R Moon, Kumar Sricharan, Kristjan Greenewald, and Alfred O Hero. Ensemble

estimation of information divergence. *Entropy*, 20(8):560, 2018.

Huzefa Neemuchwala, Alfred Hero, and Paul Carson. Image matching using alpha-entropy measures and entropic graphs. *Signal Process.*, 85(2):277–296, 2005.

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Trans. Inf. Theory*, 56(11):5847–5861, 2010.

Yung-Kyun Noh. *Generative metric learning and dimensionality reduction with f-divergences*. PhD thesis, Seoul National University, August 2011. URL http://s-space.snu.ac.kr/handle/10371/159245.

Morteza Noshad, Kevin R Moon, Salimeh Yasaei Sekeh, and Alfred O Hero. Direct estimation of information divergence using nearest neighbor ratios. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 903–907. IEEE, 2017.

Junier Oliva, Barnabás Póczos, and Jeff Schneider. Distribution to distribution regression. In *Proc. Int. Conf. Mach. Learn.*, pages 1049–1057, 2013.

Dávid Pál, Barnabás Póczos, and Csaba Szepesvári. Estimation of rényi entropy and mutual information based on generalized nearest-neighbor graphs. In *Adv. Neural Inf. Proc. Syst.*, volume 23, pages 1849–1857, 2010.

Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8):1226–1238, 2005.

Fernando Pérez-Cruz. Estimation of information theoretic measures for continuous random variables. In *Adv. Neural Inf. Proc. Syst.*, volume 22, pages 1257–1264, 2009.

Barnabás Póczos and Jeff G Schneider. On the Estimation of alpha-Divergences. *Int. Conf. Artif. Int. Statist.*, pages 609–617, 2011.

Barnabás Póczos, Liang Xiong, Dougal J Sutherland, and Jeff Schneider. Nonparametric kernel estimators for image classification. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 2989–2996. IEEE, June 2012.

Yury Polyanskiy and Yihong Wu. Lecture notes on information theory, 2019. URL http://www.stat.yale.edu/~yw562/teaching/itlectures.pdf.

Alfréd Rényi. On measures of entropy and information. In *Proc. 4th Berkeley Sympos. Math. Statist. Probab.*, volume 1, pages 547–761. Univ. California Press, Berkeley, 1961.

Walter Rudin. *Real and Complex Analysis*. McGraw-Hill Education, 1987.

Caifeng Shan, Shaogang Gong, and Peter W McOwan. Conditional Mutual Information Based Boosting for Facial Expression Recognition. In *Proc. British Mach. Vis. Conf.*, 2005.

Harshinder Singh, Neeraj Misra, Vladimir Hnizdo, Adam Fedorowicz, and Eugene Demchuk. Nearest neighbor estimates of entropy. *Am. J. Math. Manag. Sci.*, 23(3-4): 301–321, 2003.

Shashank Singh and Barnabás Póczos. Generalized exponential concentration inequality for rényi divergence estimation. In *Proc. Int. Conf. Mach. Learn.*, pages 333–341. PMLR, 2014a.

Shashank Singh and Barnabás Póczos. Exponential concentration of a density functional estimator. In *Adv. Neural Inf. Proc. Syst.*, volume 27, pages 3032–3040, 2014b.

Shashank Singh and Barnabás Póczos. Finite-sample analysis of fixed-k nearest neighbor density functional estimators. In *Adv. Neural Inf. Proc. Syst.*, volume 29, pages 1217–1225. Curran Associates, Inc., 2016.

José Martínez Sotoca and Filiberto Pla. Supervised feature selection by clustering using conditional mutual information-based distances. *Pattern Recogni.*, 43(6):2068–2081, 2010.

Kumar Sricharan, Raviv Raich, and Alfred O Hero. Estimation of nonlinear functionals of densities with confidence. *IEEE Trans. Inf. Theory*, 58(7):4135–4159, 2012.

Kumar Sricharan, Dennis Wei, and Alfred O Hero. Ensemble estimators for multivariate entropy estimation. *IEEE Trans. Inf. Theory*, 59(7):4374–4388, 2013.

J. Michael Steele. An Efron–Stein Inequality for Nonsymmetric Statistics. *Ann. Statist.*, 14(2):753–758, June 1986.

Karl-Theodor Sturm. On the geometry of metric measure spaces. *Acta Math.*, 196(1): 65–131, 2006.

Seba Susan and Madasu Hanmandlu. A non-extensive entropy feature and its application to texture classification. *Neurocomputing*, 120:214–225, 2013.

Constantino Tsallis. Possible generalization of boltzmann-gibbs statistics. *J. of Statist. Phys.*, 52(1-2):479–487, 1988.

A. B. Tsybakov and E. C. van der Meulen. Root-n Consistent Estimators of Entropy for

Densities with Unbounded Support. *Scand. Statist. Theory Appl.*, 1996.

Alexandre B Tsybakov. *Introduction to Nonparametric Estimation*. Springer Series in Statistics. Springer New York, New York, NY, 2009.

Qing Wang, Sanjeev R Kulkarni, and Sergio Verdú. Divergence estimation for multidimensional densities via k-nearest-neighbor distances. *IEEE Trans. Inf. Theory*, 55(5): 2392–2405, 2009.

Henry L Weidemann and Edwin B Stear. Entropy analysis of parameter estimation. *Inf. Control*, 14(6):493–506, 1969.

Alan Wisler, Kevin Moon, and Visar Berisha. Direct ensemble estimation of density functionals. In *Int. Conf. Acoust. Speech Signal Process.*, pages 2866–2870. IEEE, 2018.

Eric Wolsztynski, Eric Thierry, and Luc Pronzato. Minimum-entropy estimation in semi-parametric models. *Signal Process.*, 85:937–949, 2005.

Part III

Universal Information Processing

Chapter 5

Efficient Discrete Universal Denoising

5.1 Introduction

One of the simplest, yet most powerful approaches in data processing (such as compression, prediction, filtering, and estimation) of sequential data with spatiotemporal memory (text, image, biological sequences, and time series) is to first parse a given sequence according to a context model and then apply symbol-by-symbol solutions for each context independently. The discrete universal denoiser (DUDE) algorithm (Weissman et al., 2005) is a canonical example of this approach for denoising. With context size k, the DUDE algorithm is a two-sided k-th order sliding window denoiser, which decides each reconstruction symbol as the Bayes optimal response with respect to a given loss function and noise model, solely based on the counts of noisy symbols in the noisy observation sequence without any additional knowledge on the underlying sequence.

Due to its theoretical performance guarantee and low-complexity implementation, DUDE has been studied in various settings including continuous-alphabet (Sivaramakrishnan and Weissman, 2008, 2009), nonstationary (Moon and Weissman, 2009), and online (Khadivi et al., 2015) denoising. It has also found applications such as denoising DNA sequence (Lee et al., 2017) and image (Motta et al., 2005, 2011; Ordentlich et al., 2003, 2010; Sivaramakrishnan and Weissman, 2006).

Most context-based algorithms, with DUDE being no exception, however, suffer the "sparse context" problem (see, e.g., (Carpentieri et al., 2000; Motta et al., 2011)). As we increase the context size k, which is necessary to capture more spatiotemporal dependence in given data, the number of contexts increases exponentially in k and thus each context has too few samples to learn the structure of the data reliably. As this problem becomes more severe when the alphabet size is large, it poses a serious challenge on grayscale image denoising with DUDE (Buades et al., 2005; Motta et al., 2005, 2011).

One remedy to this sparse context problem is *context aggregation* that reduces the number of contexts by merging statistically or semantically similar contexts together. Image denoising using this context aggregation approach was developed as the iDUDE algorithm proposed in (Motta et al., 2005, 2011). In iDUDE, multiple contexts are explicitly aggregated based on vector quantization as well as prior assumptions on natural images previously used in lossless image compression (Carpentieri et al., 2000). The resulting denoising performance and computational complexity improves upon the naive k-context DUDE algorithm by orders of magnitude, and are comparable to other state-of-the-art grayscale image denoising algorithms.

As an alternative to an explicit reduction of a context model, one can *implicitly* aggregate contexts by allowing multiple contexts to "share" their samples. This idea was materialized recently by the Neural DUDE algorithm (Moon et al., 2016) that utilizes a neural network to learn a smooth mapping from a given context to expected losses of all single-symbol denoisers, through which contexts are effectively aggregated. Neural DUDE outperforms DUDE for a large context size k without suffering the aforementioned sparse context problem. On the downside, Neural DUDE has to learn all single-symbol denoiser losses, which becomes intractable even with a moderate alphabet size and makes it unfit for grayscale images.

In this paper, we propose a more natural and perhaps more principled approach

to implicit context aggregation, in which a simple feedforward deep neural network is trained from the given noisy image to learn a smooth mapping from each context to the conditional distribution of a noisy symbol conditioned on the context. This conditional probability is then plugged in to construct the Bayes optimal symbol-by-symbol denoiser used in DUDE and iDUDE. Compared to Neural DUDE, the neural network employed in the proposed context-aggregated universal denoiser (CUDE) algorithm scales linearly in the alphabet size, which makes it suitable for denoising of grayscale images and other larger alphabet problems. We remark that the idea of learning the contextual conditional distribution via neural networks and plugging in a corresponding Bayes optimal response to a given data processing problem is not new. For example, in the previous work (Adali et al., 1997), the conditional distribution of a binary channel information sequence was learned adaptively for channel equalization using a neural network with structure and training objective similar to ours.

Throughout this paper, we use x^n to denote a length-n sequence $(x_1, x_2, \ldots, x_n)$, and x_i^j to denote its subsequence $(x_i, x_{i+1}, \ldots, x_j)$. A random variable is denoted by an uppercase symbol, and a corresponding lowercase symbol denotes its realization. The probability mass function (pmf) of a random variable $X \in \mathcal{X}$ is denoted by $\mathsf{P}\{X = x\} = p(x)$ and is often identified as a vector in the simplex $\Delta^{|\mathcal{X}|}$. Finally, $1_z \in \{0,1\}^{|\mathcal{Z}|}$ denotes the one-hot encoding vector of $z \in \mathcal{Z}$ whose z-th coordinate is 1 and others are 0.

5.2 Problem Formulation

We first describe the problem in the one-dimensional case, and discuss how it can be generalized in higher dimensions later. We follow the standard definition of *universal denoising* in (Weissman et al., 2005). Let $\mathcal{X}$, $\mathcal{Z}$, and $\hat{\mathcal{X}}$ denote the alphabets of the clean source, the noisy observation, and the reconstruction symbol, respectively. Suppose that there is an underlying hidden sequence of clean symbols $X^n \in \mathcal{X}^n$ emitted from an

unknown stationary distribution, which is corrupted by a discrete memoryless channel $\Pi(z|x)$ to result in a noisy observation sequence Z^n. A denoiser $\hat{x}^n(z^n)$ is a mapping from Z^n to a reconstruction sequence $\hat{X}^n = \hat{x}^n(Z^n)$ with associated cumulative loss $\sum_{i=1}^n \Lambda(X_i, \hat{X}_i)$, where $\Lambda : \mathcal{X} \times \hat{\mathcal{X}} \to [0, \infty)$ is a prespecified loss function. We assume that Π is known and, when written in a matrix form, has a right inverse $\Pi^\dagger$.

We note that the aforementioned *stochastic* setting can be relaxed to the *semi-stochastic* setting, in which there is no probabilistic assumption on the clean source sequence x^n.

5.3 Review of the DUDE Algorithm

We first assume that the distribution of (X^n, Z^n) is known. For a given context size k, let $\mathbf{C}_i := (Z_{i-k}^{i-1}, Z_{i+1}^{i+k})$ be a *two-sided balanced* context consisting of k symbols on the left and k symbols on the right of the symbol Z_i. For each position $i = 1, 2, \ldots, n$, consider the Bayes optimal denoiser $\hat{x}_i^*(\mathbf{c}_i, z_i)$ based on the observation $\{\mathbf{C}_i = \mathbf{c}_i, Z_i = z_i\}$:

$$\hat{x}_i^*(\mathbf{c}_i, z_i) = \arg\min_{\hat{x} \in \hat{X}} \mathbb{E}[\Lambda(X_i, \hat{x}) | \mathbf{C}_i = \mathbf{c}_i, Z_i = z_i], \tag{5.1}$$

where the expectation is taken with respect to $p(x_i|\mathbf{c}_i, z_i)$, which can be found from $p(z_i|\mathbf{c}_i)$ by the Bayes rule and the inverse channel $\Pi^\dagger$. This denoiser can be readily shown to minimize the expected cumulative loss $\sum_{i=1}^n \mathbb{E}\Lambda(X_i, \hat{X}_i)$ among all denoisers $\hat{x}_i$ that use $z_{i-k}^{i+k} = (\mathbf{c}_i, z_i)$. Therefore, if the stationary pmf $p(z_i|\mathbf{c}_i)$ were known, the optimal denoiser could be found immediately.

Without any prior knowledge of the distribution, the DUDE algorithm follows this symbol-by-symbol Bayes optimal denoising approach by using the empirical distri-

bution

$$\hat{p}_{\text{emp}}(z|\mathbf{c}) = \frac{|\{j\colon \mathbf{c}_j = \mathbf{c}, z_j = z\}|}{|\{j\colon \mathbf{c}_j = \mathbf{c}\}|} \tag{5.2}$$

in place of the true $p(z|\mathbf{c})$ for each position i. Accordingly, the algorithm runs in two passes. In the first pass, scanning through the data once, it finds the empirical conditional pmf $\hat{p}_{\text{emp}}(z|\mathbf{c})$ in (5.2) by counting the number of occurrences of noisy symbols for each context $\mathbf{c}$. In the second pass, it finds the Bayes optimal denoiser (5.1) under $\hat{p}(x_i|\mathbf{c}_i, z_i)$, which can be computed from the empirical conditional pmf $\hat{p}_{\text{emp}}(z_i|\mathbf{c}_i)$ and the inverse channel matrix $\Pi^{\dagger}$. This computation can be performed easily by a few matrix–vector operations (see, for example, eq. (2) in (Moon et al., 2016).)

The DUDE algorithm has been shown to be *universal* in the sense that for any underlying stationary process it asymptotically attains the Bayes optimal performance, provided that k grows appropriately with n. A similar universality result has been also established for the semistochastic setting (Weissman et al., 2005).

The two-sided balanced context model can be easily extended to other context models. For example, a square-window neighborhood of side length $2k + 1$ centered at each symbol can be used for two-dimensional images. For a detailed discussion on the choice of a context model in higher dimensions, we refer the reader to (Ordentlich et al., 2011).

5.4 The Proposed CUDE Algorithm

Our CUDE algorithm consists of two steps. First, it learns the conditional distribution $p(z|\mathbf{c})$ using a neural network. It then plugs in the estimated distribution to find the symbol-by-symbol Bayes optimal denoiser (5.1), as in DUDE.

5.4.1 Conditional Distribution Learning Network

As before, suppose that a context model $\mathcal{C}$ of order k is used (e.g., the two-sided context model or the square-window context model). We introduce a feedforward fully connected neural network with multiple layers $\hat{p}_{\mathbf{w}} : \mathcal{C} \to \Delta^{|\mathcal{Z}|}$ parameterized by the weight vector $\mathbf{w}$, which is trained with the training data $\{(\mathbf{c}_i, 1_{z_i})\}_{i=1}^n$, solely based on the noisy observation sequence z^n, to learn the stationary conditional distribution $p(z|\mathbf{c})$, under the cross entropy loss function $H(p\|q) := -\sum_{z\in\mathcal{Z}} p(z) \log q(z)$. Equivalently, the network training minimizes

$$L(\mathbf{w}|z^n) := \frac{1}{n}\sum_{i=1}^{n} H(1_{z_i} \|\hat{p}_{\mathbf{w}}(z|\mathbf{c}_i)). \tag{5.3}$$

To force the output to be a proper probability distribution, the softmax layer of dimension $|\mathcal{Z}|$ is placed at the output layer.

The context aggregating behavior of our conditional distribution learning network can be explained by rewriting the objective function (5.3) as

$$\frac{1}{n}\sum_{i=1}^{n} \hat{p}_{\text{emp}}(\mathbf{c})(D(\hat{p}_{\text{emp}}(z|\mathbf{c})\|\hat{p}_{\mathbf{w}}(z|\mathbf{c})) + H(\hat{p}_{\text{emp}}(z|\mathbf{c}))).$$

Here we use $H(p\|q) = D(p \parallel q) + H(p)$, where $D(p \parallel q) = \sum_{z\in\mathcal{Z}} p(z) \log(p(z)/q(z))$ denotes the relative entropy between p and q, and $H(p) = -\sum_{z\in\mathcal{Z}} p(z) \log p(z)$ denotes the entropy of p. As the second term is independent of $\mathbf{w}$, our neural network can be trained to estimate the conditional distribution to minimize the first term, which captures the discrepancy between the empirical distribution and the trained distribution. This term converges to the conditional relative entropy $\mathbb{E}[D(p(z|\mathbf{C}) \parallel \hat{p}_{\mathbf{w}}(z|\mathbf{C}))]$ almost surely in the sample limit by Birkhoff's ergodic theorem (Cover and Thomas, 2012). Due to the finite capacity of the neural network and the continuity of the mapping $\mathbf{c} \mapsto \hat{p}_{\mathbf{w}}(z|\mathbf{c})$, the network is expected to assign similar conditional probabilities to close

contexts, effectively aggregating multiple contexts.

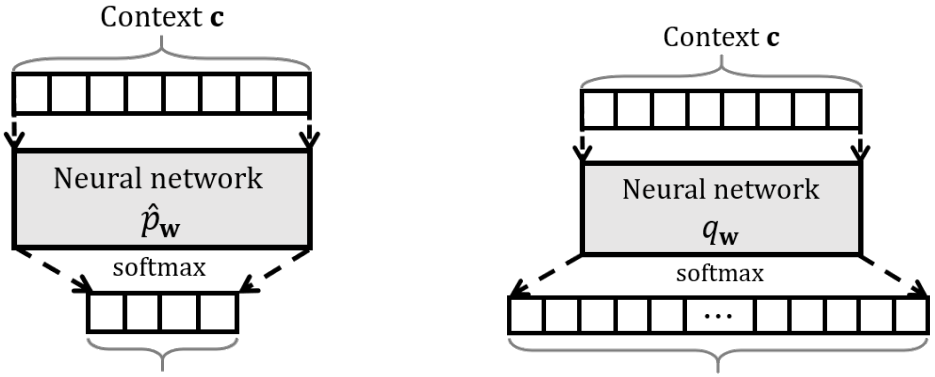
5.4.2 Context-Based Symbol-by-Symbol Denoising

After training the network, we use the trained conditional distribution $\hat{p}_{\mathbf{w}}(z|\mathbf{c})$ for symbol-by-symbol denoising by finding the Bayes optimal denoiser in (5.1). This plug-in approach provides a complete separation between probability learning and the denoising operation.

5.5 Comparison with Neural DUDE

The Neural DUDE algorithm (Moon et al., 2016) is a variant of DUDE that was designed to select the optimal symbol-by-symbol denoiser for a given context based on a neural network. Neural DUDE trains a single fully connected feedforward neural network $q_{\mathbf{w}} : \mathcal{C} \to \Delta^{|\mathcal{S}|}$, which maps a context to a probability vector over the collection $\mathcal{S} := \{s : \mathcal{Z} \to \hat{\mathcal{X}}\}$ of all single-symbol denoisers. After training the parameter $\mathbf{w}$ with the training data constructed from z^n and a new loss function over $\mathcal{Z} \times \mathcal{S}$, the output probability distribution $q_{\mathbf{w}}(s|\mathbf{c})$ is used as the *score* vector of each single-symbol denoiser for a context $\mathbf{c}$ as in classification (see, e.g., (Christopher, 2006, Ch. 5)). Neural DUDE then selects the single-symbol denoiser of the highest score and uses it to denoise the given noisy symbol.

The advantage of CUDE over Neural DUDE lies mostly in its simple and flexible plug-in architecture. CUDE uses a smaller output layer that scales linearly in the alphabet size $|\mathcal{Z}|$, while the output layer in Neural DUDE scales as $|\mathcal{S}| = |\mathcal{Z}|^{|\hat{\mathcal{X}}|}$ (see Fig. 5.5.1 for a comparison of the neural networks used in CUDE and Neural DUDE). As a concrete example, when $|\mathcal{Z}| = |\hat{\mathcal{X}}| = 4$ (quaternary image), the network for CUDE has the output layer dimension of 4, whereas the dimension for Neural DUDE is $4^4 = 256$. Hence, CUDE can be implemented in lower complexity for a large alphabet, while achieving a faster convergence to the desired performance.



(a) CUDE (b) Neural DUDE

Figure 5.5.1. Comparison of neural networks used in CUDE and Neural DUDE under the two-sided balanced context model of order $k = 4$.

5.6 Experiments

Experiments were carried out with Python 3.6 and Keras package with Theano backend (Bastien et al., 2012). We trained the networks with six hidden layers of 40 rectified linear unit (ReLU) activations for Neural DUDE and CUDE by the optimization method Adam (Kingma and Ba) following the same setting such as mini-batch size in (Moon et al., 2016). Raw alphabets were used for both cases, instead of the one-hot encoding used in (Moon et al., 2016).

To compare CUDE with DUDE and Neural DUDE, we performed denoising experiments with publicly available standard test images such as Barbara, boat, cameraman, and Lena of size 512×512 (e.g., (BM3)), scaled down to the bit depth of 2 (alphabet size 4). We chose the quaternary alphabet for our simulation because DUDE and Neural DUDE can only handle small alphabets. We considered an image as a one-dimensional sequence by raster scan, and used the balanced two-sided context model of order $k = 1, 2, \ldots, 40$. The images were corrupted by the salt and pepper (S&P) noise (Motta et al., 2011) with error probability $\delta = 10\%$ and 30%, and by the

quaternary symmetric channel (QSC) noise with error probability $\delta = 10\%$ and 30%. The squared-error loss was assumed. Fig. 5.6.1 shows the plot of PSNRs of the different context order k for the boat image corrupted by S&P noise, and CUDE consistently outperforms Neural DUDE. Denoising results for different images and noise models exhibit a similar trend, as summarized in Table 5.6.1. Note that the gain in performance as well as computational complexity would become more pronounced as the alphabet size grows.

Unlike DUDE and Neural DUDE that cannot be scaled to large alphabets due to either high complexity or the sparse context problem, CUDE can be applied directly to grayscale image denoising. To demonstrate the potential of CUDE for grayscale images, we performed a denoising experiment for the grayscale Barbara image of the original bit depth 8 corrupted by S&P noise with $\delta = 50\%$ in Fig. 5.6.2. In this experiment, we used two-dimensional square context model, which yields a better performance than one-dimensional model in general. Fig. 5.6.2(c) shows the reconstructed image using

Table 5.6.1. Comparison of denoising performance in PSNR(dB) attained by DUDE, Neural DUDE, and CUDE for quaternary scaled images corrupted by S&P or QSC noise with $\delta = 10\%$ and 30%. The number in the parentheses indicates the best order k that achieves the PSNR presented.

Noise	Algorithms	Barbara	Boat	Cameraman	Lena
S&P (10%)	DUDE	21.3 (3)	23.4 (2)	25.8 (2)	23.3 (2)
	Neural DUDE	21.9 (30)	23.7 (10)	25.8 (16)	24.0 (21)
	CUDE	**23.0** (20)	**24.4** (16)	**27.8** (5)	**25.3** (35)
S&P (30%)	DUDE	13.4 (2)	16.4 (2)	19.0 (2)	14.7 (2)
	Neural DUDE	16.3 (23)	18.0 (11)	19.0 (5)	16.8 (23)
	CUDE	**17.2** (38)	**19.1** (20)	**20.3** (17)	**17.9** (34)
QSC (10%)	DUDE	20.5 (3)	22.0 (2)	24.4 (2)	22.4 (2)
	Neural DUDE	20.7 (26)	21.9 (5)	23.9 (3)	21.9 (27)
	CUDE	**21.5** (36)	**22.6** (11)	**25.2** (10)	**23.1** (6)
QSC (30%)	DUDE	14.7 (3)	16.3 (2)	16.7 (2)	15.7 (3)
	Neural DUDE	16.3 (10)	17.8 (13)	18.7 (16)	17.6 (17)
	CUDE	**16.5** (18)	**18.2** (16)	**19.1** (15)	**17.9** (15)

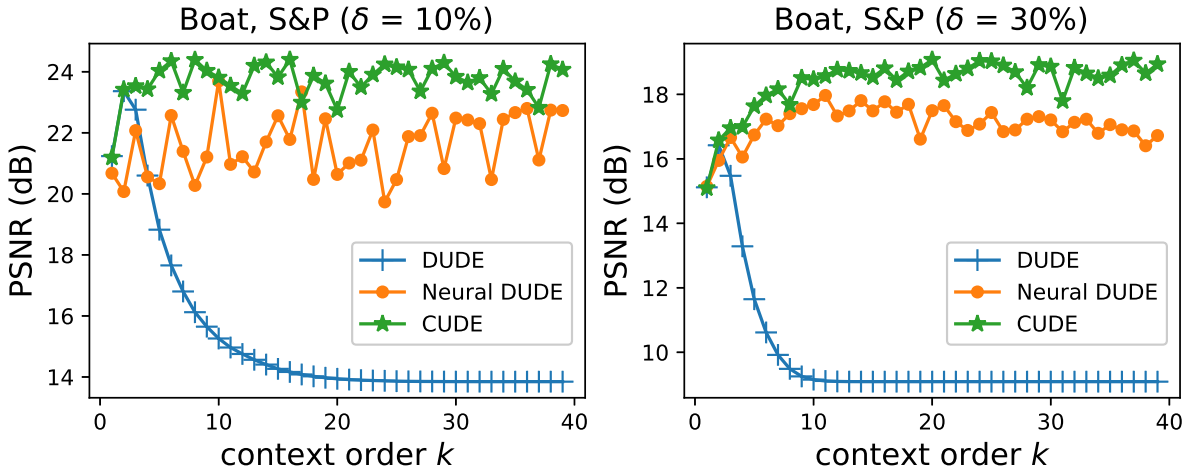


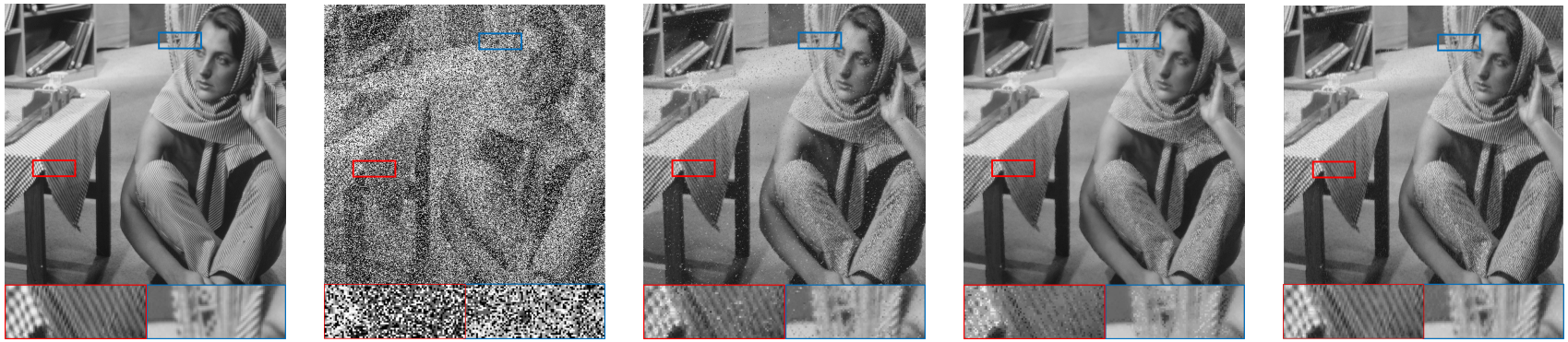
Figure 5.6.1. PSNR plot for the quaternary boat image corrupted by S&P noise ($\delta = 10\%$ and 30%) with different context orders.

CUDE under the best context order of $k = 1$ (8 pixels surrounding a given pixel), and the attained PSNR. As is clear from the image, CUDE was able to denoise the corrupted image only roughly, leaving numerous visible spots. It was generally observed that in low SNR as in this case, excessive aggregation of contaminated contexts degraded the performance.

In order to mitigate this issue, we extended the CUDE algorithm with prefiltering followed by iterated denoising. This approach was developed originally in (Motta et al., 2011), where the iterated median selective median filter (IMSM) tailored for S&P noise was used as a prefilter for initial, low-quality denoising, and a context-aggregated DUDE algorithm was used iteratively as a main denoiser. Our conditional distribution learning framework can readily incorporate prefiltered images to enhance the quality of context aggregation. Let y^n be a cleaner version of the original noisy observation z^n, obtained by prefiltering or iterated denoising. Instead of learning $p(z_i|\mathbf{c}_i)$, we can learn the conditional distribution $p(z_i|\mathbf{c}_i(y^n))$ of z_i given the corresponding context at position i in y^n. This can be implemented by training our network with $\{(\mathbf{c}_i(y^n), 1_{z_i}\}_{i=1}^n$. Under this modification, we performed IMSM prefiltering initially on the same noisy

image and iteratively applied CUDE. Fig. 5.6.2(d) shows the prefiltered image by the IMSM filter (no CUDE yet), and Fig. 5.6.2(e) shows the denoised image obtained after 5 iterations of CUDE under the context order of $k = 15$, initially starting from Fig. 5.6.2(d). Although the IMSM prefilter destroys some image structures and results in a blurry image (see the magnified patches below the image), the subsequent CUDE iterations recover the texture details in the original image. It can be also noted that, compared to CUDE-only denoising, larger contexts are utilized without performance degradation. According to our preliminary results (data not shown), this extension of CUDE achieves denoising performance comparable to that of iDUDE, especially in a low SNR regime, although further research and more extensive experiments are called for in high SNR and other noise models.

Tuning the context order can be performed by visual assessment of the resulting images. An alternative was proposed in (Moon et al., 2016) based on the observation that the estimated loss for Neural DUDE concentrates tightly around the true loss. The same phenomenon was also observed for CUDE (data not shown). A theoretical development on the CUDE loss estimator and its concentration behavior will be reported elsewhere.



(a) Original image **(b)** Noisy image (8.3dB) **(c)** CUDE w/ k=1 (24.1dB) **(d)** IMSM prefiltered image (25.6dB) **(e)** Iterated CUDE w/ k=15 (**29.9dB**)

Figure 5.6.2. Denoising of the grayscale Barbara image corrupted by S&P noise with $\delta = 50\%$. Two-dimensional square-window contexts were used. The red and blue patches specified in each image are magnified and shown below.

Acknowledgement

Chapter 5, in part, is a reprint of the material in the paper with permission: © 2018 IEEE. Jongha Ryu and Young-Han Kim, "Conditional distribution learning with neural networks and its application to universal image denoising," In *Proceedings of IEEE International Conference on Image Processing*, pp. 3214–3218, Athens, Greece, October 2018. The dissertation author was the primary investigator and author of this paper.

Bibliography

The official webpage of the BM3D algorithm. https://www.cs.tut.fi/~foi/GCF-BM3D/. Online; Accessed February-11-2018.

T. Adali, X. Liu, and M. K. Sonmez. Conditional distribution learning with neural networks and its application to channel equalization. *IEEE Trans. Signal Process.*, 45 (4):1051–1064, Apr 1997. ISSN 1053-587X. doi: 10.1109/78.564193.

F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio. Theano: new features and speed improvements. *CoRR*, abs/1211.5590, 2012. URL http://arxiv.org/abs/1211.5590.

A. Buades, B. Coll, and J. M. Morel. A Review of Image Denoising Algorithms, with a New One. *SIAM Multiscale Model. Simul.*, 4(2):490–530, 2005. ISSN 1540-3459. doi: 10.1137/040616024. URL http://epubs.siam.org/doi/10.1137/040616024.

B. Carpentieri, M. J. Weinberger, and G. Seroussi. Lossless compression of continuous-tone images. *Proc. IEEE*, 88(11):1797–1809, Nov 2000. ISSN 0018-9219. doi: 10.1109/5.892715.

M.B. Christopher. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2012.

P. Khadivi, R. Tandon, and N. Ramakrishnan. Online denoising of discrete noisy data. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 671–675, June 2015. doi: 10.1109/ISIT.2015.7282539.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. Int. Conf. Learn. Repr.*

B. Lee, T. Moon, S. Yoon, and T. Weissman. Dude-seq: Fast, flexible, and robust denoising for targeted amplicon sequencing. *PLOS ONE*, 12(7):1–25, Jul 2017. doi: 10.1371/journal.pone.0181463. URL https://doi.org/10.1371/journal.pone.0181463.

T. Moon and T. Weissman. Discrete denoising with shifts. *IEEE Trans. Inf. Theory*, 55(11): 5284–5301, 2009. ISSN 00189448. doi: 10.1109/TIT.2009.2030461.

T. Moon, S. Min, B. Lee, and S. Yoon. Neural universal discrete denoiser. In *Proc. Adv. Neural Info. Proc. Syst.*, pages 4772–4780, 2016. URL http://papers.nips.cc/paper/6497-neural-universal-discrete-denoiser.pdf.

G. Motta, E. Ordentlich, I. Ramirez, G. Seroussi, and M. J. Weinberger. The dude framework for continuous tone image denoising. In *Proc. IEEE Int. Conf. Image Proc.*, volume 3, pages III–345–8, Sep 2005. doi: 10.1109/ICIP.2005.1530399.

G. Motta, E. Ordentlich, I. Ramírez, G. Seroussi, and M. J. Weinberger. The iDUDE framework for grayscale image denoising. *IEEE Trans. Image Proc.*, 20(1):1–21, 2011. ISSN 10577149. doi: 10.1109/TIP.2010.2053939.

E. Ordentlich, G. Seroussi, S. Verdu, M. Weinberger, and T. Weissman. A discrete universal denoiser and its application to binary images. In *Proc. IEEE Int. Conf. Image Proc.*, volume 1, pages I–117–20 vol.1, Sep 2003. doi: 10.1109/ICIP.2003.1246912.

E. Ordentlich, G. Seroussi, and M. Weinberger. Modeling enhancements in the DUDE framework for grayscale image denoising. In *Proc. IEEE Inf. Theory Workshop*, pages 1–5, Jan 2010. doi: 10.1109/ITWKSPS.2010.5503145.

E. Ordentlich, M. J. Weinberger, and C. Chang. On multi-directional context sets. *IEEE Trans. Inf. Theory*, 57(10):6827–6836, 2011.

K. Sivaramakrishnan and T. Weissman. Universal denoising of continuous amplitude signals with applications to images. In *Proc. IEEE Int. Conf. Image Proc.*, pages 2609–2612, Oct 2006. doi: 10.1109/ICIP.2006.313021.

K. Sivaramakrishnan and T. Weissman. Universal denoising of discrete-time continuous-amplitude signals. *IEEE Trans. Inf. Theory*, 54(12):5632–5660, Dec 2008. ISSN 0018-9448. doi: 10.1109/TIT.2008.2006438.

K. Sivaramakrishnan and T. Weissman. A context quantization approach to universal denoising. *IEEE Trans. Signal Process.*, 57(6):2110–2129, Jun 2009. ISSN 1053-587X. doi: 10.1109/TSP.2008.2011847.

T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu, and M. J. Weinberger. Universal discrete denoising: known channel. *IEEE Trans. Inf. Theory*, 51(1):5–28, Jan 2005. ISSN 0018-9448. doi: 10.1109/TIT.2004.839518. URL http://ieeexplore.ieee.org/stampPDF/getPDF.jsp?arnumber=1377489{&}isnumber=30067.

Chapter 6

Parameter-Free Online Learning with Side Information

6.1 Introduction

In this paper, we consider the problem of online linear optimization (OLO) in a Hilbert space V with norm $\|\cdot\|$. In each round $t = 1, 2, \ldots$, a learner picks an action $\mathbf{x}_t \in V$, receives a vector $\mathbf{g}_t \in V$ with $\|\mathbf{g}_t\| \leq 1$, and suffers loss $\langle \mathbf{g}_t, \mathbf{x}_t \rangle$. In this repeated game, the goal of the learner is to keep her *cumulative regret* small with respect to any competitor $\mathbf{u}$ for any adversarial sequence $\mathbf{g}^T := \mathbf{g}_1, \ldots, \mathbf{g}_T$, where the cumulative regret is defined as the difference between the cumulative losses of the learner and $\mathbf{u} \in V$, i.e.,

$$\mathsf{Reg}_T(\mathbf{u}) := \mathsf{Reg}(\mathbf{u}; \mathbf{g}^T) := \sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{x}_t \rangle - \sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{u} \rangle.$$

Albeit simple in nature, an OLO algorithm serves as a versatile building block in machine learning algorithms (Shalev-Shwartz, 2011); for example, it can be used to solve online convex optimization.

While there exist standard algorithms such as online gradient descent (OGD) that achieve optimal regret of order $\mathsf{Reg}_T(\mathbf{u}) = O(\|\mathbf{u}\|\sqrt{T})$, these algorithms typically require tuning parameters with unknowns such as the norm $\|\mathbf{u}\|$ of a target competitor $\mathbf{u}$. For example, OGD with step size $\eta = 1/\sqrt{T}$ achieves $\mathsf{Reg}_T(\mathbf{u}) = O((1 + \|\mathbf{u}\|^2)\sqrt{T})$

for any $\mathbf{u} \in V$, while OGD with $\eta = U/\sqrt{T}$ achieves $\mathsf{Reg}_T(\mathbf{u}) = O(U\sqrt{T})$ for any $\mathbf{u} \in V$ such that $\|\mathbf{u}\| \leq U$; see, e.g., (Shalev-Shwartz, 2011). To avoid tuning parameters, several *parameter-free* algorithms have been proposed in the last decade, aiming to achieve cumulative regret of order $\tilde{O}(\|\mathbf{u}\|\sqrt{T})$ for any $\mathbf{u} \in V$ without knowing $\|\mathbf{u}\|$ a priori (McMahan and Abernethy, 2013; McMahan and Orabona, 2014; Orabona, 2013, 2014; Orabona and Pál, 2016), where $\tilde{O}(\cdot)$ hides any polylogarithmic factor in the big O notation; the extra polylogarithimic factor is known to be necessary (McMahan and Abernethy, 2013; Orabona, 2013).

While these optimality guarantees on regret seem sufficient, they may not be satisfactory in bounding the incurred loss of the algorithm, due to the limited power of the class of static competitors $\mathbf{u}$ as a benchmark. For example, consider the adversarial sequence $\mathbf{g}, -\mathbf{g}, \mathbf{g}, -\mathbf{g}, \ldots$ for a fixed vector $\mathbf{g} \in \mathbb{B} := \{\mathbf{x} \in V : \|\mathbf{x}\| \leq 1\}$. Despite the apparent structure (or predictability) in the sequence, the best achievable reward of any static competitor $\mathbf{u} \in V$ is zero for any even T. In general, the cumulative loss of a static competitor $\mathbf{u}$ is $\sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{u} \rangle = \langle \sum_{t=1}^{T} \mathbf{g}_t, \mathbf{u} \rangle$, and can be large if and only if the norm $\| \sum_{t=1}^{T} \mathbf{g}_t \|$ is large, or equivalently, when $\mathbf{g}_1, \ldots, \mathbf{g}_T$ are well *aligned*. It is not only a theoretical issue, since, for example, when we consider a practical scenario such as weather forecasting, the sequence $(\mathbf{g}_t)$ may have such a *temporal structure* that can be exploited in optimization, rather than being completely adversarial.

One remedy for this issue is to consider a larger class of competitors, which may *adapt* to the history $\mathbf{g}^{t-1} := \mathbf{g}_1, \ldots, \mathbf{g}_{t-1}$. Hereafter, we use x_t^s to denote the sequence $x_t, \ldots, x_s$ for $t \leq s$ and $x^t := x_1^t$ by convention. For instance, in the previous example, consider a competitor which can play two different actions $\mathbf{u}_{+1}$ and $\mathbf{u}_{-1}$ based on the quantization $Q(\mathbf{g}_{t-1}) = \operatorname{sgn}(\langle \mathbf{f}, \mathbf{g}_{t-1} \rangle)$ for some fixed $\mathbf{f} \in V$; for example, we chose standard vectors $\mathbf{e}_i$ for a Euclidean space V in our experiments; see Section 6.4. Then the best loss achieved by the competitor class on this sequence becomes $-(T/2)\|\mathbf{g}\|(\|\mathbf{u}_{+1}\| + \|\mathbf{u}_{-1}\|)$, which could be much smaller than 0. We remark that, from the view of binary

prediction, this example can be thought of a first-order Markov prediction, which takes only the previous time step into consideration. Hence, it is natural to consider a k-th order extension of the previous example, i.e., a competitor that adapts to the length-k sequence $Q(\mathbf{g}_{t-k}^{t-1}) := Q(\mathbf{g}_{t-k}) \dots Q(\mathbf{g}_{t-1}) \in \{1, \bar{1}\}^k$, where we define $\bar{1} := -1$.

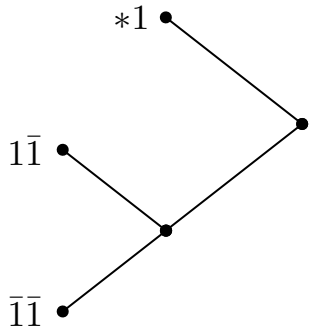


Figure 6.1.1. An example set of suffixes $\mathbf{T} = \{*1, 1\bar{1}, \bar{1}\bar{1}\}$.

We can even further sophisticate a competitor's dependence structure by allowing it to adapt to a *tree structure* (also known as a *variable-order Markov structure*) of the quantization sequence, which is widely deployed structure in sequence prediction; see, e.g., (Begleiter et al., 2004). For example, for the depth-2 quantization sequence $Q(\mathbf{g}_{t-2}^{t-1})$, rather than adapting to the all four possible states, a competitor may adapt to the suffix falls into a set of suffixes $\mathbf{T} = \{*1, 1\bar{1}, \bar{1}\bar{1}\}$ of one fewer states; here, $*$ denotes that any symbol from $\{1, \bar{1}\}$ is possible in that position. As depicted in Figure 6.1.1 for $\mathbf{T}$, in general, a suffix set has a one-to-one correspondence between a full binary tree, and is thus often identified as a tree; see Section 6.3.3 for the formal definition and further justification of the tree side information.

Since we do not know a priori which tree structure is best to adapt to, we ultimately aim to design an OLO algorithm that achieves the performance of the best tree competitor of given maximum depth $D \geq 1$. Since there are $O(2^{2^D})$ possible trees of depth at most D, it becomes challenging even for a moderate size of D. We remark that the problem of following the best tree structure in hindsight, the *tree problem* in short,

is a classical problem which has been studied in multiple areas such as information theory (Willems et al., 1995) and online learning (Freund et al., 1997), but an application of this framework to the OLO problem has not been considered in the literature.

To address this problem, we combine two technical components from online learning and information theory. Namely, we apply an information theoretic technique of following the best tree structure for universal compression, called the *context tree weighting* (CTW) algorithm invented by Willems et al. (1995), to generalize a parameter-free OLO algorithm called the *KT OLO algorithm* proposed by Orabona and Pál (2016), which is designed based on universal coin betting. Consequently, as the main result, we propose the *CTW OLO algorithm* that efficiently solves the problem with only $O(D)$ updates per round achieving nearly minimax optimal regret; see Section 6.3.3.

We motivate the proposed approach by solving two intermediate, abstract OLO problems, the one with (single) side information (Section 6.3.1) and the other with multiple side information (Section 6.3.2), and propose information theoretic OLO algorithms (i.e., product KT and mixture KT) respectively, which might be of independent interest. We remark, however, that it is not hard to convert any parameter-free algorithm to solve the abstract problems with same guarantees and complexity of the proposed solutions, using existing meta techniques such as a black-box aggregation scheme by Cutkosky (2019) with per-state extension of a base OLO algorithm; hence, the contribution of the intermediate solutions is rather purely of intellectual merit.

In Section 6.4, we experimentally demonstrate the power of the CTW OLO algorithm with real-world temporal datasets. We conclude with some remarks in Section 6.5. All proofs and discussion with related work are deferred to Appendix due to the space constraint.

Notation

Given a tuple $\mathbf{a} = (a_1, \dots, a_m)$, we use $\sum \mathbf{a} := \sum_{i=1}^m a_i$ to denote the sum of all entries in a tuple $\mathbf{a}$. For example, we write $\sum g^{t-1}$ to denote the sum of $g_1, \dots, g_{t-1}$ by identifying g^{t-1} as a tuple $(g_1, \dots, g_{t-1})$. For the empty tuple $()$, we define $\sum() := 0$ by convention. We use $|\mathbf{a}|$ to denote the number of entries of a tuple $\mathbf{a}$. For a tuple of vectors $\mathbf{u}_{1:S} := (\mathbf{u}_1, \dots, \mathbf{u}_S) \in V \times \cdots \times V$, we use $\|\mathbf{u}\|_{1:S} := (\|\mathbf{u}_1\|, \dots, \|\mathbf{u}_S\|) \in \mathbb{R}^S_{\geq 0}$ to denote the tuple of norms of each entry.

6.2 Preliminaries

We review the coin betting based OLO algorithm of Orabona and Pál (2016). From this point, we will describe all algorithms in the reward maximization framework, which is philosophically consistent with the goal of gambling, to avoid any confusion, but we will keep using the conventional naming OGD even though it is actually gradient *ascent*.[1]

6.2.1 Continuous Coin Betting and 1D OLO

Consider the following repeated gambling. Starting with an initial wealth W_0, at each round t, a player picks a *signed relative bet* $b_t \in [-1, 1]$. At the end of the round, a real number $g_t \in [-1, 1]$ is revealed as an outcome of the "continuous coin toss" and the player gains the reward $g_t b_t \mathsf{W}_{t-1}$. This game leads to the cumulative wealth

$$\mathsf{W}_t(g^t) = \mathsf{W}_0 \prod_{i=1}^t (1 + g_i b_i).$$

When $g_t \in \{\pm 1\}$, this game boils down to the standard coin betting, where the player splits her wealth into $\frac{1+b_t}{2}\mathsf{W}_{t-1}$ and $\frac{1-b_t}{2}\mathsf{W}_{t-1}$, and bets the amounts on the binary outcomes $+1$ and -1, respectively. It is well known that the standard coin betting

[1] Note that one can translate a reward maximization algorithm to an equivalent loss minimization algorithm by feeding $-\mathbf{g}_t$ instead of $\mathbf{g}_t$, and vice versa.

game is equivalent to the binary compression, or binary log-loss prediction, which have been extensively studied in information theory; see, e.g., (Cover and Thomas, 2006, Chapter 6).

Even when the outcomes g_t are allowed to take continuous values, many interesting connections remain to hold. For example, the Krichevsky and Trofimov (1981)'s (KT) probability assignment, which is competitive against i.i.d. Bernoulli models, can be translated into a betting strategy

$$b^{\mathsf{KT}}(g^{t-1}) := b_t^{\mathsf{KT}}(\textstyle\sum g^{t-1}),$$

where $b_t^{\mathsf{KT}}(x) := \frac{x}{t}$ for $x \in [-t+1, t-1]$. As a natural continuous extension of the KT probability assignment, we define the *KT coin betting potential*

$$\psi^{\mathsf{KT}}(g^t) := \psi_t^{\mathsf{KT}}(\textstyle\sum g^t) := 2^t \tilde{q}_t^{\mathsf{KT}}(\sum g^t),$$

where

$$\tilde{q}_t^{\mathsf{KT}}(x) := B\Big(\frac{t+x+1}{2}, \frac{t-x+1}{2}\Big)\Big/B\Big(\frac{1}{2}, \frac{1}{2}\Big)$$

for $x \in [-t, t]$ and $B(x, y) := \Gamma(x)\Gamma(y)/\Gamma(x+y)$ and $\Gamma(x)$ denote the Beta function and Gamma function, respectively. We remark that the interpolation for continuous values is naturally defined via the Gamma functions. This simple KT betting scheme guarantees that the cumulative wealth satisfies

$$\mathsf{W}_T(g^T) \geq \mathsf{W}_0 \psi^{\mathsf{KT}}(g^T) = \mathsf{W}_0 2^T \tilde{q}_T^{\mathsf{KT}}(\textstyle\sum g^T) \tag{6.1}$$

for any $T \geq 1$ and $g_1, \ldots, g_T \in [-1, 1]$; see the proof of Theorem 6.2.1 in Appendix. It can be easily shown that the wealth lower bound is near-optimal when compared to the best static bettor $b_t = b$ for some fixed $b \in [-1, 1]$ in hindsight, the so-called

Kelly betting (Kelly Jr., 1956). This follows as a simple consequence of the fact that the KT probability assignment is a near-optimal probability assignment for universal compression of i.i.d. sequences. In this paper, going forward the interpretation of the coin betting potential as probability assignment in the parlance of compression will prove useful.

In their insightful work, Orabona and Pál (2016) demonstrated that the universal continuous coin betting algorithm can be directly translated to an OLO algorithm with a parameter-free guarantee. By defining an *absolute betting* $w_t := b_t \mathsf{W}_{t-1}$, we can write the cumulative wealth in an additive form

$$\mathsf{W}_t(g^t) = \mathsf{W}_0 + \sum_{i=1}^{t} g_t w_t,$$

whence we interpret $\sum_{i=1}^{t} g_i w_i$ as the cumulative reward in the 1D OLO with $g_1, \dots, g_t \in [-1, 1]$. Now, if we define the KT coin betting OLO algorithm by the action

$$w_t^{\mathsf{KT}} := w^{\mathsf{KT}}(g^{t-1}) = b^{\mathsf{KT}}(g^{t-1}) \mathsf{W}_{t-1}(g^{t-1}),$$

then the "universal" wealth lower bound (6.1) with respect to any g^T can be translated to establish a "parameter-free" bound on the 1D regret

$$\mathsf{Reg}(u; g^T) := \sum_{t=1}^{T} g_t u - \sum_{t=1}^{T} g_t w_t^{\mathsf{KT}},$$

against static competitors $u \in \mathbb{R}$. Let $(\psi_T^{\mathsf{KT}})^\star \colon \mathbb{R} \to \mathbb{R}$ denote the Fenchel dual of the potential function $\psi_T^{\mathsf{KT}} \colon \mathbb{R} \to \mathbb{R}$, i.e.,

$$(\psi_T^{\mathsf{KT}})^\star(u) := \sup_{g \in \mathbb{R}} (gu - \psi_T^{\mathsf{KT}}(g)).$$

Theorem 6.2.1. *For any $g_1, \dots, g_T \in [-1, 1]$, the 1D OLO algorithm $w_t^{\mathsf{KT}} = b^{\mathsf{KT}}(g^{t-1}) \mathsf{W}_{t-1}$*

satisfies

$$\sup_{u\in\mathbb{R}}\Big\{\mathsf{Reg}(u;g^T)-\mathsf{W}_0(\psi_T^{\mathsf{KT}})^\star\Big(\frac{u}{\mathsf{W}_0}\Big)\Big\}\le\mathsf{W}_0.$$

In particular, for any $u\in\mathbb{R}$*, we have*

$$\mathsf{Reg}(u;g^T)\le\sqrt{Tu^2\ln(Tu^2/(e\sqrt{\pi}\mathsf{W}_0^2)+1)}+\mathsf{W}_0.$$

6.2.2 Reduction of OLO over a Hilbert Space to Continuous Coin Betting

This reduction can be extended for OLO over a Hilbert space V with norm $\|\cdot\|$, where we wish to maximize the cumulative reward $\sum_{t=1}^T\langle\mathbf{g}_t,\mathbf{x}_t\rangle$ for $\mathbf{g}_1,\dots,\mathbf{g}_T\in\mathbb{B}:=\{\mathbf{x}\in V\colon\|\mathbf{x}\|\le1\}$. Orabona and Pál (2016) proposed the following OLO algorithm over Hilbert space based on the continuous coin betting. For an initial wealth $\mathsf{W}_0>0$, we define the *cumulative wealth*

$$\mathsf{W}_T(\mathbf{g}^T):=\mathsf{W}_0+\sum_{t=1}^T\langle\mathbf{g}_t,\mathbf{x}_t\rangle$$

as the cumulative reward plus the initial wealth, analogously to the coin betting. If we define the *vectorial betting* given $\mathbf{g}^{t-1}$ as

$$\mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1}):=b_t^{\mathsf{KT}}(\|\textstyle\sum\mathbf{g}^{t-1}\|)\dfrac{\sum\mathbf{g}^{t-1}}{\|\sum\mathbf{g}^{t-1}\|}=\dfrac{1}{t}\sum\mathbf{g}^{t-1}$$

and define a *potential* function

$$\Psi^{\mathsf{KT}}(\mathbf{g}^t):=\psi_t^{\mathsf{KT}}(\|\sum\mathbf{g}^t\|)=2^t\tilde{q}_t^{\mathsf{KT}}(\|\textstyle\sum\mathbf{g}^t\|),$$

then the corresponding OLO algorithm ensures the wealth lower bound $\mathsf{W}_t(\mathbf{g}^t)\ge\mathsf{W}_0\Psi^{\mathsf{KT}}(\mathbf{g}^t)$, and thus the corresponding regret upper bound in the same spirit of Theorem 6.2.1.

Theorem 6.2.2 (Orabona and Pál, 2016, Theorem 3)**.** *For any* $\mathbf{g}_1, \dots, \mathbf{g}_T \in \mathbb{B}$*, the OLO algorithm* $\mathbf{w}_t^{\mathsf{KT}} = \mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1})\mathsf{W}_{t-1}$ *based on the coin betting satisfies* $\mathsf{W}_T \geq \mathsf{W}_0 \Psi^{\mathsf{KT}}(\mathbf{g}^T)$*, and moreover*

$$\sup_{\mathbf{u} \in V} \left\{ \mathsf{Reg}(\mathbf{u}; \mathbf{g}^T) - \mathsf{W}_0 (\psi_T^{\mathsf{KT}})^\star \left(\frac{\|\mathbf{u}\|}{\mathsf{W}_0} \right) \right\} \leq \mathsf{W}_0.$$

In particular, for any $\mathbf{u} \in V$*, we have*

$$\mathsf{Reg}(\mathbf{u}; \mathbf{g}^T) \leq \sqrt{T\|\mathbf{u}\|^2 \ln(T\|\mathbf{u}\|^2/(e\sqrt{\pi}\mathsf{W}_0^2) + 1)} + \mathsf{W}_0.$$

6.3 Main Results

In what follows, we will illustrate how to incorporate (multiple) sequential side information based on coin betting algorithms in OLO over Hilbert space with an analogous guarantee by extending the aforementioned algorithmic reduction and guarantee translation. In doing so, we will leverage the connection between coin betting and compression, and adopt universal compression techniques beyond the KT strategy, namely per-state adaptation (Section 6.3.1), mixture (Section 6.3.2), and context tree weighting techniques (Section 6.3.3). For each case, we will first define a potential function and introduce a corresponding vectorial betting which guarantees the cumulative wealth to be at least the desired potential function.

6.3.1 OLO with Single Side Information via Product Potential

We consider the scenario when a (discrete) side information $H = (h_t \in [S])_{t \geq 1}$ is sequentially available for some $S \geq 1$. That is, at each round t, the side information h_t is revealed before the plays. As motivated in the introduction, the canonical example is a *causal* side information based on the history $\mathbf{g}^{t-1}$ such as a quantization of $\mathbf{g}_{t-D}^{t-1}$ for some $D \geq 1$. Yet another example is side information given by an oracle with foresight such as $h_t = \operatorname{sgn}(\langle \mathbf{g}_t, \mathbf{f} \rangle)$, i.e., the sign of the correlation between a fixed vector $\mathbf{f} \in V$

and the incoming symbol $\mathbf{g}_t$, as a rough hint to the future.

We define an *adaptive competitor with respect to the side information* H, denoted as $\mathbf{u}_{1:S}[H]$ for an S-tuple $\mathbf{u}_{1:S} := (\mathbf{u}_1, \dots, \mathbf{u}_S) \in V \times \cdots \times V$, to play $\mathbf{u}_{h_t}$ at time t, and let $\mathcal{C}[H] := \{\mathbf{u}_{1:S}[H] \colon \mathbf{u}_{1:S} \in V \times \cdots \times V\}$ denote the collection of all such adaptive competitors.

We first observe that the cumulative loss incurred by an adaptive competitor $\mathbf{u}_{1:S}[H] \in \mathcal{C}[H]$ can be decomposed with respect to the *states* defined by the side information symbols, i.e.,

$$\sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{u}_{h_t} \rangle = \sum_{s=1}^{S} \Big\langle \sum_{t \in [T] : h_t = s} \mathbf{g}_t, \mathbf{u}_s \Big\rangle.$$

Hence, a naive solution is to run independent OGD algorithms for each subsequence $\mathbf{g}^t(s; h^t) := (\mathbf{g}_i \colon h_i = s, i \in [t])$ sharing the same side information $s \in [S]$; it is straightforward to show that the per-state OGD with optimal learning rates achieves the regret of order $O(\sum_{s=1}^{S} \|\mathbf{u}_s\| \sqrt{T_s})$ with knowing the competitor norms $\|\mathbf{u}\|_{1:S}$. Like the per-state OGD algorithm, we can also extend other parameter-free algorithms such as DFEG (Orabona, 2013) and AdaNormal (McMahan and Orabona, 2014) to adapt to side information; see Appendix 6.B. This is what we call the *per-state extension* of an OLO algorithm.

Here, we propose a different type of parameter-free per-state algorithm based on coin betting. To compete against any adaptive competitor from $\mathcal{C}[H]$, we define a *product KT potential function*

$$\begin{aligned} \Psi^{\mathsf{KT}}(\mathbf{g}^t; h^t) &:= \prod_{s \in [S]} \Psi^{\mathsf{KT}}(\mathbf{g}^t(s; h^t)) \\ &= \prod_{s \in [S]} \psi_{t_s}^{\mathsf{KT}}(\| \textstyle\sum \mathbf{g}^t(s; h^t) \|), \end{aligned}$$

where $t_s := |\mathbf{g}^t(s; h^t)|$ for each $s \in [S]$. Note that $\Psi^{\mathsf{KT}}(\mathbf{g}^t; h^t)$ is a function of the summa-

tions of the subsequences $(\sum \mathbf{g}^t(1;h^t),\ldots,\sum \mathbf{g}^t(S;h^t))$. For each time t, we then define the vectorial KT betting with side information h^t as the application of the vectorial KT betting onto the subsequence corresponding to the current side information symbol h_t, i.e.,

$$\mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1};h^t) := \mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1}(h_t;h^{t-1})).$$

Unlike the other per-state extensions which play independent actions for each state thus allowing straightforward analyses, the per-state KT actions

$$\mathbf{w}_t^{\mathsf{KT}}(\mathbf{g}^{t-1};h^t) = \mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1};h^t)\mathsf{W}_{t-1} \tag{6.2}$$

depend on all previous history $\mathbf{g}^{t-1}$ due to the wealth factor W_{t-1}. We can establish the following guarantee with the same line of argument in the proof of Theorem 6.2.1, by analyzing the Fenchel dual of $\Psi^{\mathsf{KT}}(\mathbf{g}^t;h^t)$. Recall that for a multivariate function $\Psi\colon \mathbb{R}^d \to \mathbb{R}$, its Fenchel dual $\Psi^\star\colon \mathbb{R}^d \to \mathbb{R}$ is defined as

$$\Psi^\star(\mathbf{y}) := \sup_{\mathbf{x}\in\mathbb{R}^d} (\mathbf{y}^T\mathbf{x} - \Psi(\mathbf{x})).$$

Theorem 6.3.1. *For any side information $H = (h_t \in [S])_{t\geq 1}$ and any $\mathbf{g}_1,\ldots,\mathbf{g}_T \in \mathbb{B}$, let $\phi^{\mathsf{KT}}_{T_{1:S}}\colon \mathbb{R}^S \to \mathbb{R}$ be the Fenchel dual of the function*

$$(f_1,\ldots,f_S) \mapsto \prod_{s\in[S]} \psi^{\mathsf{KT}}_{T_s}(f_s),$$

where $T_s := |\{t\in[T]\colon h_t = s\}|$. Then, the OLO algorithm

$$\mathbf{w}_t^{\mathsf{KT}}(\mathbf{g}^{t-1};h^t) := \mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1};h^t)\mathsf{W}_{t-1}$$

satisfies $\mathsf{W}_T \geq \mathsf{W}_0 \Psi^{\mathsf{KT}}(\mathbf{g}^T; h^T)$*, and moreover*

$$\sup_{\mathbf{u}_{1:S}} \left\{ \mathsf{Reg}(\mathbf{u}_{1:S}[H]; \mathbf{g}^T) - \mathsf{W}_0 \phi^{\mathsf{KT}}_{T_{1:S}} \left(\frac{\|\mathbf{u}\|_{1:S}}{\mathsf{W}_0} \right) \right\} \leq \mathsf{W}_0.$$

In particular, for any $\mathbf{u}_{1:S}[H] \in \mathcal{C}[H]$*,*

$$\mathsf{Reg}(\mathbf{u}_{1:S}[H]; \mathbf{g}^T) = \mathsf{W}_0 + \tilde{O}\left(\sqrt{\sum_{s=1}^{S} T_s \|\mathbf{u}_s\|^2} \right). \tag{6.3}$$

Example 6.3.2. *Recall the "easy" adversarial sequence* $\mathbf{g}^T = (\mathbf{g}, -\mathbf{g}, \mathbf{g}, \ldots, -\mathbf{g})$ *for some* $\mathbf{g} \in \mathbb{B}$ *previously considered in the introduction. For a side information* $h_t = \operatorname{sgn}(\langle \mathbf{g}_t, \mathbf{f} \rangle)$ *with some* $\mathbf{f} \in V$*, Theorem 6.3.1 states that* $\mathsf{Reg}((\mathbf{u}_+, \mathbf{u}_-); \mathbf{g}^T) = \tilde{O}((\|\mathbf{u}_+\| + \|\mathbf{u}_-\|)\sqrt{T})$*, matching the regret guarantee of the optimally tuned per-state OGD up to logarithmic factors. Overall, the regret guarantee against adaptive competitors for the per-state KT method implies a much larger overall reward than was achieved by an algorithm competing against static competitors.*

Remark 6.3.3 (Cost of noninformative side information)**.** Consider a scenario where competitors of the form $\mathbf{u}_{1:S} = (\mathbf{u}, \ldots, \mathbf{u})$ with some vector $\mathbf{u} \in V$ perform best; in this case, an algorithm without adapting to side information may suffice for optimal regret guarantees. Even in such cases with *noninformative* side information, the dominant factor in the regret remains the same as the regret guarantee with respect to the static competitor class, since $\sum_{s=1}^{S} T_s \|\mathbf{u}_s\|^2 = T\|\mathbf{u}\|^2$.

Remark 6.3.4 (Effect of large S)**.** While side information with larger S may provide more levels of granularity, too large S may degrade the performance of the per-state algorithms. Intuitively, if $S \gg 1$, it is likely that we will see each state only few times, which results in poor convergence for almost every state. These are also captured in the regret guarantee; we note that the hidden logarithmic factor of the regret bound (6.3) might incur a multiplicative factor of at most $O(\sqrt{S})$. Similarly, in the optimal regret

attained by the per-state OGD, we have $O(\sum_{s=1}^{S} \|\mathbf{u}_s\|\sqrt{T_s}) \le O(\max_{s\in[S]} \|\mathbf{u}_s\|\sqrt{ST})$.

6.3.2 OLO with Multiple Side Information via Mixture of Product Potentials

Now suppose that multiple side information sequences

$$\{H^{(m)} = (h_t^{(m)} \in S^{(m)})_{t\ge 1} \colon m \in [M]\}$$

are sequentially available; for example, each $H^{(m)}$ can be either constructed based on a different quantizer $Q_m \colon V \to \{1, \bar{1}\}$ and/or based on the history $\mathbf{g}_{t-D_m}^{t-1}$ of different lengths $D_m \ge 0$, each of which aims to capture a different structure of $(\mathbf{g}_t)$. In this setting, we aim to minimize the *worst* regret among all possible side information, i.e.,

$$\max_{m\in[M]} \mathsf{Reg}(\mathbf{u}_{1:S^{(m)}}[H_m]; \mathbf{g}^T) = \sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{w}_t\rangle - \min_{m\in[M]} \sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{u}_{h_{mt}}^{(H)}\rangle, \tag{6.4}$$

which is equivalent to aiming to follow the best side information in hindsight.

We first remark that Cutkosky (2019) recently proposed a simple black-box meta algorithm that combines multiple OLO algorithms achieving the best regret guarantee, which can also be applied to solving this multiple side information problem. For example, for algorithms $(\mathcal{A}_m)_{m\in[M]}$ each of which play an action $\mathbf{w}_t^{(m)}$, the meta algorithm $\mathcal{A}$ which we refer to the *addition* plays $\mathbf{w}_t = \sum_{m=1}^{M} \mathbf{w}_t^{(m)}$ and guarantees the regret

$$\mathsf{Reg}_T^{\mathcal{A}}(\mathbf{u}) \le \varepsilon + \min_{m\in[M]} \mathsf{Reg}_T^{\mathcal{A}_m}(\mathbf{u}),$$

provided that $\mathcal{A}_m$'s suffer at most constant regret ε against $\mathbf{u} = 0$; the same guarantee also hold for adaptive competitors.

Rather, we propose the following information theoretic solution. For each side information sequence $H^{(m)}$, we can apply the per-state KT algorithm from the previous

section, which guarantees the wealth lower bound $\mathsf{W}_0\Psi^{\mathsf{KT}}(\mathbf{g}^t;(h^{(m)})^t)$. To achieve the best among the per-state KT algorithms, we consider the *mixture potential*

$$\Psi^{\mathsf{mix}}(\mathbf{g}^t;\mathbf{h}^t)=\sum_{m=1}^{M} w_m\Psi^{\mathsf{KT}}(\mathbf{g}^t;(h^{(m)})^t)$$

for some $w_1,\dots,w_M>0$ such that $\sum_{m=1}^M w_m=1$. Here, $\mathbf{h}_t:=(h_t^{(1)},\dots,h_t^{(M)})$ denotes the side information vector revealed at time t. When there exists no prior belief on how useful each side information is, one can choose the uniform weight $w_1=\ldots=w_M=1/M$ by default. Now, define the *vectorial mixture betting* given $\mathbf{g}^{t-1}$ and $\mathbf{h}^t$ as

$$\begin{aligned}\mathbf{v}^{\mathsf{mix}}(\mathbf{g}^{t-1};\mathbf{h}^t)&:=\frac{\mathbf{u}^{\mathsf{mix}}(\mathbf{g}^{t-1};\mathbf{h}^t)}{\Psi^{\mathsf{mix}}(\mathbf{g}^{t-1};\mathbf{h}^{t-1})},\quad\text{where}\\ \mathbf{u}^{\mathsf{mix}}(\mathbf{g}^{t-1};\mathbf{h}^t)&:=\sum_{m=1}^{M} w_m\Psi^{\mathsf{KT}}(\mathbf{g}^{t-1};(h^{(m)})^{t-1})\mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1};(h^{(m)})^t),\end{aligned}$$

and finally define the *mixture OLO* algorithm by the action

$$\mathbf{w}_t^{\mathsf{mix}}(\mathbf{g}^{t-1};\mathbf{h}^t):=\mathbf{v}^{\mathsf{mix}}(\mathbf{g}^{t-1};\mathbf{h}^t)\mathsf{W}_{t-1}. \tag{6.5}$$

In the language of gambling, the mixture strategy bets by distributing her wealth based on the weights w_m's to strategies, each of which is tailored to a side information sequence, and thus can guarantee at least w_m times the cumulative wealth attained by the m-th strategy following $H^{(m)}$ for any $m\in[M]$.

Theorem 6.3.5. *For any side information $H^{(1)},\dots,H^{(M)}$ and any $\mathbf{g}_1,\dots,\mathbf{g}_T\in\mathbb{B}$, the mixture OLO algorithm* (6.5) *satisfies $\mathsf{W}_T\ge\mathsf{W}_0\Psi^{\mathsf{mix}}(\mathbf{g}^T;\mathbf{h}^T)$, and moreover for any $m\in[M]$, we have*

$$\sup_{\mathbf{u}_{1:S^{(m)}}}\left\{\mathsf{Reg}(\mathbf{u}_{1:S^{(m)}}[H^{(m)}];\mathbf{g}^T)-w_m\mathsf{W}_0\phi^{\mathsf{KT}}_{T_{1:S^{(m)}}}\left(\frac{\|\mathbf{u}\|_{1:S^{(m)}}}{w_m\mathsf{W}_0}\right)\right\}\le w_m\mathsf{W}_0.$$

In other words, for any m and any $\mathbf{u}_{1:S^{(m)}}$, we have

$$\mathsf{Reg}(\mathbf{u}_{1:S^{(m)}}[H_m]; \mathbf{g}^T) = w_m \mathsf{W}_0 + \tilde{O}\left(\sqrt{\left(\ln \frac{1}{w_m}\right) \sum_{s_m=1}^{S_m} T_{s_m}^{(H_m)} \|\mathbf{u}_{s_m}^{(H_m)}\|^2}\right).$$

Remark 6.3.6 (Cost of mixture)**.** A mixture strategy adapts to any available side information with the cost of replacing W_0 with $w_m \mathsf{W}_0$ in the regret guarantee for each $m \in [M]$. Since the dependence of regret on W_0 scales as $O(\sqrt{\ln(1 + 1/\mathsf{W}_0)} + \mathsf{W}_0)$ from Theorem 6.3.1, a small w_m may degrade the quality of the regret guarantee by only a small multiplicative factor $O(\sqrt{\ln(1/w_m)})$.

Remark 6.3.7 (Comparison to the addition technique)**.** While the mixture algorithm attains a similar guarantee to the addition technique (Cutkosky, 2019), it is only applicable to coin betting based algorithms and requires a rather sophisticated aggregation step. Thus, if there are only moderate number of side information sequences, the addition of per-state parameter-free algorithms suffices. The merit of mixture will become clear in the next section in the tree side information problem of combining $O(2^{2^D})$ many components for a depth parameter $D \geq 1$, while a naive application of the addition technique to the tree problem is not feasible due to the number of side information; see Section 6.5 for an alternative solution with the addition technique.

6.3.3 OLO with Tree Side Information

In this section, we formally define and study a tree-structured side information H, which was illustrated in the introduction. We suppose that there exists an auxiliary binary sequence $\Omega = (\omega_t \in \{\pm 1\})_{t \geq 1}$, which is revealed one-by-one at the *end* of each round; hence, a learner has access to ω^{t-1} when deciding an action at round t. In the motivating problem in the introduction, such an auxiliary sequence was constructed as $\omega_t := Q(\mathbf{g}_t)$ with a fixed binary quantizer $Q \colon V \to \{\pm 1\}$.

Markov Side Information

Given $\Omega = (\omega_t)_{t\geq 1}$, the most natural form of side information is the *depth-D Markov side information* $h_t := \omega_{t-D}^{t-1} \in \{\pm 1\}^D$, i.e., the last D bits of $(\omega_t)_{t\geq 1}$—note that it can be mapped into a perfect binary tree of depth D with 2^D possible states.

Example 6.3.8. *As an illustrative application of the mixture algorithm and a precursor to the tree side information problem, suppose that we wish to compete with any Markov side information of depth $\leq D$. Then, there are $D+1$ different side information, one for each depth $d = 0, \ldots, D$; for simplicity, assume uniform weights $w_d = 1/(D+1)$ for each depth d. Then, Theorem 6.3.5 guarantees that the mixture OLO algorithm* (6.5) *satisfies, for any depth $d = 0, \ldots, D$,*

$$\mathsf{Reg}(\mathbf{u}_{1:2^d}^{(d)}; \mathbf{g}^T) = \frac{\mathsf{W}_0}{D+1} + \tilde{O}\left(\sqrt{\ln(D+1)\sum_{s=1}^{2^d} T_s^{(d)} \|\mathbf{u}_s^{(d)}\|^2}\right)$$

for any competitor $\mathbf{u}_{1:2^d}^{(d)} \in V^{2^d}$, where we identify 2^d possible states by $1, \ldots, 2^d$ and $T_s^{(d)}$ is the number of time steps with s as side information.

While a larger D can capture a longer dependence in the sequence, however, the performance of a per-state algorithm could significantly degrade due to the exponential number of states as pointed out in Remark 6.3.4.

Tree-Structured Side Information

The limitation of Markov side information motivates a general *tree-structured side information* (or tree side information in short). Informally, we say that a sequence has a *depth-D tree structure* if the state at time t depends on at most D of the previous occurrences, corresponding to a full binary tree of depth D; see Figure 6.1.1. This degree of freedom allows to consider different lengths of history for each state, leading to the terminology *variable-order Markov structure*, as opposed to the previous *fixed-order Markov*

structure. If an underlying structure is approximately captured by a tree structure of depth D with the number of leaves far fewer than 2^D, the corresponding per-state algorithm can enjoy a much lower regret guarantee.

We now formally define a tree side information. We say that a string $\omega_{1-l}\omega_{2-l}\dots\omega_0$ is a *suffix* of a string $\omega'_{1-l'}\omega'_{2-l'}\dots\omega'_0$, if $l \le l'$ and $\omega_{-i} = \omega'_{-i}$ for all $i \in \{0,\dots,l-1\}$. Let λ denote the empty string. We define a *(binary) suffix set* $\mathbf{T}$ as a set of binary strings that satisfies the following two properties (Willems et al., 1995): (1) Properness: no string in $\mathbf{T}$ is a suffix of any other string in $\mathbf{T}$; (2) Completeness: every semi-infinite binary string $\dots h_{t-2}h_{t-1}h_t$ has a suffix from $\mathbf{T}$. Since there exists an one-to-one correspondence between a binary suffix set and a full binary tree, we also call $\mathbf{T}$ a *suffix tree*. Given $D \ge 0$, let $\mathscr{T}_{\le D}$ denote the set of all suffix trees of depth at most D.

For a suffix tree $\mathbf{T} \in \mathscr{T}_{\le D}$, we define a *tree side information* $H_{\mathbf{T};\Omega}$ with respect to $\mathbf{T}$ and $\Omega = (\omega_t)_{t\ge 1}$ as the matching suffix from the auxiliary sequence. We can also identify h_t, the tree side information defined by $\mathbf{T}$ at time t, with a unique leaf node $s_t^{\mathbf{T}} \in \mathbf{T}$. For example, if a suffix set $\mathbf{T}$ consists of all possible 2^D binary strings of length $D \ge 1$, then it boils down to the fixed-order Markov case $h_t = \omega_{t-D}^{t-1}$.

For a single tree $\mathbf{T}$, the goal is to keep the regret

$$\mathsf{Reg}(\mathbf{u}[\mathbf{T}]; \mathbf{g}^T) := \sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u}_{s_t^{\mathbf{T}}}^{\mathbf{T}} \rangle$$

small for any competitor $\mathbf{u}[\mathbf{T}] := (\mathbf{u}_s^{\mathbf{T}})_{s\in\mathbf{T}}$. In the next two subsections, we aim to follow the performance of the *best suffix tree* of depth at most D, or equivalently, to keep the worst regret $\max_{\mathbf{T}\in\mathscr{T}_{\le D}} \mathsf{Reg}_{\mathcal{A}}(\mathbf{u}[\mathbf{T}]; \mathbf{g}^t)$ small for any collection of competitors $(\mathbf{u}[\mathbf{T}])_{\mathbf{T}\in\mathscr{T}_{\le D}}$.

Remark 6.3.9 (Matching Lower Bound)**.** When the auxiliary sequence Ω is constructed from a binary quantizer Q with the history $\mathbf{g}^{t-1}$ as mentioned earlier, we can show an optimality of the per-state KT algorithm in Section 6.3 for a single tree by establishing a

matching regret lower bound extending the technique of Orabona (2019, Theorem 5.12); see Appendix 6.C.2.

Below, we will use the *tree potential* with respect to $\mathbf{T}$ and Ω defined as

$$\Psi^{\mathsf{KT}}(\mathbf{g}^t; \mathbf{T}, \Omega) := \prod_{s \in \mathbf{T}} \Psi^{\mathsf{KT}}(\mathbf{g}^t(s; \Omega)),$$

where we write $s \in \mathbf{T}$ for any leaf node s of the tree $\mathbf{T}$ with a slight abuse of notation and we define

$$\mathbf{g}^t(s; \Omega) := (\mathbf{g}_i \colon s \text{ is a suffix of } \omega_{i-D}^{i-1},\ 1 \le i \le t).$$

From now on, we will hide any dependence on Ω whenever the omission does not incur confusion.

Context Tree Weighting for OLO with Tree Side Information

To compete against the best competitor adaptive to *any* tree side information of depth $\le D$, a natural solution is to consider a mixture of all tree potentials; note, however, that there are doubly-exponentially many $O(2^{2^D})$ possible suffix trees of depth $\le D$, and thus it is not computationally feasible to compute such a mixture naively. Instead, inspired by the context tree weighting (CTW) probability assignment of Willems et al. (1995), we analogously define the CTW potential as $\Psi^{\mathsf{CTW}}(\mathbf{g}^t) := \Psi_{\lambda}^{\mathsf{CTW}}(\mathbf{g}^t)$ with a recursive formula

$$\Psi_s^{\mathsf{CTW}}(\mathbf{g}^t) := \begin{cases} \frac{1}{2}\Psi_s^{\mathsf{KT}}(\mathbf{g}^t) + \frac{1}{2}\Psi_{\bar{1}s}^{\mathsf{CTW}}(\mathbf{g}^t)\Psi_{1s}^{\mathsf{CTW}}(\mathbf{g}^t) & \text{if } |s| < D \\ \Psi_s^{\mathsf{KT}}(\mathbf{g}^t) & \text{if } |s| = D \end{cases} \tag{6.6}$$

for any binary string s of length $\le D$ and $\Psi_s^{\mathsf{KT}}(\mathbf{g}^t) := \Psi^{\mathsf{KT}}(\mathbf{g}^t(s))$. Conceptually, this recursion can be performed over the perfect suffix tree of depth D, which we denote by $\mathcal{T}_D$ and call the context tree of depth D; see Figure 6.3.1 for the context tree of depth

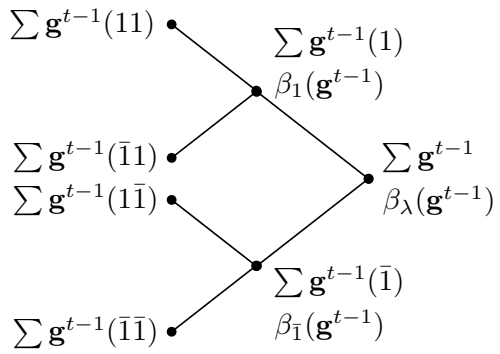


Figure 6.3.1. A context tree of depth 2.

$D = 2$. Following the same logic of Willems et al. (1995), one can easily show that

$$\Psi^{\mathsf{CTW}}(\mathbf{g}^t) = \sum_{\mathbf{T} \in \mathscr{T}_{\leq D}} w(\mathbf{T}) \Psi^{\mathsf{KT}}(\mathbf{g}^t; \mathbf{T})$$

for $w(\mathbf{T}) = 2^{-\Gamma_D(\mathbf{T})}$, where $\Gamma_D(\mathbf{T}) := 2|\mathbf{T}| - 1 - |\{s \in \mathbf{T} \colon |s| = D\}|$ is a complexity measure of a full binary tree $\mathbf{T}$ of depth $\leq D$, $|\mathbf{T}|$ denotes the number of leaf nodes of a full binary tree $\mathbf{T}$, and $\mathscr{T}_{\leq D}$ denotes the set of all suffix trees of depth $\leq D$.

For a path ρ from the root to a leaf node of $\mathcal{T}_D$ and a full binary tree $\mathbf{T}$, we let $s_{\mathbf{T}}(\rho)$ denote the unique leaf node of $\mathbf{T}$ that intersects with the path ρ. We also define $\mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1}; \mathbf{T}) := \mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1}(s_{\mathbf{T}}(\omega_{t-D}^{t-1})))$. Then, based on the construction of the vectorial betting for a mixture potential in Section 6.3.2, we define the vectorial CTW betting

$$\begin{aligned} \mathbf{v}^{\mathsf{CTW}}(\mathbf{g}^{t-1}) &:= \frac{\mathbf{u}^{\mathsf{CTW}}(\mathbf{g}^{t-1})}{\Psi^{\mathsf{CTW}}(\mathbf{g}^{t-1})}, \qquad \text{where} \\ \mathbf{u}^{\mathsf{CTW}}(\mathbf{g}^{t-1}) &:= \sum_{\mathbf{T} \in \mathscr{T}_{\leq D}} w(\mathbf{T}) \Psi^{\mathsf{KT}}(\mathbf{g}^{t-1}; \mathbf{T}) \mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1}; \mathbf{T}), \end{aligned} \tag{6.7}$$

then we define the CTW OLO algorithm as the action

$$\mathbf{w}^{\mathsf{CTW}}(\mathbf{g}^{t-1}) := \mathbf{v}^{\mathsf{CTW}}(\mathbf{g}^{t-1}) \mathsf{W}_{t-1}(\mathbf{g}^{t-1}). \tag{6.8}$$

By Theorem 6.3.5, we readily have the regret guarantee of the CTW OLO algorithm as follows:

Corollary 6.3.10. *Let $D \geq 0$ be fixed. For any $\mathbf{g}_1, \dots, \mathbf{g}_T \in \mathbb{B}$, the CTW OLO algorithm* (6.8) *satisfies $\mathsf{W}_T \geq \mathsf{W}_0 \Psi^{\mathsf{CTW}}(\mathbf{g}^T)$. Moreover, we have*

$$\mathsf{Reg}(\mathbf{u}[\mathbf{T}]; \mathbf{g}^T) = w(\mathbf{T})\mathsf{W}_0 + \tilde{O}\left(\sqrt{\left(\ln \frac{1}{w(\mathbf{T})}\right) \sum_{s \in \mathbf{T}} T_s^{\mathbf{T}} \|\mathbf{u}_s^{\mathbf{T}}\|^2}\right)$$

for any tree $\mathbf{T} \in \mathscr{T}_{\leq D}$, where $T_s^{\mathbf{T}}$ denotes the number of occurrences of a side information symbol $s \in \mathbf{T}$ with respect to the tree side information $H_{\mathbf{T};\Omega}$.

Hence, the CTW OLO algorithm (6.8) can tailor to the best tree side information in hindsight. Now, the remaining question is: can we *efficiently* compute the vectorial CTW betting (6.7)? As a first attempt, the summation over the trees $\mathbf{T} \in \mathscr{T}_{\leq D}$ in (6.7) can be naively computed via a similar recursive formula as (6.6). We define

$$\rho(\omega_{t-D}^{t-1}) := \{\lambda, \omega_{t-1}, \dots, \omega_{t-D}^{t-1}\}$$

and call the *active nodes* given the side information suffix ω_{t-D}^{t-1}.

Proposition 6.3.11. *For each node s of $\mathcal{T}_D$, define*

$$\begin{aligned}
\mathbf{u}_s^{\mathsf{CTW}}(\mathbf{g}^{t-1}) &:= \begin{cases} \frac{1}{2}\Psi_s^{\mathsf{KT}}(\mathbf{g}^{t-1})\mathbf{v}_s^{\mathsf{KT}}(\mathbf{g}^{t-1}) + \frac{1}{2}\mathbf{u}_{\bar{1}s}^{\mathsf{CTW}}(\mathbf{g}^{t-1})\mathbf{u}_{1s}^{\mathsf{CTW}}(\mathbf{g}^{t-1}) & \text{if } |s| < D, \\ \Psi_s^{\mathsf{KT}}(\mathbf{g}^{t-1})\mathbf{v}_s^{\mathsf{KT}}(\mathbf{g}^{t-1}) & \text{if } |s| = D, \end{cases} \\
\mathbf{v}_s^{\mathsf{KT}}(\mathbf{g}^{t-1}) &:= \begin{cases} \mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1}(s)) & \text{if } s \in \rho(\omega_{t-D}^{t-1}) \\ 1 & \text{otherwise.} \end{cases}
\end{aligned} \tag{6.9}$$

Then, the recursion is well-defined, and $\mathbf{u}_\lambda^{\mathsf{CTW}}(\mathbf{g}^{t-1}) = \mathbf{u}^{\mathsf{CTW}}(\mathbf{g}^{t-1})$.

While the recursions (6.6) and (6.9) take $O(2^D)$ steps for computing a mixture of $O(2^{2^D})$ many tree potentials, they are still not feasible as an online algorithm even for a

moderate D. In the next section, we show that the per-round time complexity $O(2^D)$ can be significantly improved to $O(D)$ by exploiting the tree structure further.

The Efficient CTW OLO Algorithm with $O(D)$ Steps Per Round

(1) Compute $\mathbf{v}^{\mathsf{CTW}}$ in $O(D)$ steps

The key idea is that, given the suffix ω_{t-D}^{t-1}, the vector betting $\mathbf{v}^{\mathsf{CTW}} = \mathbf{u}^{\mathsf{CTW}}/\Psi^{\mathsf{CTW}}$ can be computed efficiently via the recursive formulas (6.6) and (6.9), by only traversing the active nodes $\rho(\omega_{t-D}^{t-1}) = \{\lambda, \omega_{t-1}, \dots, \omega_{t-D}^{t-1}\}$ in the context tree $\mathcal{T}_D$. In order to do so, we define

$$\beta_s(\mathbf{g}^{t-1}) := \frac{\Psi_s^{\mathsf{KT}}(\mathbf{g}^{t-1})}{\Psi_{1s}^{\mathsf{CTW}}(\mathbf{g}^{t-1})\Psi_{1s}^{\mathsf{CTW}}(\mathbf{g}^{t-1})} \tag{6.10}$$

for every *internal* node s of $\mathcal{T}_D$.

Proposition 6.3.12. *Define*

$$\mathbf{v}_{s_d}^{\mathsf{CTW}}(\mathbf{g}^{t-1}) := \begin{cases} \dfrac{\beta_{s_d}(\mathbf{g}^{t-1})}{\beta_{s_d}(\mathbf{g}^{t-1})+1}\mathbf{v}_{s_d}^{\mathsf{KT}}(\mathbf{g}^{t-1}) + \dfrac{1}{\beta_{s_d}(\mathbf{g}^{t-1})+1}\mathbf{v}_{s_{d+1}}^{\mathsf{CTW}}(\mathbf{g}^{t-1}) & \text{if } d < D \\ \mathbf{v}_{s_D}^{\mathsf{KT}}(\mathbf{g}^{t-1}) & \text{if } d = D \end{cases} \tag{6.11}$$

for $s_d = \omega_{t-d}^{t-1} \in \mathcal{T}_D$, $d = 0, \dots, D$. Then, $\mathbf{v}^{\mathsf{CTW}}(\mathbf{g}^{t-1}) = \mathbf{v}_{\lambda}^{\mathsf{CTW}}(\mathbf{g}^{t-1})$.

Hence, if we can store $\sum \mathbf{g}^{t-1}(s)$ and the value $\beta_s(\mathbf{g}^{t-1})$ as defined in (6.10) for every node s of $\mathcal{T}_D$, we can compute $\mathbf{v}^{\mathsf{CTW}}$ in $O(D)$.

(2) Update β_s in $O(D)$ steps

Upon receiving $\mathbf{g}_t$, we need to update $\beta_{s_d}(\mathbf{g}^{t-1})$ as

$$\beta_{s_d}(\mathbf{g}^{t}) = \beta_{s_d}(\mathbf{g}^{t-1})\frac{\Psi_{s_d}^{\mathsf{KT}}(\mathbf{g}^{t})}{\Psi_{s_d}^{\mathsf{KT}}(\mathbf{g}^{t-1})}\frac{\Psi_{s_{d+1}}^{\mathsf{CTW}}(\mathbf{g}^{t-1})}{\Psi_{s_{d+1}}^{\mathsf{CTW}}(\mathbf{g}^{t})} \tag{6.12}$$

for each $s_d = \omega_{t-d}^{t-1} \in \mathcal{T}_D$. Here, the ratio $\Psi_{s_d}^{\mathsf{CTW}}(\mathbf{g}^{t})/\Psi_{s_d}^{\mathsf{CTW}}(\mathbf{g}^{t-1})$ can be also computed efficiently while traversing the path $\rho(\omega_{t-D}^{t-1})$ from the leaf node s_D to the root $s_0 = \lambda$, based on the following recursion:

Proposition 6.3.13. *For each node $s_d = \omega_{t-d}^{t-1} \in \mathcal{T}_D$, $d = 0, \ldots, D$,*

$$\frac{\Psi_{s_d}^{\mathsf{CTW}}(\mathbf{g}^t)}{\Psi_{s_d}^{\mathsf{CTW}}(\mathbf{g}^{t-1})} = \begin{cases} \dfrac{\beta_{s_d}(\mathbf{g}^{t-1})}{\beta_{s_d}(\mathbf{g}^{t-1})+1} \dfrac{\Psi_{s_d}^{\mathsf{KT}}(\mathbf{g}^t)}{\Psi_{s_d}^{\mathsf{KT}}(\mathbf{g}^{t-1})} & \\ \quad + \dfrac{1}{\beta_{s_d}(\mathbf{g}^{t-1})+1} \dfrac{\Psi_{s_{d+1}}^{\mathsf{CTW}}(\mathbf{g}^t)}{\Psi_{s_{d+1}}^{\mathsf{CTW}}(\mathbf{g}^{t-1})} & \text{if } d < D \cdot \\ \dfrac{\Psi_{s_D}^{\mathsf{KT}}(\mathbf{g}^t)}{\Psi_{s_D}^{\mathsf{KT}}(\mathbf{g}^{t-1})} & \text{if } d = D \end{cases} \tag{6.13}$$

Hence, updating β_s's can be also performed efficiently in $O(D)$ time. The space complexity of this algorithm is $O(DT)$, since there can be at most D nodes activated for the first time at each round. The complete algorithm is summarized in Algorithm 6.D.3 in Appendix.

6.4 Experiments

To validate the motivation of this work and demonstrate the power of the proposed algorithms in online convex optimization, we performed online linear regression with absolute loss following Orabona and Pál (2016). We observed, however, that the datasets considered therein do not contain any temporal dependence and thus the proposed algorithms did not prove useful (data not shown). Instead, we chose two real-world temporal datasets (Beijing PM2.5 (Liang et al., 2015) and Metro Interstate Traffic Volume (Hogue, 2019)) from the UCI machine learning repository (Dua and Graff, 2019). All details including data preprocessing can be found in Appendix 6.E and the code that fully reproduce the results is available at https://github.com/jongharyu/olo-with-side-information.

To construct auxiliary sequences, we used the *canonical binary quantizers* $Q_{\mathbf{e}_i}$, where $\mathbf{e}_i$ denotes the i-th standard vector. We first ran the per-state versions of OGD, AdaNormal (McMahan and Orabona, 2014), DFEG (Orabona, 2013), and KT with Markov side information of different depths and ran the CTW algorithm for the maxi-

mum depth ranging $0, 1, 3 \ldots, 11$. We optimally tuned the per-state OGD using only a single rate for all states due to the prohibitively large complexity of the optimal grid search; see Figures 6.E.1(a) and 6.E.2(a) in Appendix. While the per-state KT consistently showed the best performance, the performance degraded as we used too deep Markov side information beyond some threshold for all algorithms. In Figures 6.E.1(b) and 6.E.2(b) in Appendix, CTW often achieved even better performance than the best performance achieved by KT across the different choices of quantizer, also being robust to the choice of the maximum depth.

In practice, however, we do not know which dimension to quantize a priori. Hence, we showed the performance of the combined CTW algorithms over all d quantizers aggregated by either the mixture or the addition—conceptually, the mixture of CTWs can be viewed as a *context forest weighting*. As a benchmark, we also ran the combined KT algorithms over all d quantizers for each depth. In Figure 6.4.1, we summarized the per-coordinate results by taking the best performance over all quantizers; see the first five dashed lines in the legend. While these are only hypothetical which were not attained by an algorithm, surprisingly, the combined CTW algorithms over different quantizers, either by the mixture or the addition of Cutkosky (2019), achieved the hypothetically best performance (plotted solid).

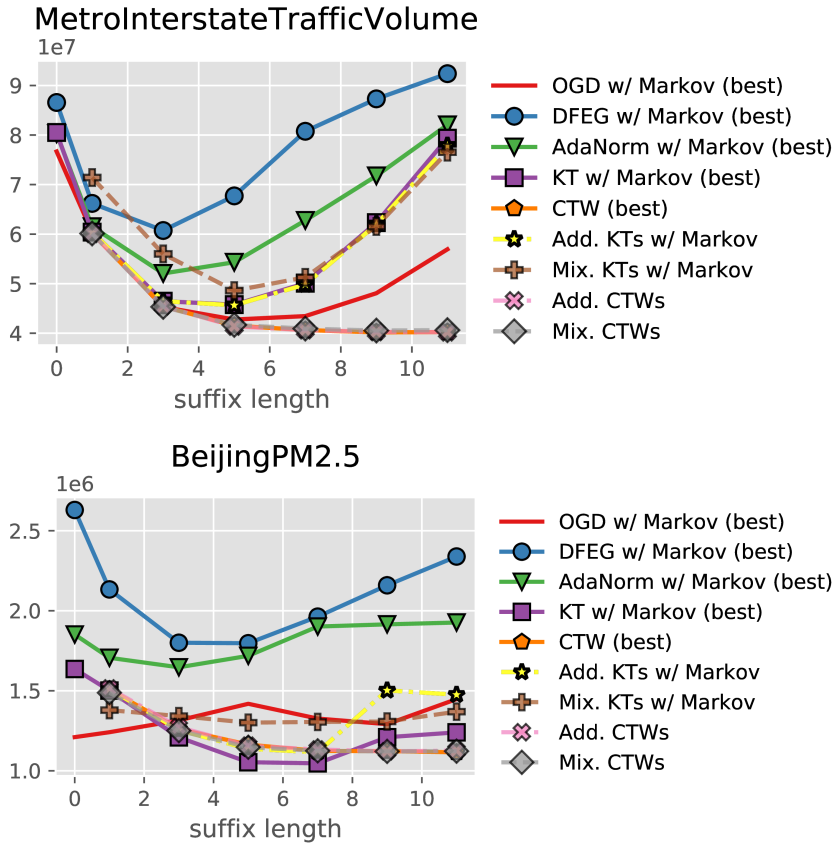


Figure 6.4.1. Summary of the experiments.

6.5 Concluding Remarks

Aiming to leverage a temporal structure in the sequence g^n, we developed the CTW OLO algorithm that can efficiently adapt to the best tree side information in hindsight by combining a universal coin betting based OLO algorithm and universal compression (or prediction) techniques from information theory. Experimental results demonstrate that the proposed framework can be effective in solving real-life online convex optimization problems.

The key technical contribution of the paper is to consider the product and mixture potentials, motivated from information theory, and to adapt the CTW algorithm of Willems et al. (2006) to online linear optimization in Hilbert spaces. Main technical difficulties lie in analyzing the product potential (Proposition 6.C.15) and properly

invoking Rissanen's lower bound in Theorem 6.C.7 to establish the optimality.

We remark that an anonymous reader of an earlier version of this manuscript proposed a simpler alternative approach based on a meta algorithm that recasts any parameter-free OLO algorithm for tree-structured side information. The idea is to combine the specialist framework of Freund et al. (1997) and apply the addition technique of Cutkosky (2019). Running a base OLO algorithm at each node of a context tree as a specialist, the meta algorithm adds up the outputs of the specialists on the active path at each round and updates them at the end of the round. This approach achieves a similar regret guarantee of the CTW OLO (Corllary 6.3.10) with the same complexity. A detailed study is beyond the scope of this paper and thus left as future work.

Appendix

6.A Related Work

There have been several parameter-free methods proposed for OLO in Hilbert space (McMahan and Orabona, 2014; Orabona, 2013, 2014; Orabona and Pál, 2016) as well as learning with expert advice (LEA) (Chaudhuri et al., 2009; Chernov and Vovk, 2010; Foster et al., 2015; Freund and Schapire, 1997; Koolen and Van Erven, 2015; Luo and Schapire, 2015; Orabona and Pál, 2016); see also (Orabona, 2019, Chapter 9) and the references therein. A parallel line of work on parameter-free methods considers the case when the maximum norm of $\mathbf{g}_t$ (often referred to as the *Lipschitz constant*), which is assumed to be 1 throughout in this paper, is unknown but the competitor norm $\|\mathbf{u}\|$ is known (Cutkosky and Boahen, 2017; Duchi et al., 2011). Recently, Chen et al. (2021); Zhang et al. (2021) studied a similar setting in this paper, albeit establishing guarantees only for bounded domains. We remark that AdaNormalHedge (Luo and Schapire, 2015) is a parameter-free LEA algorithm which can compete with mixtures of forcasters with side information, in particular tree experts via mixtures of sleeping experts; for example, Kuzborskij and Cesa-Bianchi (2020) used AdaNormalHedge with tree experts for binary classification with absolute loss. For a comprehensive overview of these parameter-free methods, see the tutorial (Orabona and Cutkosky, 2020).

The connection between OLO and gambling was shown by Orabona and Pál (2016), where they also described a reduction for LEA. This idea was also applied to training deep neural networks (Orabona and Tommasi, 2017). While the proposed

algorithms in this paper are against *stationary* competitors, Jun et al. (2017) proposed a coin betting based OLO algorithm against nonstationary competitors characterized by a sequence of vectors $\mathbf{u}_1, \ldots, \mathbf{u}_T$ such that have at most m change points. Van der Hoeven et al. (2018, Section 5 and particularly Theorem 9) establishes a connection between the exponential weights (EW) algorithm and the coin-betting scheme. Earlier on in the paper, in Section 2 the interpretation of compression as a special case of EW with $\eta = 1$ is provided as well. Similarly, Jun and Orabona (2019) utilize such a connection as well. To the best of our knowledge, however, we did not find a clear bridge constructed between compression and coin-betting methods in either, even though a careful examination of the mathematical details may hint toward this connection.

Universal compression, which is a classical topic in information theory, aims to compress sequences with no (or very little) statistical assumptions. In the last century, there have been several techniques proposed that can compete against the best i.i.d. compressor (Krichevsky and Trofimov, 1981; Rissanen, 1984; Xie and Barron, 1997), finite state compressor (Ziv and Lempel, 1977) and tree compressor (Willems et al., 1995). The CTW probability assignment invented by (Willems et al., 1995) has been one of the most successful and widely used universal compression techniques. Beyond compression, this technique has been applied to estimation of directed information (Jiao et al., 2013), universal portfolios (Kozat et al., 2008), and reinforcement learning (Messias and Whiteson, 2018), to name a few. The efficient CTW OLO algorithm presented in Section 6.3.3 is in the spirit of the processing betas algorithm proposed by Willems et al. (2006) for computing the predictive conditional probability induced by the CTW probability assignment (Willems et al., 1995). Cesa-Bianchi and Lugosi (2006, Section 5.3) also presented a CTW-based Hedge algorithm for LEA; see bibliographic remarks therein for other applications of CTW to learning problems.

A related line of recent work on online learning with hints (Bhaskara et al., 2020a,b; Dekel et al., 2017) considers a scenario where the learner receives a vector

$\mathbf{h}_t$ with $\|\mathbf{h}_t\| = 1$ such that $\langle \mathbf{h}_t, \mathbf{g}_t/\|\mathbf{g}_t\| \rangle \geq \alpha > 0$ as a "hint" to the future. However, our setting is not directly comparable, since we only consider a finite side information and this line of work aims to establish small regret $o(\sqrt{T})$ measured with respect to static competitors. We also remark that Rakhlin and Sridharan (2013) studied the problem of OLO when $\mathbf{g}_t$ is modelled as a "predictable" sequence, in the sense that $\mathbf{g}_t = M(\mathbf{g}^{t-1}) + \mathbf{n}_t$ with some adversarial noise $\mathbf{n}_t$ with a (possibly randomized) function M; yet, they considered static competitors unlike this work.

6.B Per-State Extensions of Existing Algorithms

Here we present per-state versions of OGD and two existing parameter-free OLO algorithms: the dimension-free exponentiated gradient algorithm (DFEG) (Orabona, 2013) and the adaptive normal algorithm (AdaNormal) (McMahan and Orabona, 2014).

Following the original problem setting in (Orabona, 2013), we describe the per-state DFEG only for online linear regression. Consider a loss function $\ell(\hat{y}, y)$, which is convex and L-Lipschitz in its first argument. At each round t, a learner picks $\mathbf{w}_t \in V$. A nature then reveals $(\mathbf{x}_t, y_t) \in V \times \mathbb{R}$, and the learner suffers loss $\ell_t(\mathbf{w}_t) := \ell(\hat{y}_t, y_t)$, where $\hat{y}_t := \langle \mathbf{w}_t, \mathbf{x}_t \rangle$. Note that the DFEG algorithm requires a norm of the instance $\|\mathbf{x}_t\|$ to form an action $\mathbf{w}_t$.

We remark that these two algorithms are also guaranteed to incur essentially the same order of regret without tuning learning rate. Also, while the per-state KT OLO algorithm serves as a base algorithm in the CTW OLO algorithm, to be a fair comparison, the two algorithms can be also used as a base in the specialist framework to solve the tree side information problem, as noted in Section 6.5. There are, however, two minor disadvantages we can observe. First of all, the DFEG algorithm is tailored to the online linear regression problem, while the per-state KT OLO and AdaptiveNormal algorithms can be applied to a general OLO problem. Second, while the KT OLO has

Algorithm 6.B.1. Per-state Dimension-free Exponentiated Gradient (Orabona, 2013) for online regression

procedure PERSTATEDFEG($L, \delta, 0.882 \leq a \leq 1.109$)
  Initialize $\boldsymbol{\theta}^{(s)} \leftarrow 0 \in V, H^{(s)} \leftarrow \delta$ for each $s \in [S]$
  for $1 \leq t \leq T$ **do**
    Receive $h_t \in [S]$ and $\|\mathbf{x}_t\|$
    Update $H^{(h_t)} \leftarrow H^{(h_t)} + L^2 \max\{\|\mathbf{x}_t\|, \|\mathbf{x}_t\|^2\}$
    Set $\alpha_t \leftarrow a(H^{(h_t)})^{1/2}, \beta_t \leftarrow (H^{(h_t)})^{3/2}$
    if $\|\boldsymbol{\theta}^{(h_t)}\| = 0$ **then**
      Set $\mathbf{w}_t \leftarrow 0$
    else
      Set $\mathbf{w}_t \leftarrow \frac{\boldsymbol{\theta}^{(h_t)}}{\beta_t \|\boldsymbol{\theta}^{(h_t)}\|} \exp(\frac{\|\boldsymbol{\theta}^{(h_t)}\|}{\alpha_t})$
    end if
    Receive $(\mathbf{x}_t, y_t)$ and incur loss $\ell_t(\mathbf{w}_t)$
    Update $\boldsymbol{\theta}^{(h_t)} \leftarrow \boldsymbol{\theta}^{(h_t)} - \partial \ell_t(\langle \mathbf{w}_t, \mathbf{x}_t \rangle) \mathbf{x}_t$
  end for
end procedure

only one hyperparameter, the initial wealth W_0, the above two per-state algorithms have two hyperparameters (except the Lipschitz constant), which may need to be chosen or tuned in practice.

6.C Deferred Technical Materials

6.C.1 Proofs for Section 6.2

Proof of Theorem 6.2.1

We note that all statements in Section 6.2 originally appeared in (Orabona and Pál, 2016). The proofs given here are rephrased and simplified from (Orabona and Pál, 2016).

Before we prove Theorem 6.2.1, we state some key properties of the KT potential function ψ^{KT}.

Proposition 6.C.1. *For each $t \geq 1$ and any $g_1, \ldots, g_t \in [-1, 1]$, the followings hold:*

(a) (Coordinatewise convexity) $g \mapsto \psi^{\mathsf{KT}}(g^{t-1} g)$ is convex for $g \in [-1, 1]$.

Algorithm 6.B.2. Per-state AdaptiveNormal (McMahan and Orabona, 2014) for OLO with side information

procedure PERSTATEADANORMAL($L, a \geq \frac{3L^2\pi}{4}, \epsilon$)
  Initialize $\boldsymbol{\theta}^{(s)} \leftarrow 0 \in V$ for each $s \in [S]$
  for $1 \leq t \leq T$ **do**
    Receive $h_t \in [S]$
    if $\|\boldsymbol{\theta}^{(h_t)}\| = 0$ **then**
      Set $\mathbf{w}_t \leftarrow 0$
    else
      Set $\mathbf{w}_t \leftarrow \epsilon \frac{\boldsymbol{\theta}^{(h_t)}}{\|\boldsymbol{\theta}^{(h_t)}\|} \frac{1}{2L\ln^2(t+1)} \{\exp(\frac{(\|\boldsymbol{\theta}^{(h_t)}\|+L)^2}{2at}) - \exp(\frac{(\|\boldsymbol{\theta}^{(h_t)}\|-L)^2}{2at})\}$
    end if
    Receive $\mathbf{g}_t$ and incur loss $\langle \mathbf{g}_t, \mathbf{w}_t \rangle$
    Update $\boldsymbol{\theta}^{(h_t)} \leftarrow \boldsymbol{\theta}^{(h_t)} - \mathbf{g}_t$
  end for
end procedure

(b) (Consistency) $\psi^{\mathsf{KT}}(g^{t-1}) = \frac{1}{2}(\psi^{\mathsf{KT}}(g^{t-1}1) + \psi^{\mathsf{KT}}(g^{t-1}\bar{1}))$.

(c) (The relation of signed betting and potential)

$$b^{\mathsf{KT}}(g^{t-1}) = \frac{\psi^{\mathsf{KT}}(g^{t-1}1) - \psi^{\mathsf{KT}}(g^{t-1}\bar{1})}{\psi^{\mathsf{KT}}(g^{t-1}1) + \psi^{\mathsf{KT}}(g^{t-1}\bar{1})} = \frac{\psi^{\mathsf{KT}}(g^{t-1}1) - \psi^{\mathsf{KT}}(g^{t-1}\bar{1})}{\psi^{\mathsf{KT}}(g^{t-1})}.$$

(d) For any $x \in [0, t)$, $x(\psi_t^{\mathsf{KT}})''(x) \geq (\psi_t^{\mathsf{KT}})'(x)$.

Proof. Recall $\tilde{q}_t^{\mathsf{KT}}(x) := B(\frac{t+x+1}{2}, \frac{t-x+1}{2})/B(\frac{1}{2}, \frac{1}{2})$ and

$$\psi^{\mathsf{KT}}(g^t) := \psi_t^{\mathsf{KT}}(\textstyle\sum g^t) := 2^t \tilde{q}_t^{\mathsf{KT}}(\sum g^t).$$

(a) and (d) follow from the properties of the Gamma function $\Gamma(\cdot)$; for details, see (Orabona and Pál, 2016, Lemma 12) and the proof therein. (b) and (c) can be easily verified by the definition of the KT potential ψ^{KT}. □

We remark that the relation (b) can be understood as a continuous extension of the consistency of $\tilde{q}^{\mathsf{KT}}$ as a joint probability over a binary sequence $g^t \in \{-1, 1\}^t$.

Further, in view of the relation (c), the signed bet b^{KT} is a continuous extension of the prequential probability $\tilde{q}^{\mathsf{KT}}(\cdot|g^{t-1})$ induced by the joint probability assignment $\tilde{q}^{\mathsf{KT}}(g^t)$.

We now show the following single round bound.

Lemma 6.C.2. *For any $t \geq 1$ and $g_1, \dots, g_t \in [-1, 1]$, we have*

$$(1 + g_t b_t^{\mathsf{KT}}(g^{t-1}))\psi^{\mathsf{KT}}(g^{t-1}) \geq \psi^{\mathsf{KT}}(g^t).$$

Proof. By the definition of coin betting potentials, we have

$$\begin{aligned}
&(1 + g_t b^{\mathsf{KT}}(g^{t-1}))\psi^{\mathsf{KT}}(g^{t-1}) \\
&\overset{(i)}{\geq} (1 + g_t b^{\mathsf{KT}}(g^{t-1}))\frac{1}{2}(\psi^{\mathsf{KT}}(g^{t-1}1) + \psi^{\mathsf{KT}}(g^{t-1}\bar{1})) \\
&\overset{(ii)}{=} \Big(1 + g_t \frac{\psi^{\mathsf{KT}}(g^{t-1}1) - \psi^{\mathsf{KT}}(g^{t-1}\bar{1})}{\psi^{\mathsf{KT}}(g^{t-1}1) + \psi^{\mathsf{KT}}(g^{t-1}\bar{1})}\Big)\frac{1}{2}(\psi^{\mathsf{KT}}(g^{t-1}1) + \psi^{\mathsf{KT}}(g^{t-1}\bar{1})) \\
&= \frac{1 + g_t}{2}\psi^{\mathsf{KT}}(g^{t-1}1) + \frac{1 - g_t}{2}\psi^{\mathsf{KT}}(g^{t-1}\bar{1}) \\
&\overset{(iii)}{\geq} \psi^{\mathsf{KT}}(g^t).
\end{aligned}$$

where (i), (ii), and (iii) follow from (b), (c), and (a) in Proposition 6.C.1, respectively. □

While the above lemma establishes the lower bound on the cumulative wealth, we then need the following statement that connects regret and wealth via convex duality. We remark that this relation is the key statement that motivates all coin betting based algorithms.

Proposition 6.C.3 (McMahan and Orabona, 2014, (Orabona and Pál, 2016, Lemma 1))**.** *Let $\Phi \colon V \to \mathbb{R}$ be a convex function and let $\Phi^\star \colon V \to \mathbb{R} \cup \{+\infty\}$ denote its Fenchel conjugate function. For any $\mathbf{g}_1, \dots, \mathbf{g}_T \in V^\star$ and any $\mathbf{w}_t, \dots, \mathbf{w}_T \in V$, we have*

$$\sup_{\mathbf{u} \in V}\{\mathsf{Reg}(\mathbf{u}; \mathbf{g}^T) - \Phi(\mathbf{u})\} = -\sum_{t=1}^{T}\langle \mathbf{g}_t, \mathbf{w}_t\rangle + \Phi^\star\Big(\sum_{t=1}^{T}\mathbf{g}_t\Big),$$

where $\mathsf{Reg}(\mathbf{u}; \mathbf{g}^T) := \sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{u} - \mathbf{w}_t \rangle$.

Proof. By definition of Fenchel dual, we have

$$\begin{aligned}
\sup_{\mathbf{u} \in V} \{\mathsf{Reg}(\mathbf{u}; \mathbf{g}^T) - \Phi(\mathbf{u})\} &= \sup_{\mathbf{u} \in V} \Big\{ \sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{u} - \mathbf{w}_t \rangle - \Phi(\mathbf{u}) \Big\} \\
&= -\sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t \rangle + \sup_{\mathbf{u} \in V} \Big\{ \Big\langle \sum_{t=1}^T \mathbf{g}_t, \mathbf{u} \Big\rangle - \Phi(\mathbf{u}) \Big\} \\
&= -\sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t \rangle + \Phi^\star \Big(\sum_{t=1}^T \mathbf{g}_t \Big).
\end{aligned}$$

□

Now we are ready to prove Theorem 6.2.1.

Proof of Theorem 6.2.1. We first show the wealth lower bound $\mathsf{W}_t \geq \mathsf{W}_0 \psi^{\mathsf{KT}}(g^t)$ stated in (6.1) by induction on t. Suppose that $\mathsf{W}_{t-1} \geq \mathsf{W}_0 \psi^{\mathsf{KT}}(g^{t-1})$. Then,

$$\begin{aligned}
\mathsf{W}_t &= \mathsf{W}_{t-1} + g_t w_t \\
&= (1 + b^{\mathsf{KT}}(g^{t-1}) g_t) \mathsf{W}_{t-1} \\
&\overset{(a)}{\geq} (1 + b^{\mathsf{KT}}(g^{t-1}) g_t) \mathsf{W}_0 \psi^{\mathsf{KT}}(g^{t-1}) \\
&\overset{(b)}{\geq} \mathsf{W}_0 \psi^{\mathsf{KT}}(g^t),
\end{aligned}$$

where (a) follows from the induction hypothesis and (b) follows from Lemma 6.C.2.

The wealth lower bound can be converted into the desired regret bound by Proposition 6.C.3. That is, we have

$$\sup_{u \in \mathbb{R}} \{\mathsf{Reg}(u; g^T) - \phi(u)\} = -\sum_{t=1}^T g_t w_t + \mathsf{W}_0 \psi^{\mathsf{KT}}(g^T) \leq \mathsf{W}_0,$$

where $\phi \colon \mathbb{R} \to \mathbb{R}$ is a convex function such that its conjugate function $\phi^\star \colon \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ is equal to $\mathsf{W}_0 \psi_T^{\mathsf{KT}}(\sum g^t)$. Since $x \mapsto \psi_T^{\mathsf{KT}}(x)$ is a convex, proper, closed function, one can check that $\phi(u) = \mathsf{W}_0 (\psi_T^{\mathsf{KT}})^\star(\frac{u}{\mathsf{W}_0})$ using Lemma 6.C.10. □

Proof of Theorem 6.2.2

As in 1D OLO case, we first show the following single round bound.

Lemma 6.C.4. *For any* $\mathbf{g}_1, \dots, \mathbf{g}_t \in \mathbb{B}$*, we have*

$$(1 + \langle \mathbf{g}_t, \mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1})\rangle)\Psi^{\mathsf{KT}}(\mathbf{g}^{t-1}) \geq \Psi^{\mathsf{KT}}(\mathbf{g}^t).$$

Proof. Let $\mathbf{f}_{t-1} := \sum \mathbf{g}^{t-1}$. Consider

$$\begin{aligned}
&(1+\langle \mathbf{g}_t, \mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1})\rangle)\Psi^{\mathsf{KT}}(\mathbf{g}^{t-1}) - \Psi^{\mathsf{KT}}(\mathbf{g}^t) \\
&\quad = \Psi^{\mathsf{KT}}(\mathbf{g}^{t-1}) + \langle \mathbf{g}_t, \mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1})\rangle\Psi^{\mathsf{KT}}(\mathbf{g}^{t-1}) - \Psi^{\mathsf{KT}}(\mathbf{g}^t) \\
&\quad = \psi_{t-1}^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|) + \left\langle \mathbf{g}_t, b_t^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|)\frac{\mathbf{f}_{t-1}}{\|\mathbf{f}_{t-1}\|}\right\rangle \psi_{t-1}^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|) - \psi_t^{\mathsf{KT}}(\|\mathbf{f}_{t-1} + \mathbf{g}_t\|) \\
&\quad \overset{(a)}{\geq} \psi_{t-1}^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|) + \min_{r\in\{\pm 1\}} \{r\|\mathbf{g}_t\| b_t^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|)\psi_{t-1}^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|) - \psi_t^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\| + r\|\mathbf{g}_t\|)\} \\
&\quad = \min_{r\in\{\pm 1\}} \{(1 + r\|\mathbf{g}_t\| b_t^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|))\psi_{t-1}^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|) - \psi_t^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\| + r\|\mathbf{g}_t\|)\} \\
&\quad \geq \min_{g\in[-1,1]} \{(1 + g b_t^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|))\psi_{t-1}^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|) - \psi_t^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\| + g)\} \\
&\quad \overset{(b)}{\geq} 0.
\end{aligned}$$

Here, we apply Lemma 6.C.8 since ψ_t^{KT} satisfies $x(\psi_t^{\mathsf{KT}})''(x) \geq (\psi_t^{\mathsf{KT}})'(x)$ for all $x \in [0, t)$, to have (a) by plugging in $\mathbf{u} \leftarrow \mathbf{g}_t$, $\mathbf{v} \leftarrow \mathbf{f}_{t-1}$, $c(\|\mathbf{u}\|, \|\mathbf{v}\|) \leftarrow \frac{b_t^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|)}{\|\mathbf{f}_{t-1}\|}\psi_{t-1}^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|)$, and $h(\cdot) \leftarrow \psi_t^{\mathsf{KT}}(\cdot)$. (b) follows from the single round bound for 1D case established in Lemma 6.C.2. □

The proof of Theorem 6.2.2 now follows similarly to that of Theorem 6.2.1.

Proof of Theorem 6.2.2. We show $\mathsf{W}_t \geq \mathsf{W}_0\Psi^{\mathsf{KT}}(\mathbf{g}^t)$ by induction on t. For $t = 0$, it trivially holds. For $t \geq 1$, assume that $\mathsf{W}_{t-1} \geq \mathsf{W}_0\Psi^{\mathsf{KT}}(\mathbf{g}^{t-1})$ holds. Then, we have

$$\mathsf{W}_t = \langle \mathbf{g}_t, \mathbf{w}_t^{\mathsf{KT}}\rangle + \mathsf{W}_{t-1}$$

$$
\begin{aligned}
&= (1 + \langle \mathbf{g}_t, \mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1})\rangle)\mathsf{W}_{t-1} \\
&\overset{(a)}{\geq} (1 + \langle \mathbf{g}_t, \mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1})\rangle)\mathsf{W}_0\Psi^{\mathsf{KT}}(\mathbf{g}^{t-1}) \\
&\overset{(b)}{\geq} \mathsf{W}_0\Psi^{\mathsf{KT}}(\mathbf{g}^t).
\end{aligned}
$$

Here, (a) follows from the induction hypothesis and (b) follows from the above lemma. The regret bound follows by the same logic of the 1D case using Proposition 6.C.3 with the additional application of Lemma 6.C.9, which implies that $(\psi_t^{\mathsf{KT}})^\star(\mathbf{u}) = (\psi_t^{\mathsf{KT}})^\star(\|\mathbf{u}\|)$. □

6.C.2 Proofs for Section 6.3

Proof of Theorem 6.3.1

The following statement generalizes Proposition 6.C.3 for static competitors to adaptive competitors.

Proposition 6.C.5. *Let $\Phi\colon V \times \cdots \times V \to \mathbb{R}$ be a convex function and let $\Phi^\star\colon V \times \cdots \times V \to \mathbb{R} \cup \{+\infty\}$. For any side information sequence $H = (h_t)_{t\geq 1}$, any $\mathbf{g}_1, \ldots, \mathbf{g}_T \in V^\star$, and any $\mathbf{w}_t, \ldots, \mathbf{w}_T \in V$, we have*

$$
\begin{aligned}
&\sup_{\mathbf{u}_{1:S}\in V\times\cdots V} \{\mathsf{Reg}(\mathbf{u}_{1:S}[H]; \mathbf{g}^T) - \Phi(\mathbf{u}_{1:S})\} \\
&= -\sum_{t=1}^{T}\langle \mathbf{g}_t, \mathbf{w}_t\rangle + \Phi^\star\Big(\sum_{t\in[T]:h_t=1}\mathbf{g}_t, \ldots, \sum_{t\in[T]:h_t=S}\mathbf{g}_t\Big),
\end{aligned}
$$

where $\mathsf{Reg}(\mathbf{u}_{1:S}[H]; \mathbf{g}^T) := \sum_{s=1}^{S}\sum_{t\in[T]:h_t=S}\langle \mathbf{g}_t, \mathbf{u}_s - \mathbf{w}_t\rangle$.

Proof. By definition of Fenchel dual, we have

$$
\begin{aligned}
&\sup_{\mathbf{u}_{1:S}\in V\times\cdots V} \{\mathsf{Reg}(\mathbf{u}_{1:S}[H]; \mathbf{g}^T) - \Phi(\mathbf{u}_{1:S})\} \\
&= \sup_{\mathbf{u}_{1:S}\in V\times\cdots V} \Big\{\sum_{s=1}^{S}\sum_{t\in[T]:h_t=s}\langle \mathbf{g}_t, \mathbf{u}_s - \mathbf{w}_t\rangle - \Phi(\mathbf{u}_{1:S})\Big\}
\end{aligned}
$$

$$= -\sum_{t=1}^{T}\langle \mathbf{g}_t, \mathbf{w}_t\rangle + \sup_{\mathbf{u}_{1:S}\in V\times\cdots V}\Big\{\sum_{s=1}^{S}\Big\langle \sum_{t\in[T]:h_t=s}\mathbf{g}_t, \mathbf{u}_s\Big\rangle - \Phi(\mathbf{u}_{1:S})\Big\}$$
$$= -\sum_{t=1}^{T}\langle \mathbf{g}_t, \mathbf{w}_t\rangle + \Phi^\star\Big(\sum_{t\in[T]:h_t=1}\mathbf{g}_t, \ldots, \sum_{t\in[T]:h_t=S}\mathbf{g}_t\Big). \qquad \square$$

We are now ready to prove Theorem 6.3.1.

Proof of Theorem 6.3.1. Since the vectorial betting $\mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1}; h^t)$ only affects the component potential $\Psi^{\mathsf{KT}}(\mathbf{g}^t(h_t; h^{t-1}))$ by construction, the wealth lower bound readily follows from the same argument in the proof of Theorem 6.2.2. Now, we observe that

$$\Psi^{\mathsf{KT}}(\mathbf{g}^T; h^T) = 2^T \prod_{s\in[S]} \tilde{q}^{\mathsf{KT}}_{T_s}(\|\textstyle\sum \mathbf{g}^T(s; h^T)\|),$$

where $T_s := |\{t \in [T] \colon h_t = s\}|$. Since $\tilde{q}^{\mathsf{KT}}_T(x) \geq \frac{1}{2^T e\sqrt{\pi}}\frac{1}{\sqrt{T}}e^{\frac{2x^2}{T}}$ for $T \geq 1$ by (Orabona and Pál, 2016, Lemma 14), we have

$$\Psi^{\mathsf{KT}}(\mathbf{g}^T; h^T) \geq \Big(\frac{1}{e\sqrt{\pi}}\Big)^{S'} \frac{1}{\sqrt{T'_1 \cdots T'_S}} \exp\Big(\sum_{s=1}^{S}\frac{2\|\sum \mathbf{g}^T(s; h^T)\|^2}{T'_s}\Big),$$

where $S' := \sum_{s=1}^{S} 1\{T_s \geq 1\}$ and $T'_s := T_s \vee 1$. Applying Propositions 6.C.5 and 6.C.15 then establishes the regret upper bound. $\square$

Proof of Theorem 6.3.5

We show $\mathsf{W}_t \geq \mathsf{W}_0\Psi^{\mathsf{mix}}(\mathbf{g}^t; \mathbf{h}^t)$ by induction on t. For $t = 0$, it trivially holds. For $t \geq 1$, assume that $\mathsf{W}_{t-1} \geq \mathsf{W}_0\Psi^{\mathsf{mix}}(\mathbf{g}^{t-1}; \mathbf{h}^{t-1})$ holds. Then, we have

$$\begin{aligned}
\mathsf{W}_t &= \langle \mathbf{g}_t, \mathbf{w}^{\mathsf{mix}}_t(\mathbf{g}^{t-1}; \mathbf{h}^t)\rangle + \mathsf{W}_{t-1} \\
&= (1 + \langle \mathbf{g}_t, \mathbf{v}^{\mathsf{mix}}(\mathbf{g}^{t-1}; \mathbf{h}^t)\rangle)\mathsf{W}_{t-1} \\
&\overset{(a)}{\geq} (1 + \langle \mathbf{g}_t, \mathbf{v}^{\mathsf{mix}}(\mathbf{g}^{t-1}; \mathbf{h}^t)\rangle)\mathsf{W}_0\Psi^{\mathsf{mix}}(\mathbf{g}^{t-1}; \mathbf{h}^{t-1})
\end{aligned}$$

$$\overset{(b)}{\geq} \mathsf{W}_0 \Psi^{\mathsf{mix}}(\mathbf{g}^t; \mathbf{h}^t).$$

Here, (a) follows from the induction hypothesis, and (b) follows from the construction of $\mathbf{v}^{\mathsf{mix}}(\mathbf{g}^{t-1}; \mathbf{h}^t)$. The regret guarantee for $m \in [M]$ readily follows from the construction of the mixture potential, which guarantees $\mathsf{W}_T \geq w_m \mathsf{W}_0 \Psi^{\mathsf{KT}}(g^T; (h^{(m)})^T)$. □

Matching lower bounds for tree side information

We first require the following theorem from (Orabona, 2019).

Theorem 6.C.6 (Orabona, 2019, Theorem 5.11)**.** *Suppose that an OLO algorithm satisfies that for each $t \geq 0$*

$$\sup_{\mathbf{g}^t \in \mathbb{B}^t} \mathsf{Reg}(\mathbf{0}; \mathbf{g}^t) = -\inf_{\mathbf{g}^t \in \mathbb{B}^t} \sum_{i=1}^{t} \langle \mathbf{g}_i, \mathbf{w}_i \rangle \leq \mathsf{W}_0^{(t)} \tag{6.14}$$

with some nondecreasing sequence $(\mathsf{W}_0^{(t)})_{t \geq 0}$. Then, for each $T \geq 1$, there exists $\mathbf{v}_1, \ldots, \mathbf{v}_T \in \mathbb{B}$ such that

$$\mathbf{w}_t = \mathbf{v}_t\Big(\mathsf{W}_0^{(T)} + \sum_{i=1}^{t-1} \langle \mathbf{g}_i, \mathbf{w}_i \rangle\Big) \quad \textit{for all } t \in [T].$$

For a binary quantizer $Q \colon \mathbb{B} \to \{\pm 1\}$, let $H_{\mathbf{T},Q}$ denote the tree side information with respect to a tree $\mathbf{T}$ and an auxiliary sequence $\Omega = (\omega_t)_{t \geq 1}$ with $\omega_t = Q(\mathbf{g}_t)$.

Theorem 6.C.7. *Let $V = \mathbb{R}^d$ be the d-dimensional Euclidean space. Suppose that a binary quantizer $Q \colon \mathbb{B} \to \{\pm 1\}$ satisfies $Q(\mathbf{e}_j) = 1$ and $Q(-\mathbf{e}_j) = -1$ for some $j \in [d]$. For T sufficiently large, for any* causal *OLO algorithm that satisfies the condition* (6.14) *in Theorem 6.C.6, for any binary suffix tree $\mathbf{T}$, there exist a sequence $\mathbf{g}_1, \ldots, \mathbf{g}_T \in \mathbb{B}$ and a competitor $(\mathbf{u}_s^*)_{s \in \mathbf{T}}[H_{\mathbf{T},Q}] \in \mathcal{M}(H_{\mathbf{T},Q})$ such that*

$$\mathsf{Reg}((\mathbf{u}_s^*)_{s \in \mathbf{T}}[H_{\mathbf{T},Q}]); \mathbf{g}^T) \geq \sqrt{\sum_{s \in \mathbf{T}} T_s \|\mathbf{u}_s^*\|_2^2 \ln\Big(\frac{(T/|\mathbf{T}|)^{|\mathbf{T}|}}{(\mathsf{W}_0^{(T)})^2} \sum_{s \in \mathbf{T}} T_s \|\mathbf{u}_s^*\|_2^2 + 1\Big)} + \mathsf{W}_0^{(T)}.$$

Proof. Without loss of generality, assume that the binary quantizer $Q \colon \mathbb{B} \to \{\pm 1\}$ satisfies $Q(\mathbf{e}_1) = 1$ and $Q(-\mathbf{e}_1) = -1$. For a binary sequence $c^T \in \{\pm 1\}^T$, we set

$\mathbf{g}_t = (c_t, 0, \dots, 0)$ for $c_t \in \{\pm 1\}$, so that $\langle \mathbf{g}_t, \mathbf{w}_t \rangle = c_t x_{t1}$. Then, by Theorem 6.C.6, we can write

$$x_{t1} = v_{t1}\Big(\mathsf{W}_0^{(T)} + \sum_{i=1}^{t-1} \langle \mathbf{g}_i, \mathbf{w}_i \rangle\Big) = v_{t1}\Big(\mathsf{W}_0^{(T)} + \sum_{i=1}^{t-1} c_i x_{i1}\Big)$$

for some v_{t1} such that $|v_{t1}| \le 1$. Hence, the OLO problem with any causal algorithms satisfying (6.14) with respect to the 1D sequences $\mathbf{g}^T$ can be equivalently viewed as the 1D coin betting with initial wealth $\mathsf{W}_0 = \mathsf{W}_0^{(T)}$.

Now, we state the celebrated Rissanen's lower bound for universal compression in the form of the wealth upper bound for the coin betting. Rissanen (1996) showed that for any probability assignment $q(x^T)$ on a binary sequence $x^T \in \{0,1\}^T$, there exists a sequence $\tilde{x}^T \in \{0,1\}$ such that

$$q(\tilde{x}^T) \le e^{-\frac{|\mathbf{T}|}{2} \ln \frac{T}{|\mathbf{T}|}} \max_{p_{\mathbf{T}}} p_{\mathbf{T}}(\tilde{x}^T),$$

where the maximum is over all possible tree sources $p_{\mathbf{T}}$ with the underlying tree $\mathbf{T}$. This can be translated into the wealth upper bound for the standard coin betting with binary outcomes $c_t \in \{\pm 1\}$ thanks to the equivalence between the coin betting and universal compression: for any continuous coin betting algorithm which plays a relative bet $b_t \in [-1, 1]$ at time t, there exists a binary sequence $\tilde{c}^T \in \{\pm 1\}^T$ such that

$$\begin{aligned}
\frac{\mathsf{W}_T}{\mathsf{W}_0} = \prod_{t=1}^{T} (1 + b_t \tilde{c}_t) &\le \Big(\frac{|\mathbf{T}|}{T}\Big)^{\frac{|\mathbf{T}|}{2}} \prod_{s \in \mathbf{T}} \max_{b_s \in [-1,1]} \prod_{t \in [T]: h_t = s} (1 + b_s \tilde{c}_t) \\
&\overset{(a)}{\le} \Big(\frac{|\mathbf{T}|}{T}\Big)^{\frac{|\mathbf{T}|}{2}} \prod_{s \in \mathbf{T}} \exp\Big(\frac{\ln 2}{T'_s} \Big(\sum_{t \in [T]: h_t = s} \tilde{c}_t \Big)^2 \Big), \\
&= f\big((\textstyle\sum \tilde{c}^T(s; H_{\mathbf{T},Q}))_{s \in \mathbf{T}}\big), \qquad (6.15)
\end{aligned}$$

where h_t denotes the suffix of the sequence c^{t-1} with respect to $\mathbf{T}$ at time t, $T'_s := T_s \vee 1$, $T_s := |\{t \in [T] \colon h_t = s\}|$, $f((x_s)_{s \in \mathbf{T}}) := \prod_{s \in \mathbf{T}} h_s(x_s)$, and $h_s(x_s) = \beta_s \exp(\frac{x_s^2}{2\alpha_s})$ with

$\alpha_s = \frac{2T_{s'}}{\ln 2}$, and $\beta_s = \sqrt{|\mathbf{T}|/T}$. Here, (a) follows by Lemma 6.C.16.

For the adversarial coin sequence $(\tilde{c}_t)_{t\geq 1}$ satisfying (6.15), we now define $\mathbf{g}_t := (\tilde{c}_t, 0, \ldots, 0)$. Then, we have

$$\begin{aligned}
\mathsf{W}_0^{(T)} + \sum_{t=1}^{T} \langle \tilde{\mathbf{g}}_t, \mathbf{w}_t \rangle &= \mathsf{W}_0^{(T)} + \sum_{t=1}^{T} \tilde{c}_t x_{t1} \\
&\leq \mathsf{W}_0^{(T)} f\big((\textstyle\sum \tilde{c}^T(s; H_{\mathbf{T},Q}))_{s\in\mathbf{T}} \big) \\
&= \sum_{s\in\mathbf{T}} \big(\textstyle\sum \tilde{c}^T(s; H_{\mathbf{T},Q}) \big) u_s^* - \mathsf{W}_0^{(T)} f^\star\Big(\Big(\frac{|u_s^*|}{\mathsf{W}_0^{(T)}} \Big)_{s\in\mathbf{T}} \Big) \\
&= \sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{u}_{h_t}^* \rangle - \mathsf{W}_0^{(T)} f^\star\Big(\Big(\frac{\|\mathbf{u}_s^*\|_2}{\mathsf{W}_0^{(T)}} \Big)_{s\in\mathbf{T}} \Big),
\end{aligned}$$

where $(u_s^*)_{s\in\mathbf{T}} = \mathsf{W}_0^{(T)} \nabla f((\sum \tilde{c}^T(s; H_{\mathbf{T},Q}))_{s\in\mathbf{T}})$ and $\mathbf{u}_s^* := (u_s^*, 0, \ldots, 0)$ for each $s \in \mathbf{T}$. Rearranging the terms, we have

$$\begin{aligned}
\mathsf{Reg}((\mathbf{u}_s^*)_{s\in\mathbf{T}}[H_{\mathbf{T},Q}]); \mathbf{g}^T) &= \sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{u}_{h_t}^* \rangle - \sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{w}_t \rangle \\
&\geq \mathsf{W}_0^{(T)} + \mathsf{W}_0^{(T)} f^\star\Big(\Big(\frac{\|\mathbf{u}_s^*\|_2}{\mathsf{W}_0^{(T)}} \Big)_{s\in\mathbf{T}} \Big). \qquad \square
\end{aligned}$$

Proof of Proposition 6.3.11

We use a backward induction over the depth $|s|$ to show that the recursion is well-defined. First, if $|s| = D$, $\mathbf{u}_s^{\mathsf{CTW}}(\mathbf{g}^{t-1}) = \Psi_s^{\mathsf{KT}}(\mathbf{g}^{t-1}) \mathbf{v}_s^{\mathsf{KT}}(\mathbf{g}^{t-1})$. By definition of $\mathbf{v}_s^{\mathsf{KT}}(\mathbf{g}^{t-1})$, $\mathbf{u}_s^{\mathsf{CTW}}(\mathbf{g}^{t-1})$ is a vector if s is the active node at depth D, and a scalar otherwise. Now, for $d \leq D-1$, assume that $\mathbf{u}_{s'}^{\mathsf{CTW}}(\mathbf{g}^{t-1})$ is a scalar if s' is an active node and a vector otherwise for any $|s'| = d+1$ (induction hypothesis). Consider any node s of $\mathcal{T}_D$ with $|s| = d$. If s is an active node, then $\mathbf{u}_{\bar{1}s}^{\mathsf{CTW}}(\mathbf{g}^{t-1}) \mathbf{u}_{1s}^{\mathsf{CTW}}(\mathbf{g}^{t-1})$ is a vector by the induction hypothesis, since exactly one of $\bar{1}s$ and $1s$ is active. Hence, $\mathbf{u}_s^{\mathsf{CTW}}(\mathbf{g}^{t-1})$ is a vector. If s is not an active node, then, $\mathbf{u}_{\bar{1}s}^{\mathsf{CTW}}(\mathbf{g}^{t-1}) \mathbf{u}_{1s}^{\mathsf{CTW}}(\mathbf{g}^{t-1})$ is a scalar by the induction hypothesis, since neither of $\bar{1}s$ and $1s$ is active. Hence, $\mathbf{u}_s^{\mathsf{CTW}}(\mathbf{g}^{t-1})$ is a scalar. This completes the

induction and thus the recursion is well-defined for all nodes s.

The claim $\mathbf{u}_\lambda^{\mathsf{CTW}}(\mathbf{g}^{t-1}) = \mathbf{u}^{\mathsf{CTW}}(\mathbf{g}^{t-1})$ can be checked by a similar induction argument. □

Proof of Proposition 6.3.12

We claim that $\mathbf{v}_s^{\mathsf{CTW}}(\mathbf{g}^{t-1}) = \frac{\mathbf{u}_s^{\mathsf{CTW}}(\mathbf{g}^{t-1})}{\Psi_s^{\mathsf{CTW}}(\mathbf{g}^{t-1})}$ for any $s = s_d = \omega_{t-d}^{t-1} \in \mathcal{T}_D$, $d = 0, \dots, D$. This trivially holds for the leaf node $s_D = \omega_{t-D}^{t-1}$. For the internal nodes s_d with $d < D$, by plugging in the recursive formulas of $\mathbf{u}^{\mathsf{CTW}}(\mathbf{g}^{t-1})$ and $\Psi^{\mathsf{CTW}}(\mathbf{g}^{t-1})$, we can write

$$\frac{\mathbf{u}^{\mathsf{CTW}}(\mathbf{g}^{t-1})}{\Psi^{\mathsf{CTW}}(\mathbf{g}^{t-1})} = \frac{\beta_s(\mathbf{g}^{t-1})}{\beta_s(\mathbf{g}^{t-1})+1}\mathbf{v}_s^{\mathsf{KT}}(\mathbf{g}^{t-1}) + \frac{1}{\beta_s(\mathbf{g}^{t-1})+1}\frac{\mathbf{u}_{\bar{1}s}^{\mathsf{CTW}}(\mathbf{g}^{t-1})}{\Psi_{\bar{1}s}^{\mathsf{CTW}}(\mathbf{g}^{t-1})}\frac{\mathbf{u}_{1s}^{\mathsf{CTW}}(\mathbf{g}^{t-1})}{\Psi_{1s}^{\mathsf{CTW}}(\mathbf{g}^{t-1})}.$$

It is now enough to show that

$$\frac{\mathbf{u}_{s'}^{\mathsf{CTW}}(\mathbf{g}^{t-1})}{\Psi_{s'}^{\mathsf{CTW}}(\mathbf{g}^{t-1})} = 1 \text{ for } s' = \overline{\omega_{t-1-|s|}}s.$$

This holds since $\mathbf{u}_s^{\mathsf{CTW}} = \Psi_s^{\mathsf{CTW}}$ for any off-path node $s \notin \rho(\omega_{t-D}^{t-1})$ by definition (6.9). □

Proof of Proposition 6.3.13

Similar to the processing betas algorithm (Willems et al., 2006), we only need to show that

$$\frac{\Psi_{\overline{\omega_{t-1-|s|}}s}^{\mathsf{CTW}}(\mathbf{g}^t)}{\Psi_{\overline{\omega_{t-1-|s|}}s}^{\mathsf{CTW}}(\mathbf{g}^{t-1})} = 1 \text{ for } s' = \overline{\omega_{t-1-|s|}}s \text{ for any } s \notin \rho(\omega_{t-D}^{t-1}).$$

Since the new symbol $\mathbf{g}_t$ is added to a node s if and only if $s \in \rho(\omega_{t-D}^{t-1})$, if $s \notin \rho(\omega_{t-D}^{t-1})$, then the CTW potential on the node s will not be updated. This proves the claim. □

6.C.3 Technical Lemmas

Lemma 6.C.8 (Orabona and Pál, 2016, Lemma 10)**.** *Let $h\colon (-a, a) \to \mathbb{R}$ be an even, twice differentiable function that satisfies $xh''(x) \geq h'(x)$ for all $x \in [0, a)$. Let $c\colon [0, \infty) \times [0, \infty) \to$*

$\mathbb{R}$ *be an arbitrary function. If* $u, v \in \mathcal{H}$ *satisfy* $\|u\| + \|v\| < a$*, then*

$$c(\|u\|, \|v\|) \cdot \langle u, v \rangle - h(\|u + v\|) \geq \min_{r \in \{\pm 1\}} \{rc(\|u\|, \|v\|)\|u\|\|v\| - h(\|u\| + r\|v\|)\}.$$

Proof sketch. It is easy to check that the inequality holds if $u = 0$ or $v = 0$. Hence, we assume $u, v \neq 0$. With $\alpha := \langle u, v \rangle / (\|u\|\|v\|)$, we can write the left hand side of the desired inequality as

$$f(\alpha) := c(\|u\|, \|v\|)\|u\|\|v\|\alpha - h(\sqrt{\|u\|^2 + \|v\|^2 + 2\alpha\|u\|\|v\|}).$$

Since the function h is assumed to be even, it is equivalent to showing that

$$\inf_{\alpha \in [-1,1]} f(\alpha) = \min\{f(+1), f(-1)\}.$$

By using the condition $xh''(x) \geq h'(x)$, one can easily show that f is concave by checking $f''(\alpha) \leq 0$, which concludes the proof. □

Lemma 6.C.9 (Bauschke and Combettes, 2011, Example 13.7)**.** *Let* $\phi \colon \mathbb{R} \to (-\infty, +\infty]$ *be even. Then* $(\phi \circ \|\cdot\|)^\star = \phi^\star \circ \|\cdot\|$*.*

Lemma 6.C.10 (Orabona, 2019, Lemma 5.8)**.** *Let* f *be a function and let* $f^\star$ *be its Fenchel conjugate. For* $a > 0$ *and* $b \in \mathbb{R}$*, the Fenchel conjugate of* $g(x) = af(x) + b$ *is* $g^\star(z) = af^\star(z/a) - b$*.*

Lemma 6.C.11 (Orabona, 2019, Theorem 5.8)**.** *For a convex, proper, closed function* $h \colon \mathbb{R}^d \to (-\infty, +\infty]$*, we have* $\langle \theta, x \rangle \geq h(x) + h^\star(\theta)$*, where the equality is attained if and only if* $x \in \partial h^\star(\theta)$*.*

Since $f(x) \geq h(x)$ for any $x \in \mathbb{R}$ implies $f^\star(u) \geq h^\star(u)$ for any $u \in \mathbb{R}$, it is enough to find the conjugate dual of a function $h(x) = \beta \exp(\frac{x^2}{2\alpha})$ for $\alpha, \beta > 0$.

The *Lambert function* $W\colon (-1/e, \infty) \to [0, \infty)$ is defined by the equation $x = W(x)e^{W(x)}$ for $x \geq 0$.

Lemma 6.C.12 (Orabona and Pál, 2016, Lemma 17)**.** *For $x \geq 0$,*

$$0.6321 \ln(x+1) \leq W(x) \leq \ln(x+1).$$

Remark 6.C.13. Here, $0.6321\ldots \approx 1/b^*$, where b^* is the solution to the equation

$$\frac{eb}{(e+1)b+1} = \frac{b}{(b+1)\ln(b+1)}.$$

Proposition 6.C.14 (Orabona and Pál, 2016, Lemma 18)**.** *For $h(x) = \beta \exp(\frac{x^2}{2\alpha})$ with $\alpha, \beta > 0$,*

$$h^\star(y) = y\sqrt{\alpha W\Big(\frac{\alpha y^2}{\beta^2}\Big)} - \beta \exp\Big(\frac{1}{2}W\Big(\frac{\alpha y^2}{\beta^2}\Big)\Big) = y\sqrt{\alpha}\Big(\sqrt{W\Big(\frac{\alpha y^2}{\beta^2}\Big)} - \sqrt{\frac{1}{W(\frac{\alpha y^2}{\beta^2})}}\Big).$$

In particular,

$$h^\star(y) \leq y\sqrt{\alpha \ln\Big(\frac{\alpha y^2}{\beta^2}+1\Big)} - \beta.$$

For a generalization with the product potential, we also have the following proposition.

Proposition 6.C.15. *Define $f_i(y_i) = \beta_i \exp(\frac{y_i^2}{2\alpha_i})$ with $\alpha_i, \beta_i > 0$ for each $i \in S$, and define $f(y_1, \ldots, y_S) = f_1(y_1)\cdots f_S(y_S)$. Then, we have*

$$f^\star(y_1, \ldots, y_S) = \sqrt{\alpha_1 y_1^2 + \ldots + \alpha_S y_S^2}\Big(\sqrt{W\Big(\frac{\alpha_1 y_1^2 + \ldots + \alpha_S y_S^2}{\beta_1^2 \cdots \beta_S^2}\Big)} - \frac{1}{\sqrt{W\Big(\frac{\alpha_1 y_1^2 + \ldots + \alpha_S y_S^2}{\beta_1^2 \cdots \beta_S^2}\Big)}}\Big).$$

In particular,

$$f^\star(y_1, \ldots, y_S) \leq \sqrt{(\alpha_1 y_1^2 + \ldots + \alpha_S y_S^2) \ln\Big(\frac{\alpha_1 y_1^2 + \ldots + \alpha_S y_S^2}{\beta_1^2 \cdots \beta_S^2} + 1\Big)} - \beta_1 \cdots \beta_S$$

Proof. For the sake of simplicity, we prove only for $S = 2$. The proof can be generalized to any $S \geq 2$ with little modification. To find

$$f^\star(y_1, y_2) = \sup_{x_1, x_2} (y_1 x_1 + y_2 x_2 - f_1(x_1) f_2(x_2)),$$

we consider the stationarity conditions

$$\frac{\partial}{\partial x_i}(y_1 x_1 + y_2 x_2 - f_1(x_1) f_2(x_2)) = 0$$

for $i \in \{1, 2\}$, which leads to

$$\begin{cases} y_1 &= f_1'(x_1) f_2(x_2), \\ y_2 &= f_1(x_1) f_2'(x_2). \end{cases}$$

Since $f_i'(x) = \frac{x}{\alpha_i} f_i(x)$, we have

$$\begin{cases} y_1 &= \frac{x_1}{\alpha_1} f_1(x_1) f_2(x_2), \\ y_2 &= \frac{x_2}{\alpha_2} f_1(x_1) f_2(x_2). \end{cases}$$

Manipulating the equations, we have

$$\Big(\frac{x_1^2}{\alpha_1} + \frac{x_2^2}{\alpha_2}\Big) \exp\Big(\frac{x_1^2}{\alpha_1} + \frac{x_2^2}{\alpha_2}\Big) = \frac{\alpha_1 y_1^2 + \alpha_2 y_2^2}{\beta_1^2 \beta_2^2},$$

which leads to

$$\frac{x_1^2}{\alpha_1} + \frac{x_2^2}{\alpha_2} = W\Big(\frac{\alpha_1 y_1^2 + \alpha_2 y_2^2}{\beta_1^2 \beta_2^2}\Big).$$

Hence,

$$f(x_1^*, x_2^*) = \beta_1\beta_2 \exp\Big(\frac{1}{2} W\Big(\frac{\alpha_1 y_1^2 + \alpha_2 y_2^2}{\beta_1^2 \beta_2^2}\Big)\Big) = \sqrt{\frac{\alpha_1 y_1^2 + \alpha_2 y_2^2}{W(\frac{\alpha_1 y_1^2 + \alpha_2 y_2^2}{\beta_1^2 \beta_2^2})}}.$$

Finally, we can compute

$$y_1 x_1^* + y_2 x_2^* = \frac{\alpha_1 y_1^2 + \alpha_2 y_2^2}{f(x_1^*, x_2^*)} = \sqrt{(\alpha_1 y_1^2 + \alpha_2 y_2^2) W\Big(\frac{\alpha_1 y_1^2 + \alpha_2 y_2^2}{\beta_1^2 \beta_2^2}\Big)},$$

whence

$$\begin{aligned} f^\star(y_1, y_2) &= y_1 x_1^* + y_2 x_2^* - f(x_1^*, x_2^*) \\ &= \sqrt{\alpha_1 y_1^2 + \alpha_2 y_2^2}\Big(\sqrt{W\Big(\frac{\alpha_1 y_1^2 + \alpha_2 y_2^2}{\beta_1^2 \beta_2^2}\Big)} - \frac{1}{\sqrt{W(\frac{\alpha_1 y_1^2 + \alpha_2 y_2^2}{\beta_1^2 \beta_2^2})}}\Big). \end{aligned} \qquad \square$$

Lemma 6.C.16 (Orabona, 2019, Lemma 9.4). *For any $T \geq 1$ and any $c^T \in [-1, 1]^T$, we have*

$$\max_{b \in [-1,1]} \prod_{t \in [T]} (1 + b c_t) \leq \exp\Big(\frac{\ln 2}{T} (\sum c^T)^2\Big).$$

6.D The CTW OLO Algorithm

See Algorithm 6.D.3.

6.E Experiment Details and Additional Figures

Problem setting

We applied the proposed OLO algorithms to solve the online linear regression problem as described in Appendix 6.B especially with absolute loss $\ell_t(\mathbf{w}_t) = |\langle \mathbf{w}_t, \mathbf{x}_t \rangle - y_t|$, where $\mathbf{w}_t$ denotes the action of an OLO algorithm and $\mathbf{x}_t$ denotes the feature vector.

Algorithm 6.D.3. CTW OLO algorithm

Parameters maximum depth $D \geq 1$, auxiliary sequence $\Omega = (\omega_t)_{t \geq 1}$, initial wealth $\mathsf{W}_0 > 0$.

procedure CTWOLO(D, Ω, W_0)
  Initialize a context tree $\mathcal{T}_D$ of depth D with $G_s \leftarrow \phi$ and $\beta_s \leftarrow 1$ for each $s \in \mathcal{T}_D$
  for each $t = 1, 2, \ldots$ **do**
    Compute $\mathbf{v}^{\mathsf{CTW}}(\mathbf{g}^{t-1}) = \mathbf{v}^{\mathsf{CTW}}_{\lambda}(\mathbf{g}^{t-1})$ by computing, for $s_0, \ldots, s_D \in \rho(\omega_{t-D}^{t-1})$,

$$\mathbf{v}^{\mathsf{CTW}}_{s_d}(\mathbf{g}^{t-1}) \leftarrow \begin{cases} \frac{\beta_{s_d}(\mathbf{g}^{t-1})}{\beta_{s_d}(\mathbf{g}^{t-1})+1} \mathbf{v}^{\mathsf{KT}}_{s_d}(\mathbf{g}^{t-1}) + \frac{1}{\beta_{s_d}(\mathbf{g}^{t-1})+1} \mathbf{v}^{\mathsf{CTW}}_{s_{d+1}}(\mathbf{g}^{t-1}) & \text{if } d < D \\ \mathbf{v}^{\mathsf{KT}}_{s_D}(\mathbf{g}^{t-1}) & \text{if } d = D \end{cases} \tag{6.11}$$

    Set $\mathbf{w}^{\mathsf{CTW}}_t(\mathbf{g}^{t-1}) \leftarrow \mathbf{v}^{\mathsf{CTW}}(\mathbf{g}^{t-1}) \mathsf{W}_{t-1}$
    Receive $\mathbf{g}_t$ and update the cumulative wealth $\mathsf{W}_t \leftarrow \mathsf{W}_{t-1} + \langle \mathbf{g}_t, \mathbf{w}^{\mathsf{CTW}}_t(\mathbf{g}^{t-1}) \rangle$
    Update $G_s \leftarrow G_s + \mathbf{g}_t$ and update β_s for $s_d = \omega_{t-d}^{t-1}$, $d = 0, \ldots, D-1$, as

$$\beta_{s_d}(\mathbf{g}^{t-1}) \leftarrow \beta_{s_d}(\mathbf{g}^{t}) = \beta_{s_d}(\mathbf{g}^{t-1}) \frac{\Psi^{\mathsf{KT}}_{s_d}(\mathbf{g}^{t})}{\Psi^{\mathsf{KT}}_{s_d}(\mathbf{g}^{t-1})} \frac{\Psi^{\mathsf{CTW}}_{s_{d+1}}(\mathbf{g}^{t-1})}{\Psi^{\mathsf{CTW}}_{s_{d+1}}(\mathbf{g}^{t})}, \tag{6.12}$$

      where

$$\frac{\Psi^{\mathsf{CTW}}_{s_d}(\mathbf{g}^{t})}{\Psi^{\mathsf{CTW}}_{s_d}(\mathbf{g}^{t-1})} = \begin{cases} \frac{\beta_{s_d}(\mathbf{g}^{t-1})}{\beta_{s_d}(\mathbf{g}^{t-1})+1} \frac{\Psi^{\mathsf{KT}}_{s_d}(\mathbf{g}^{t})}{\Psi^{\mathsf{KT}}_{s_d}(\mathbf{g}^{t-1})} + \frac{1}{\beta_{s_d}(\mathbf{g}^{t-1})+1} \frac{\Psi^{\mathsf{CTW}}_{s_{d+1}}(\mathbf{g}^{t})}{\Psi^{\mathsf{CTW}}_{s_{d+1}}(\mathbf{g}^{t-1})} & \text{if } d < D \\ \frac{\Psi^{\mathsf{KT}}_{s_D}(\mathbf{g}^{t})}{\Psi^{\mathsf{KT}}_{s_D}(\mathbf{g}^{t-1})} & \text{if } d = D \end{cases} \tag{6.13}$$

      for $s_d = \omega_{t-d}^{t-1}$, $d = 0, \ldots, D$
    Receive ω_t
  end for
end procedure

Hence, we linearized the convex loss and fed the subgradient $\partial \ell_t(\mathbf{w}_t) = \operatorname{sgn}(\langle \mathbf{w}_t, \mathbf{x}_t \rangle - y_t)\mathbf{x}_t$ to an OLO algorithm.

Data preprocessing

For each dataset, we linearly interpolated any missing values. We discarded time stamps as well as some categorical features such as `cbwd` of Beijing PM2.5 and `weather_description` of Metro Inter State Traffic Volume, and binarized the others, if possible, such as `holiday`, `weather_main`, and `snow_1h` of Metro Inter State Traffic Volume. We also applied a logarithmic mapping $x \mapsto \ln(1 + x)$ for the features `lws`,

`ls`, `lr` of Beijing PM2.5 and applied another logarithmic mapping $x \mapsto \ln x$ to the feature `rain_1h`, to make the features more suitable for linear regression. We then normalized each feature $\tilde{\mathbf{x}}_t$ so that $\|\tilde{\mathbf{x}}_t\|_2 = 1$ and added all-one coordinates as the bias component with an additional scaling by $1/\sqrt{2}$. After this preprocessing step, we obtained 7-dimensional feature vectors for both datsets. See the attached Python code for the details in Supplementary Material.

Computing resource

All experiments were run on a single laptop with a CPU Intel(R) Core(TM) i7-9750H CPU 2.60GHz with 12 (logical) cores and 16GB of RAM.

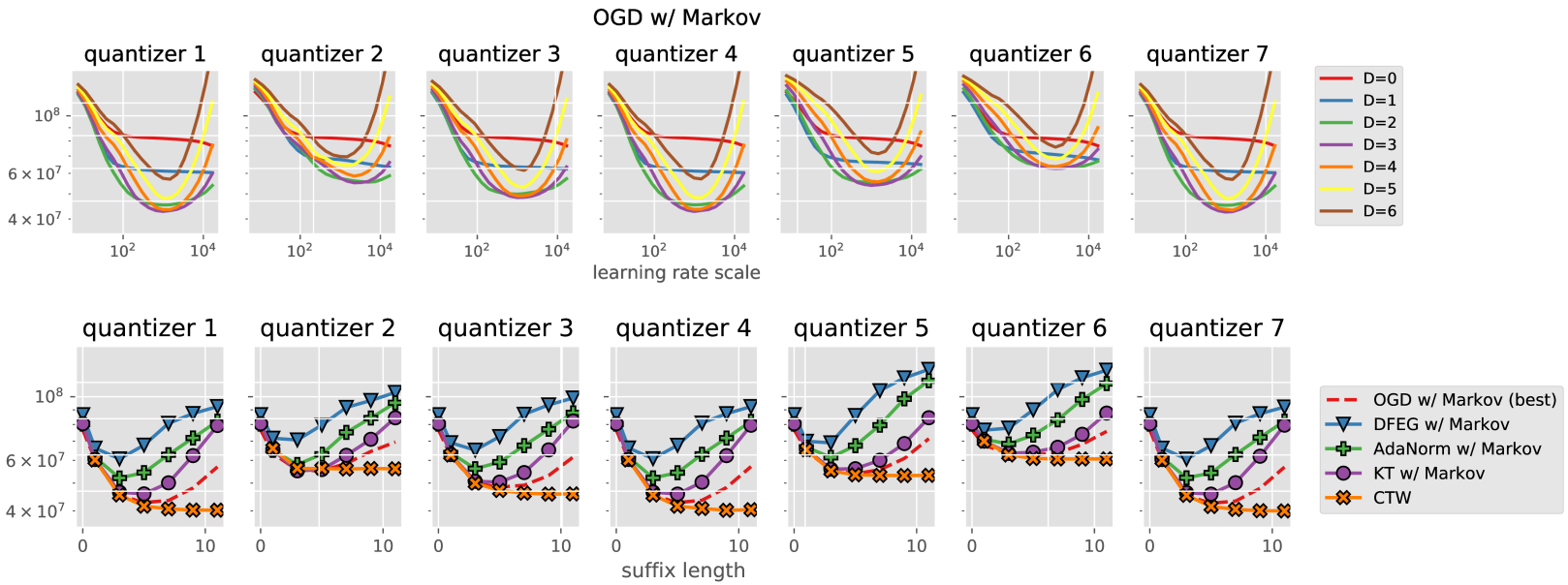


Figure 6.E.1. Metro Inter State Traffic Volume dataset (Hogue, 2019). The y-axes represent cumulative losses. (a) Performance of per-state OGD adaptive to Markov side information with various learning rate scales. (b) Performance of parameter-free algorithms.

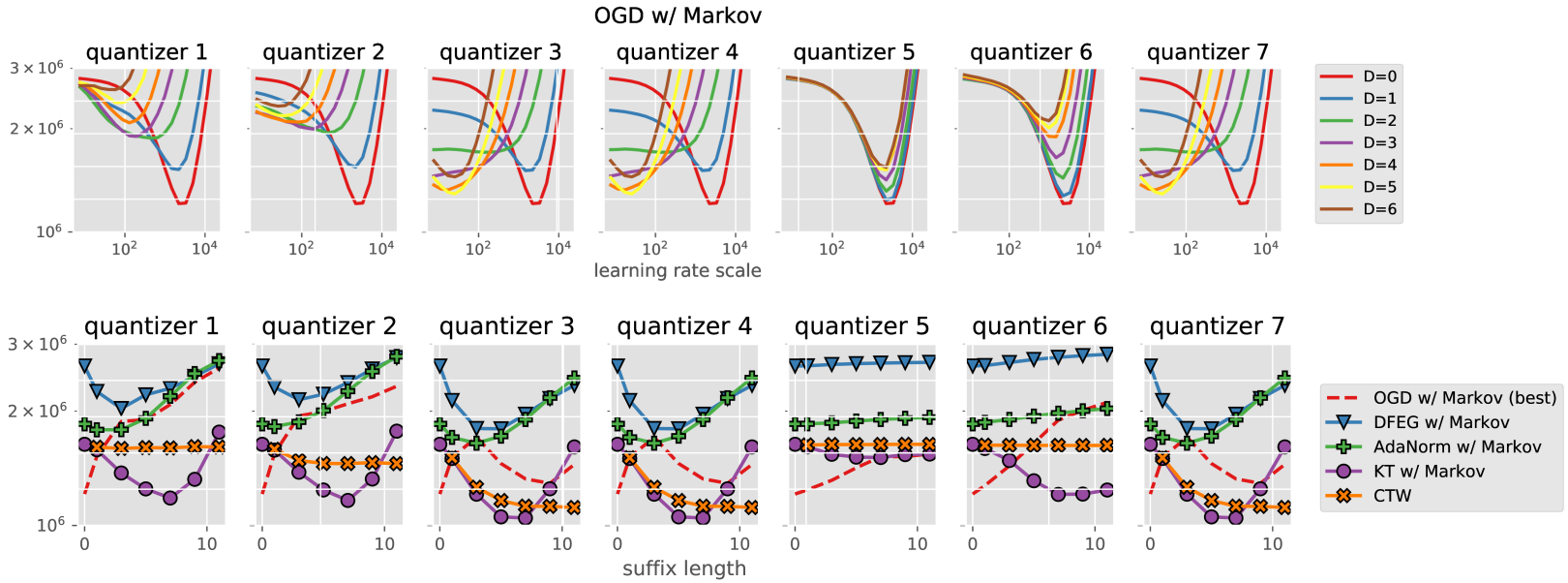


Figure 6.E.2. Beijing PM2.5 dataset (Liang et al., 2015). See the caption of Figure 6.E.1 for details.

Acknowledgement

Chapter 6, in part, is a reprint of the material in the paper: J. Jon Ryu, Alankrita Bhatt, and Young-Han Kim, "Parameter-Free Online Linear Optimization with Side Information via Universal Coin Betting," In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, A Virtual Conference, March 2022. The dissertation author was the primary investigator and author of this paper. This work was supported in part by the National Science Foundation under Grant CCF-1911238

Bibliography

Heinz H Bauschke and Patrick L Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer, 2011.

Ron Begleiter, Ran El-Yaniv, and Golan Yona. On prediction using variable order Markov models. *J. Artif. Intell. Res.*, 22:385–421, 2004.

Aditya Bhaskara, Ashok Cutkosky, Ravi Kumar, and Manish Purohit. Online learning with imperfect hints. In *Proc. Int. Conf. Mach. Learn.*, pages 822–831. PMLR, 2020a.

Aditya Bhaskara, Ashok Cutkosky, Ravi Kumar, and Manish Purohit. Online linear optimization with many hints. *arXiv preprint arXiv:2010.03082*, 2020b.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

Kamalika Chaudhuri, Yoav Freund, and Daniel Hsu. A parameter-free hedging algorithm. In *Adv. Neural Inf. Proc. Syst.*, volume 22. Curran Associates, Inc., 2009.

Liyu Chen, Haipeng Luo, and Chen-Yu Wei. Impossible tuning made possible: A new expert algorithm and its applications. *arXiv preprint arXiv:2102.01046*, 2021.

Alexey Chernov and Vladimir Vovk. Prediction with advice of unknown number of experts. In *Proc. Uncertain. Artif. Intell.*, 2010.

Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2006.

Ashok Cutkosky. Combining online learning guarantees. In *Conf. Learn. Theory*, pages 895–913. PMLR, 2019.

Ashok Cutkosky and Kwabena Boahen. Online learning without prior information. In *Conf. Learn. Theory*, pages 643–677. PMLR, 2017.

Ofer Dekel, Arthur Flajolet, Nika Haghtalab, and Patrick Jaillet. Online learning with a

hint. In *Adv. Neural Inf. Proc. Syst.*, volume 30, pages 5299–5308. Curran Associates, Inc., 2017.

Dheeru Dua and Casey Graff. UCI Machine Learning Repository, 2019. URL http://archive.ics.uci.edu/ml.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12(7), 2011.

Dylan J Foster, Alexander Rakhlin, and Karthik Sridharan. Adaptive online learning. In *Adv. Neural Inf. Proc. Syst.*, volume 28, pages 3375–3383. Curran Associates, Inc., 2015.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1): 119–139, 1997.

Yoav Freund, Robert E Schapire, Yoram Singer, and Manfred K Warmuth. Using and combining predictors that specialize. In *Proc. Annu. ACM Symp. Theory Comput.*, pages 334–343, 1997.

John Hogue. Metro interstate traffic volume data set, 2019. URL https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Volume.

Jiantao Jiao, Haim H Permuter, Lei Zhao, Young-Han Kim, and Tsachy Weissman. Universal estimation of directed information. *IEEE Trans. Inf. Theory*, 59(10):6220–6242, 2013.

Kwang-Sung Jun and Francesco Orabona. Parameter-free online convex optimization with sub-exponential noise. In *Conf. Learn. Theory*, pages 1802–1823. PMLR, 2019.

Kwang-Sung Jun, Francesco Orabona, Stephen Wright, and Rebecca Willett. Online learning for changing environments using coin betting. *Electron. J. Stat.*, 11(2):5282–5310, 2017.

John Larry Kelly Jr. A new interpretation of information rate. *IRE Trans. Inf. Theory*, 3(2): 185–189, 1956.

Wouter M Koolen and Tim Van Erven. Second-order quantile methods for experts and combinatorial games. In *Conf. Learn. Theory*, pages 1155–1175. PMLR, 2015.

Suleyman S Kozat, Andrew C Singer, and Andrew J Bean. Universal portfolios via context trees. In *Proc. IEEE Int. Conf. Acoust. Speech. Signal Process.*, pages 2093–2096. IEEE, 2008.

Raphail Krichevsky and Victor Trofimov. The performance of universal encoding. *IEEE Trans. Inf. Theory*, 27(2):199–207, 1981.

Ilja Kuzborskij and Nicolò Cesa-Bianchi. Locally-adaptive nonparametric online learning. In *Adv. Neural Inf. Proc. Syst.*, volume 33, 2020.

Xuan Liang, Tao Zou, Bin Guo, Shuo Li, Haozhe Zhang, Shuyi Zhang, Hui Huang, and Song Xi Chen. Assessing Beijing's PM2.5 pollution: Severity, weather impact, APEC and winter heating. *Proc. R. Soc. A*, 471(2182):20150257, 2015.

Haipeng Luo and Robert E Schapire. Achieving all with no parameters: AdaNormalHedge. In *Conf. Learn. Theory*, pages 1286–1304. PMLR, 2015.

H Brendan McMahan and Jacob Abernethy. Minimax optimal algorithms for unconstrained linear optimization. In *Adv. Neural Inf. Proc. Syst.*, volume 26. Curran Associates, Inc., 2013.

H Brendan McMahan and Francesco Orabona. Unconstrained online linear learning in Hilbert spaces: Minimax algorithms and normal approximations. In *Conf. Learn. Theory*, pages 1020–1039. PMLR, 2014.

João V Messias and SA Whiteson. Dynamic-depth context tree weighting. In *Adv. Neural Inf. Proc. Syst.*, volume 31. Curran Associates, Inc., 2018.

Francesco Orabona. Dimension-free exponentiated gradient. In *Adv. Neural Inf. Proc. Syst.*, volume 26, pages 1806–1814. Curran Associates, Inc., 2013.

Francesco Orabona. Simultaneous model selection and optimization through parameter-free stochastic learning. *arXiv preprint arXiv:1406.3816*, 2014.

Francesco Orabona. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.

Francesco Orabona and Ashok Cutkosky. ICML 2020 tutorial on parameter-free online optimization. Websites: https://parameterfree.com/icml-tutorial/, https://icml.cc/Conferences/2020/Schedule?showEvent=5753, 2020.

Francesco Orabona and Dávid Pál. Coin betting and parameter-free online learning. In *Adv. Neural Inf. Proc. Syst.*, volume 29. Curran Associates, Inc., 2016.

Francesco Orabona and Tatiana Tommasi. Training deep networks without learning rates through coin betting. In *Adv. Neural Inf. Proc. Syst.*, volume 30. Curran Associates, Inc., 2017.

Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. In *Conf. Learn. Theory*, pages 993–1019. PMLR, 2013.

Jorma Rissanen. Universal coding, information, prediction, and estimation. *IEEE Trans. Inf. Theory*, 30(4):629–636, 1984.

Jorma J Rissanen. Fisher information and stochastic complexity. *IEEE Trans. Inf. Theory*, 42(1):40–47, 1996.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Found. Trends Mach. Learn.*, 4(2):107–194, 2011.

Dirk Van der Hoeven, Tim van Erven, and Wojciech Kotłowski. The many faces of exponential weights in online learning. In *Conf. Learn. Theory*, pages 2067–2092. PMLR, 2018.

Frans MJ Willems, Yuri M Shtarkov, and Tjalling J Tjalkens. The context-tree weighting method: Basic properties. *IEEE Trans. Inf. Theory*, 41(3):653–664, 1995.

Frans MJ Willems, Tjalling J Tjalkens, and Tanya Ignatenko. Context-tree weighting and maximizing: Processing betas. In *Proc. UCSD Inf. Theory Appl. Workshop*, 2006.

Qun Xie and Andrew R Barron. Minimax redundancy for the class of memoryless sources. *IEEE Trans. Inf. Theory*, 43(2):646–657, 1997.

Lijun Zhang, Guanghui Wang, Jinfeng Yi, and Tianbao Yang. A simple yet universal strategy for online convex optimization. *arXiv preprint arXiv:2105.03681*, 2021.

Jacob Ziv and Abraham Lempel. A universal algorithm for sequential data compression. *IEEE Trans. Inf. Theory*, 23(3):337–343, 1977.