Title

The curse of dimension in nonparametric regression

Author
Kpotufe, Samory

Publication Date
2010

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA, SAN DIEGO

The curse of dimension in nonparametric regression

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

in

Computer Science

by

Samory Kpotufe

Committee in charge:

Professor Sanjoy Dasgupta, Chair
Professor Ian Abramson
Professor Ery-Arias Castro
Professor Yoav Freund
Professor Lawrence Saul

2010

The dissertation of Samory Kpotufe is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2010

DEDICATION

To my mother, Louise Ayovi Dossou-Lawson, whose strength in life I have tried to emulate, to my step-father, Amen Boevi Lawson, whose patience I've always admired, to my father, Prosper Papato Kpotufe, whose free spirit I inherited.

EPIGRAPH

I see a pattern, but my imagination cannot picture the maker of that pattern. I see a clock, but I cannot envision the clockmaker. The human mind is unable to conceive of the four dimensions, so how can it conceive of a God, before whom a thousand years and a thousand dimensions are as one? —Albert Einstein.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I thank God or whatever is out there, for my belief in a higher reality helps me deal with life.

I am grateful to people whose support allowed me to pursue and complete my Ph. D. First, my family: parents, siblings and cousins, including those that became family later in life. I thank Anna Amagou for her support in the crucial first years back in school. I thank Jade Power for her true friendship over the last years. I thank my undergraduate professors Jim Hagler, and George Edwards for their guidance and for encouraging me to continue my education.

Special thanks to my Ph. D advisor Sanjoy Dasgupta for all his help and advice throughout graduate school. I thank my Ph. D committee, Yoav Freund, Lawrence Saul, Ian Abramson, and Ery Arias-Castro for their time and advice.

Some of the results appearing in this dissertation are based on collabrations with Sanjoy Dasgupta and Nakul Verma. Some of this work has been published or submitted:

- Portions of Chapter 2 and Chapter 4 appear in
  N. Verma, S. Kpotufe, S. Dasgupta, "Which spatial partition trees are adaptive to intrinsic dimension?", *Uncertainty in Artificial Intelligence*, 2009.

- Most of Chapter 5 and portions of the Appendix appear in
  S. Kpotufe, S. Dasgupta, "A tree-based regressor that adapts to intrinsic dimension", Journal of Computer and System Sciences, Special Issue on Learning Theory, (Invited Submission).

This work was supported by the National Science Foundation (under grants IIS-0347646, IIS-0713540, and IIS-0812598) and by a fellowship from the Engineering Institute at the Los Alamos National Laboratory.

VITA

1999	B. S. in Mathematics, University of Denver, Denver, CO
2007	M. S. in Computer Science, University of California, San Diego, CA
2010	Ph. D. in Computer Science, University of California, San Diego, CA

PUBLICATIONS

S. Kpotufe, S. Dasgupta, “A tree-based regressor that adapts to intrinsic dimension”, Journal of Computer and System Sciences, Special Issue on Learning Theory, (Invited Submission).

S. Kpotufe, “Escaping the curse of dimension with a tree-based regressor”, *Conference on Learning Theory*, 2009.

N. Verma, S. Kpotufe, S. Dasgupta, “Which spatial partition trees are adaptive to intrinsic dimension?”, *Uncertainty in Artificial Intelligence*, 2009.

S. Kpotufe, “Fast, smooth and adaptive regression in metric spaces”, *Neural Information Processing Systems*, 2009.

ABSTRACT OF THE DISSERTATION

The curse of dimension in nonparametric regression

by

Samory Kpotufe

Doctor of Philosophy in Computer Science

University of California, San Diego, 2010

Professor Sanjoy Dasgupta, Chair

We consider the problem of *nonparametric regression*, consisting of learning an arbitrary mapping $f : \mathcal{X} \to \mathcal{Y}$ from a data set of (X, Y) pairs in which the Y values are corrupted by noise of mean zero. This statistical task is known to be subject to a so-called "curse of dimension": if $\mathcal{X} \subset \mathbb{R}^D$, and if the only smoothness assumption on f is that it satisfies a Lipschitz condition, it is known that any estimator based on n data points will have an error rate (risk) of $\Omega(n^{-2/(2+D)})$. In other words a data size exponential in D is required to approximate f, which is unfeasible even for relatively small D.

Fortunately, high-dimensional data often has low-intrinsic complexity (e.g. manifold data, sparse data) and some nonparametric regressors perform better in such situations. This dissertation presents and analyzes various fast regressors that escape the curse of dimension in situations where data has low-intrinsic complexity. These nonparametric regressors, namely tree and tree-kernel-hybrid regressors, have strong theoretical guarantees which are verifiable on a wide range of real-world data.

Chapter 1

Introduction

1.1 Nonparametric regression

Given a set of data points (X, Y), where $Y = f(X) + \text{noise}$ (of mean zero), is it possible to infer the unknown function f? This is the statistical problem of *regression* (illustrated in Figure 1.1). This problem is of increasing importance today: due to the tremendous growth of information technology, there is a growing need for procedures that can automatically extract patterns from information databases that are prohibitively large for humans to handle. One of the fundamental ways for dealing with this "pattern-recognition" task is regression estimation. Formally, given n data pairs (X, Y) we want to construct a function f_n (the regressor) which approximates f.

The following simple examples are meant to give a sense of the wide practical relevance of the regression problem.

Example 1 (Marketing)**.** *A company is interested in predicting the amount of profit (Y) it gains by marketing a certain product to particular customers (X), where each customer is described by a list of attributes such as income, age, location, etc. Given data collected from n previous customers, it builds a model $Y = f_n(X)$ that is used to assess the potential profits on future customers with similar attributes.*

Example 2 (Epidemiology)**.** *A United Nations epidemiologist is tasked with as-*

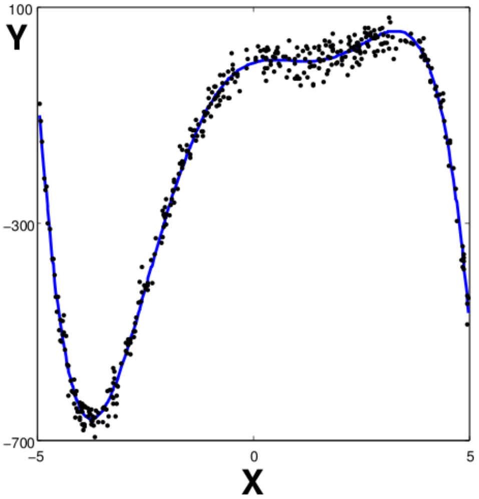


Figure 1.1: Regression data $Y = f(X) + \text{noise}$, $f(\cdot)$ shown in blue.

sessing the spread (Y*) of a new virus in some part (*X*) of the world, where parts of the world are described by attributes such as poverty level, sanitation level, age spread, etc. To accomplish the task, the epidemiologist examines historical data on the spread of similar viruses, and builds a model* $Y = f_n(X)$ *that is used to predict the number of people that might be affected if no action is taken.*

Example 3 (Robot Learning)**.** *The field of Robot Learning aims to develop robotic tools that can learn to perform new tasks. For example, in obstacle avoidance, a set of obstacles is presented to a robotic vehicle, which is then shown how to avoid it. Next time it sees a similar obstacle it should be able to avoid it on its own. Here the* X *variable consists of descriptive features of an obstacle. The* Y *value is the path to be taken to avoid the obstacle.*

Another example is that of learning robot dynamics. Suppose we want to derive the right amount of force to apply in various places of a robotic arm to move it from a state to the other, where each state describes the physical position, speed and acceleration of the arm joints. Here experts can be called on to analyze the arm and derive the right dynamics equations. This might be time consuming and expensive. A cheaper alternative is to train the robotic arm as follows. The X *variable is the description of the current state and the desired state. The* Y

variable is a description of the various forces to apply. Some initial experiments are ran where various forces are applied and the movements from one state to the next are recorded. Using this data, a model $Y = f_n(X)$ of the dynamics is built which can now be used to move the arm in the future.

Example 4 (Farming)**.** *A farmer is interested in predicting the crop yield (Y) in various parcels (X) of land, where each parcel is described by its mineral contents. The farmer collects data over the years from which he builds a model $Y = f_n(X)$ that enables him to better predict crop yields in subsequent years.*

Depending on the application domain, one might have some knowledge about the form of the function f being estimated. For instance, we might know that f is linear in X, i.e. $f(X) = b{\cdot}X$ for some unknown parameter b, or that f can be well approximated by such a linear function. The regression task then consists of estimating the parameter b, and this easily done with a bit of linear algebra (see e.g. [DHS01]). Unfortunately there is often little or no a priori information about f in practice, and therefore no parametric form might be assumed. This motivates the development of *nonparametric* regression procedures that can infer fairly arbitrary functions f from data. Formally, we say that a regression approach is nonparametric if it only assumes that the function f belongs to some *infinite* dimensional class. In contrast, the set of linear functions is a vector space of finite dimension coinciding with that of X.

1.1.1 Estimation error

Given the goal of approximating f with the regressor f_n, we have to agree on a notion of approximation error. But first we need a bit of formalism.

The data $(\mathbf{X}, \mathbf{Y}) \doteq \{(X_i, Y_i)\}_{i=1}^n$ are assumed to be drawn i.i.d from a distribution over a joint input-output space $\mathcal{X} \times \mathcal{Y}$. The input space $\mathcal{X}$ is usually assumed to be a subset of $\mathbb{R}^D$, i.e. X is a vector of D features (attributes in the above examples). The output space $\mathcal{Y}$ is assumed to be a subset of $\mathbb{R}^{D'}$, and is a random vector satisfying $\mathbb{E}\left[Y|X = x\right] = f(x)$, i.e. $Y = f(X) + \eta$ where the noise vector η has mean $0 \in \mathbb{R}^{D'}$.

Suppose $g : \mathcal{X} \to \mathcal{Y}$ is some estimate of f. We define its l_2 *pointwise risk at* $x \in \mathcal{X}$ to be $R(g,x) \doteq \mathbb{E}_{Y|X=x} \|Y - g(x)\|^2$ and its *integrated risk* to be $R(g) \doteq \mathbb{E}_X R(g,X)$. Standard manipulations show that

$$\begin{aligned} R(g,x) &= R(f,x) + \|g(x) - f(x)\|^2 \\ \therefore R(g) &= R(f) + \mathbb{E}_X \|g(X) - f(X)\|^2 . \end{aligned}$$

Thus, the pointwise excess risk of $g(x)$ over $f(x)$ is simply $\|f(x) - g(x)\|^2$. We are interested in the *integrated excess risk* of the regressor f_n, namely

$$\|f_n - f\|^2 \doteq R(f_n) - R(f) = \mathbb{E}_X \|f_n(X) - f(X)\|^2 . \tag{1.1}$$

The excess risk decomposes nicely into bias and variance terms, which will be useful in a lot of the analysis in this dissertation by allowing us to focus on these terms separately. Let $f_n(x)$ be any regression estimate and define

$$\widetilde{f}(x) \doteq \mathop{\mathbb{E}}_{\mathbf{Y}|\mathbf{X}} f_n(x),$$

that is the conditional expectation of the estimate, for $\mathbf{X}$ fixed. A bit of algebra yields the decomposition:

$$\mathop{\mathbb{E}}_{\mathbf{Y}|\mathbf{X}} \|f_n(x) - f(x)\|^2 = \mathop{\mathbb{E}}_{\mathbf{Y}|\mathbf{X}} \left\|f_n(x) - \widetilde{f}_n(x)\right\|^2 + \left\|\widetilde{f}_n(x) - f(x)\right\|^2 . \tag{1.2}$$

The excess risk (1.1) will be our notion of error of f_n relative to f. It is well known that there exists universally consistent regressors such as kernel regressors, i.e. $\|f_n - f\| \to 0$[1]. However, in practice we are more interested in assessing the number of samples required for a good estimate, in other words, we need to characterize the rate at which $\|f_n - f\|$ goes to 0 relative to n. This is where we encounter the curse of dimension.

1.2 Curse of dimension

In order to guarantee that a regressor f_n converges to f at a reasonable rate, we need f to be reasonably smooth[2]. A common smoothness assumption,

[1] The limit is to be understood in the wide sense of convergence of random variables. Consistency can be shown under mild conditions such as boundedness of X and bounded second moment for Y (see e.g. [GKKW02])

[2] If f is allowed to be arbitrary, then the convergence rate of f_n to f can be arbitrarily slow, i.e. for any sequence $\{a_n\}$, $a_n \to 0$, there exists an f and a distribution on $\mathcal{X} \times \mathcal{Y}$ such that

which we use throughout this dissertation, is that f is Lipschitz, i.e. there exists λ unknown such that $\forall x, x', \|f(x) - f(x')\| \leq \lambda \|x - x'\|$. Under this assumption, a number of regressors can be shown to satisfy

$$\mathbb{E}_{\mathbf{X},\mathbf{Y}} \|f_n - f\|^2 \leq O(n^{-2/(2+D)}),$$

where D is the dimension of the input space $\mathcal{X}$. This is for instance the rate for kernel regressors, k-NN regressors, and tree-based regressors [GKKW02]. This is quite a slow rate since it implies that we need a sample size n exponential in D in order to approximate f. Unfortunately, D is often so high (say $D > 30$) in modern applications that $n > 2^D$ is impractical.

In order to get an intuition as to the reason for such a rate, consider that nonparametric approaches such as the aforementioned operate by approximating the target function locally (on its domain $\mathcal{X}$) by simpler functions. There is necessarily some local errors and these errors aggregate globally. Thus to approximate the entire function well, we need to do well in *most* local areas. Suppose for instance that the target function is well approximated by constants in regions of radius at most $0 < r < 1$. In how many ways can we divide up the domain $\mathcal{X}$ into smaller regions of radius at most r? If $\mathcal{X}$ is D-dimensional then the smallest such partition is of size $O(r^{-D})$. We will need data points to fall into each such region if we hope to do well locally everywhere. In other words, we will need a data size exponential in D.

Unfortunately it turns out this is the best we can hope for no matter the regression approach, a grim fact formalized in Theorem 5 below. The minimax rates established by the theorem were first obtained by Stone in [Sto80, Sto82].

Theorem 5 (Paraphrasing Theorem 3.2 of [GKKW02])**.** *Let $\mathcal{D}_\lambda$ be the class of distributions of (X, Y) such that $f : [0,1]^D \mapsto \mathbb{R}$ is λ-Lipschitz, X is uniformly distributed on $[0,1]^D$, $Y = f(X) + \eta$, where $\eta \sim \mathcal{N}(0,1)$ and X is independent of η. Then there exists $C > 0$ independent of λ such that*

$$\varliminf_{n\to\infty} \inf_{f_n} \sup_{\mathcal{D}(X,Y)\in\mathcal{D}_\lambda} \frac{\|f_n - f\|^2}{\lambda^{2d/(2+d)} n^{-2/(2+d)}} > C,$$

$\lim_{n\to\infty} \mathbb{E}_{\mathbf{X},\mathbf{Y}} \|f_n - f\|^2 / a_n > 1$ (see [GKKW02]).

where the inf *is taken over all regressors f_n (viewed as maps from the sample space $(\mathcal{X} \times \mathcal{Y})^n$ to the set of functions from $[0,1]^D \mapsto \mathbb{R}$) and the* sup *is taken over data-distributions in $\mathcal{D}_\lambda$.*

Notice that a similar statement is automatically implied for any class of distributions containing $\mathcal{D}_\lambda$. Thus this theorem paints a rather gloomy picture for nonparametric regression in high-dimensional spaces. Fortunately there is hope in the fact that, in practice, many high-dimensional datasets actually lie near a subspace of lower-complexity than indicated by the dimension D. This is discussed in the following section.

1.3 Intrinsic dimension

The curse of dimension would appear to rule out nonparametric approaches for the increasingly high-dimensional data sets that arise in modern applications. In image retrieval, or text classification, or genomic analysis, for instance, the number of features, or dimensions, of X can easily grow to tens of thousands, or more. However, in many of these cases, it is believed that the dimensionality is large only in the superficial sense of there being many coordinates, whereas the true degrees of freedom are much smaller in number. This might occur, for example, because of strong dependencies between the features and/or because some features might be irrelevant. In this sense a high-dimensional dataset might have low *intrinsic* dimension. For the time being, we use the term “intrinsic dimension” informally to describe the inherent complexity of a dataset. In Chapter 2 we present various formalisms that capture this notion. For now, let us go over a few examples of datasets whose inherent complexity is far lower than indicated by their ambient dimension.

Example 6 (Linear data)**.** *A classical example of a high-dimensional dataset with low-intrinsic dimension is that of a dataset $\mathcal{X} \subset \mathbb{R}^D$ that lies near a d-dimensional affine subspace ($d < D$). In this case Principal Component Analysis (PCA [DHS01]) can be used to re-map $\mathcal{X}$ into $\mathbb{R}^d$ while preserving most information.*

Example 7 (Sparse data). *This is a situation where each vector $x \in \mathcal{X} \subset \mathbb{R}^D$ has at most $d \ll D$ non-zero coordinates. Consider for instance a dataset consisting of black and white images of handwritten digits. Although each image is described by many pixels, few pixels are black in each (these are the nonzero coordinates), in other words the number of pixels D is an overestimate of the true complexity of the data.*

Another example of sparse data arises in document classification. The most common way of representing a document is as a vector with one coordinate per word, which describes whether or not that word appears in the document (or the number of times the word appears, or some function thereof). The dimensionality D is therefore the size of the vocabulary, which is typically in the tens of thousands. However, any given document only contains a few hundred (or so) words, and thus most of its vector is zero: it is sparse. In a sense, the intrinsic dimension d of the data is the average number of non-zero entries, which is much smaller than D.

Example 8 (Manifold data). *A speech signal is typically represented by a high-dimensional time series: the signal is broken into overlapping windows, and a variety of filters is applied within each window. Even richer representations can be obtained by using more filters, or by concatenating vectors corresponding to consecutive windows. In this way, the dimensionality D can be made arbitrarily high. However, the physical system can alternatively be described by just a few ($d \ll D$) parameters specifying the configuration of the speaker's vocal apparatus. These are the true degrees of freedom of the data, and as they vary, the high-dimensional representation traces out a d-dimensional submanifold of $\mathbb{R}^D$. It is generally believed that sensory data generated from physical structures with few degrees of freedom appear high-dimensional while conforming to a low-dimensional manifold structure.*

Many methods have been developed to handle the case where data lies near a manifold. These methods, termed *manifold learning*, consist of embedding the nonlinear data into a lower dimensional space while preserving key properties such as interpoint distances [RS00, BN03, TSL00]. Manifold learning might be used as a preprocessing step to regression. However the approach does not easily yield theoretical guarantees in a distribution-free regression setting. Our interest is in circumventing the embedding step and automatically adapting to low intrinsic dimension while operating in the original space $\mathbb{R}^D$.

1.4 Overview of results and related work

How do we benefit from situations where high-dimensional data has low-intrinsic dimension? As mentioned earlier, the common approach is to remap the data into a lower-dimensional space and perform regression in this space. If the data lies near a linear-subspace then PCA might be used to remap it. Otherwise if it lies near a manifold, non-linear embedding techniques from manifold learning might be used. If instead the data is more complex, e.g. a collection of manifolds as is the case with sparse data, it is no longer clear how to properly reduce dimension. Thus the embedding approach is limited to datasets that have a certain degree of regularity. Unfortunately, even in cases where the data has enough regularity (e.g. it lies on a smooth manifold), the embedding approach does not easily yield theoretical guarantees for regression. In Chapter 3 we discuss the various reasons for this lack of guarantees, most important of which being that manifold learning techniques generally do not embed the whole data space $\mathcal{X}$, but only embed the training data $\mathbf{X}$.

Our interest is in circumventing the embedding step and automatically adapting to low intrinsic dimension while operating in the original space $\mathbb{R}^D$. *Adaptivity to intrinsic dimension* is the main subject of this dissertation, and it refers to the ability of a regressor to operate in $\mathbb{R}^D$ while achieving a convergence rate that depends just on the intrinsic dimension of the data.

We investigate and compare various formalisms of intrinsic dimension in Chapter 2, and we develop a simple estimator which we use to verify that many high-dimensional real-world data do have low-intrinsic dimension. Some of the formal notions of intrinsic dimension presented in Chapter 2 are broad enough to capture the many different situations discussed earlier such as sparse data and manifold data. In this dissertation we work primarily with one such broad notion called the Assouad dimension, and we demonstrate several efficient regressors that are adaptive to intrinsic dimension.

Here we emphasize efficiency in contrast with more traditional regressors such as kernel and k-NN regressors. Kernel regression was recently shown to be adaptive (Bikel and Li [BL06]), while there is evidence that k-NN (k Nearest Neighbor) regression might also be adaptive (Kulkarni and Posner [KP95] [3], also see Chapter 3). These traditional methods can be expensive in practice. Either kernel weights must be computed at many training points, resulting in an $\Omega(n)$ evaluation time, or the k_n nearest neighbors of a query point must be located, where k_n is optimally chosen as a root of n [GKKW02]. This sort of time complexity can be a burden in practice considering that nonparametric regression usually depends upon large data sizes n for accuracy. Hence the appeal for adaptive regressors that are efficient, as in operating in $O(\log n)$ time.

Natural candidates for efficient adaptive regressors are tree-based regressors. A tree-based regressor works by building a hierarchical partition (the tree) of the space $\mathcal{X}$ and learns simple functions such as constants in the leaf cells of the tree. An attractive property of this estimator is that $f_n(x)$ can be evaluated by simply navigating down to the leaf containing x, which takes time proportional to the height of the tree, often just $O(\log n)$. This computational efficiency, and an overall ease of use, have motivated a variety of tree partition methods such as CART, dyadic trees, and k-d trees [GN05, SN06a, DGL96a], but none of these had been shown to adapt to intrinsic dimension in its regression risk.

In Chapter 4 we introduce tree-based regression in general and we argue that a tree-based regressor is adaptive to intrinsic dimension if it decreases the diameter of data within its cells at a fast rate that depends on this dimension. First, the link between data diameter decrease rate and the performance of a tree-based regressor is established empirically on real-world datasets. Then, building on earlier work of Dasgupta and Freund [DF08a], we establish the rate at which various trees decrease data diameter in terms of (intrinsic) dimension. Many (unusual) trees

[3] The cited work does not directly treat the problem of adaptivity but expresses the convergence rate of Nearest Neighbor regression ($k = 1$) in terms of the box dimension, which is a known measure of intrinsic dimension.

are guaranteed to decrease diameter at a fast rate when data is intrinsically low-dimensional, but more common methods such as dyadic trees and k-d trees can be shown not to have such guarantees.

In Chapter 5 we formalize the intuition developed earlier in Chapter 4, namely that fast data diameter decrease implies good regression rates for tree-based regressors. Data diameters are unstable quantities, i.e. might vary a lot from sample to sample. Thus they present a unique difficulty for regression analysis. We develop novel techniques to relate data diameters to the bias of a tree-based estimator. These techniques apply generally although in Chapter 4 we focus our analysis on a particular tree, the Random Projection tree (RP tree), a randomized variant of k-d trees first analyzed by Dasgupta and Freund in the context of vector quantization [DF08a]. We show that, given data from a space $\mathcal{X} \subset \mathbb{R}^D$ of Assouad dimension d, an RP tree regressor attains an excess risk of $O(n^{-2/(2+k)})$ where $k = O(d \log d)$ describes the rate at which RP tree decreases data diameter.

Another family of efficient adaptive regressors covered in this dissertation is that of tree-kernel hybrids which combine aspects of both tree-based and kernel regressors. The motivation for this family of methods is that, while adaptive tree-based regressors perform well relative to non-adaptive tree methods, they still lag relative to the regression accuracy of kernel methods, as seen both in practice and in the achievable theoretical bounds. The excess risk of a kernel regressor in terms of Assouad dimension is $O(n^{-2/(2+d)})$ while the rate for an RP tree for example is $O(n^{-2/(2+Cd \log d)})$. We show in Chapter 6 that by combining elements of both tree-based and kernel regression, we can achieve regression accuracy comparable to that of kernel regression while maintaining most of the time efficiency of tree-based regression. In particular, if $\mathcal{X}$ has Assouad dimension d, some tree-kernel hybrids achieve an excess risk of $O(n^{-2/(2+d)})$ with a time complexity of $C \log n$ where C depends on d. These tree-kernel hybrids offer many other parctical benefits discussed in Chapter 6. In particular, a very appealing aspect of these regressors is that they provide tunable parameters that allow the practitioner to attain tradeoffs between time-efficiency and accuracy as appropriate for their application.

1.5 Open questions and extensions

1.5.1 Higher order smoothness assumptions

Suppose we knew the regression function f satisfies higher order smoothness conditions than mere Lipschitz conditions. For instance f might be p times differentiable ($p > 1$ an integer) with all partial derivatives of order p bounded. It is well known that the minimax regression rates in this case are of the form $n^{-2p/(2p+D)}$ [GKKW02]. To attain this rate, one has to learn a polynomial of degree $p - 1$ in the neighborhood of the query x as opposed to learning a simple constant as is the case with the methods presented in this dissertation. We would expect adaptive rates to then take the form $n^{-2p/(2p+d)}$ where d is an appropriate notion of intrinsic dimension.

It turns out that the techniques used to establish adaptive rates in this dissertation can be extended to higher order smoothness assumptions using higher order local polynomials for the regression estimates. This is because higher order smoothness assumptions only affect the form of the bias of the estimator, and the variance maintains the same form. As we will see throughout the dissertation, the (intrinsic) dimension mainly appears in the estimator's variance (for the estimators considered here). In this sense, we believe that most of the discussion in this dissertation also provides insight into regression with higher order polynomials.

1.5.2 More practical notions of dimension

The notions of intrinsic dimension used in this work are meant to capture the worst-case intrinsic complexity over any subset of the space $\mathcal{X}$ (see Chapter 2). However, it is conceivable that while some data set might be high dimensional in some parts of space, it could have low intrinsic dimension in other parts of space. The situation can be even more complex: the intrinsic dimension of the data might depend not only on the region of space but also on the scale at which the data is being examined; furthermore the intrinsic dimension in some region of space might not be relevant if this region has low mass under the data distribution. How do we formalize these ideas, and can we develop regression procedures that can identify these situations and benefit from them? These questions require further investigation.

Chapter 2

Intrinsic dimension

There are many ways to formalize intrinsic dimension. We aim to work with as broad a notion as possible which captures the various scenarios discussed in the introduction where high-dimensional data has low intrinsic complexity.

In this dissertation we work primarily with a notion called the *Assouad dimension*, which is defined for any set of data points in $\mathbb{R}^D$ (or in fact, in any metric space). What makes it particularly attractive is that it generalizes both the notion of manifold dimension and that of sparsity, while at the same time being amenable to the kinds of analysis that arise in algorithm design. There exist many other related notions and we overview some of them in the following section.

Before diving into formal notions of intrinsic dimension, we need to familiarize ourselves with the following definitions which will come in handy throughout this dissertation.

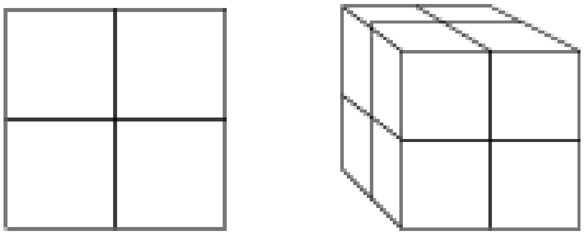
Definition 9. *(Covers, packings, and nets)*

– $\mathbf{Q} \subset A$ *is an* r*-cover of* $A \subset \mathbb{R}^D$ *if for all* $x \in A$*, there exists* $i \in [n]$*, such that* $\|x - q\| < r$*.*

– $\mathbf{Q} \subset A$ *is an* r*-packing if for all* $q, q' \in \mathbf{Q}$*,* $\|q - q'\| \geq r$*.*

– $\mathbf{Q} \subset A$ *is an* r*-net of* $A \subset \mathbb{R}^D$ *if it is an* r*-cover and an* r*-packing.*

We will sometimes talk about the covering of a set by balls. This is simply to say that the set is in the union of these balls.



(a) $2^{\mathbf{2}}$ squares (b) $2^{\mathbf{3}}$ cubes

Figure 2.1: A hypercube in $\mathbb{R}^d$ can be covered by 2^d hypercubes of half the side length.

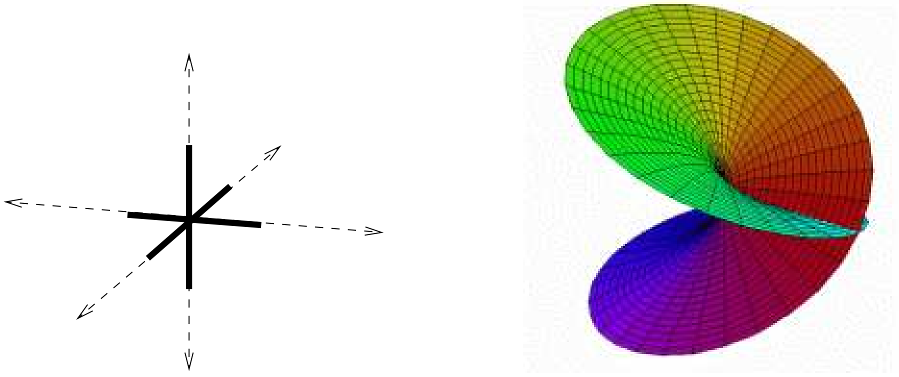
Figure 2.1 serves to give an intuition about how the above definitions are useful in formalizing intrinsic dimension. Consider a natural two-dimensional object such as a square. A square can be covered by $4 = 2^2$ squares of half its side length. Similarly, a cube can be covered by $8 = 2^3$ cubes of half its side length. Notice that the exponent always corresponds to the natural dimension of the object. Many of the formalisms of intrinsic dimension in the following sections will consider the way in which the space can be covered by balls of small radius. Notions such as packing and nets are related and yield crude bounds on cover sizes, as stated in the following lemma.

Lemma 10. *Let $r > 0$. The size of an r-net of a set A is at least the size of an r-cover of A (by definition), and at most the size of an $r/2$-cover of A.*

Proof. Let $\mathbf{Q}$ and $\mathbf{Q}'$ be respectively an r-net and a minimal $r/2$-cover of $\mathcal{A}$. Consider the set of open balls of radius $r/2$ centered at points in $\mathbf{Q}'$. Each of these balls contains at most one point from $\mathbf{Q}$ since these points are at least r-apart and the balls have diameter less than r. However, each point in $\mathbf{Q}'$ is in one of these balls since they cover A. It follows that $|\mathbf{Q}| \leq \mathbf{Q}'$. □

2.1 Formalisms of intrinsic dimension

The question of characterizing the intrinsic dimension of a data space $\mathcal{X}$ has aroused keen interest in many different scientific communities, and has given



(a) Sparse data set.

(b) 2-dimensional manifold.

Figure 2.2: Examples of data with low Assouad dimension.

rise to a variety of definitions. Here are four of the most successful such notions, arranged in decreasing order of generality:

- Covering dimension
- Assouad dimension
- Manifold dimension
- Affine dimension

The most general is the *covering dimension:*

Definition 11 (Covering or box dimension)**.** *The smallest d for which there is a constant $C > 0$ such that for any $\epsilon > 0$, $\mathcal{X}$ has an ϵ-cover of size $C(1/\epsilon)^d$.*

This notion lies at the heart of much of empirical process theory. Although it permits many kinds of analysis and is wonderfully general, for our purposes it falls short on one count: for nonparametric estimators, we often need small covering numbers for $\mathcal{X}$, but also for individual *neighborhoods* of $\mathcal{X}$. Thus we would like this same covering condition (with the same constant C) to hold for all L_2-balls in $\mathcal{X}$. This additional stipulation yields the *Assouad dimension:*

Definition 12 (Assouad or doubling dimension)**.** *The smallest d such that for any (Euclidean) ball $B \subset \mathbb{R}^D$, $X \cap B$ can be covered by 2^d balls of half the radius.*

At the bottom end of the spectrum is the *affine dimension*, which is simply the smallest d such that $\mathcal{X}$ is contained in a d-dimensional affine subspace of $\mathbb{R}^D$. It is a tall order to expect this to be smaller than D, although we may hope that $\mathcal{X}$ lies close to such a subspace. A more general hope is that $\mathcal{X}$ lies on (or close to) a d-dimensional Riemannian submanifold of $\mathbb{R}^D$. The manifold assumption is clearly more general since If $\mathcal{X}$ has an affine dimension of d, it certainly has manifold dimension at most d.

Similarly, low Assouad dimension implies small covering numbers:

Lemma 13. *If $\mathcal{X}$ has diameter C and Assouad dimension d, then for any $\epsilon > 0$, it has an ϵ-cover of size at most $(2C/\epsilon)^d$.*

Proof. Applying the doubling condition recursively, $\mathcal{X}$ can be covered by one ball of radius C, 2^d balls of radius $C/2$, 2^{2d} balls of radius $C/4$, and so on. □

The only nontrivial containment result is that if $\mathcal{X}$ is a d-dimensional Riemannian submanifold with bounded curvature, then sufficiently small neighborhoods of $\mathcal{X}$ (where this neighborhood radius depends on the curvature) have Assouad dimension $O(d)$. This result (see Lemma 14 of Section 2.2) is formalized and proved in [DF08b]. This result requires a clean formulation of curvature. [NSW06] has recently suggested formulation in which the curvature is captured by a single value which they call the *condition number* of the manifold. Similar notions have earlier been used in the computational geometry literature [AB98].

Lemma 14. *[DF08a] If a d-dimensional Riemannian submanifold of $\mathbb{R}^D$ has bounded condition number $\tau < \infty$, then its neighborhoods of radius $< 1/\tau$ have Assouad dimension $O(d)$.*

The containment is strict: there is a substantial gap between manifolds of bounded curvature and sets of low Assouad dimension, on account of the smoothness properties of the former. This divide is not just a technicality but has important algorithmic implications. For instance, a variant of the Johnson Lindenstrauss lemma states that when a d-dimensional manifold (of bounded curvature) is projected onto a random subspace of dimension $O(d/\epsilon^2)$, then all interpoint distances are preserved within 1 ± ϵ factor [BW07], [Cla07]. This does not hold for sets of Assouad dimension d [IN07]. In fact, the Assouad dimension, and hence the covering dimension, capture the complexity of more general sets with less regularity than a smooth manifold. For instance $\mathcal{X}$ might be made up of many pieces of low-intrinsic dimension with no structural restriction. For instance a set of n points can always be covered by n balls, and therefore has Assouad dimension at most $\log n$ (where the logarithm is taken base two). Also a sparse dataset can be viewed as a finite collection of hyperplanes and thus has low Assouad dimension. We have the following two lemmas.

Lemma 15. *Suppose sets $S_1, \ldots, S_n$ each have Assouad dimension $\leq d$. Then $S_1 \cup \cdots \cup S_n$ has Assouad dimension at most $d + \log n$.*

Proof. Pick any ball B; by hypothesis $B \cap S_i$ can be covered by 2^d balls of half the radius. Therefore $B \cap (S_1 \cup \cdots \cup S_n)$ can be covered by $n \cdot 2^d$ such balls. □

Lemma 16. *Suppose that $S \subset \mathbb{R}^D$ is k-sparse: that is, each point in S has at most k nonzero coordinates. Then S has Assouad dimension at most $c_o k + k \log D$.*

Proof. S is contained within the union of $\binom{D}{k} \leq D^k$ subspaces of dimension k: pick which k coordinates, out of D, will be nonzero, and consider the subspace in which the remaining coordinates are forced to zero. It is well known (see [Cla05]) that each of these subspaces has Assouad dimension at most $c_o k$. Lemma 15 then bounds the increase in dimension from taking the union of the subspaces. □

2.2 Intrinsic dimension of real-world data

As mentioned earlier, most of the theoretical guarantees in this dissertation will be given in terms of intrinsic dimension, most often Assouad dimension. We therefore need to make sure that these sorts of notions developed in the last section actually capture the complexity of real-world data. Unfortunately, quantities such as covering and doubling dimension are hard to estimate on data. This is because they require estimating the *smallest* r-covers for given r's which itself is a difficult problem. The size of the minimum r-cover of a set A is (by definition) bounded by the size of an r-net of this set which is easily obtained by farthest-first traversal (see Chapter 6). However this can yield a poor estimate as it is just a coarse upper-bound.

Given the difficulty in estimating covering and Assouad dimensions of real-world data, we instead relate these notions to other verifiable properties of data such as *local flatness*. In the previous section we have related the Assouad dimension to that of manifolds and other geometrical structures which are locally flat. Here we present a formalism for local flatness (called *covariance dimension*) which is verifiable real-world datasets. The following notion of local flatness is borrowed from similar definitions in [DF08a].

Definition 17. *Let μ be any measure over $\mathbb{R}^D$ and let S be its covariance matrix. We say that μ has covariance dimension (d, ϵ) if the largest d eigenvalues of S account for $(1-\epsilon)$ fraction of its trace. That is, if the eigenvalues of S are $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D$, then*

$$\lambda_1 + \cdots + \lambda_d \geq (1-\epsilon)(\lambda_1 + \cdots + \lambda_D).$$

A distribution has covariance dimension (d, ϵ) if all but an ϵ fraction of its variance is concentrated in a d-dimensional affine subspace. Equivalently, the projection of the distribution onto this subspace leads to at most an ϵ total loss in squared distances. It is, in general, too much to hope that an entire data distribution would have low covariance dimension. But we might hope that this property holds *locally*; or more precisely, that all (or most) sufficiently-small neighborhoods have low covariance dimension. At this stage, we could make this definition more complicated by quantifying the "most" or "sufficiently small" (as [DF08b] did to some extent), but it will turn out that we don't need to do this in order to state the results in this dissertation, so we leave things as they are.

Intuitively, the local covariance condition lies somewhere between manifold dimension and Assouad dimension, although it is more general in that it merely requires points to be close to a locally flat set, rather than exactly on it.

Covariance dimension is an intuitive notion, and recalls standard constructs in statistics such as mixtures of factor analyzers. It is instructive to see how it might be estimated from samples, and whether there is evidence that many data sets do exhibit local flatness as formalized by the covariance dimension.

First let's set our expectations properly. Even if data truly lies near a low-dimensional manifold structure, this property would only be apparent at a certain *scale*, that is, when considering neighborhoods whose radii lie within an appropriate range. For larger neighborhoods, the data set might seem slightly higher dimensional: the union of a slew of local low-dimensional subspaces. And for smaller neighborhoods, all we would see is pure noise, and the data set would seem full-dimensional.

Thus we will empirically estimate covariance dimension at different resolutions. First, we determine the diameter Δ of the dataset $\mathbf{X}$ by computing the maximum interpoint distance, and we choose multiple values $r \in [0, \Delta]$ as our different scales (radii). For each such radius r, and each data point point $x \in \mathbf{X}$, we compute the covariance matrix of the data points lying in the ball $B(x, r)$, and we determine (using a standard eigenvalue computation) how many dimensions suffice for capturing a $(1 - \epsilon)$ fraction of the variance. In our experiments, we try $\epsilon = 0.1$ and 0.01. We then take the dimension at scale r (call it $d(r)$) to be average of all these values (over x).

How can we ascertain that our estimate $d(r)$ is indicative of the underlying covariance dimension at resolution r? If the balls $B(x, r)$ are so small as to contain very few data points, then the estimate $d(r)$ is not reliable. Thus we also keep track of $n(r)$, the average number of data points within the balls $B(x, r)$ (averaged over x). Roughly, we can expect $d(r)$ to be a reliable estimate if $n(r)$ is an order of magnitude larger than $d(r)$.

Results

Figure 2.3 plots $d(r)$ against r for several data sets. The numerical annotations on each curve represent the values $n(r)$. Loosely speaking, the larger the ratio $n(r)/d(r)$, the higher our confidence in the estimate.

Noisy swiss-roll: This is a noisy version of the ever-popular "swiss roll" (depicted below). In small neighborhoods, it is noise that dominates, and thus the

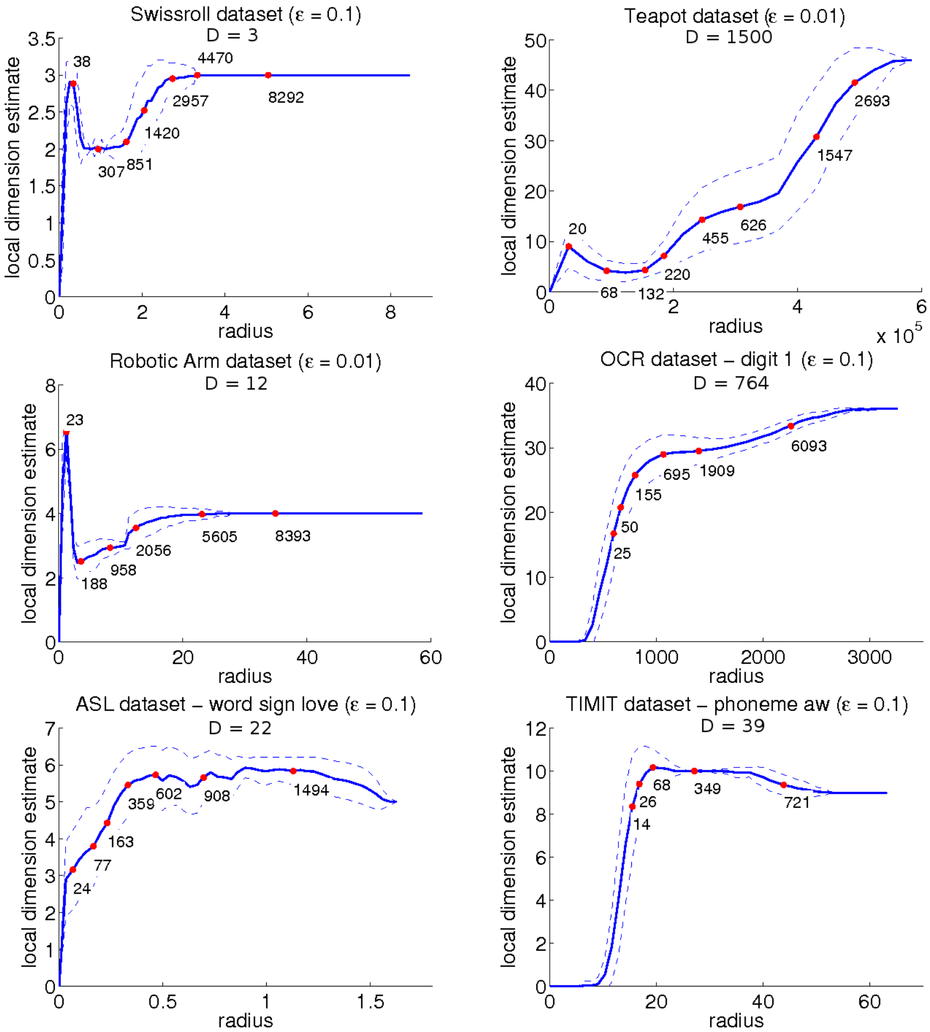
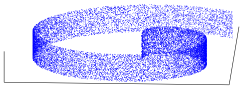
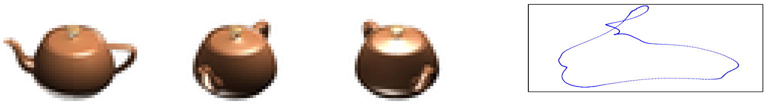


Figure 2.3: Local Covariance Dimension Estimates for Various Datasets. We fix ϵ and we report the average intrinsic dimension estimates for balls of varying radii centered at the data points. The bold line shows the dimension estimate as a function of radius, with dashed lines giving standard deviations over the different balls for each radius. The numeric annotations are average numbers of datapoints falling in balls of the specified radius.

data appear full-dimensional. In larger neighborhoods, the two-dimensional structure emerges: notice that the neighborhoods have very large numbers of points, so that we can feel very confident about the estimate of the local covariances. In even larger neighborhoods, we capture a significant chunk of the swiss roll and again revert to three dimensions.



Rotating teapot: This consists of images of a rotating teapot, each 30×50 pixels in size. Thus the ambient dimension D is 1500, although the points lie close to a one-dimensional manifold. Some of the images are shown below along with a 2-dimensional PCA of the dataset which clearly shows the 1-dimensional manifold structure. The intrinsic dimension experiments clearly identifies a low-dimensional structure at a small scale, although in the figure, the $d(r)$ values seem to be 3 or 4 rather than 1, but in any case much lower than D.



Robotic arm: The data consists of noisy measurements from 12 sensors placed on a robotic arm with two joints. Thus the ambient dimension is 12, but there are only two underlying degrees of freedom. The estimate $d(r)$ is on average close to 2.

OCR: This is the MNIST OCR dataset of handwritten digits, where we just pick the subset of handwritten "1". Each datapoint is a 28×28 pixels image and so the ambient dimension D is 784. However, the intrinsic dimensionality according to the estimates is between 20 and 30.

ASL: This is the Australian Sign Language time-series dataset from UCI Machine Learning Repository [Kad02] where we just pick out the subset for the "love" sign. The time series in this dataset are obtained from sensors on a glove. After some processing, the ambient dimension is 22, however we estimate low intrinsic dimension of between 3 and 5.

TIMIT: This is the popular TIMIT dataset of speech frames, which contains recordings of 600 speakers reading text; each recording is then broken up into 25 milliseconds windows each corresponding to a spoken phoneme. Standard Mel-frequency cepstal coefficients (MFCC) are then computed on each window to obtain a 39 dimensional vector. Here we pick the subset of "aw" phoneme. Although the ambient dimension is 39, we estimate the intrinsic dimension at around 10.

Portions of this chapter appear in:

– N. Verma, S. Kpotufe, S. Dasgupta, "Which spatial partition trees are adaptive to intrinsic dimension?", *Uncertainty in Artificial Intelligence*, 2009.

Chapter 3

Traditional solutions to the curse of dimension

Traditional solutions to the curse of dimension involve either preprocessing the data by reducing its dimension, or operating an adaptive regressor such as kernel methods in the original space. We investigate these two approaches in the following sections.

3.1 Dimensionality reduction

A simple approach to the curse of dimension is to reduce dimension by remapping the data space $\mathcal{X}$ to a lower-dimensional space $\mathcal{Z}$ and then perform regression in this final space (Figure 3.1). One of the earliest forms of dimen-

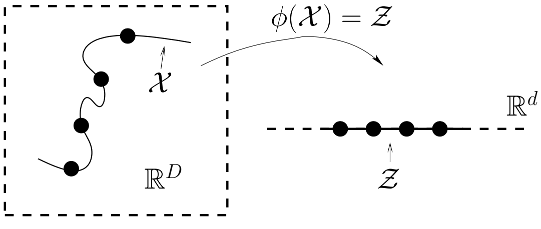


Figure 3.1: Embedding: $\mathcal{X} \subset \mathbb{R}^D$ gets remapped to $\mathcal{Z} \subset \mathbb{R}^d$, where we assume $d \ll D$. Regression can then be performed in the space $\mathcal{Z}$ provided the entire space $\mathcal{Z}$ (as opposed to just the training data) gets remapped.

sionality reduction technique is PCA [DGL96a] which consists of remapping the data $\mathcal{X} \subset \mathbb{R}^D$ to a subset $\mathcal{Z}$ of the subspace spanned by its top d principal components. This preserves most interpoint distances if the data is close to linear, i.e. the variance of the data in the subspace of $\mathbb{R}^D$ orthogonal to $\mathcal{Z}$ is negligible. Unfortunately, this cannot be expected in general, and although data might be intrinsically low-dimensional, it might lie near a non-linear manifold. This realization has motivated a wide body of work termed *manifold learning* which seeks to transform data from $\mathbb{R}^D$ to a lower-dimensional space while preserving important structure; key early results are [RS00, TSL00, BN03]. The type of structure being preserved varies with the procedures and it is not clear a priori whether such structures are the relevant information to be preserved for regression. Next we look at a few early examples to give the reader an idea of structures that manifold learning algorithms aim to preserve.

The Local Linear Embedding (LLE) method of Roweis and Saul [RS00] seeks to preserve local linearity, i.e.. if each point in a small set of nearby points can be expressed as a linear combination of the others (this would happen in small regions of a manifold) then they should be mapped to a set of points that can similarly be expressed as a linear combination of each other using the same weights. The Isomap method of TenenBaum et. al [TSL00] seeks to preserve geodesic distances, i.e. the distance between points in the lower-dimensional space should correspond to the geodesic distances of their image on the manifold. The Laplacian Eigenmaps of Belkin and Niyogi aims to preserve local information, i.e. nearby points should be mapped to nearby points. This is probably most directly related to the types of information useful in nonparametric classification and regression, as many learning methods are based on the assumption that nearby X points have similar Y values (e.g. the Lipschitz assumptions on the regression function f discussed earlier in the introduction).

The main problem however in using these methods as a preprocessing step to regression is that they typically only map the training data to a new space and don't automatically provide a map for the entire space $\mathcal{X}$. In other words, when we get a new query point x, we do not know where to map it in the new low-dimensional space so we cannot perform regression in this new space. This problem has spurred the development of new methods called *out-of-sample extensions* algorithms (e.g. [BPV$^+$03]) which, given a new sample X, seek to predict its embedding Z from pairs $\{(X_i, Z_i)\}_{i=1}^n$ of initial samples and their embeddings by a manifold learning procedure. Unfortunately, this prediction task itself operates on $\mathcal{X} \subset \mathbb{R}^D$ and is thus subject to the curse of dimension.

Perhaps for the above reasons, there exist no result in our knowledge that guarantees good regression risks for a procedure consisting of manifold learning followed by regression in the low-dimensional space. In the next section we investigate some simple distance preserving conditions under which a dimensionality reduction technique will yield good guarantees while used as a preprocessing step to regression.

3.1.1 Guaranteed regression rates after dimensionality reduction

Consider a dimensionality reduction technique $\phi : \mathcal{X} \mapsto \mathbb{R}^d$ where $\mathcal{X} \subset \mathbb{R}^D$, and $d \leq D$. Let $\mathcal{Z}$ be the image of $\mathcal{X}$ under ϕ. Assume that ϕ preserves distance, i.e. there exists $0 < \epsilon < 1$ such that $\forall x, x' \in \mathcal{X}$

$$(1 - \epsilon) \|x - x'\| \leq \|\phi(x) - \phi(x')\| \leq (1 + \epsilon) \|x - x'\| . \tag{3.1}$$

Notice that the above imply that ϕ is a bijection from $\mathcal{X}$ to $\mathcal{Z}$. If a dimensionality reduction method preserves distance in this way, then it can be used to preprocess data for regression and yields guaranteed regression rates that depend just on the dimension d of the embedding space $\mathcal{Z}$. This follows easily as explained below.

Suppose the regression function f satisfy a Lipschitz condition of the form

$$\forall x, x' \in \mathcal{X}, \|f(x) - f(x')\| \leq \lambda \|x - x'\| .$$

Now let

$$g(z) \doteq \mathbb{E}[Y|z] = \mathbb{E}[f(\phi^{-1}(z)) + \eta|z] = f(\phi^{-1}(z)),$$

be the regression function over the new space $\mathcal{Z}$. Then for all $z, z' \in \mathcal{Z}$, g satisfies the Lipschitz condition

$$\begin{aligned}\|g(z) - g(z')\| =& \left\|f(\phi^{-1}(z)) - f(\phi^{-1}(z'))\right\| \\ \le& \lambda \left\|\phi^{-1}(z) - \phi^{-1}(z'))\right\| \\ \le& \frac{\lambda}{1-\epsilon} \|z - z'\| .\end{aligned}$$

Now let g_n be a regressor in $\mathbb{R}^d$ such that if the regression function g is λ'-Lipschitz, g_n satisfies

$$\mathbb{E}\|g_n - g\|^2 \le C \cdot \Delta_{\mathcal{Z}}^{2d/(2+d)} \left(\frac{\Delta_{\mathcal{Y}}^2}{\lambda'^2 n}\right)^{2/(2+d)},$$

where $\Delta_{\mathcal{Z}}$ is the diameter of $\mathcal{Z}$. Note that most common regressors such as tree-based, kernel, k-NN regressors satisfy this property[1] (see e.g. [GKKW02]). Let $f_n(x) \doteq g_n(\phi(x))$ be the regressor obtained by learning g_n on a transformed sample $\{(Z_i, Y_i)\}_{i=1}^n = \{(\phi(X_i), Y_i)\}_{i=1}^n$. We automatically have:

$$\begin{aligned}\mathbb{E}\|f_n - f\|^2 =& \mathop{\mathbb{E}}_{\mathbf{X},\mathbf{Y},X} \|f_n(X) - f(X)\|^2 \\ =& \mathop{\mathbb{E}}_{\mathbf{X},\mathbf{Y},X} \|g_n(\phi(X)) - g(\phi(X))\|^2 \\ =& \mathbb{E}\|g_n - g\|^2 \\ \le& C \cdot ((1+\epsilon)\Delta_{\mathcal{X}})^{2d/(2+d)} \left(\frac{\Delta_{\mathcal{Y}}^2}{(\lambda/(1-\epsilon))^2 n}\right)^{2/(2+d)} .\end{aligned}$$

The obvious question now is whether there exists dimensionality reduction methods that preserve distance as in (3.1) for the entire data space $\mathcal{X}$ (as opposed to just a sample from $\mathcal{X}$). The only results we know with this sort of guarantee are those of [BW07], and [Cla07] where ϕ is a random projection operator. It is shown in these results that if $\mathcal{X}$ is a d-dimensional manifold, then a random projection to $\mathbb{R}^k$ preserves distances with distortion $(1 \pm \epsilon)$ distortion as in (3.1), provided $k = O((d/\epsilon^2)\log(V\tau^{-1}\epsilon^{-1}))$ where V is the volume of the manifold and τ is a quantity (called the condition number) that describes the curvature of the

[1] These are so-called oracle bounds that hold under the optimal parameter setting of the algorithm, e.g. the optimal bandwidth setting for a kernel regressor, which unfortunately is unknown. However, a good parameter setting can be found in practice through cross-validation and the resulting risk bound is only slightly worse than the optimal (see e.g. [LW07]).

manifold. As previously mentioned in Chapter 2 this sort of result does not hold for general data sets of low intrinsic dimension (e.g. Assouad dimension) as it requires a fair amount of geometric regularity.

3.2 Known adaptive regressors

In this section we discuss kernel regression and k-NN regression. As mentioned in the introduction, kernel regression is known to be adaptive [BL06] to intrinsic dimension, while there is strong evidence that k-NN might also be. Unfortunately, both these regression approaches can be expensive in practice since, for any new prediction, they require time consuming searches through the training data.

The efficient adaptive methods that are the main subjects of this dissertation all guarantee an $O(\log n)$ time complexity, where the O notation might hide constants that depend on the intrinsic dimension of the data.

3.2.1 Kernel regression

A kernel regression estimate $f_n(x)$ is obtained by averaging the Y values of sample points according to how close they are to the query x. That is, it takes the form $f_n(x) = \sum_{i=1}^{n} w_i(x) Y_i$ where the weight $w_i(x)$ is defined via a *kernel* function. The analysis in this section is restricted to the following types of kernels:

Definition 18 (Admissible kernels)**.** *The kernel* $K : [0, \infty) \mapsto [0, \infty)$ *is a nonincreasing function that is highest and positive at* 0*, and is* 0 *on* $[1, \infty)$*.*

Examples of such kernels are the *naive* kernel $K(u) = \mathbb{1}[u < 1]$, the *triangle* kernel $K(u) = (1 - u)_+$, and the popular *Epanechnikov* kernel $K(u) = \frac{3}{4}(1 - u^2)_+$. This definition excludes the Gaussian kernel $K(u) = e^{-u^2}$ but the analysis in this section still gives us insight about regression with this kernel since its approximately satisfies the above conditions: it is non increasing and its value is approximately 0 for u sufficiently large.

Fix a kernel K, and a *bandwidth* parameter $h > 0$. The kernel weights are defined as

$$w_i(x) = \frac{K(\|x - X_i\|/h)}{\sum_{j=1}^n K(\|x - X_j\|/h)},$$

provided the ball $B(x, h)$ contains sample points from $\mathbf{X} = \{X_i\}_{i=1}^n$ (so that the weights in the denominator are not all 0), otherwise w_i is undefined and we just set the estimate $f_n(x)$ to some value in the range of Y, say $\bar{Y}$ the average Y value in the dataset.

In the naive implementation of kernel regression, evaluation takes time $\Omega(n)$ since weights have to be computed for all points. However there exist many heuristics which generally combine fast proximity search procedures with other elaborate methods for approximating the kernel weights (see e.g. [LG08, AMS97]). These heuristics do not guarantee a better time complexity since this depends on the distribution of training points around the query point. For example, assuming that the regression function is Lipschitz, the optimal bandwidth for kernel regression is of the form $h \approx n^{-1/(2+d)}$ (see Theorem 21 below), and we would expect $O(nh^d) \approx n^{2/(2+d)}$ points in the ball $B(x, h)$. In other words a time complexity better than a root of n is impossible.

The adaptivity of kernel regression to intrinsic dimension was shown recently by Bikel and Li in [BL06] where they establish the asymptotic pointwise risk[2] of such a regressor in terms of manifold dimension. In this section we present a simple finite sample analysis of the integrated excess risk and show that kernel regression is adaptive to Assouad dimension, which as argued in Chapter 2 is a more general notion of intrinsic dimension. We will later rely on insights from this analysis to derive and understand *tree-kernel hybrids* procedures in Chapter 6, an important subject of this dissertation.

We'll proceed by bounding the bias and variance separately in the following two lemmas, and then combining these bounds in Theorem 21. We note that, to simplify notation we will often use the shorthand $K(x, x', h)$ to denote $K(\|x - x'\|/h)$.

Lemma 19 (Variance at x). *Fix* $\mathbf{X}$*, and let* $0 < h < \Delta_{\mathcal{X}}^2$*. Consider* $x \in \mathcal{X}$ *such*

[2] This commonly refers to the pointwise risk (at a point x) for large data sizes n.

that $\mathbf{X} \cap B(x, h/2) \neq \emptyset$. *We have*

$$\mathop{\mathbb{E}}_{\mathbf{Y}|\mathbf{X}} \left\| f_n(x) - \widetilde{f}_n(x) \right\|^2 \leq \frac{2K(0)\sigma_Y^2}{K(1/2) \cdot n\mu_n\left(B(x, h/2)\right)}.$$

Proof. It is easily verified that, for independent random vectors v_i with expectation $\mathbf{0}$, $\mathbb{E}\left\|\sum_i v_i\right\|^2 = \sum_i \mathbb{E}\left\|v_i\right\|^2$. We apply this fact twice in the inequalities below, given that, conditioned on $\mathbf{X}$ and $\mathbf{Q} \subset \mathbf{X}$, the Y_i values are mutually independent. We have

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathbf{Y}|\mathbf{X}} \left\| f_n(x) - \widetilde{f}_n(x) \right\|^2 &= \mathop{\mathbb{E}}_{\mathbf{Y}|\mathbf{X}} \left\| \sum_{i \in [n]} w_i(x) \left(Y - \mathop{\mathbb{E}}_{\mathbf{Y}|\mathbf{X}} Y \right) \right\|^2 \\
&\leq \sum_{i \in [n]} w_i^2(x) \mathop{\mathbb{E}}_{\mathbf{Y}|\mathbf{X}} \left\| Y - \mathop{\mathbb{E}}_{\mathbf{Y}|\mathbf{X}} Y \right\|^2 = \sum_{i \in [n]} w_i^2(x) \sigma_Y^2 \\
&\leq \left(\max_{i \in [n]} w_i(x) \sigma_Y^2 \right) \sum_{i \in [n]} w_i = \max_{i \in [n]} w_i(x) \sigma_Y^2 \\
&= \max_{i \in [n]} \frac{K(x, x_i, h)\sigma_Y^2}{\sum_{j \in [n]} K(x, x_j, h)} \\
&\leq \frac{K(0)\sigma_Y^2}{\sum_{j \in [n]} K(x, x_j, h)}.
\end{aligned} \tag{3.2}$$

To bound the fraction in (3.2), we lower-bound the denominator as:

$$\begin{aligned}
\sum_{i \in [n]} K(x, x_i, h) &\geq \sum_{x_i \in B(x, h/2) \cap \mathbf{X}} K(x, x_i, h) \\
&\geq \sum_{x_i \in B(x, h/2) \cap \mathbf{X}} K\left(1/2\right) = K\left(1/2\right) \cdot n\mu_n(B(x, h/2)).
\end{aligned}$$

Plug this last inequality into (3.2) and conclude. □

Lemma 20 (Bias at x)**.** *Fix* $\mathbf{X}$, *and let* $0 < h < \Delta_{\mathcal{X}}^2$. *Consider* $x \in \mathcal{X}$ *such that* $\mathbf{X} \cap B(x, h) \neq \emptyset$. *We have*

$$\left\| \widetilde{f}_n(x) - f(x) \right\|^2 \leq \lambda^2 h^2.$$

Proof. We have

$$\begin{aligned}\left\|\widetilde{f}_n(x) - f(x)\right\|^2 &= \left\|\sum_{i\in[n]} w_i(x)\left(f(X_i) - f(x)\right)\right\|^2 \\ &\leq \sum_{i\in[n]} w_i(x)\left\|f(X_i) - f(x)\right\|^2 \\ &\leq \sum_{i\in[n]} w_i(x)\lambda^2 h^2 = \lambda^2 h^2\end{aligned}$$

where the first inequality is obtained from a Jensen's inequality on the norm square.

□

Theorem 21. *Assume the data space $\mathcal{X}$ has Assouad dimension d. There exists $C > 0$ depending on d and $K(0)/K(1/2)$, such that for any $0 < h < \Delta_{\mathcal{X}}^2$, the kernel regressor $f_n = f_n(h)$ satisfies*

$$\mathop{\mathbb{E}}_{(\mathbf{X},\mathbf{Y})} \left\|f_n - f\right\|^2 \leq C\frac{\Delta_{\mathcal{Y}}^2 \left(\Delta_{\mathcal{X}}/h\right)^d}{n} + \lambda^2 h^2.$$

Thus for $h = C'\Delta_{\mathcal{X}}^{d/(2+d)}\left(\frac{\Delta_{\mathcal{Y}}^2}{\lambda^2 n}\right)^{1/(2+d)}$ we have

$$\mathop{\mathbb{E}}_{(\mathbf{X},\mathbf{Y})} \left\|f_n - f\right\|^2 \leq C''\Delta_{\mathcal{X}}^{2d/(2+d)}\left(\frac{\Delta_{\mathcal{Y}}^2}{\lambda^2 n}\right)^{2/(2+d)}.$$

Proof. Applying Fubini's theorem, the expected excess risk, $\mathbb{E}_{(\mathbf{X},\mathbf{Y})}\left\|f_n - f\right\|^2$, can be written as

$$\mathop{\mathbb{E}}_{X}\mathop{\mathbb{E}}_{(\mathbf{X},\mathbf{Y})} \left\|f_n(X) - f(X)\right\|^2\left(\mathbb{1}\left[\mu_n(B(X,h/2)) > 0\right] + \mathbb{1}\left[\mu_n(B(X,h/2)) = 0\right]\right).$$

By lemmas 19 and 20, and (1.2), we have for $X = x$ fixed,

$$\begin{aligned}&\mathop{\mathbb{E}}_{(\mathbf{X},\mathbf{Y})} \left\|f_n(x) - f(x)\right\|^2 \mathbb{1}\left[\mu_n(B(x,h/2)) > 0\right] \\ &\quad\leq C_1 \mathop{\mathbb{E}}_{\mathbf{X}}\left[\frac{\sigma_Y^2\mathbb{1}\left[\mu_n(B(x,h/2)) > 0\right]}{n\mu_n(B(x,h/2))}\right] + \lambda^2 h^2 \\ &\quad\leq C_1\left(\frac{2\sigma_Y^2}{n\mu(B(x,h/2))}\right) + \lambda^2 h^2 \qquad (3.3)\end{aligned}$$

where for the last inequality we used the fact that (see lemma 4.1 of [GKKW02]) for a binomial $b(n,p)$,

$$\mathbb{E}\left[\frac{\mathbb{1}\left[b(n,p)>0\right]}{b(n,p)}\right] \leq \frac{2}{np}.$$

For the case where $B(x,h/2)$ is empty, we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(\mathbf{X},\mathbf{Y})} \|f_n(x)-f(x)\|^2 \mathbb{1}\left[\mu_n(B(x,h/4))=0\right] \\
&\quad\leq \Delta_{\mathcal{Y}}^2 \mathop{\mathbb{E}}_{\mathbf{X}} \mathbb{1}\left[\mu_n(B(x,h/2))=0\right] = \Delta_{\mathcal{Y}}^2 \left(1-\mu(B(x,h/2)\right)^n \\
&\quad\leq \Delta_{\mathcal{Y}}^2 e^{-n\mu(B(x,h/2))} \leq \frac{\Delta_{\mathcal{Y}}^2}{n\mu(B(x,h/2))}.
\end{aligned} \tag{3.4}$$

Combining (3.4) and (3.3), we can then bound the expected excess risk as

$$\mathop{\mathbb{E}}_{(\mathbf{X},\mathbf{Y})} \|f_n - f\|^2 \leq \frac{(C_1+1)\Delta_{\mathcal{Y}}^2}{n} \mathop{\mathbb{E}}_{X}\left[\frac{1}{\mu(B(X,h/2))}\right] + \lambda^2 h^2. \tag{3.5}$$

The expectation on the r.h.s is bounded using a standard covering argument (see e.g. [GKKW02]). Let $\{z_i\}_1^N$ be an $\frac{h}{4}$-cover of $\mathcal{X}$. Notice that for any z_i, $x \in B(z_i,h/4)$ implies $B(x,h/2) \supset B(z_i,h/4)$. We therefore have

$$\begin{aligned}
\mathop{\mathbb{E}}_{X}\left[\frac{1}{\mu(B(X,h/2))}\right] &\leq \sum_{i=1}^{N} \mathop{\mathbb{E}}_{X}\left[\frac{\mathbb{1}\left[X\in B(z_i,h/4)\right]}{\mu(B(X,h/2))}\right] \\
&\leq \sum_{i=1}^{N} \mathop{\mathbb{E}}_{X}\left[\frac{\mathbb{1}\left[X\in B(z_i,h/4)\right]}{\mu(B(z_i,h/4))}\right] \\
&= N \leq C_2\left(\frac{\Delta_{\mathcal{X}}}{h}\right)^d, \text{ where } C_2 \text{ depends just on } d.
\end{aligned}$$

□

In the above theorem, the optimal bandwidth is expressed in terms of quantities such as d that we do not generally know. How to properly set the bandwidth is the subject of many works (e.g. [Sta89, CA91, DvdL05]) and it is not clear whether we can always get close to the optimal bandwidth. However, in practice one just employs cross-validation over a range of bandwidth settings. If this range contains a setting that is close to the optimal then simple concentration bounds tell us that the resulting regressor will have an excess risk in the same order as that in the theorem. For example, disregarding other terms, we can insure that the range contains bandwidths of the form $n^{-1/(2+d)}$ simply by trying all possible values of d (see e.g. [LW07]). The resulting risk will be optimal in its dependence on the dimension although it can worsen in terms of the other variables such as λ.

3.2.2 k-NN regression

A k-NN regression estimate $f_n(x)$ is obtained by averaging the Y values of the k nearest neighbors of the query x in the sample $\mathbf{X}$. Let $X_{(i)}(x)$ and $Y_{(i)}(x)$ denote the i'th nearest neighbor and its corresponding Y value. We have

$$f_n(x) = \frac{1}{k}\sum_{i=1}^{k} Y_{(i)}(x).$$

Evaluation consists of finding all the k nearest neighbors and so it takes time at least $\Omega(k)$. Unfortunately, to insure good regression rates, k has to be chosen large as a root of n [GKKW02], and this can be expensive considering that nonparametric regression in general requires large data sizes n.

We now turn to the question of adaptivity. For the case $k = 1$, Kulkarni and Posner [KP95] show that $\mathbb{E}\left\|f_n - f\right\|^2$ converges to $R(f)$ at a rate that depend just on the covering dimension of $\mathcal{X}$. Unfortunately 1-NN is not consistent as for consistency we need k to grow with n (see [GKKW02]).

The adaptivity of k-NN to intrinsic dimension is an open problem. However we list this method in this Section on adaptive procedures as there are reasons to believe that they are indeed adaptive (at least in their asymptotic rates). One simple such reason is that locally at a point x, they behave like a kernel regressor with a naive kernel, using a bandwidth h corresponding to the radius of the smallest ball around x of mass k/n. We hope to formalize this intuition and close the question of adaptivity of k-NN regression to intrinsic dimension.

Chapter 4

Tree-based regressors and data diameters

A tree-based regression scheme takes as input a data set of n pairs (X, Y), with $X \in \mathbb{R}^D$, and then works (typically) in two phases.

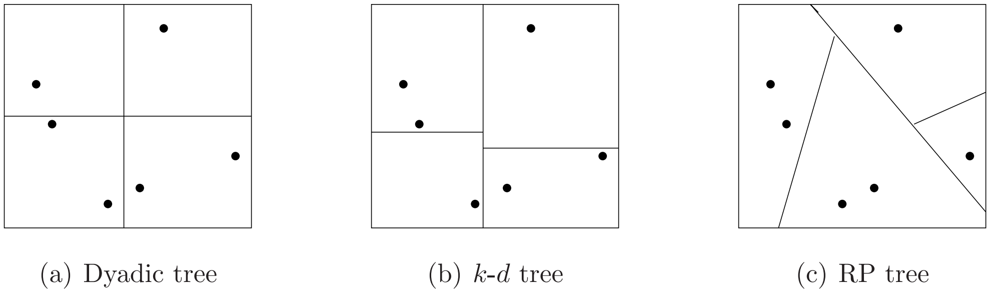
1. It builds a tree T each of whose nodes corresponds to a *cell* (region) of $\mathbb{R}^D$.

   The root node is all of $\mathbb{R}^D$; and each internal node's cell is the disjoint union of the cells of its two children.

2. It prunes the trees to some T', and fits a simple (constant, or at rate continuous) function to the data in each leaf of T'.

   The cells corresponding to the leaves of T' are a partition of $\mathbb{R}^D$, and the collection of these local estimates, one per cell, form a piecewise continuous function f_n.

An attractive property of this estimator is that $f_n(x)$ can be evaluated by simply navigating down to the leaf containing x, which takes time proportional to the height of the tree, often just $O(\log n)$. This computational efficiency, and an overall ease of use, have motivated a variety of tree partition methods (Figure 4.1) such as CART, dyadic trees, and *k*-*d* trees [GN05, SN06a, DGL96a], but none of these has been shown to adapt to intrinsic dimension in its regression risk.



(a) Dyadic tree

(b) *k*-*d* tree

(c) RP tree

Figure 4.1: Spatial partitioning induced by various splitting rules. Two levels of the tree are shown for each. For the dyadic tree, each region is split at the midpoint along a coordinate direction. The *k*-*d* tree splits at (or near) the median of the (projected) data along a coordinate direction. The RP tree split at (or near) the median of the (projected) data along a random direction.

In this chapter, we investigate the adaptivity of tree methods to intrinsic dimension. First we tie adaptivity to the speed at which the diameter of the data in each cell of a tree decreases as one goes from the root down to the leaves. We will see empirically that trees that decrease this data diameter quickly tend to yield better regressors. We then investigate the rate at which various trees decrease data diameter as a function of the intrinsic diameter of the data space. Later in Chapter 5 we show a formal link between the risk of a tree-based regressor and the rate at which it decrease the diameter of data within it cells.

4.1 Spatial partition trees

Spatial partition trees conform to a simple template:

```
Procedure PartitionTree(dataset $A \subset \mathcal{X}$)
    if $|A| \leq MinSize$ then
        return leaf
    end
    else
        $(A_{\text{left}}, A_{\text{right}}) \leftarrow SplitAccordingToSomeRule(A)$
        $LeftTree \leftarrow PartitionTree(A_{\text{left}})$
        $RightTree \leftarrow PartitionTree(A_{\text{right}})$
    end
    return $(LeftTree, RightTree)$
```

Different types of trees are distinguished by their splitting criteria. Here are some common varieties:

- **Dyadic tree:** Pick a coordinate direction and splits the data at the midpoint along that direction. One generally cycles through all the coordinates as one moves down the tree.
- ***k*-D tree:** Pick a coordinate direction and splits the data at the median along that direction. One often chooses the coordinate with largest spread.
- **Random Projection (RP) tree:** Split the data at the median along a random direction chosen from the surface of the unit sphere.
- **Principal Direction (PD or PCA) tree:** Split at the median along the principal eigenvector of the covariance matrix.
- **Two Means (2M) tree:** Pick the direction spanned by the centroids of the 2-means solution, and split the data as per the cluster assignment.

4.2 Bias-Variance tradeoff: standard intuition

Based on the training set, we will construct a partition $\mathcal{A}$ of $\mathcal{X}$ (or more precisely, of $\mathbb{R}^D$, since $\mathcal{X}$ is unknown), and we will estimate f, in a piecewise manner, as the average Y value in each cell of this partition (or some arbitrary value if no point falls in the cell). It is standard to decompose the error of the estimator into two parts.

$$\begin{aligned}\text{bias} &\equiv \text{in expectation, how well does the average } Y \text{ approximate } f?\\ \text{variance} &\equiv \text{how unstable is the average } Y \text{ within a cell?}\end{aligned}$$

Traditionally, the analysis of bias is based on the *physical diameters* of cells $A \in \mathcal{A}$,

$$\Delta(A) \doteq \max_{x,x' \in A} \|x - x'\|$$

(see, for instance, [GN05, SN06a, DGL96a]). Suppose $\Delta(A)$ is small, then if the query point x falls in A, we know x is close to all the data within A, and we can therefore expect that $f(x)$ should be close to the Y values of data within A. Thus, bias can be controlled by making sure cells have small physical diameter so that f itself does not vary much within a cell. However, such cells are typically found deep down the tree and likely contain very few points. In other words, cells A with small $\Delta(A)$ tend to yield high variance estimates. However cells A with a lot of data in them tend to be found higher up the tree and therefore have large $\Delta(A)$, implying large bias.

The most crucial aspect of tree-based regression is how to pick a good partition $\mathcal{A}$ in the tree that allows a good tradeoff between bias and variance. The standard intuition is therefore to pick a partition somewhere mid-level in the tree so that the cell diameters are sufficiently small and the cells still contain a reasonable number of points. This intuition works fine for a lot of common partition rules such as dyadic trees and k-d trees. However, for splitting rules such as RP tree, PD tree, the cells are irregular polytopes whose physical diameters are hard to assess, and worse, whose physical diameters might not decrease at all. A new approach is therefore needed, and is the subject of Section 4.3 below.

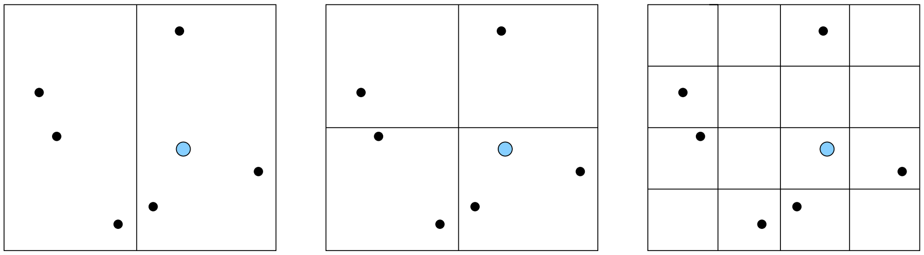


Figure 4.2: Bias-Variance tradeoff. The query point is shown in blue. The left partition yields a high bias estimator, while the right partition yields a high variance estimator, we therefore might settle for the tradeoff offered by the middle partition.

4.3 Bias-Variance tradeoff: new intuition

4.3.1 Data-diameter

The generalization behavior of a spatial partitioning has traditionally been analyzed in terms of the physical diameter of the individual cells (see, for instance, [DGL96b, SN06b]). But this kind of diameter is hard to analyze for general convex cells. Instead we consider more flexible notions that measure the diameter of *data within the cell*. We will later show that such measures are sufficient for giving generalization bounds for the regression risk.

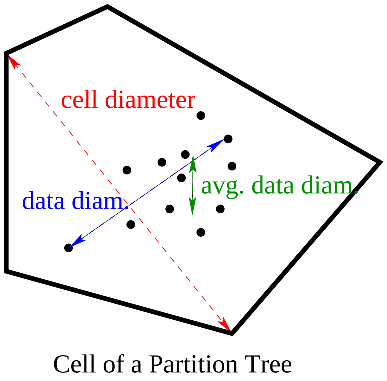


Figure 4.3: Various Notions of Diameter

For any cell A, we will use two types of data diameter: the maximum distance between data points in A, denoted $\Delta_n(A)$, and the average interpoint distance among data in A, denoted $\Delta_{n,a}(A)$ (Figure 4.3).

Let $\mathbf{X} = \{X_1, \ldots, X_n\}$ be a data set drawn from $\mathcal{X}$, and let μ_n be the empirical distribution that assigns equal weight to each of these points. Consider a collection $\mathcal{A}$ of disjoint subsets A of $\mathcal{X}$. For each such cell A, we can look at its maximum (data) diameter as well as its average (data) diameter; these are, respectively,

$$\Delta_n(A) \doteq \max_{x,x' \in A \cap \mathbf{X}} \|x - x'\|$$

$$\Delta_{n,a}(A) \doteq \frac{1}{(n\mu_n(A))} \left(\sum_{x,x' \in A \cap \mathbf{X}} \|x - x'\|^2 \right)^{1/2}$$

(for the latter it turns out to be a big convenience to use squared Euclidean distance.) We can also average these quantities all over cells $A \in \mathcal{A}$:

$$\Delta_n(\mathcal{A}) \doteq \left(\frac{\sum_{A \in \mathcal{A}} \mu_n(A) \Delta_n^2(A)}{\sum_{A \in \mathcal{A}} \mu_n(A)} \right)^{1/2}$$

$$\Delta_{n,a}(\mathcal{A}) \doteq \left(\frac{\sum_{A \in \mathcal{A}} \mu_n(A) \Delta_{n,a}^2(A)}{\sum_{A \in \mathcal{A}} \mu_n(A)} \right)^{1/2} .$$

As mentioned before, physical cell diameters may not decrease at all with some partitioning rules such as RP tree, but the data diameter is bound to decrease whenever the rule insures that the number of points per cell goes down at some steady rate.

4.3.2 Diameter-decrease rate and adaptivity to intrinsic dimension

In this work, we build upon the new intuition that, for good bias-variance tradeoff, we just need data diameter (under any of the above formalisms) to decrease quickly from the root down. This way, the tree would contain partitions which have cells containing many points (good variance) and whose data diameters are also small (which, as we will see, implies good bias). This intuition checks out against real-world data. In Figure 4.4 we see the general trend that the tree methods that decrease data diameter ($\Delta_n(\cdot)$, and $\Delta_{n,a}(\cdot)$) fastest, also produce the best regressors at all levels.

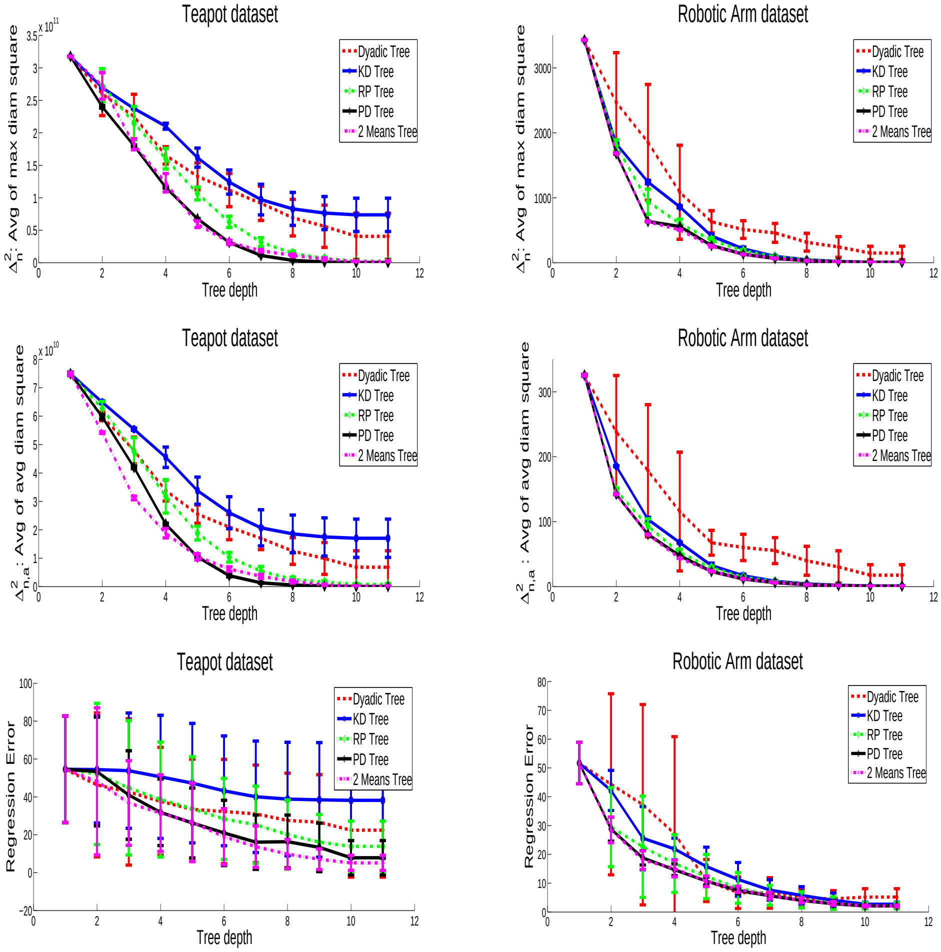


Figure 4.4: Data diameter decrease rates vs regression errors for various tree methods on two datasets. We report for each level of the tree, the average data diameter ($\Delta_n(\mathcal{A})$ and $\Delta_{n,a}(\mathcal{A})$) for the partition $\mathcal{A}$ defined by the cells at that level. The reported regression error (for the regressor defined over the same partition $\mathcal{A}$) is the l_2 risk evaluated on a test sample. The observed trend is that the tree methods that decrease data diameter fastest also attain better regression risk.

In the experiments of Figure 4.4 we implement the trees as follows: dyadic trees – fix a permutation and cycle through the coordinates, k-D trees – determine the spread over each coordinate by computing the coordinate vise diameter and picking the coordinate with maximum diameter, RP trees – pick the direction that results in the largest diameter decrease from a bag of 20 random directions, PD trees – pick the principal direction in accordance to the data falling in each node of the tree, 2M trees – solve 2-means via the Lloyd's method and pick the direction spanned by the centroids of the 2-means solution.

As it turns out, some tree methods decrease data diameter at a rate that depend on the intrinsic dimension of the data. In other words, if the data has low intrinsic dimension, some tree methods achieve better data diameter decrease rates, and hence, given the observed trend, attain better regression risks. In the next sections we establish the rate at which various tree methods decrease data diameter. We follow this by an analysis of the RP tree method (Chapter 5) where we formally link the diameter decrease rate to the regression risk of the tree estimator.

4.4 Data diameter decrease rates: low covariance dimension

4.4.1 Irregular splitting rules

This section considers the RPTree, PDtree, and 2Mtree splitting rules. The nonrectangular partitions created by these trees turn out to be adaptive to the local dimension of the data: the decrease in average diameter resulting from a given split depends just on the eigenspectrum of the data in the local neighborhood, irrespective of the ambient dimension. In all the lemmas below, the covariance dimension is defined under the empirical distribution μ_n.

For the analysis, we consider a slight variant of these trees, in which an alternative type of split is used whenever the data in the cell has outliers (here, points that are much farther away from the mean than the typical distance-from-mean).

```
Procedure split(region $A \subset \mathcal{X}$)
    if $\Delta_n^2(A) \geq c \cdot \Delta_{n,a}^2(A)$ then
        //SPLIT BY DISTANCE: remove outliers.
        $A_{\text{left}} \leftarrow \{x \in A, \|x - \nu_{n,A}\| \leq \text{median}\{\|z - \nu_{n,A}\| : z \in \mathbf{X} \cap A\}\}$;
    end
    else
        //SPLIT BY PROJECTION: no outliers.
        Choose a unit direction $v \in \mathbb{R}^D$ and a threshold $t \in \mathbb{R}$.
        $A_{\text{left}} \leftarrow \{x \in A,\ x \cdot v \leq t\}$;
    end
    $A_{\text{right}} \leftarrow A \setminus A_{\text{left}}$;
```

The *distance split* is common to all three rules, and serves to remove outliers. It is guaranteed to reduce maximum data diameter by a constant fraction:

Lemma 22 (Lemma 12 of [DF08b])**.** *Suppose $\Delta_n^2(A) > c \cdot \Delta_{n,a}^2(A)$, so that A is split by* distance *under any instantiation of procedure* `split`. *Let $\mathcal{A} = \{A_1, A_2\}$ be the resulting split. We have*

$$\Delta_n^2(\mathcal{A}) \leq \left(\frac{1}{2} + \frac{2}{c}\right) \Delta_n^2(A).$$

We consider the three instantiations of procedure `split` in the following three sections, and we bound the decrease in diameter after a single split in terms of the local spectrum of the data.

RPtree

For RPtree, the direction v is picked randomly, and the threshold t is the median of the projected data.

The diameter decrease after a split depends just on the parameter d of the local covariance dimension, for ϵ sufficiently small.

Lemma 23 (Theorem 4 of [DF08b])**.** *There exist constants $0 < c_1, c_2 < 1$ with the following property. Suppose $\Delta_n^2(A) \leq c \cdot \Delta_{n,a}^2(A)$, so that A is split by* projection *into $\mathcal{A} = \{A_1, A_2\}$ using the RPtree split. If $A \cap \mathbf{X}$ has covariance dimension*

(d, c_1), then

$$\mathbb{E}\left[\Delta_{n,a}^2(\mathcal{A})\right] < (1 - c_2/d)\Delta_{n,a}^2(A),$$

where the expectation is over the choice of direction.

PDtree

For PDtree, the direction v is chosen to be the principal eigenvector of the covariance matrix of the data, and the threshold t is the median of the projected data.

The diameter decrease after a split depends on the local spectrum of the data. Let A be the current cell being split, and suppose the covariance matrix of the data in A has eigenvalues $\lambda_1 \geq \cdots \geq \lambda_D$. If the covariance dimension of A is (d, ϵ), define

$$k \doteq \frac{1}{\lambda_1}\sum_{i=1}^{d} \lambda_i, \tag{4.1}$$

By definition, $k \leq d$.

The diameter decrease after the split depends on k^2, the worst case being when the data distribution in the cell has heavy tails (example omitted for want of space). In the absence of heavy tails (condition (4.2)), we obtain a faster diameter decrease rate that depends just on k. This condition holds for any logconcave distribution (such as a Gaussian or uniform distribution), for instance. The decrease rate of k could be much better than d in situations where the first eigenvalue is dominant; and thus in such situations PD trees could do a lot better than RP trees.

Lemma 24. *There exist constants $0 < c_1, c_2 < 1$ with the following property. Suppose $\Delta_n^2(A) \leq c \cdot \Delta_{n,a}^2(A)$, so that A is split by* projection *into $\mathcal{A} = \{A_1, A_2\}$ using the PDtree split. If $A \cap \mathbf{X}$ has covariance dimension (d, c_1), then*

$$\Delta_{n,a}^2(\mathcal{A}) < (1 - c_2/k^2)\Delta_{n,a}^2(A),$$

where k is as defined in (4.1).

If in addition the empirical distribution on $A \cap \mathbf{X}$ satisfies (for any $s \in \mathbb{R}$ and some $c_0 \geq 1$)

$$\mathbb{E}_A[(X \cdot v - s)^2] \leq c_0 \left(\mathbb{E}_A[X \cdot v - s]\right)^2 \tag{4.2}$$

we obtain a faster decrease where

$$\Delta^2_{n,a}(\mathcal{A}) < (1 - c_2/k)\Delta^2_{n,a}(A).$$

Proof. The argument is based on the following fact which holds for any bi-partiton $\mathcal{A} = \{A_1, A_2\}$ of A (see lemma 15 of [DF08b]):

$$\Delta^2_{n,a}(A) - \Delta^2_{n,a}(\mathcal{A}) = 2\mu(A_1) \cdot \mu(A_2) \left\| \nu_{n,A_1} - \nu_{n,A_2} \right\|^2.$$

We start with the first part of the statement with no assumption on the data distribution. Let $\widetilde{x} \in \mathbb{R}$ be the projection of $x \in A \cap \mathbf{X}$ to the principal direction. WLOG assume that the median on the principal direction is 0. Notice that

$$\begin{aligned}\left\| \nu_{n,A_1} - \nu_{n,A_2} \right\| &\geq \mathbb{E}\left[\widetilde{x} \,|\widetilde{x} > 0\right] - \mathbb{E}\left[\widetilde{x} \,|\widetilde{x} \leq 0\right] \\ &\geq \max\left\{\mathbb{E}\left[\widetilde{x} \,|\widetilde{x} > 0\right], -\mathbb{E}\left[\widetilde{x} \,|\widetilde{x} \leq 0\right]\right\}\end{aligned}$$

where the expectation is over x chosen uniformly at random from $A \cap \mathbf{X}$. The claim is therefore shown by bounding the r.h.s below by $O(\Delta_a(A)/k$ and applying equation (4.3).

We have $\mathbb{E}\left[\widetilde{x}^2\right] \geq \lambda_1$, so either $\mathbb{E}\left[\widetilde{x}^2|\widetilde{x} > 0\right]$ or $\mathbb{E}\left[\widetilde{x}^2|\widetilde{x} \leq 0\right]$ is greater than λ_1. Assume WLOG that it is the former. Let $\widetilde{m} = \max\{\widetilde{x} > 0\}$. We have that

$$\lambda_1 \leq \mathbb{E}\left[\widetilde{x}^2|\widetilde{x} > 0\right] \leq \mathbb{E}\left[\widetilde{x} \,|\widetilde{x} > 0\right]\widetilde{m},$$

and since $\widetilde{m}^2 \leq c\Delta^2_{n,a}(A)$, we get

$$\mathbb{E}\left[\widetilde{x} \,|\widetilde{x} > 0\right] \geq \frac{\lambda_1}{\Delta_a(A)\sqrt{c}}.$$

Now, by the assumption on covariance dimension,

$$\lambda_1 = \frac{\sum_{i=1}^d \lambda_i}{k} \geq (1 - c_1)\frac{\sum_{i=1}^D \lambda_i}{k} = (1 - c_1)\frac{\Delta^2_{n,a}(A)}{2k}.$$

We therefore have (for appropriate choice of c_1) that

$$\mathbb{E}\left[\widetilde{x} \,|\widetilde{x} > 0\right] \geq \Delta_a(A)/4k\sqrt{c},$$

which concludes the argument for the first part.

For the second part, assumption (4.2) yields

$$\begin{aligned}\mathbb{E}\left[\widetilde{x}\,|\widetilde{x}>0\right]-\mathbb{E}\left[\widetilde{x}\,|\widetilde{x}\leq 0\right] &= 2\mathbb{E}\left|\widetilde{x}\right|\geq 2\sqrt{\frac{\mathbb{E}\left|\widetilde{x}\right|^2}{c_0}}\\ &\geq 2\sqrt{\frac{\lambda_1}{c_0}} = 2\sqrt{\frac{\Delta^2_{n,a}\left(A\right)}{4c_0k}}.\end{aligned}$$

We finish up by appealing to equation (4.3). □

2Mtree

For 2Mtree, the direction $v = \nu_{n,A_1} - \nu_{n,A_2}$ where $A = \{A_1, A_2\}$ is the bisection of A that minimizes the 2-means cost. The threshold t is the half point between the two means.

The 2-means cost can be written as

$$\sum_{i\in[2]}\sum_{x\in A_i\cap\mathbf{X}}\left\|x-\nu_{n,A_i}\right\|^2 = \frac{n}{2}\Delta^2_{n,a}\left(\mathcal{A}\right).$$

Thus, the 2Mtree (assuming an exact solver) minimizes $\Delta^2_{n,a}\left(\mathcal{A}\right)$. In other words, it decreases diameter at least as fast as RPtree and PDtree. Note however that, since these are greedy procedures, the decrease in diameter over multiple levels may not be superior to the decrease attained with the other procedures.

Lemma 25. *Suppose* $\Delta^2_n\left(A\right) \leq c\cdot\Delta^2_{n,a}\left(A\right)$*, so that* A *is split by* projection *into* $\mathcal{A} = \{A_1, A_2\}$ *using the 2Mtree split. There exists constants* $0 < c_1, c_2 < 1$ *with the following property. Assume* $A \cap \mathbf{X}$ *has covariance dimension* (d, c_1)*. We then have*

$$\Delta^2_{n,a}\left(\mathcal{A}\right) < (1 - c_2/d')\Delta^2_{n,a}\left(A\right),$$

where $d' \leq \min\{d, k^2\}$ *for general distributions, and* d' *is at most* k *for distributions satisfying (4.2).*

Diameter Decrease Over Multiple Levels

The diameter decrease parameters d, k^2, k, d' in lemmas 23, 24, 25 above are a function of the covariance dimension of the data in the cell A being split. The covariance dimensions of the cells may vary over the course of the splits implying that the decrease rates may vary. However, we can bound the overall diameter decrease rate over multiple levels of the tree in terms of the worst case rate attained over levels.

Lemma 26 (Diameter decrease over multiple levels)**.** *Suppose a partition tree is built by calling* `split` *recursively (under any instantiation). Assume furthermore that every node $A \subset \mathcal{X}$ of the tree satisfies the following: let $\mathcal{A} = \{A_1, A_2\}$ represent the child nodes of A, we have for some constants $0 < c_1, c_2 < 1$ and $\kappa \leq D$ that*

(i) If A is split by distance, $\Delta_n^2(\mathcal{A}) < c_1 \Delta_n^2(A)$.

(ii) If A is split by projection, $\mathbb{E}\left[\Delta_{n,a}^2(\mathcal{A})\right] < (1 - c_2/\kappa)\Delta_{n,a}^2(A)$.

Then, there exists a constant C such that the following holds: let $\mathcal{A}_l$ be the partition of $\mathcal{X}$ defined by the nodes at level l, we have

$$\mathbb{E}\left[\Delta_{n,a}^2(\mathcal{A}_l)\right] \leq \mathbb{E}\left[\Delta_n^2(\mathcal{A}_l)\right] \leq \frac{1}{2^{\lfloor l/C\kappa \rfloor}} \Delta_n^2(\mathcal{X}).$$

In all the above, the expectation is over the randomness in the algorithm for **X** *fixed.*

Proof. Fix **X**. Consider the r.v. X drawn uniformly from **X**. Let the r.v.s $A_i = A_i(X), i = 0 \cdots l$ denote the cell to which X belongs at level i in the tree. Define $I(A_i) \doteq \mathbb{1}\left[\Delta_n^2(A_i) \leq c\Delta_{n,a}^2(A_i)\right]$.

Let $\mathcal{A}_l$ be the partition of $\mathcal{X}$ defined by the nodes at level l, we'll first show that $\mathbb{E}\left[\Delta_n^2(\mathcal{A}_l)\right] \leq \frac{1}{2}\Delta_n^2(\mathcal{X})$ for $l = C\kappa$ for some constant C. We point out that $\mathbb{E}\left[\Delta_n^2(\mathcal{A}_l)\right] = \mathbb{E}\left[\Delta_n^2(A_l)\right]$ where the last expectation is over the randomness in the algorithm and the choice of X.

To bound $\mathbb{E}\left[\Delta_n^2(A_l)\right]$, note that one of the following events must hold:

(a) $\exists\, 0 \leq i_1 < \cdots < i_m < l,\ m \geq \frac{l}{2}, I\left(A_{i_j}\right) = 0$

(b) $\exists\, 0 \leq i_1 < \cdots < i_m < l,\ m \geq \frac{l}{2}, I\left(A_{i_j}\right) = 1$

Let's first condition on event (a). We have

$$\mathbb{E}\left[\Delta_n^2\left(A_l\right)\right] \;\leq\; \mathbb{E}\left[\Delta_n^2\left(A_{i_m+1}\right)\right] = \mathbb{E}\left[\mathbb{E}\left[\Delta_n^2\left(A_{i_m+1}\right)|A_{i_m}\right]\right],$$

and since by the assumption, $\mathbb{E}\left[\Delta_n^2\left(A_{i_m+1}\right)|A_{i_m}\right] \leq c_1\Delta_n^2\left(A_{i_m}\right)$ we get that

$$\mathbb{E}\left[\Delta_n^2\left(A_l\right)\right] \leq c_1\mathbb{E}\left[\Delta_n^2\left(A_{i_m}\right)\right].$$

Applying the same argument recursively on i_j, $j = m, (m-1), \ldots, 1$, we obtain

$$\mathbb{E}\left[\Delta_n^2\left(A_l\right)\right] \leq c_1^m \cdot \mathbb{E}\left[\Delta_n^2\left(A_{i_1}\right)\right] \leq c_1^{l/2}\Delta_n^2\left(\mathcal{X}\right).$$

Now condition on event (b). Using the fact that $\mathbb{E}\left[\Delta_{n,a}^2\left(A_i\right)\right]$ is non-increasing in i (see [DF08b]), we can apply a similar recursive argument as above to obtain that $\mathbb{E}\left[\Delta_{n,a}^2\left(A_{i_m}\right)\right] \leq \left(1 - c_2/\kappa\right)^{m-1}\mathbb{E}\left[\Delta_{n,a}^2\left(A_{i_1}\right)\right]$. It follows that

$$\begin{aligned}\mathbb{E}\left[\Delta_n^2\left(A_l\right)\right] &\leq \mathbb{E}\left[\Delta_n^2\left(A_m\right)\right] \leq c\mathbb{E}\left[\Delta_{n,a}^2\left(A_m\right)\right]\\ &\leq c\left(1 - \frac{c_2}{\kappa}\right)^{l/2-1}\Delta_n^2\left(\mathcal{X}\right).\end{aligned}$$

Thus, in either case we have

$$\mathbb{E}\left[\Delta_n^2\left(A_l\right)\right] \leq \max\left\{c_1^{l/2},\, c\left(1 - c_2/\kappa\right)^{l/2-1}\right\} \cdot \Delta_n^2\left(\mathcal{X}\right)$$

and we can verify that there exists C such that the r.h.s above is at most $\frac{1}{2}\Delta_n^2\left(\mathcal{X}\right)$ for $l \leq C\kappa$. Thus, we can repeat the argument over every $C\kappa$ levels to obtain the statement of the lemma. □

So if every split decreases average diameter at a rate controlled by κ as defined above, then it takes at most $O(\kappa\log(1/\varepsilon))$ levels to decrease average diameter down to an ε fraction of the original diameter of the data. Combined with lemmas 23, 24, 25, we see that the three rules considered will decrease diameter at a fast rate whenever the covariance dimensions in local regions are small.

4.4.2 Axis parallel splitting rules

It was shown in [DF08b] that axis-parallel splitting rules do not always adapt to data that is intrinsically low-dimensional. They exhibit a data set in $\mathbb{R}^D$

that has low Assouad dimension $O(\log D)$, and where k-d trees (and also, it can be shown, dyadic trees) require D levels to halve the data diameter.

The adaptivity of axis-parallel rules to covariance dimension is unclear. But they *are* guaranteed to decrease diameter at a rate depending on D. The following result states that it takes at most $O(D(\log D)\log(1/\varepsilon))$ levels to decrease average diameter to an ε fraction of the original data diameter.

Lemma 27. *Suppose a partition tree is built using either k-d tree or dyadic tree by cycling through the coordinates. Let $\mathcal{A}_l$ be the partition of $\mathcal{X}$ defined by the nodes at level l. Then we have*

$$\Delta^2_{n,a}\left(\mathcal{A}_l\right) \leq \Delta^2_n\left(\mathcal{A}_l\right) \leq \frac{D}{2^{\lfloor l/D \rfloor}}\Delta^2_n\left(\mathcal{X}\right).$$

Proof. We assume that the procedure builds the tree by cycling through the coordinates (a single coordinate is used at each level).

Suppose a cell A is split into $\mathcal{A} = \{A_1, A_2\}$ along some coordinate e_i. Then the average diameter along coordinate i decreases under either split satisfies

$$\frac{1}{\mu(A)}\left(\mu(A_1)\Delta^2_n\left(A_1 \cdot e_i\right) + \mu(A_2)\Delta^2_n\left(A_2 \cdot e_i\right)\right) \leq \frac{1}{2}\Delta^2_n\left(A \cdot e_i\right).$$

To see this, notice that the masses of the resulting cells are halved under *k-d* tree splits (we assume that n is a power of 2), while the diameters are halved under the dyadic tree splits.

We can derive an upper bound on the diameter decrease rate over multiple levels as follows. Let $X \sim \mathcal{U}(\mathbf{X})$, and let A_l be the cell to which $\widetilde{X}$ belongs at level $l \geq 0$ in the tree (built by either procedure). Let $l \geq 1$, if we condition on the event that the split at level $l-1$ is along coordinate i, we have by the above argument that

$$\mathbb{E}_X\left[\Delta^2_n\left(A_l \cdot e_i\right)\right] \leq \frac{1}{2}\mathbb{E}_X\left[\Delta^2_n\left(A_{l-1} \cdot e_i\right)\right].$$

No matter the coordinate used for the previous split, we always have

$$\mathbb{E}_X\left[\Delta^2_n\left(A_l \cdot e_i\right)\right] \leq \mathbb{E}_X\left[\Delta^2_n\left(A_{l-1} \cdot e_i\right)\right],$$

and it follows that after a multiple of D levels we have

$$\mathbb{E}_X\left[\Delta_n^2\left(A_l \cdot e_i\right)\right] \leq \frac{1}{2^{l/D}} \Delta_n^2\left(\mathcal{X} \cdot e_i\right),$$

for all $i \in [D]$. Summing over all coordinates, we then get

$$\mathbb{E}_X\left[\Delta_n^2\left(A_l\right)\right] \leq \mathbb{E}_X\left[\sum_i^D \Delta_n^2\left(A_l \cdot e_i\right)\right] \leq \frac{D}{2^{l/D}} \Delta(\mathcal{X}).$$

To conclude, notice that $\mathbb{E}_X\left[\Delta_n^2\left(A_l\right)\right]$ is exactly $\Delta_n^2\left(\mathcal{A}_l\right)$ where $\mathcal{A}_l$ is the partition defined by the nodes at level l. □

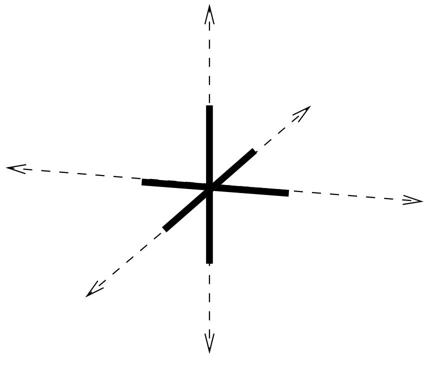
4.5 Data diameter decrease rates: low Assouad dimension

4.5.1 Limitations of axis-parallel rules

Consider the data space,

$$S = \bigcup_{i \neq j} \{te_i \pm \varepsilon e_j : -1 \leq t \leq 1\},\ i, j \in [D],$$

where e_i is the unit vector in the ith coordinate direction, and $\varepsilon > 0$ is some small constant serving as noise. S is an extreme case of a (noisy) sparse data set: each point has at most two nonzero coordinates, and one such coordinates is ε close to zero.



It is not hard to see that tree structures with axis-parallel splits (such as k-d trees and dyadic trees) would require at least D levels to halve the diameter of S;

that is, any tree with fewer levels would contain leaf cells of diameter greater than one. Thus halving the diameter would require 2^D data points, which is prohibitive for large D.

However, by using a richer class of splits, cell size can be decreased a lot quicker. By Lemma 15, S has Assouad dimension $d \leq O(\log D)$, and it is shown in [DF08a] that an RP tree halves the diameter in just $O(d \log d)$ levels, no matter the distribution over the data space. A version of this result is presented below in the next section.

This example suggests that, depending on the distribution μ on $\mathcal{X}$, regression based on axis-parallel cells might require a data size (n) exponential in D in order to attain low risk, whereas regression based on RP splits might do better, requiring resources that depend just on the intrinsic dimension d. However, there is an interesting subtlety. We show in Theorem 50 (Appendix) that the excess risk of a dyadic tree regressor depends on D only in the form of a leading constant 2^D, and not in the exponent of n. That is, for $n \geq 2^D$, the risk looks like $O\left(n^{-2/(2+d)}\right)$. This is a curse of dimension that emerges in a finite-sample analysis but not necessarily in an asymptotic analysis. All our results on RP tree regression in this paper are finite-sample convergence rates which depend just on d even for small n.

4.5.2 Decrease rate for the RP tree

Here we work with a modified version of the RP tree called `basicRPtree`. It takes as input a region $A_0 \subset \mathcal{X}$, and $\Delta > 0$, and (using a dataset $\mathbf{X}$ drawn from $\mathcal{X}$) builds a tree rooted at A_0 whose leaves form a partition $\mathcal{A}$ of A_0 such that $\Delta_n(\mathcal{A}) < \Delta$. We will soon bound the height of this tree in terms of the Assouad dimension of $\mathcal{X}$.

In a random projection (RP) tree [DF08a], each cell is split by a random hyperplane; specifically, a random direction is chosen from the surface of the unit sphere, and then the cell is split along that direction, at the median plus a small random perturbation. As a result of this perturbation, the two halves of the cell might not contain an equal number of points, and, in some cases, might be severely imbalanced. To keep the tree balanced (which insures that its has height

```
Procedure basicRPtree($A_0 \subset \mathcal{X}$, $\Delta$)
    $\mathcal{A}_0 \leftarrow \{A_0\}$;
    for $i \leftarrow 1$ to $\infty$ do
        if $\Delta_n(\mathcal{A}_{i-1}) \leq \Delta$ and $i$ is odd then
            return;
        end
        Choose a random direction $v \sim \mathcal{N}\left(0, \frac{1}{D} I_D\right)$;
        Choose a random $\tau \sim \mathcal{U}[-1,1] \cdot \frac{6}{\sqrt{D}} \Delta_n(A_0)$;
        foreach cell $A \in \mathcal{A}_{i-1}$ do
            if $i$ is odd then
                $t \leftarrow \text{median}\{z^\top v : z \in \mathbf{X} \cap A_0\} + \tau$ ; // Noisy splits
            else
                $t \leftarrow \text{median}\{z^\top v : z \in \mathbf{X} \cap A\}$; // Median splits
            end
            $A_{\text{left}} \leftarrow \{x \in A, x^\top v \leq t\}$;
            $A_{\text{right}} \leftarrow A \setminus A_{\text{left}}$;
            if $(A_{\text{left}} \cap \mathbf{X})$ and $(A_{\text{right}} \cap \mathbf{X})$ are both nonempty then
                (children of $A$) $\leftarrow A_{\text{left}}, A_{\text{right}}$ ;
            end
        end
        $\mathcal{A}_i \leftarrow$ partition of $A_0$ defined by the leaves of the current tree;
    end
```

at most $O(\log n)$) we alternate the RP split with another type of bisection that splits exactly at the median. Thus, if the tree is grown to l levels, we are assured that each cell contains at most a $2^{-l/2}$ fraction of the original data set; hence the overall depth of the tree must be $O(\log n)$.

Lemma 28. *There is an absolute constant C' for which the following holds. Let $A \subset \mathbb{R}^D$ and suppose $A \cap \mathbf{X}$ has Assouad dimension d. Then with probability at least $1/2$ over the randomization within the algorithm,*
`basicRPtree`$(A, \Delta_n(A)/2)$ *returns a tree of depth at most $C' d \log d$.*

Proof. The proof is a direct consequence of Lemma 9 of [DF08a] applied to the "noisy" splits at alternating levels in procedure `basicRPtree`.

Lemma 9 of [DF08a] states the following:

> Let $r = \Delta_n(A)/512\sqrt{d}$ and consider an r-cover of A; now consider pairs of balls $B = B(z, r)$, $B' = B(z', r)$, where z, z' are in the cover and $\|z - z'\| \geq \frac{1}{2}\Delta_n(A) - 2r$. Every "noisy" split has a constant probability of separating $B \cap \mathbf{X}$ and $B' \cap \mathbf{X}$.

Notice that `basicRPtree` stops if for all such pairs, no leaf of the tree contains points from both $B \cap \mathbf{X}$ and $B' \cap \mathbf{X}$.

Fix such a pair B and B'. By Lemma 9 of [DF08a], the probability that some cell at level i contains points from both $B \cap \mathbf{X}$ and $B' \cap \mathbf{X}$ goes down exponentially with i. A union bound over at most $(O(d)^d)$ such pairs yields the statement of our lemma. □

Portions of this chapter appear in:

– N. Verma, S. Kpotufe, S. Dasgupta, "Which spatial partition trees are adaptive to intrinsic dimension?", *Uncertainty in Artificial Intelligence*, 2009.

Chapter 5

Adaptive regression rates for the RPtree

In this chapter we analyze the recently-proposed *random projection tree* (RP tree), which uses random hyperplanes to partition space (Figure 4.1(c)). Previous work has analyzed RP trees for unsupervised learning, and established that they are adaptive to intrinsic dimension when used in this way [DF08a, GLZ08, VKD09]. Here we explore their use in regression and show formally that, because they decrease the diameter of the data within their cells quickly, they have risk that depend just on the intrinsic dimension of data, namely on the Assouad dimension. The results in this chapter validate the intuition of the previous chapter (Chapter 4) that fast data diameter decrease rate imply good regression risk, and more importantly, adaptive rates.

We develop novel tools for the analysis of bias. As previously mentioned, the bias of a tree estimator is typically analyzed in terms of the physical diameter of its cells (see, for instance, Chapter 20 of [DGL96a]). However, this can be worked out only when the cells have simple shapes like hyper-rectangles. For example, the cells of an RP tree are irregular convex polytopes, and their diameters might not systematically decrease while moving down the tree. What we do instead is to track the diameter of the *data* within each cell, and we develop new techniques to relate these empirical *data diameters* to the estimator's bias. Our method takes the focus away from the cells' physical diameters, opening the door to richer partitioning rules with nontrivial cell structure.

5.1 Detailed overview of results

5.1.1 Building the regression tree

A tree-based regressor works in two phases.

1. The data space is split into some partition $\mathcal{A}$.

2. A regressor is learned as a piecewise continuous function over the cells of $\mathcal{A}$.

In this work we'll consider a piecewise constant regressor over $\mathcal{A}$, defined as follows: for any $x \in \mathcal{X}$, let $\mathcal{A}(x)$ be the cell of $\mathcal{A}$ to which x belongs, and set

$$f_{n,\mathcal{A}}(x) \doteq \frac{\sum_{i=1}^{n} Y_i \cdot \mathbb{1}\left[X_i \in \mathcal{A}(x)\right]}{n \cdot \mu_n(\mathcal{A}(x))}$$

if $\mu_n(\mathcal{A}(x)) > 0$ (that is, if the cell $\mathcal{A}(x)$ contains at least one training point). If $\mathcal{A}(x) \cap \mathbf{X}$ is empty, then a default setting $f_{n,\mathcal{A}}(x) = y_o$ is used instead, for some $y_o \in \mathcal{Y}$. We will often refer to the final regressor as f_n when the partition $\mathcal{A}$ used for the estimate is clear from context.

The first phase of the regression algorithm implicitly builds a tree, each of whose nodes corresponds to a region of $\mathbb{R}^D$. Each node has two children whose regions are a partition of its own. We will also associate each such cell A with the data points $A \cap \mathbf{X}$ that happen to fall in it.

All the splitting is done by random hyperplanes, and thus each cell is a convex region of $\mathbb{R}^D$. The precise details are given in Procedure `basicRPtree` of Section 4.5.2. In order to boost the probability that this procedure returns a short tree, we call it repeatedly in Procedure `coreRPtree`. This last procedure is the key subroutine for our main tree building procedure called `adaptiveRPtree`. Procedure `coreRPtree` operates as follows (via calls to `basicRPtree`):

- It takes as input a region $A \subset \mathbb{R}^D$ (or more precisely, the data points that fall in this region).

```
Procedure adaptiveRPtree(sample $\mathbf{X} \subset \mathbb{R}^D$, $0 < \delta < 1$)
  $\mathcal{A}^0 \leftarrow \mathbb{R}^D$;
  for $i \leftarrow 1$ to $\infty$ do
      foreach cell $A \in \mathcal{A}^{i-1}$ do
          (subtree rooted at $A$) $\leftarrow$ coreRPtree ($A$, $\Delta_n(A)/2, \delta$);
      end
      $\mathcal{A}^i \leftarrow$ partition of $\mathbb{R}^D$ defined by the leaves of the current tree;
      level $(\mathcal{A}^i) \leftarrow \max_{A \in \mathcal{A}^i}$ level $(A)$ ; // level = depth in tree
      // There are two options for stopping and returning a partition.
      Option 1:  Cross-validation
      if $\Delta_n (\mathcal{A}^i) = 0$ or level $(\mathcal{A}^i) \geq 2 \log n$ then
          Define $R'_n(\cdot)$ as the empirical risk on a validation sample
          $(\mathbf{X}', \mathbf{Y}')$ of size $n$;
          $\mathcal{A}^* \leftarrow \underset{\mathcal{A} \in \{\mathcal{A}^0, \ldots, \mathcal{A}^i\}}{\text{argmin}} R'_n(f_{n,\mathcal{A}})$;
          return $f_n \doteq f_{n,\mathcal{A}^*}$;
      end
      Option 2:  Automatic stopping
      $\alpha(n) \leftarrow \left(\log^2 n\right) \log\log(n/\delta) + \log(1/\delta)$;
      if $\Delta_n^2 (\mathcal{A}^i) \leq \Delta_n^2 (\mathcal{A}^0) \cdot (\alpha(n)/n) \cdot 2^{\text{level}(A^i)}$ then
          $\mathcal{A}^* \leftarrow \underset{\mathcal{A} \in \{\mathcal{A}^{i-1}, \mathcal{A}^i\}}{\text{argmin}} \left( \frac{\alpha(n)}{n} \cdot |\mathcal{A}| + \Delta_n^2 (\mathcal{A}) \right)$;
          return $f_n \doteq f_{n,\mathcal{A}^*}$;
      end
  end
```

```
Procedure coreRPtree($A \subset \mathcal{X}$, $\Delta$, $\delta$)
  Call basicRPtree($A, \Delta$) (see Section 4.5.2) $\log(3n/\delta)$ times and
  return the shortest tree.
```

- By recursive splits, it builds a tree whose root corresponds to A and whose leaves form a partition of A, call it $\mathcal{A}$, such that $\Delta_n(\mathcal{A}) \leq \Delta_n(A)/2$.

- If A has zero diameter (for instance, if it contains one point), then the procedure leaves it untouched. Otherwise, a tree is returned whose leaves contain at most $\lceil |A \cap \mathbf{X}|/2 \rceil$ points.

The main tree building algorithm is Procedure `adaptiveRPtree`. It starts with a single node $\mathcal{A}^0$ for all of $\mathbb{R}^D$, and then grows a tree in measured steps. At each stage, the current set of leaves constitute a partition $\mathcal{A}^i$ of $\mathbb{R}^D$, whose cells have diameter $\Delta_n(\mathcal{A}^i) \leq 2^{-i}\Delta_n(\mathbb{R}^D)$. Then the subroutine `coreRPtree` is called on each leaf to yield an even finer partition $\mathcal{A}^{i+1}$.

This process is stopped when each cell of the current partition is sufficiently small that the bias is controlled, but also has sufficiently many data points in it that the variance is controlled. How can the right stopping point be identified? We present two options.

1. *Automatic stopping.* We return a partition as soon as the data diameters of cells are small enough relative to tree size.

2. *Cross-validation.* Here, we grow a large tree and then prune it using a separate validation sample $(\mathbf{X}', \mathbf{Y}')$, also of size n, drawn from the same underlying distribution. To prune, an intermediate partition $\mathcal{A}^i$ is chosen which minimizes the empirical risk

$$R'_n(g) \doteq \frac{1}{n} \sum_{i \in [n]} \left\| Y'_i - g(X'_i) \right\|^2 .$$

The automatic stopping option requires no validation sample and is computationally faster. As we'll see, its risk bound is only slightly worse than that of the cross-validation option.

Regardless of which stopping rule is employed, it follows from the properties of `coreRPtree` that the final tree has height at most $2 \log 2n$ and the number of partitions $\mathcal{A}^i$ generated is at most $\log 2n$. These are important properties which we emphasize in the following remark.

Remark 29. *Given the implementation of* `coreRPtree`*, the tree returned by procedure* `adaptiveRPtree` *has the following properties:*

- *The number of data points in a cell (node) at level i is at most half the number contained in its ancestor at level $i-2$. Taking rounding effects into consideration, this means that by level $2(1+\log n)$, each cell will contain at most one point. Thus the entire tree built by* `adaptiveRPtree` *has depth at most $2\log 2n$.*

- *By construction, each node contains at least one data point. Therefore, there are at most n leaves and $n-1$ internal nodes.*

- *Since the tree has height at most $2\log 2n = \log 4n^2$, a total of at most $8n^2\log(3n/\delta)$ random directions are required to build the entire tree.*

5.1.2 Main Results

The excess risk of the tree-based regressor can be expressed in terms of the rate at which diameters decrease from the root down. We have the following definition.

Definition 30. *Given a sample $\mathbf{X}$, we say that* `coreRPtree` *attains a diameter decrease rate of k on $\mathbf{X}$ for $k \geq d$, if every call to it in the second loop of the main procedure* `adaptiveRPtree` *returns a tree of depth at most k.*

The main theorem below builds upon the following result which establishes the diameter decrease rate attained by the algorithm.

Lemma 31 (Corollary to Lemma 28)**.** *Let C' be as in Lemma 28. Suppose $\mathcal{X}$ has Assouad dimension d and fix $\mathbf{X} \subset \mathcal{X}$. With probability at least $1-\delta/3$ over the randomness in the algorithm,* `adaptiveRPtree` *attains a diameter decrease rate $k \leq C'd\log d$ on $\mathbf{X}$.*

Proof. The procedure `adaptiveRPtree` grows the tree in blocks: it starts with a single node (cell) that contains all of $\mathbf{X}$ and then repeatedly expands one of its current leaf nodes A into the subtree produced by the call `coreRPtree`$(A, \Delta_n(A), 2)$.

Consider any such A. Since $\mathcal{X}$ has Assouad dimension d, so does $A \cap \mathbf{X} \subset \mathcal{X}$; we can therefore apply Lemma 28. Procedure `coreRPtree` calls `basicRPtree` $\log(3n/\delta)$ times and returns the smallest tree; thus the probability that this tree has depth $> C' d \log d$ is at most $\delta/(3n)$.

How many nodes A are expanded in this way? Any A with data diameter zero (for instance, containing just one point) is untouched by `coreRPtree`; on the other hand, any A with nonzero diameter will certainly get expanded (on account of the median split, if nothing else). Thus `coreRPtree` is invoked at most once on each internal node of the tree. There are at most n leaf nodes and thus at most $n-1$ internal nodes. A union bound over them yields an overall probability of failure at most $\delta/3$. □

The following is the main result of this chapter.

Theorem 32. *Assume that $\mathcal{X}$ has Assouad dimension d. There exist constants C, C' independent of d and $\mu(\mathcal{X})$, such that the following hold. Pick any $\delta > 0$ and define*

$$\alpha(n) \doteq \left(\log^2 n\right) \log\log(n/\delta) + \log(1/\delta).$$

With probability at least $1-\delta$:

(a) `coreRPtree` *attains a diameter decrease rate of $k \leq C' d \log d$.*

(b) If the automatic stopping option is used, the excess risk of the regressor is

$$\|f_n - f\|^2 \leq C \cdot \left(\Delta_{\mathcal{Y}}^2 + \lambda^2\right)\left(\Delta_{\mathcal{X}}^2 + 1\right) \cdot \left(\frac{\alpha(n)}{n}\right)^{2/(2+k)}.$$

(c) If the cross-validation option is used and

$$n \geq \max\left\{\alpha(n), \lambda^2 \Delta_{\mathcal{X}}^2 / \Delta_{\mathcal{Y}}^2, \alpha(n) \Delta_{\mathcal{Y}}^2 / \lambda^2 \Delta_{\mathcal{X}}^2\right\},$$

then the excess risk of the regressor is

$$\begin{aligned} \|f_n - f\|^2 \quad &\leq \quad C \cdot (\lambda \Delta_{\mathcal{X}})^{2k/(2+k)} \left(\frac{\Delta_{\mathcal{Y}}^2 \cdot \alpha(n)}{n}\right)^{2/(2+k)} \\ &\qquad + 2\Delta_{\mathcal{Y}}^2 \sqrt{\frac{\log\log n + \log 8/\delta}{2n}}. \end{aligned}$$

The two stopping options yield similar bounds in terms of the dependence on n and d; however the cross-validation bound has a better dependence on λ, $\Delta_{\mathcal{X}}$, and $\Delta_{\mathcal{Y}}$.

In section 5.2, we lay out the key tools for the rest of the analysis, culminating in a risk bound in terms of data diameter. In section 5.3, we investigate the two stopping rules, and bound the excess risk of the final regressor in terms of the observed diameter decrease rate.

The algorithm takes an input a permissible failure probability δ. There are three sources of failure, and we apportion each of them a $\delta/3$ probability: failure to build a tree with the desired height and diameter decrease rate; an $(\mathbf{X}, \mathbf{Y})$ sampling failure in which either the empirical masses of cells do not accurately represent their true masses or the y-values within cells have non-representative averages; and an $(\mathbf{X}', \mathbf{Y}')$ sampling failure in the cross-validation step.

Parts (a), (b), and (c) of Theorem 32 result from Lemma 31, Lemma 43, and Lemma 41 respectively.

5.2 Risk bound for $f_{n,\mathcal{A}}$

In this section we develop the necessary tools to bound the excess risk of $f_{n,\mathcal{A}}$, where $\mathcal{A}$ is an RP tree partition, that is, $\mathcal{A}$ is defined by the leaves of the tree returned by `adaptiveRPtree`.

5.2.1 Generic decomposition of pointwise excess risk

We start the analysis with a standard decomposition of the excess risk into bias and variance terms. Let $\mathcal{A}$ be any partition of $\mathcal{X}$, on which the regressor $f_{n,\mathcal{A}}$ is defined. Recall that we denote by $\mathcal{A}(x)$ the cell of $\mathcal{A}$ containing x.

A useful intermediary between $f_{n,\mathcal{A}}$ and the target f is the following function on $\mathcal{X}$:

$$\widetilde{f}_{n,\mathcal{A}}(x) \quad \doteq \quad \frac{\sum_{i=1}^{n} f(X_i)\mathbb{1}\left[X_i \in \mathcal{A}(x)\right]}{n\mu_n(\mathcal{A}(x))}$$

if $\mu_n(\mathcal{A}(x)) \neq 0$; otherwise $\widetilde{f}_{n,\mathcal{A}}(x) = y_o \in \mathcal{Y}$. Notice that $\widetilde{f}_{n,\mathcal{A}}(x)$ is just the conditional expectation of the estimate given $\mathbf{X}$ fixed. Both $f_{n,\mathcal{A}}$ and $\widetilde{f}_{n,\mathcal{A}}$ are constant within any cell $A \in \mathcal{A}$; we will therefore overload notation and occasionally write these quantities as $f_{n,\mathcal{A}}(A)$ and $\widetilde{f}_{n,\mathcal{A}}(A)$, respectively.

Similar to the decomposition of the expected excess risk in (1.2), the pointwise excess risk at x can be bounded as

$$\begin{aligned} \|f_{n,\mathcal{A}}(x) - f(x)\|^2 &\leq \left(\left\|f_{n,\mathcal{A}}(x) - \widetilde{f}_{n,\mathcal{A}}(x)\right\| + \left\|\widetilde{f}_{n,\mathcal{A}}(x) - f(x)\right\|\right)^2 \\ &\leq 2\underbrace{\left\|f_{n,\mathcal{A}}(\mathcal{A}(x)) - \widetilde{f}_{n,\mathcal{A}}(\mathcal{A}(x))\right\|^2}_{\text{variance}} \\ &\quad + 2\underbrace{\left\|\widetilde{f}_{n,\mathcal{A}}(x) - f(x)\right\|^2}_{\text{bias}^2}. \end{aligned} \tag{5.1}$$

In the next two lemmas, we separately bound the variance and the bias.

Lemma 33 (Variance)**.** *Fix any partition $\mathcal{A}$ and a sample $\mathbf{X} = \{X_1, \ldots, X_n\} \subset \mathcal{X}$. Suppose the Y_i are now drawn according to their conditional distribution given X_i. Pick any $\delta > 0$. Then with probability at least $1 - \delta$, for every cell $A \in \mathcal{A}$ with $\mu_n(A) > 0$:*

$$\left\|f_{n,\mathcal{A}}(A) - \widetilde{f}_{n,\mathcal{A}}(A)\right\|^2 \leq \Delta_{\mathcal{Y}}^2 \cdot \frac{2 + \ln(|\mathcal{A}|/\delta)}{n\mu_n(A)}.$$

Proof. For any cell $A \in \mathcal{A}$, let $I(A) = \{1 \leq i \leq n : X_i \in A\}$ be the indices of points falling in that cell. Then $\mu_n(A) = |I(A)|/n$, and

$$\left\|f_n(A) - \widetilde{f}_n(A)\right\| = \left\|\frac{1}{|I(A)|}\sum_{i \in I(A)} (Y_i - f(X_i))\right\|.$$

Changing any single Y_i value alters this expression by at most $\Delta_{\mathcal{Y}}/|I(A)|$. We can therefore use McDiarmid's inequality to assert that with probability at least $1 - \delta/|\mathcal{A}|$ over the choice of the Y_i's,

$$\left\|f_n(A) - \widetilde{f}_n(A)\right\| \leq \mathbb{E}\left\|f_n(A) - \widetilde{f}_n(A)\right\| + \Delta_{\mathcal{Y}} \cdot \sqrt{\frac{\ln(|\mathcal{A}|/\delta)}{2|I(A)|}}.$$

The expectation can be bounded as follows:

$$\begin{aligned}
\mathbb{E}\left\|f_n(A) - \widetilde{f}_n(A)\right\| &\leq \left(\mathbb{E}\left\|f_n(A) - \widetilde{f}_n(A)\right\|^2\right)^{1/2} \\
&= \frac{1}{|I(A)|}\left(\mathbb{E}\left\|\sum_{i \in I(A)} (Y_i - f(X_i))\right\|^2\right)^{1/2} \\
&= \frac{1}{|I(A)|}\left(\sum_{i \in I(A)} \mathbb{E}\left\|Y_i - f(X_i)\right\|^2\right)^{1/2} \\
&\leq \frac{1}{|I(A)|}\left(|I(A)|\Delta_{\mathcal{Y}}^2\right)^{1/2} \;=\; \frac{\Delta_{\mathcal{Y}}}{\sqrt{|I(A)|}}.
\end{aligned}$$

The first line uses Jensen's inequality. The third uses the fact that the vectors $v_i = Y_i - f(X_i)$ are independent random vectors with zero expectation, so that $\mathbb{E}\left\|\sum_i v_i\right\|^2 = \sum_i \mathbb{E}\left\|v_i\right\|^2$.

We conclude with a union bound over all nonempty $A \in \mathcal{A}$. □

Lemma 34 (Bias)**.** *Fix any partition $\mathcal{A}$ and any set of n points $\mathbf{X} = \{X_i\}_{i=1}^n \subset \mathcal{X}$. For any $x \in \mathcal{X}$ with $\mu_n(\mathcal{A}(x)) > 0$,*

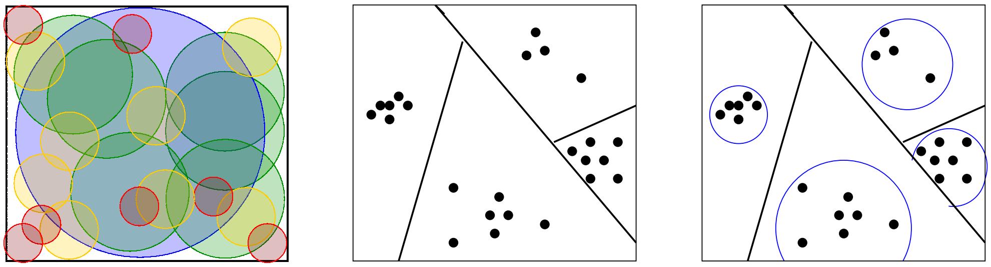
$$\left\|\widetilde{f}_{n,\mathcal{A}}(x) - f(x)\right\| \leq \lambda \cdot \Delta\left(\mathcal{A}(x)\right).$$

Proof. Let $A = \mathcal{A}(x)$, so that

$$\begin{aligned}
\left\|\widetilde{f}_{n,\mathcal{A}}(x) - f(x)\right\| &= \left\|\frac{\sum_{i=1}^n (f(X_i) - f(x))\mathbb{1}\left[X_i \in A\right]}{n\mu_n(A)}\right\| \\
&\leq \frac{\sum_{i=1}^n \left\|f(X_i) - f(x)\right\| \mathbb{1}\left[X_i \in A\right]}{n\mu_n(A)} \\
&\leq \frac{\sum_{i=1}^n \lambda \left\|X_i - x\right\| \mathbb{1}\left[X_i \in A\right]}{n\mu_n(A)} \leq \lambda \cdot \Delta\left(A\right),
\end{aligned}$$

where the second inequality uses the Lipschitz condition on $f(\cdot)$. □

What we have at this point is a fairly standard bias-variance decomposition of the risk. It contains two quantities that non-trivial to bound in our context: the empirical weights of cells, $\mu_n(A)$; and, more importantly, their physical diameters $\Delta(A)$.



(a) Cover $\mathcal{B}$ (b) Partition $\mathcal{A}$ (c) Partition $\mathcal{A}'$

Figure 5.1: We start with a cover $\mathcal{B}$ of $\mathcal{X}$ with balls of different size; then, we see the data and obtain a partition $\mathcal{A}$; and finally we substitute $\mathcal{A}$ with an alternate partition $\mathcal{A}'$, by intersecting the cells of $\mathcal{A}$ with balls of $\mathcal{B}$.

To relate the empirical masses $\mu_n(A)$ to their true values $\mu(A)$, we could use a uniform large deviation bound. A naive such bound would involve terms in D, since each cell is an intersection of hyperplanes. To avoid such a dependency, we make heavy use of the fact that the *directions* of the hyperplanes are chosen at random, independent of the data points, and that the data are consulted only to determine the *displacements* of the boundaries along these directions.

The bigger challenge is to handle cell diameters. The bound on bias involves the physical diameters $\Delta(A)$ of cells, and these might not decrease gracefully down the tree. So we create an alternate partition $\mathcal{A}'$ with the following properties:

- Each cell of $\mathcal{A}$ is the union of two cells of $\mathcal{A}'$.
- Every cell in $\mathcal{A}'$ is either void of data points (and thus has low probability under μ and can be disregarded) or else has a physical diameter that is roughly the same as its data diameter.

This construction lets us upper-bound the bias in terms of the data diameters $\Delta_n(A)$ of cells, which are easier to quantify and to control.

5.2.2 An alternate partition

Although the algorithm works with a partition $\mathcal{A}$ built from recursive hyperplane splits, and the regressor is defined using this partition, for purposes of the analysis only we will also consider an alternate, related partition $\mathcal{A}'$. This $\mathcal{A}'$ will be designed so that $f_{n,\mathcal{A}'}$ is equivalent to $f_{n,\mathcal{A}}$ on most of $\mathcal{X}$, but has the advantage that its cells are well-behaved as explained at the end of the previous section.

$\mathcal{A}'$ is obtained by intersecting the cells of $\mathcal{A}$ with balls or complements-of-balls from a fixed, pre-defined collection $\mathcal{B}$ (Figure 5.1). Specifically, let $\mathcal{B}_i$ be a cover of $\mathcal{X}$ by balls of radius $\Delta_{\mathcal{X}}/2^i$. Take a variety of scales: $i = 0, 1, 2, \ldots, I = \lfloor \log n^{2/(2+d)} \rfloor$. Then $\mathcal{B}$ is the union of all these balls of different sizes, blown up by a factor of 4:

$$\mathcal{B} = \bigcup_{i=0}^{I} \{4B : B \in \mathcal{B}_i\}.$$

at The partition $\mathcal{A}'$ is created by replacing each cell $A \in \mathcal{A}$ by two cells A'_1, A'_2 as follows:

- If $A \cap \mathbf{X} = \emptyset$, then set $A'_1 = A$ and $A'_2 = \emptyset$.

- Otherwise, set $i = \min\{I, \lceil \log(\Delta_{\mathcal{X}}/\Delta_n(A)) \rceil\}$; we'll find a ball $B \in \mathcal{B}_i$ such that $A \cap \mathbf{X}$ is contained in $4B$. To this end, pick any $x \in A \cap \mathbf{X}$, and pick the ball $B \in \mathcal{B}_i$ whose center z is closest to x. Then $A \cap \mathbf{X} \subset 4B$ because $\forall x' \in A \cap \mathbf{X}$,

$$\begin{aligned} \|z - x'\| &\leq \|z - x\| + \|x - x'\| \\ &\leq 2^{-i}\Delta_{\mathcal{X}} + \Delta_n(A) \\ &\leq 2^{-i}\Delta_{\mathcal{X}} + 2^{-(i-1)}\Delta_{\mathcal{X}} \quad \leq \quad 4 \cdot 2^{-i}\Delta_{\mathcal{X}} \end{aligned}$$

  (using the fact that $i - 1 \leq \log(\Delta_{\mathcal{X}}/\Delta_n(A))$, whereby $\Delta_n(A) \leq 2^{-(i-1)}\Delta_{\mathcal{X}}$). Define $A'_1 = A \cap 4B$ and $A'_2 = A \setminus A'_1$.

$\mathcal{A}'$ is the collection of all such A'_1, A'_2, over $A \in \mathcal{A}$. What makes this refined partition valuable is that the average physical diameter of its cells can be upper-bounded by the empirical data diameters of cells in $\mathcal{A}$.

Lemma 35 (Diameters of $\mathcal{A}'$)**.** *Let $\mathcal{A}$ be a partition of $\mathcal{X}$ and define $\mathcal{A}'$ as above. Then*

$$\sum_{A' \in \mathcal{A}'} \mu_n(A')\Delta^2(A') \leq 64\Delta_n^2(\mathcal{A}) + 256 n^{-4/(2+d)}\Delta_{\mathcal{X}}^2.$$

Proof. Pick any cell $A \in \mathcal{A}$ for which $A \cap \mathbf{X} \neq \emptyset$. This cell is broken into two pieces in $\mathcal{A}'$: a set A_1' with $\mu_n(A_1') = \mu_n(A)$ and a set A_2' with $\mu_n(A_2') = 0$. Specifically, $A_1' = A \cap 4B$, where B is a ball of radius $2^{-i}\Delta_{\mathcal{X}}$, for $i = \min\{I, \lceil \log(\Delta_{\mathcal{X}}/\Delta_n(A)) \rceil\}$. It follows that A_1' has diameter at most $8 \cdot 2^{-i}\Delta_{\mathcal{X}} \leq 8 \max\{2^{-I}\Delta_{\mathcal{X}}, \Delta_n(A)\}$.

This bound makes it natural to divide the cells of $\mathcal{A}$ into two groups: $\mathcal{A}_+ = \{A \in \mathcal{A} : \Delta_n(A) > 2^{-I}\Delta_{\mathcal{X}}\}$; and $\mathcal{A} \setminus \mathcal{A}_+$. Then

$$\begin{aligned}
\sum_{A' \in \mathcal{A}'} \mu_n(A')\Delta^2(A') &= \sum_{A \in \mathcal{A}_+} \mu_n(A)\Delta^2(A_1') + \sum_{A \in \mathcal{A} \setminus \mathcal{A}_+} \mu_n(A)\Delta^2(A_1') \\
&\leq \sum_{A \in \mathcal{A}_+} 64\mu_n(A)\Delta_n^2(A) + \sum_{A \in \mathcal{A} \setminus \mathcal{A}_+} 64\mu_n(A)2^{-2I}\Delta_{\mathcal{X}}^2 \\
&\leq 64\Delta_n^2(\mathcal{A}) + 256 n^{-4/(2+d)}\Delta_{\mathcal{X}}^2.
\end{aligned}$$

□

5.2.3 Bounding the empirical masses of cells

In order to bound the integrated excess risk, we will need the empirical masses of cells, $\mu_n(A')$, to be close to their true masses, $\mu(A')$. In particular, this will allow us to disregard cells that are empty of data since they will have little effect on the integrated excess risk.

The uniform convergence bounds we use are based on the following standard notion of *shatter coefficient*, which describes the complexity of a (potentially infinite) collection of subsets of $\mathbb{R}^D$. In our case, each such subset is a cell.

Definition 36. *Let n be some positive integer, and let $\mathcal{C}$ be a class of subsets of $\mathbb{R}^D$. The n-shatter coefficient of $\mathcal{C}$, denoted $\mathcal{S}(\mathcal{C}, n)$, is the largest possible size of a collection of sets obtained by intersecting sets of $\mathcal{C}$ with a sample $\mathbf{X}$ of size n. That is,*

$$\mathcal{S}(\mathcal{C}, n) \doteq \max_{|\mathbf{X}|=n} |\{C \cap \mathbf{X} : C \in \mathcal{C}\}|.$$

For example, suppose $D = 1$ and $\mathcal{C}$ is the set of all half lines, that is, intervals of the form $(-\infty, t]$ or $[t, +\infty)$. For any set of n distinct points $\mathbf{X} = \{x_1, \ldots, x_n\}$ where (without loss of generality) $x_1 < \cdots < x_n$, the intersection of these points with half lines consists of all subsets of the form $\{x_1, \ldots, x_i\}$ or $\{x_i, \ldots, x_n\}$. Therefore $\mathcal{S}(\mathcal{C}, n) = 2n$.

The following theorem of Vapnik and Chervonenkis gives uniform rates of convergence for empirical masses over a class $\mathcal{C}$, using the $2n$-shattering coefficient of $\mathcal{C}$.

Lemma 37 (Relative VC bounds [VC71]). *Let $\mathcal{C}$ be a class of subsets of $\mathbb{R}^D$. Pick any $\delta > 0$. Suppose a sample of size n is drawn independently at random from a distribution μ over $\mathbb{R}^D$, with resulting empirical distribution μ_n. Then with probability at least $1 - \delta$ over the choice of sample, all $C \in \mathcal{C}$ satisfy*

$$\mu(C) \;\; \leq \;\; \mu_n(C) + 2\sqrt{\mu_n(C)\frac{\ln \mathcal{S}(\mathcal{C}, 2n) + \ln(4/\delta)}{n}} + 4\frac{\ln \mathcal{S}(\mathcal{C}, 2n) + \ln(4/\delta)}{n}.$$

where $\mathcal{S}(\mathcal{C}, 2n)$ is the $2n$-shatter coefficient of $\mathcal{C}$.

Recall that in our algorithm, we use the data sample $\mathbf{X}$ to generate a tree that contains various candidate partitions $\mathcal{A}^i$, and that eventually one of these partitions is chosen, and a regressor is defined on it. We would like to argue that for any $\mathcal{A} = \mathcal{A}^i$, the empirical mass of each cell of $\mathcal{A}'$ is close to its true mass. How should the class $\mathcal{C}$ in lemma 37 be defined?

Since the tree has height at most $2 \log 2n$ (remark 29 of Section 5.1.1) and the splits are by hyperplanes, one option is to let $\mathcal{C}$ consist of all convex sets that are intersections of at most $2 \log 2n$ halfspaces and a ball in $\mathcal{B}$ or the complement of such a ball. This works, but yields a class whose complexity depends on the ambient dimension D. Instead, we exploit the fact that the *random directions* of the hyperplanes used in the tree can be chosen before seeing $\mathbf{X}$ (removing that source of randomness), whereas their *displacements* depend on the random sample $\mathbf{X}$. We can thus define a class of lower complexity independent of D.

Lemma 38 (Masses of cells of $\mathcal{A}'$). *There is a constant C_0 such that the following holds. Pick any $\delta > 0$. With probability at least $1 - \delta$ over the choice of $\mathbf{X}$ and the randomness in the algorithm, we have that for any partition $\mathcal{A} = \mathcal{A}^i$ generated* *during the construction of the tree, every cell $A' \in \mathcal{A}'$ satisfies*

$$\begin{aligned} \mu(A') &\leq \mu_n(A') + 2\sqrt{\mu_n(A')\frac{\mathcal{V} + \ln(4/\delta)}{n}} + 4\frac{\mathcal{V} + \ln(4/\delta)}{n}, \text{ where} \qquad (5.2) \\ \mathcal{V} &\leq C_0 \log n(\log n + \text{loglog}(1/\delta)). \end{aligned}$$

Proof. Suppose without loss of generality that during the construction of the tree, all random directions (for hyperplane splits) are picked from a fixed collection $\mathcal{P}$ without replacement. How big should $\mathcal{P}$ be so that there are enough directions to choose from? The implementation of `coreRPtree` ensures that $|\mathcal{P}| \leq 8n^2 \log(3n/\delta)$ is sufficient (see remark 29 of section 5.1.1). Now fix such a collection $\mathcal{P}$ and let $\mathcal{H}_\mathcal{P}$ be the class of half spaces of $\mathbb{R}^D$ defined by hyperplanes normal to the directions in $\mathcal{P}$. For any tree partition $\mathcal{A}$, each cell of $\mathcal{A}$ is the intersection of at most $2\log 2n$ elements of $\mathcal{H}_\mathcal{P}$ since the tree is guaranteed to have height at most $2\log 2n$ (remark 29). Each cell of $\mathcal{A}'$ is the intersection of a ball or the complement of a ball in $\mathcal{B}$ with a cell of $\mathcal{A}$.

All such cells therefore belong to the following class of subsets of $\mathbb{R}^D$:

$$\mathcal{C} = \left\{ h : h = h_0 \cap \left(\bigcap_{l=1}^{2\log 2n} h_l \right), \ h_0 \text{ or } h_0^\complement \text{ is in } \mathcal{B}, h_l \in \mathcal{H}_\mathcal{P} \right\}.$$

We now proceed to bounding $\mathcal{S}(\mathcal{C}, 2n)$, the $2n$-shatter coefficient of $\mathcal{C}$.

Given $2n$ sample points and a direction $v \in \mathcal{P}$, there are at most $4n$ possible intersections of the sample with halfspaces normal to v. Therefore

$$\begin{aligned} \mathcal{S}(\mathcal{C}, 2n) &\leq 2|\mathcal{B}|(4n|\mathcal{P}| + 1)^{2\log 2n} \\ &\leq 2|\mathcal{B}|\left(32n^3 \log(3n/\delta) + 1\right)^{2\log 2n}. \end{aligned}$$

Since $\mathcal{X}$ has Assouad dimension d, we have $|\mathcal{B}| \leq \sum_{i=0}^{I} 2^{di} \leq 2n^{2d/(2+d)}$. The proof is completed by letting $\mathcal{V} = \log \mathcal{S}(\mathcal{C}, 2n)$ for $\mathcal{P}$ fixed, and then appealing to Lemma 37. □

5.2.4 A bound on the integrated excess risk in terms of data diameters

Lemma 39 (Integrated excess risk)**.** *There exists a constant C_1 independent of d and μ such that the following holds. Define $\alpha(n) \doteq \left(\log^2 n\right) \operatorname{loglog}(1/\delta) + \log(1/\delta)$. With probability at least $1 - \delta/3$ over the choice of $(\mathbf{X}, \mathbf{Y})$ and the randomness in the algorithm, for all partitions $\mathcal{A} = \mathcal{A}^i$ obtained during the execution of* `adaptiveRPtree`*,*

$$\|f_{n,\mathcal{A}} - f\|^2 \ \leq \ C_1 \left(\Delta^2_{\mathcal{Y}} \, |\mathcal{A}| \, \frac{\alpha(n)}{n} + \lambda^2 \left(\Delta^2_n \left(\mathcal{A}\right) + n^{-4/(2+d)} \Delta^2_{\mathcal{X}} \right) \right).$$

Proof. Define $\delta' = \delta/(6 \log 2n)$. By Lemma 38 we have that with probability at least $1 - \delta'$ over the randomness in the algorithm and the choice of $\mathbf{X}$, equation (5.2) — with δ' substituted for δ — holds for all cells $A' \in \mathcal{A}'$, where $\mathcal{A} = \mathcal{A}^i$ is any partition obtained during the construction of the tree and $\mathcal{V} \leq C_0 \log n(\log n + \operatorname{loglog}(1/\delta'))$. Let's assume that this condition holds, and fix $\mathbf{X}$. Henceforth we will randomize only over the choice of $\mathbf{Y}$.

Pick any partition $\mathcal{A} = \mathcal{A}^i$ obtained by `adaptiveRPtree`. The integrated excess risk can be decomposed over $\mathcal{A}'$ as follows:

$$\|f_{n,\mathcal{A}} - f\|^2 = \sum_{A' \in \mathcal{A}'} \int_{A'} \|f_{n,\mathcal{A}}(x) - f(x)\|^2 \, \mu(dx).$$

We next divide the cells of $\mathcal{A}'$ into two groups: those of significant mass, whose bias and variance must be controlled, and those of negligible mass, whose contribution to the overall risk can be ignored even if it is the worst possible. Specifically, set

$$\mathcal{A}'_{>} \doteq \left\{ A' \in \mathcal{A}', \mu_n(A') \geq \frac{\mathcal{V} + \ln(4/\delta')}{n} \right\}, \text{ and } \mathcal{A}'_{<} \doteq \mathcal{A}' \setminus \mathcal{A}'_{>}.$$

From equation (5.2), every $A' \in \mathcal{A}'_{>}$ satisfies $\mu(A') \leq 7\mu_n(A')$ while every $A' \in \mathcal{A}'_{<}$ has $\mu(A') \leq 7(\mathcal{V} + \ln(4/\delta'))/n$.

Given this upper bound on the masses of cells in $\mathcal{A}'_{<}$, their integrated risk is

$$\begin{aligned}\sum_{A' \in \mathcal{A}'_<} \int_{A'} \|f_{n,\mathcal{A}}(x) - f(x)\|^2 \mu(dx) &\leq \sum_{A' \in \mathcal{A}'_<} \Delta_{\mathcal{Y}}^2 \cdot \mu(A') \\ &\leq 7\Delta_{\mathcal{Y}}^2 \cdot |\mathcal{A}'| \cdot \frac{\mathcal{V} + \ln(4/\delta')}{n}. \qquad (5.3)\end{aligned}$$

Now for the integration over $\mathcal{A}'_>$. Each cell $A' \in \mathcal{A}'_>$ holds exactly the same data points as its counterpart $A \in \mathcal{A}$; thus $f_{n,\mathcal{A}}$ and $f_{n,\mathcal{A}'}$ coincide on A'. We first apply (5.1), and then use Lemmas 33 and 34 to assert that with probability at least $1 - \delta'$ over the choice of $\mathbf{Y}$,

$$\begin{aligned}&\sum_{A' \in \mathcal{A}'_>} \int_{A'} \|f_{n,\mathcal{A}}(x) - f(x)\|^2 \mu(dx) \\ &= \sum_{A' \in \mathcal{A}'_>} \int_{A'} \|f_{n,\mathcal{A}'}(x) - f(x)\|^2 \mu(dx) \\ &\leq \sum_{A' \in \mathcal{A}'_>} 2\lambda^2 \Delta^2(A') \cdot \mu(A') + \sum_{A' \in \mathcal{A}'_>} 2\Delta_{\mathcal{Y}}^2 \cdot \frac{2 + \ln(|\mathcal{A}'|/\delta')}{n\mu_n(A')} \cdot \mu(A') \\ &\leq \sum_{A' \in \mathcal{A}'_>} 2\lambda^2 \Delta^2(A') \cdot 7\mu_n(A') + \sum_{A' \in \mathcal{A}'_>} 2\Delta_{\mathcal{Y}}^2 \cdot \frac{2 + \ln(|\mathcal{A}'|/\delta')}{n\mu_n(A')} \cdot 7\mu_n(A') \\ &\leq 14\lambda^2 \sum_{A' \in \mathcal{A}'_>} \mu_n(A')\Delta^2(A') + 14\Delta_{\mathcal{Y}}^2 |\mathcal{A}'| \cdot \frac{2 + \ln(|\mathcal{A}'|/\delta')}{n}. \qquad (5.4)\end{aligned}$$

We can simplify $\ln|\mathcal{A}'|$ to $O(\log n)$ since the tree has at most n leaves. By combining the bounds in (5.3) and (5.4), and absorbing various constants into a single C_o, we get

$$\begin{aligned}\|f_{n,\mathcal{A}} - f\|^2 \leq C_o \Bigg(&\Delta_{\mathcal{Y}}^2 |\mathcal{A}| \frac{\log^2 n + \log n \operatorname{loglog} 1/\delta' + \log(1/\delta')}{n} \\ &+ \lambda^2 \sum_{A' \in \mathcal{A}'} \mu_n(A')\Delta^2(A') \Bigg).\end{aligned}$$

To finish up, we call on lemma 35 to bound the summation on the right, and then take a union bound over the $\leq \log 2n$ possible partitions $\mathcal{A} = \mathcal{A}^i$. □

5.3 Risk of final regressor $f_n \doteq f_{n,\mathcal{A}^*}$

Recall that the `adaptiveRPtree` procedure starts with a partition $\mathcal{A}^0$ that has a single cell containing all the data, and then grows the tree to get increasingly finer partitions $\mathcal{A}^1, \mathcal{A}^2, \ldots$, where the data diameter of each $\mathcal{A}^i$ is half that of $\mathcal{A}^{i-1}$. Recall also that the *diameter decrease rate*, denoted k, is defined to be the maximum increase in tree depth during each of these individual growth spurts.

The tree is not grown indefinitely. To see this, note that the implementation of `coreRPtree` ensures that all cells at some level down the hierarchy would eventually have a single data point in them (see remark 29). In other words, $\Delta_n\left(\mathcal{A}^i\right) = 0$ eventually, at which point either of the two stopping criteria would hold.

Once the tree is constructed, a partition $\mathcal{A}^* = \mathcal{A}^i$ is chosen and a regressor is built on it. We now bound the excess risk of $f_n \doteq f_{n,\mathcal{A}^*}$ in terms of the diameter decrease rate achieved during `adaptiveRPtree`.

To get some insight into the form of the final risk bound, pretend for a moment that $\Delta_{\mathcal{X}}$, $\Delta_{\mathcal{Y}}$, and λ are all 1. Consider a partition $\mathcal{A}$ induced by the tree. If $\Delta_n(\mathcal{A}) = \zeta$, we would expect that the data diameter has been halved roughly $\log(1/\zeta)$ times. Since each halving grows the tree by $\leq k$ levels, $\mathcal{A}$ has depth at most $k\log(1/\zeta)$ in the tree, implying also that $|\mathcal{A}| \leq (1/\zeta)^k$. Plugging these values into the bound of Lemma 39, we get $\|f_{n,\mathcal{A}} - f\|^2 \lesssim \zeta^{-k}/n + \zeta^2$. Setting $\zeta = n^{-1/(2+k)}$ then gives the optimal bound $\|f_{n,\mathcal{A}^*} - f\|^2 \lesssim n^{-2/(2+k)}$.

In the analysis, a few basic facts will repeatedly be used. First, because such successive partition halves the data diameter,

$$\Delta_n(\mathcal{A}^i) \quad \leq \quad 2^{-i}\Delta_n(\mathcal{A}^0). \tag{5.5}$$

Second, by definition of diameter decrease rate, each halving grows the tree by $\leq k$ levels:

$$\text{level}\left(\mathcal{A}^i\right) \quad \leq \quad ki. \tag{5.6}$$

5.3.1 Risk bound for cross-validation option

For the cross-validation option, we begin by arguing that the tree contains at least one good partition $\mathcal{A}^i$, such that both $\Delta_n(\mathcal{A}^i)$ and $|\mathcal{A}^i|$ are reasonably small. The shrinkage in diameter, $\Delta_n(\mathcal{A}^i)/\Delta_n(\mathcal{A}^0)$, is roughly

$$\zeta \doteq \left(\frac{\Delta_{\mathcal{Y}}^2}{\lambda^2 \Delta_{\mathcal{X}}^2} \cdot \frac{\alpha(n)}{n}\right)^{1/(2+k)}$$

(recall $\alpha(n) = (\log^2 n) \log\log(n/\delta) + \log(1/\delta)$.) The analysis requires an unusual, albeit benign, lower bound on the number of samples n, the purpose of which is to ensure that n^2 exceeds both $(1/\zeta)^k$ and $(1/\zeta)^{2+d}$.

Lemma 40 (Existence of a good pruning)**.** *Suppose* `adaptiveRPtree` *is run with the cross-validation option, and yields a sequence of partitions $\mathcal{A}^0, \mathcal{A}^1, \ldots$ with a diameter decrease rate of k. Define*

$$\zeta \doteq \left(\frac{\Delta_{\mathcal{Y}}^2}{\lambda^2 \Delta_{\mathcal{X}}^2} \cdot \frac{\alpha(n)}{n}\right)^{1/(2+k)}$$

If $n \geq \max\left\{\alpha(n), \lambda^2\Delta_{\mathcal{X}}^2/\Delta_{\mathcal{Y}}^2, \alpha(n)\Delta_{\mathcal{Y}}^2/\lambda^2\Delta_{\mathcal{X}}^2\right\}$, then there exists $i \geq 0$ such that $\Delta_n\left(\mathcal{A}^i\right) \leq 2\zeta \cdot \Delta_n\left(\mathcal{X}\right)$ and $|\mathcal{A}^i| \leq (1/\zeta)^k$.

Proof. Consider the largest i at which $\operatorname{level}(\mathcal{A}^i) < k\log(1/\zeta)$. Then $|\mathcal{A}^i| \leq (1/\zeta)^k$. In bounding $\Delta_n(\mathcal{A}^i)$, there are two cases to consider.

Case 1: $\mathcal{A}^{i+1}$ is part of the tree. Then its level is $\geq k\log(1/\zeta)$, implying that $i+1 \geq \log(1/\zeta)$ (by (5.6)) and therefore that $i \geq \log(1/2\zeta)$, whereupon (by (5.5)) $\Delta_n(\mathcal{A}^i) \leq 2\zeta\Delta_n(\mathcal{A}^0)$.

Case 2: $\mathcal{A}^{i+1}$ is not part of the tree; that is, $\mathcal{A}^i$ satisfies the stopping criteria, i.e. either $\Delta_n(\mathcal{A}^i) = 0$ or $\operatorname{level}(\mathcal{A}^i) \geq 2\log n$. The lower bound on n ensures that $\operatorname{level}(\mathcal{A}^i) < k\log(1/\zeta) \leq 2\log n$. Therefore $\Delta_n(\mathcal{A}^i) = 0$. □

Next, we argue that cross-validation will find a partition that isn't too much worse than the $\mathcal{A}^i$ of Lemma 40.

Lemma 41. *There exists an absolute constant C (independent of d and μ), such that the following holds. Under the hypotheses of Lemma 40, with probability at* *least $1-2\delta/3$ over $(\mathbf{X},\mathbf{Y})$ and the randomness in the algorithm, the excess risk of the final regressor is bounded by*

$$\begin{aligned}\|f_n - f\|^2 &\leq C\cdot(\lambda\Delta_{\mathcal{X}})^{2k/(2+k)}\left(\Delta_{\mathcal{Y}}^2\cdot\frac{\alpha(n)}{n}\right)^{2/(2+k)}\\ &\quad + 2\Delta_{\mathcal{Y}}^2\sqrt{\frac{\log\log n + \log 4/\delta}{2n}}.\end{aligned}$$

Proof. Let $\mathcal{A}^i$ and ζ be as in Lemma 40. By applying Lemma 39 and then Lemma 40, we have with probability at least $1-\delta/3$ that

$$\begin{aligned}\left\|f_{n,\mathcal{A}^i} - f\right\|^2 &\leq C_1\left(\Delta_{\mathcal{Y}}^2\left|\mathcal{A}^i\right|\frac{\alpha(n)}{n} + \lambda^2\left(\Delta_n^2\left(\mathcal{A}^i\right) + n^{-4/(2+d)}\Delta_{\mathcal{X}}^2\right)\right)\\ &\leq C_1\left(\Delta_{\mathcal{Y}}^2\cdot\zeta^{-k}\frac{\alpha(n)}{n} + 5\lambda^2\zeta^2\Delta_{\mathcal{X}}^2\right) \leq C_2\lambda^2\Delta_{\mathcal{X}}^2\zeta^2.\end{aligned}$$

To analyze the cross validation phase, we fix the partitions $\mathcal{A}^j$ obtained from procedure `adaptiveRPtree`; there at most $\log 2n$ of these. Applying McDiarmid's inequality to the empirical risk, we see that with probability at least $1-\delta/3$ over the choice of $(\mathbf{X}',\mathbf{Y}')$, each $\mathcal{A}^j$ satisfies

$$\left|R\left(f_{n,\mathcal{A}^j}\right) - R_n'\left(f_{n,\mathcal{A}^j}\right)\right| \leq \Delta_{\mathcal{Y}}^2\sqrt{\frac{\ln(\log 2n) + \ln 3/\delta}{2n}}.$$

Thus if $f_n \doteq f_{n,\mathcal{A}^*}$ is the empirical risk minimizer,

$$\|f_n - f\|^2 \leq C_2\lambda^2\Delta_{\mathcal{X}}^2\zeta^2 + 2\Delta_{\mathcal{Y}}^2\sqrt{\frac{\log\log n + \log 4/\delta}{2n}}$$

with probability at least $1-2\delta/3$. □

5.3.2 Risk bound for automatic stopping option

The automatic criterion stops growing the tree as soon as

$$\frac{\Delta_n^2\left(\mathcal{A}^i\right)}{\Delta_n^2\left(\mathcal{A}^0\right)} \leq \frac{\alpha(n)}{n}2^{\text{level}\left(\mathcal{A}^i\right)},$$

at which point either $\mathcal{A}^i$ or $\mathcal{A}^{i-1}$ is chosen as the final partition $\mathcal{A}^*$. The shrinkage in diameter is expected to be roughly a factor of

$$\zeta \doteq \left(\frac{\alpha(n)}{n}\right)^{1/(2+k)},$$

corresponding to a depth of $k\log(1/\zeta)$ in the tree. In particular, if $\text{level}\left(\mathcal{A}^i\right) \geq k\log(1/\zeta)$ then the stopping criterion holds, because then $i \geq \text{level}\left(\mathcal{A}^i\right)/k \geq \log(1/\zeta)$ (recall (5.6)) and $\Delta_n(\mathcal{A}^i) \leq 2^{-i}\Delta_n(\mathcal{A}^0) \leq \zeta\Delta_n(\mathcal{A}^0)$ (recall (5.5)), whereupon

$$\frac{\Delta_n^2\left(\mathcal{A}^i\right)}{\Delta_n^2\left(\mathcal{A}^0\right)} \leq \zeta^2 = \frac{\alpha(n)}{n}\left(\frac{1}{\zeta}\right)^k \leq \frac{\alpha(n)}{n}2^{\text{level}\left(\mathcal{A}^i\right)}.$$

Lemma 42 (Properties of $\mathcal{A}^*$)**.** *Suppose the automatic stopping option is used, and that* `adaptiveRPtree` *attains a diameter decrease rate of k on* $\mathbf{X}$*. Define $\zeta \doteq \left(\frac{\alpha(n)}{n}\right)^{1/(2+k)}$ and assume $n \geq \alpha(n)$. Then, the final partition $\mathcal{A}^*$ retained for regression satisfies*

$$\left(\frac{\alpha(n)}{n}\cdot|\mathcal{A}^*| + \Delta_n^2\left(\mathcal{A}^*\right)\right) \leq \left(4\Delta_n^2\left(\mathcal{X}\right)+1\right)\zeta^2.$$

Proof. Let $\mathcal{A}^0, \mathcal{A}^1, \ldots$ be the partitions found by `adaptiveRPtree`, and suppose the stopping criterion holds for $\mathcal{A}^i$. We consider two cases:

Case 1: $\text{level}\left(\mathcal{A}^i\right) \leq k\log(1/\zeta)$. Then $|\mathcal{A}^i| \leq (1/\zeta)^k$ and by the stopping condition

$$\frac{\Delta_n^2\left(\mathcal{A}^i\right)}{\Delta_n^2\left(\mathcal{A}^0\right)} \leq \frac{\alpha(n)}{n}2^{\text{level}\left(\mathcal{A}^i\right)} \leq \frac{\alpha(n)}{n}\left(\frac{1}{\zeta}\right)^k = \zeta^2.$$

Case 2: $\text{level}\left(\mathcal{A}^i\right) > k\log(1/\zeta)$. Then $ki \geq \text{level}\left(\mathcal{A}^i\right) \geq k\log(1/\zeta)$, implying that $i-1 \geq \log(1/2\zeta)$, whereupon $\Delta_n(\mathcal{A}^{i-1}) \leq 2\zeta\Delta_n(\mathcal{A}^0)$ (recall (5.5)). Moreover, since the stopping condition doesn't hold for $\mathcal{A}^{i-1}$ we have (by the discussion preceding the lemma) that $\text{level}\left(\mathcal{A}^{i-1}\right) < k\log(1/\zeta)$.

In either case at least one of $\mathcal{A}^i$ and $\mathcal{A}^{i-1}$ has size at most $(1/\zeta)^k$ and diameter at most $2\zeta\cdot\Delta_n(\mathcal{A}^0)$. It follows that

$$\begin{aligned}\min_{j\in\{i-1,\,i\}}\left(\frac{\alpha(n)}{n}\cdot\left|\mathcal{A}^j\right| + \Delta_n^2\left(\mathcal{A}^j\right)\right) &\leq \frac{\alpha(n)}{n}\cdot\zeta^{-k} + 4\zeta^2\cdot\Delta_n^2\left(\mathcal{X}\right)\\ &= \left(4\Delta_n^2\left(\mathcal{X}\right)+1\right)\zeta^2,\end{aligned}$$

which concludes the argument. □

Lemma 43. *There exists an absolute constant C (independent of d and μ), such that the following holds. Suppose the automatic stopping option is used and that* `adaptiveRPtree` *achieves a diameter decrease rate of $k \geq d$ on* $\mathbf{X}$*. With probability*

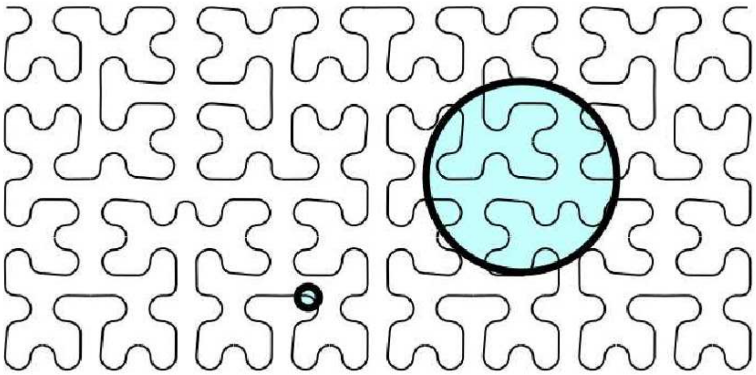


Figure 5.2: Hilbert space filling curve: the dimension depends on the scale at which the set is examined. Image obtained from [DF08a].

at least $1-\delta/3$ *over* $(\mathbf{X},\mathbf{Y})$ *and the randomness in the algorithm, the excess risk of the regressor is bounded by*

$$\|f_n - f\|^2 \leq C \cdot \left(\Delta_{\mathcal{Y}}^2 + \lambda^2\right)\left(\Delta_{\mathcal{X}}^2 + 1\right) \cdot \left(\frac{\alpha(n)}{n}\right)^{2/(2+k)}.$$

Proof. For $n \leq \alpha(n)$, the bound on the excess risk holds vacuously. We assume henceforth that $n > \alpha(n)$. Let $\zeta \doteq \left(\frac{\alpha(n)}{n}\right)^{1/(2+k)}$. By first applying Lemma 39 then Lemma 42, we have with probability at least $1-\delta$ that

$$\begin{aligned}
\|f_{n,\mathcal{A}^*} - f\|^2 &\leq C_1\left(\Delta_{\mathcal{Y}}^2\,|\mathcal{A}^*|\,\frac{\alpha(n)}{n} + \lambda^2\left(\Delta_n^2\left(\mathcal{A}^*\right) + n^{-4/(2+d)}\Delta_{\mathcal{X}}^2\right)\right)\\
&\leq C_1\left(\Delta_{\mathcal{Y}}^2 + \lambda^2\right)\left(|\mathcal{A}^*|\,\frac{\alpha(n)}{n} + \left(\Delta_n^2\left(\mathcal{A}^*\right) + n^{-4/(2+d)}\Delta_{\mathcal{X}}^2\right)\right)\\
&\leq C_1\left(\Delta_{\mathcal{Y}}^2 + \lambda^2\right)\left(\left(4\Delta_{\mathcal{X}}^2 + 1\right)\zeta^2 + \zeta^2\Delta_{\mathcal{X}}^2\right)\\
&\leq C\left(\Delta_{\mathcal{Y}}^2 + \lambda^2\right)\left(\Delta_{\mathcal{X}}^2 + 1\right)\zeta^2,
\end{aligned}$$

which concludes the argument. □

5.4 Final remarks

We have demonstrated a tree regressor that performs well in scenarios where the data space $\mathcal{X} \subset \mathbb{R}^D$ has low Assouad dimension $d \ll D$. In such cases, the integrated excess risk is roughly of the form $n^{-2/(2+k)}$ for $k = O(d \log d)$, and has no dependence on the ambient dimension D. But this still leaves room for improvement: is there an efficient tree-based regressor that achieves the optimal rate, $n^{-2/(2+d)}$?

We show in Chapter 6 that tree-kernel hybrid methods achieve this rate in general metric spaces. Moreover, in that chapter the usual $O(n)$ evaluation time of kernel methods is reduced to $O(2^d \log n)$ using a special tree data structure. This is a significant improvement, though slower than the $O(\log n)$ evaluation time of a tree regressor.

Another set of open questions concerns the data model. Assouad dimension is fairly general while at the same time being amenable to analysis. However, it has some shortcomings that motivate exploration into alternative notions of intrinsic dimension. First of all, it is natural to allow the dimensionality of a data set to depend on the *scale* at which it is being examined. The set in Figure 5.2, for instance, looks two-dimensional from a distance but one-dimensional when restricted to smaller neighborhoods. And realistically, at even smaller neighborhood sizes, it would be full-dimensional because of white noise. At the very least, we would like to be able to handle data sets that have low intrinsic dimension only when restricted to neighborhoods of a certain radius. In the Appendix, we show how to extend our results to such a setting.

Finally we emphasize the fact that our results in this chapter are obtained for relatively small n. As previously mentioned, (see Section 4.5.1) it is possible that a regression method attains rates that are adaptive to intrinsic dimension for large n (see e.g. Theorem 50 of the Appendix concerning dyadic trees), but this is not helpful in practice since large data sizes (n) is exactly what we want to avoid.

Most of this chapter appear in:

– S. Kpotufe, S. Dasgupta, "A tree-based regressor that adapts to intrinsic dimension", Journal of Computer and System Sciences, Special Issue on Learning Theory, (Invited Submission).

Chapter 6

Tree-kernel hybrid regressors

Tree-based regression and kernel regression are two popular approaches to nonparametric regression, each with its benefits and drawbacks. The main benefit of tree-based regression is that the regression estimate can be computed fast in time $O(\log n)$, where n is the training sample size. Kernel regression has worse evaluation time of $\Omega(n)$ but often yields better estimates than tree-based methods [Tor97, Bre96].

Can we combine aspects of tree-based and kernel regression in order to *guarantee* prediction performance comparable to that of kernel regression and a fast computation time as in tree-based regression? A simple hybrid approach considered here is to start with a partition of the input space as in tree-based methods, and use a kernel to average regression estimates from many cells of the partition. The estimate from each cell is weighted according to the distance from the query x to some *center* point of the cell (Figure 6.1).

The choice of center points is crucial if we want to guarantee both improved prediction performance over tree-methods and improved time performance over kernel methods. In this chapter we lay down a set of theoretical principles to help guide these choices, and we discuss many other practical benefits of tree-kernel hybrids.

How do we choose partition centers so as to attain good prediction performance comparable to kernel methods? We start by examining the theoretical rates at which kernel methods converge to the regression function. Suppose the input data X lies in a space $\mathcal{X} \subset \mathbb{R}^D$ of low intrinsic dimension as is often the case with modern data (e.g. manifold, sparse data). Kernel methods attain convergence rates of the form $n^{-1/O(d)}$ ([BL06], see also Chapter 3), where d is some notion of intrinsic dimension such as the Assouad dimension. Tree-based methods, however, typically have convergence rates of the form $n^{-1/O(D)}$ even in cases where the data is intrinsically low-dimensional [GKKW02, Kpo09]. The results of Chapter 5 on RPtree regression alleviate this problem, but the dependence on intrinsic dimension is still not optimal as in the case of kernel approaches. We show in Section 6.1.1 that if the centers are chosen sufficiently close to the data relative to the kernel bandwidth, tree-kernel hybrids achieve a rate of the form $n^{-1/O(d)}$ as good as that of kernel regression.

In order to guarantee an $O(\log n)$ prediction time, the centers are also required to be sufficiently far from each other relative to the kernel bandwidth. Here the O notation hides constants that depend on the intrinsic dimension of the input data.

We provide two detailed instantiations of these ideas in Section 6.2. This is followed by a practical discussion of the various benefits of tree-kernel hybrids (Section 6.3). Specifically, in addition to providing a good time-quality trade-off, tree-kernel hybrids yield smooth models which are useful in domains such as robotic control; they provide the flexibility of selecting the bandwidth parameter locally in order to adapt to different regression complexities in different regions of space; finally, they facilitate global bandwidth selection in that the building process is itself informative about the location of the optimal bandwidth on the real line. Throughout the discussion in Section 6.3, these benefits are highlighted with empirical evaluations in the application domains of vision, robotic control, and out-of-sample extension for MDS embeddings.

6.1 Intuition behind tree-kernel hybrids

We are now ready to discuss the two main goals of tree-kernel hybrids, namely improved estimates and good evaluation time.

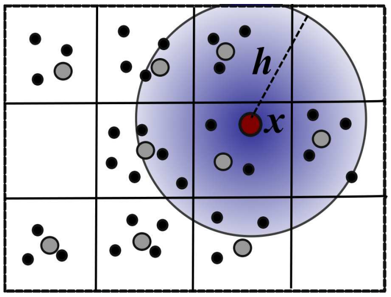


Figure 6.1: Tree-kernel hybrids at a high level: given a bandwidth $h > 0$, the estimate $f_n(x)$ is a weighted average of the Y contributions of the cells whose centers (gray points) fall in the ball $B(x, h)$. The kernel is assumed to assign 0 weight outside of $B(x, h)$.

6.1.1 Achieving good estimation quality

In order to understand how to choose cell centers to guarantee good estimation quality for tree-kernel hybrids, we start by looking at the sources of complexity for nonparametric regression.

Many variables contribute to the complexity of nonparametric regression, more precisely, to the excess risk $\|f_n - f\|^2$ of the regressor f_n. The three most important variables identified in the literature are the *smoothness* of the regression function $f(x) = \mathbb{E}[Y|x]$, the *Y-variance* $\sigma_Y^2 = \mathbb{E}\left[\|Y - f(x)\|^2 |x\right]$, and the *dimensionality* D of the input space $\mathcal{X}$. Suppose for example that f is λ-lipschitz[1], then any regressor f_n has error rate $\|f_n - f\|^2$ no better than $(\sigma_Y^2/\lambda^2 n)^{2/(2+D)}$ [Sto82]. Clearly, the dimension has the most radical effect on the quality of estimates. With kernel regressors, D in the above rate can be replaced by $d \ll D$, where d is the doubling dimension (see Chapter 3). It is therefore not surprising that kernel regressors, being adaptive to intrinsic dimension, tend to yield better estimates than tree-based regressors.

Adaptivity to intrinsic dimension

Here we would like to understand how dimensionality affects the expected risk $\mathbb{E}\|f_n - f\|^2$ of a regressor f_n. It is well known that this error is the sum of the variance of f_n and its square bias [GKKW02]. The dimensionality of the input

[1] $\forall x, x' \in \mathcal{X}, \|f(x) - f(x')\| \leq \lambda \|x - x'\|$.

space affects the variance of tree-based and kernel regressors differently. Understanding this will provide good intuition towards designing tree-kernel hybrids.

First, let's look at the variance of a tree-based regressor. Let $\mathrm{var}\left(f_n(x)\right)$ denote the variance of the estimate at x, i.e. $\mathrm{var}\left(f_n(x)\right) \doteq \mathbb{E}\left\|f_n(x) - \mathbb{E}\left[f_n(x)\right]\right\|^2$, where the randomness is over the sampling of $\mathbf{Y}$ for $\mathbf{X}$ fixed. The integrated variance $\int_{\mathcal{X}} \mathrm{var}\left(f_n(x)\right)\, d\mu(x)$ of a tree-based estimator defined over a partition $\mathcal{A}$ has the form $|\mathcal{A}|\, \sigma_Y^2/n$ [GKKW02], where $|\mathcal{A}|$ is the number of cells in $\mathcal{A}$. Suppose the diameter of data within a cell is h, and that the space $\mathcal{X}$ has diameter 1. Then under classical partitioning procedures (e.g. k-d trees, dyadic trees) $|\mathcal{A}|$ could be as large as h^{-D} even when the data has low intrinsic dimension ([DF08a], see also Chapter 4). Remember however that some recently analyzed procedures such as the Random Partitioning (RP) tree alleviate this problem: RPtree yields partitions of size just $h^{-O(d \log d)}$ where d is the doubling dimension of $\mathcal{X}$ ([DF08a], also covered in Chapter 4). However, we would ideally want $|\mathcal{A}|$ to be just $C \cdot h^{-d}$ since we know a partition of such size exists (apply the definition of doubling dimension recursively).

There is a sense in which kernel regression can be viewed as operating over such a partition of size at most $C \cdot h^{-d}$. If two points x and x' are close, their regression estimates must also be close since the two balls $B(x, h)$ and $B(x', h)$ contain nearly the same samples. In other words most balls are equivalent up to small differences in the estimates they yield, and technically, this equivalence class forms an implicit cover of the space. This point can be gleaned from the proof of Theorem 21 of Chapter 3. We make the point more concrete below.

Let $f_{n,h}(\cdot)$ be a kernel regressor using a bandwidth parameter h, and for simplicity let the kernel be a box kernel, i.e. $f_{n,h}(x)$ just averages the Y values of the points in $B(x, h)$. Let $Z = \{z_i\}_1^N$ be a minimal $h/2$-cover of $\mathcal{X}$ of size $N \leq C \cdot h^{-d}$ where d is the doubling dimension of $\mathcal{X}$. The kernel regressor $f_{n,h}$ acts approximately like a tree-based regressor $f_{n,\mathcal{A}_Z}$ operating on the Voronoi partition $\mathcal{A}_Z$ defined by the centers in Z. This can be seen by considering the biases and variances of these two regressors. First, both have bias $O(h)$. Next, we consider the variance $\mathrm{var}\left(f_{n,h}(x)\right)$ at x, given the randomness in sampling $\mathbf{X}$ and $\mathbf{Y}$. This variance is of the form $\sigma_Y^2/\left(n \cdot \mu(B(x,h))\right)$ [GKKW02], where μ is the marginal measure over $\mathcal{X}$. This is at most $\sigma_Y^2/\left(n \cdot \mu(B(z(x),h/2))\right)$ where $z(x)$ is the closest center in Z to x (notice that $B(z(x),h/2) \subset B(x,h)$). Therefore, the integrated variance of $f_{n,h}(\cdot)$ is at most

$$\begin{aligned}\int_{\mathcal{X}} \frac{\sigma_Y^2}{n \cdot \mu(B(z(x),\frac{h}{2}))}\,d\mu(x) &\leq \sum_{z\in Z}\int_{B(z,h/2)} \frac{\sigma_Y^2}{n \cdot \mu(B(z,\frac{h}{2}))}\,d\mu(x) \\ &= \frac{N\sigma_Y^2}{n} = \frac{|\mathcal{A}_Z|\,\sigma_Y^2}{n} \leq \frac{C \cdot h^{-d}\sigma_Y^2}{n},\end{aligned}$$

where the r.h.s is just the integrated variance of $f_{n,\mathcal{A}_Z}$. Thus, what makes kernel regression adaptive to intrinsic dimension is that the set of balls $B(x,h)$ form an equivalence class which implicitly covers the space in a near-optimal fashion.

Similarly, with tree-kernel hybrids we want most balls $B(x,h)$ to be equivalent up to small differences in the estimates they yield. Now however, if centers are far from the points they represent, then close-by centers can contribute very different Y values. In such a case, two close points x and x' might get quite different estimates because the balls $B(x,h)$ and $B(x',h)$ contain different centers with very different Y contributions. If instead the centers form an $O(h)$-cover of the training data $\mathbf{X}$, then we can expect as before that the set of balls $B(x,h)$ form an implicit cover of small size relative to d. We have the following lemma.

Lemma 44. *Consider a partition $\mathcal{A}$ of $\mathcal{X}$, and denote by $\mathbf{Q}_\mathcal{A}$ a set of centers for the cells of $\mathcal{A}$. For $q \in \mathbf{Q}_\mathcal{A}$, let n_q denote the number of training points from $\mathbf{X}$ falling in the cell of $\mathcal{A}$ corresponding to q and let $\bar{Y}_q$ denote the average Y value in this cell. Let the kernel $K(u)$ be a non increasing function of $u \in [0,\infty)$; K is positive on $u \in [0,1)$, maximal at $u = 0$, and is 0 for $u \geq 1$. Given a bandwidth $h > 0$, define a tree-kernel hybrid regressor as*

$$\begin{aligned} f_n(x) &= \sum_{q\in\mathbf{Q}_\mathcal{A}} w_q(x)\bar{Y}_q, \text{ where} \qquad (6.1) \\ w_q &= \frac{n_q(K(\|x-q\|/h)+\epsilon)}{\sum_{q'\in\mathbf{Q}_\mathcal{A}} n_{q'}(K(\|x-q'\|/h)+\epsilon)},\end{aligned}$$

where the positive constant $\epsilon \leq K(1/2)/n^2$ ensures the ratio is well defined. Assume the regression function f is λ-lipschitz. Then, provided $\mathbf{Q}_\mathcal{A}$ is an $(h/4)$-cover *of the sample* $\mathbf{X}$*, we have*

$$\mathop{\mathbb{E}}_{(\mathbf{X},\mathbf{Y}),X} \|f_n(X) - f(X)\|^2 \leq C \frac{\sigma_Y^2 \cdot (h/\Delta_{\mathcal{X}})^{-d}}{n} + 2\lambda^2 h^2,$$

where d *is the doubling dimension of* $\mathcal{X}$*,* $\Delta_{\mathcal{X}}$ *is its diameter, and the constant* C *depends just on* $K(\cdot)$ *and* d*.*

As a corollary, if $h \approx n^{-1/(2+d)}$ then the convergence rate is of the order of $n^{-2/(2+d)}$. This result is proved in Section 6.4.

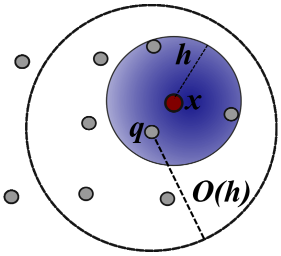
6.1.2 Maintaining a fast evaluation time

Evaluating a tree-kernel hybrid regressor at a point x has time complexity lower-bounded by the number of cell centers that fall in $B(x, h)$ and how fast we can identify these centers. To ensure that not too many such centers fall in $B(x, h)$ we just need to maintain the centers far apart. More precisely, given a bandwidth h, we use a partition of the space such that the centers are αh apart, $\alpha \in (0, 1)$, i.e. the centers form an *αh-packing*. This ensures that the number of centers in $B(x, h)$ are at most a constant that depends on α and the intrinsic dimension of the space, no matter the position of x. This is emphasized in the following lemma.

Lemma 45. *Suppose* $\mathcal{X}$ *has doubling dimension* d*. Let* $\mathbf{Q}$ *be a set of centers forming an* αh*-packing,* $\alpha \in (0, 1)$*. Then* $\max_{x \in \mathcal{X}} |\mathbf{Q} \cap B(x, h)| \leq (4/\alpha)^d$.

Proof. It is well known that an r-packing $\mathbf{Q}$ has size at most that of an $r/2$-cover of the space [Cla05]. Now applying the definition of doubling dimension recursively yields the lemma. □

Now it's left to make sure that $\mathbf{Q} \cap B(x, h)$ can be identified quickly. Suppose we can quickly find a point $q \in \mathbf{Q}$ such that $\|x - q\| \leq O(h)$. Then by a simple triangle inequality we can see that $\mathbf{Q} \cap B(x, h)$ is entirely contained in a ball $B(q, O(h)) \cap \mathbf{Q}$ around q. We therefore only have to compute weights for points falling in this ball around q. This is made fast by identifying such balls around every $q \in \mathbf{Q}$ before estimation time.



Notice that, by Lemma 45 above, $B(q, O(h)) \cap \mathbf{Q}$ has bounded size independent of n. We just have to find such a q close to x in $O(\log n)$ time. This is made clear in the implementation details of the following section.

6.2 Implementation details: Two types of tree-kernel hybrids

By Lemma 44 and Lemma 45 above, we want the centers to be both an αh-cover of $\mathbf{X}$ (for accuracy) and an αh-packing (for speed), i.e. an αh-net of $\mathbf{X}$, where $\alpha \in (0, 1)$ is assumed fixed. The first type of hybrids follows these propositions closely and operates on r-nets $\mathbf{Q}_r$ of $\mathbf{X}$. The other type only uses *approximate* r-nets of $\mathbf{X}$ obtained by merging the cells of a tree.

Algorithm 6: Building an r-net over points $\{x_i\}$

```
Input: points $\{x_i\}$, $r$.
Initialize $\mathbf{Q}_r \leftarrow \{x_1\}$.
Define $\|x_i - \mathbf{Q}_r\| \doteq \min_{x_j \in \mathbf{Q}_r} \|x_i - x_j\|$.
while $\max_{\{x_i\}} \|x_i - \mathbf{Q}_r\| \geq r$ do
  Add $x \doteq \text{argmax}_{\{x_i\}} \|x_i - \mathbf{Q}_r\|$ to $\mathbf{Q}_r$.
end while
Return: $r$-net $\mathbf{Q}_r$.
```

6.2.1 Tree-kernel hybrids using r-nets

Here we construct r-nets $\mathbf{Q}_r$ at different scales $r \in \{\Delta_{\mathcal{X}}/c^i\}_{i=0}^{O(\log n)}$, $1 < c \leq 2$ (see Procedure 1). Each $\mathbf{Q}_r$ is the set of centers of the Voronoi partition it induces, i.e. $q \in \mathbf{Q}_r$ is the center of the set $\{x \in \mathcal{X} : \|x - q\| \leq \|x - q'\|, q' \in \mathbf{Q}_r\}$ (ties are handled whichever way is appropriate). Data points in $\mathbf{X}$ are thus assigned to centers in $\mathbf{Q}_r$. For each $q \in \mathbf{Q}_r$ we define n_q as the number of points assigned to q, and we let $\bar{Y}_q$ be the average Y value of these points. The estimate $f_n(x)$ is then obtained as in Procedure 2.

Algorithm 7: Tree-kernel hybrids via r-nets

```
**Input:** $\{\mathbf{Q}_r\}$ for $r \in \{\Delta_{\mathcal{X}}/c^i\}_{i=0}^{O(\log n)}$, $h$, the query point $x$.
$r \leftarrow$ largest $r \in \{\Delta_{\mathcal{X}}/c^i\}_{i=0}^{O(\log n)}$ s.t. $r \leq \alpha h$.
$q \leftarrow \operatorname*{argmin}_{q' \in \mathbf{Q}_r} \|x - q'\|$.
$\mathbf{Q}_q \leftarrow \{\mathbf{Q}_r \cap B(q, 4r/\alpha)\}$.    // pre-computed
Define $\bar{Y} \doteq \frac{1}{n} \sum_{y_i \in \mathbf{Y}} y_i$.            // pre-computed
**Return:**
    $f_n(x) \leftarrow \frac{\sum_{q \in \mathbf{Q}_q} n_q K(\|x - q\|/h) \bar{Y}_q + \epsilon n \bar{Y}}{\sum_{q' \in \mathbf{Q}_q} n_{q'} K(\|x - q'\|/h) + \epsilon n}$.
```

For fast estimation we have to quickly find the closest $q \in \mathbf{Q}_r$ to x (see discussion in Section 6.1.2). Observe that finding the closest q can be done using a generic nearest neighbor (NN) search procedure such as the cover-tree or the navigating-net [BKL06, KL04]. These NN procedures guarantee to return q in time $2^{O(d)} \log |Q_r| \leq 2^{O(d)} \log n$, where d is the doubling dimension of the data. Combined with the evaluation over $\mathbf{Q}_q$, the whole evaluation is guaranteed to take time at most $2^{O(d)} \log n$ by Lemma 45, for small values of α. We can see from Lemma 44 that values of α around 1/4 are sensible and this is also empirically supported (see Section 6.3).

In light of Lemma 44 we can expect tree-kernel hybrids using r-nets to be competitive with kernel regression in terms of the quality of estimates. We will see that this also holds empirically.

6.2.2 Tree-kernel hybrids using cell merging procedure

The cell merging procedure described here emphasizes evaluation time while the above r-nets method emphasizes accuracy. Interestingly, cell merging can be applied to any partitioning tree method.

Using your favorite partitioning procedure, build a hierarchy of nested partitions $\{\mathcal{A}^i\}_{i=0}^{O(\log n)}$ of the input space $\mathcal{X}$. For each partition $\mathcal{A}$ in the hierarchy, we define a hybrid regressor as follows. Start with a set of centers $\{a_i\}$ over the cells of $\mathcal{A}$ (here we use the mean X value in each cell). Informed by Lemma 44, we would want these centers to form an approximate r-cover of the sample $\mathbf{X}$, for some r. We can for instance let r be the average distance between points in a cell (as in Procedure 3). Now, in order to apply Lemma 45 and ensure fast estimation, we need the cell centers to form an r-packing, however some centers might be too close to each other. We therefore apply the following cell merging procedure:

> Build an r-net $\mathbf{Q}_{\mathcal{A}} \subset \{a_i\}$ of the centers $\{a_i\}$, as in Procedure 1. Merge cell i with cell j if $a_j = \operatorname{argmin}_{q \in \mathbf{Q}_{\mathcal{A}}} \|a_i - q\|$.

For each $q \in \mathbf{Q}_{\mathcal{A}}$ let n_q be the total number of points assigned to q after the merge, and let $\bar{Y}_q$ be the average Y value of these points. The hybrid regressor is now defined as in Procedure 3.

The center a_x in Procedure 3 is found in $O(\log n)$ time by traversing the tree. Finding q is then a simple lookup. Thus, applying Lemma 45 we see that estimation time is $O(\log n + 2^{O(d)})$ in the worst case (again we treat α as a constant since sensible settings are around 1/4). The conditions of Lemma 44 are now only loosely met since the centers now only form an approximate cover of the data. We will see however that the estimates from these types of hybrid regressors are superior in quality to those obtained using the corresponding tree-based regressors on various datasets.

6.3 Practical benefits of tree-kernel hybrids

In this section we discuss various practical benefits of tree-kernel hybrids, namely time-accuracy tradeoff, local bandwidth selection, smoothness, and auto-

```
Algorithm 8: Tree-kernel hybrids via cell merging
    Input: tree $\{\mathcal{A}^i\}_i$, level $j$, query point $x$.
    Define $r^2 \doteq \frac{1}{2} \max_{A \in \mathcal{A}^j} \sum_{X_i, X_j \in A} \|X_i - X_j\|^2$.
    $h \leftarrow r/\alpha$.
    Let $\mathbf{Q}_{\mathcal{A}^j}$ be an $r$-net over centers of $\mathcal{A}^j$.
    $a_x \leftarrow$ center of the cell of $\mathcal{A}^j$ to which $x$ belongs.
    $q \leftarrow \operatorname*{argmin}_{q' \in \mathbf{Q}_{\mathcal{A}^j}} \|a_x - q'\|$.
    $\mathbf{Q}_q \leftarrow \{\mathbf{Q}_{\mathcal{A}^j} \cap B(q, 2r/\alpha)\}$.    // pre-computed
    Define $\bar{Y} \doteq \frac{1}{n} \sum_{y_i \in \mathbf{Y}} y_i$.
    Return:
        $f_n(x) \leftarrow \frac{\sum_{q \in \mathbf{Q}_q} n_q K(\|x - q\|/h) \bar{Y}_q + \epsilon n \bar{Y}}{\sum_{q' \in \mathbf{Q}_q} n_{q'} K(\|x - q'\|/h) + n\epsilon}$.
```

matic bandwidth range selection. These various benefits are illustrated through empirical evaluations on datasets chosen from the following real-world application domains: vision, robotic control, and out-of-sample extension for MDS embeddings.

6.3.1 Tradeoff between estimation time and accuracy

Here we showcase the effectiveness of tree-kernel hybrids on two prediction tasks. The first task is taken from the computer vision domain. Given 30 × 50-pixels images of a rotating teapot, the goal is to predict the angle of orientation (which is corrupted by Gaussian noise of variance 1). The second task is taken from robotic controls. The goal here is to learn the inverse dynamics for the movement of a robotic arm. That is, we want to predict the torque required to move a robotic arm to a given state, where each state is a 21-dimensional vector of position, velocity and acceleration [VS00, RW06]. Different torques are applied at 7 joint positions, but we only predict the first torque value in our experiments. In a basic experiment we draw a test sample, and draw training samples of different sizes. This basic experiment is repeated several times and the average error and time over all tests are reported. We use a test size of 180 points for the teapots experiments, and a test size of 360 points for the robotic torque.

For baseline comparisons, we use a fast implementation of kernel regression: a cover-tree [BKL06] is used to quickly identify $B(x, h) \cap \mathbf{X}$. The same cover-tree implementation is used for the r-nets-hybrids. In all the implementations we use a triangle kernel i.e. $K(u) = (1 - |u|)_+$. We also compare against k-d tree and RP tree regression as baseline for time performance. For every k-d tree split we pick the coordinate (out of D) with maximum variance. For RPtree, we simply pick a random direction. We note that one can improve on this procedure by picking the best direction out of many random directions (as done in the experiments of Chapter 4), but picking a single direction is sufficient for the purpose of the current experiment, namely showing that cell-merging hybrids improve on the estimates from the original trees.

For performance evaluations we report the root mean squared error (RMSE) on a test sample; for the tree-based regressors we report the minimum error over all tree levels, for the tree-kernel hybrids we report the minimum error over all levels and over a linear sweep of 100 bandwidth choices ranging from the diameter of the training sample down to the minimum interpoint distance; for kernel regression, the minimum error over the same bandwidth sweep is reported. For time performance we just report the average wall clock time on test evaluations. Results are reported in Figure 6.2.

Tradeoff under global bandwidth

The r-nets-hybrids (left column of Figure 6.2) perform as well as kernel regression for $\alpha = 0.25$ while consistently achieving better time. For larger values of α, the centers in $\mathbf{Q}_{\alpha h}$ are further from the data and are therefore poorer surrogates so prediction performance decreases; however these centers are far enough apart that fewer of them fall in the ball $B(x, h)$ so evaluation is faster.

The cell-merging-hybrids (center and right column of Figure 6.2) do not perform as well as the r-nets because they don't cover the space as well, being limited by the initial partition tree they are built upon. However they consistently

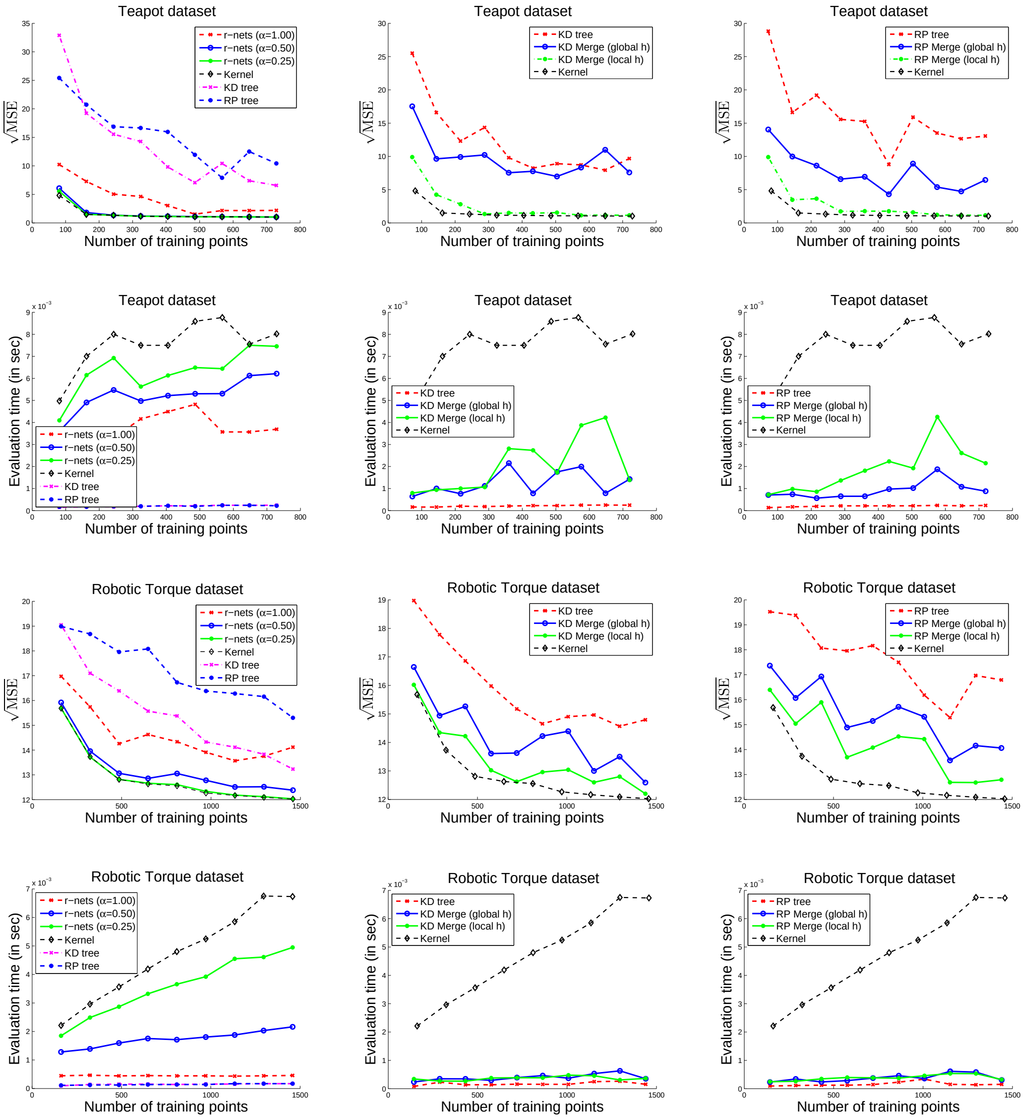


Figure 6.2: Performance vs. time for tree-kernel hybrids. First column shows the results for r-nets-hybrids. The next two columns are cell-merge-hybrids built using k-d trees and RP trees respectively.

perform better than the initial partition tree while attaining comparable evaluation time.

Improved accuracy with local bandwidths

In order to adapt to different regression complexities in different regions of space, we did the following: pick a level of the initial tree (level 2 in our experiments), and use a different bandwidth in each region defined by the partition at that level. Note that this is a much cheaper alternative to the local bandwidth methods for kernel regression, since these methods typically require the bandwidth parameter to be determined at evaluation time [Sta89]; in our case the local bandwidths can be pre-learned and saved for later evaluations. We can see that (Figure 6.2 center and right columns) using local bandwidth further improves the quality of estimates of the cell-merging-hybrids while approximately maintaining the evaluation time.

6.3.2 Smoothness of the learned function

For certain applications, we want the regressor to output a smooth function over the input domain. For instance in the torque prediction task for robotic arm, it is undesirable to have wildly varying torque values for close-by states as this could damage the physical machinery. Even though kernel methods output smooth functions, they are too slow especially for real-time tasks such as robotic control. The r-nets-hybrids provide a fast and smooth alternative. While the cell-merging-hybrids improve over the smoothness of the initial tree method they build upon, they are only smooth over regions of high density. Figure 6.3 shows predicted torque values for various methods as the input is varied over a line segment between the farthest two sample points. Observe that the cell-merging-hybrid produces a smooth curve only at the interior of the line as we pass through dense regions.

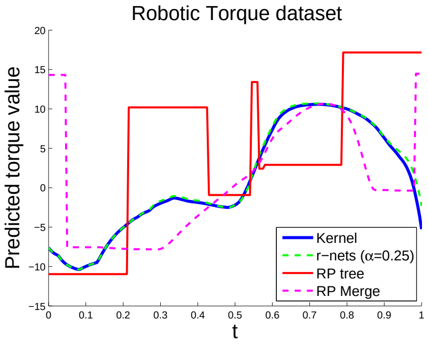


Figure 6.3: Predicted torque values over the line segment $\{x_1 + t(x_2 - x_1)\}$, $0 < t < 1$, where x_1, x_2 are the farthest two sample points.

6.3.3 Automatic bandwidth range selection

The sizes of the r-nets used in tree-kernel hybrids are informative about the location of the optimal bandwidth parameter on the real line. This is evident when one observes that intrinsic dimension is defined in terms of cover sizes (Chapter 2), and that this dimension is the most important factor in the optimal bandwidth (see discussion of Lemma 44). Remember that from Lemma 44 the MSE of these estimators breaks down into variance and bias terms as

$$C\frac{\sigma_Y^2 \cdot (h/\Delta_{\mathcal{X}})^{-d}}{n} + 2\lambda^2 h^2, \tag{6.2}$$

and is minimized by setting $h^* \approx \left(\sigma_Y^2/\lambda^2 n\right)^{-1/(2+d)}$, corresponding to the point where the variance and bias terms are essentially equal. Given our implementation of r-nets-hybrids, we aim to reduce the search space over h by automatically inferring a small range containing good settings of h.

We use the following heuristic: first, since we typically don't know the parameters in (6.2), we instead work with the simpler equation $|Q_h|/n + h^2$, knowing that $|Q_h| \leq C \cdot (h/\Delta_{\mathcal{X}})^{-d}$ (see proof of Lemma 45); now notice that as one varies h from $\Delta_{\mathcal{X}}$ down to 0, the variance term $|Q_h|/n$ starts out smaller than h^2 (for large n) and at some point (hopefully corresponding to good settings of h), the variance term becomes larger than the bias term h^2. Given candidate values $h \in \{\Delta_{\mathcal{X}}/c^i\}_{i=0}^{\infty}$, we just return the interval between the first h_1 where the variance term becomes smaller than the bias term and the last h_2 for which the variance term is larger. This sort of heuristic is not new [Kpo09] and it can be shown that

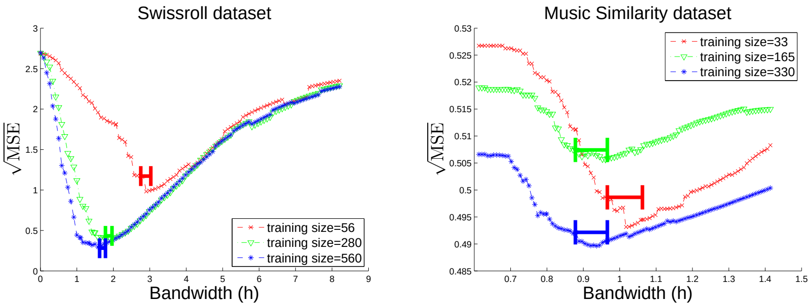


Figure 6.4: Predicted bandwidth intervals for MDS out-of-sample extension.

the interval $[h_1, h_2]$ must contain values in the order of h^*, provided σ_Y^2 and λ are bounded.

An application area where we can expect σ_Y^2 and λ to be small is that of out-of-sample extension for embeddings. Here we view the embedding of data points into some Euclidean space as a mapping f corrupted by noise. The task of out-of-sample extension is to predict the embedding of future points using the embeddings of previous training points. The embedding method used here is the classical Multidimensional Scaling (MDS). If the embedding is stable, which is desirable, then σ_Y^2 is small; also λ in this case quantifies how much interpoint distances are distorted by the embedding, and therefore can be expected to be small since MDS attempts to preserve distances.

The heuristic works remarkably well in our experiments (see Figure 6.4). We report the RMSE (after re-alignment of eigenvectors - see e.g. [BPV$^+$03]) for various values of h and various training sizes for two datasets. The datasets are the popular Swiss-roll dataset [BPV$^+$03] and the aset400 music similarity dataset with MFCC kernel [ML09]. The automatic-range returned for the optimal bandwidth typically includes the optimal value, hence reducing the search-space for h over the line. Here the r-nets are built at scales $r \in \{\Delta_{\mathcal{X}}/c^i\}_{i=0}^{O(\log n)}$, $c = 1.1$.

6.4 Analysis

In this section we go over the analysis that leads to the statement of Lemma 44. We first analyze the risk for a fixed choice of h, then we give guarantees for a simple cross-validation procedure for choosing a good h. We note that the analysis in this section also holds for any metric space $\mathcal{X}$, with the Assouad dimension defined the usual way using the given metric instead of the Euclidean metric.

6.4.1 Risk bound for h fixed

Throughout this section we assume $0 < h < \Delta_{\mathcal{X}}$ and we let $\mathbf{Q} = \mathbf{Q}_{h/4}$. We'll bound the risk for $f_{n,\mathbf{Q}}$ for any fixed choice of h. The results in this section only require the fact that $\mathbf{Q}$ is a cover of the data and thus preserves local information.

We'll proceed by bounding the bias and variance separately in the following two lemmas, and then combining these bounds in Lemma 44. We let μ denote the marginal measure over $\mathcal{X}$ and μ_n denote the corresponding empirical measure. As before we let $\widetilde{f}_{n,\mathbf{Q}}(x) \doteq \mathbb{E}_{\mathbf{Y}|\mathbf{X}}\, f_{n,\mathbf{Q}}(x)$ denote the conditional expectation of the estimate given $\mathbf{X}$ fixed. We will often use the shorthand notation $K(x, x', h)$ to denote $K(\|x - x'\|/h)$.

Lemma 46 (Variance at x). *Fix the sample $\mathbf{X}$, let $\mathbf{Q}$ be an $\frac{h}{4}$-cover of $\mathbf{X}$, and $0 < h < \Delta_{\mathcal{X}}$. Consider $x \in \mathcal{X}$ such that $\mathbf{X} \cap (B(x, h/4)) \neq \emptyset$. We have*

$$\mathop{\mathbb{E}}_{\mathbf{Y}|\mathbf{X}} \left\| f_{n,\mathbf{Q}}(x) - \widetilde{f}_{n,\mathbf{Q}}(x) \right\|^2 \leq \frac{2K(0)\sigma_Y^2}{K(1/2) \cdot n\mu_n\left(B(x, h/4)\right)}.$$

Proof. It is easily verified that, for independent random vectors v_i with expectation $\mathbf{0}$, $\mathbb{E}\left\|\sum_i v_i\right\|^2 = \sum_i \mathbb{E}\left\|v_i\right\|^2$. We apply this fact twice in the inequalities below, given that, conditioned on $\mathbf{X}$ and $\mathbf{Q} \subset \mathbf{X}$, the Y_i values are mutually independent and so are the $\bar{Y}_q$ values. We have

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathbf{Y}|\mathbf{X}} \left\| f_{n,\mathbf{Q}}(x) - \widetilde{f}_{n,\mathbf{Q}}(x) \right\|^2 &= \mathop{\mathbb{E}}_{\mathbf{Y}|\mathbf{X}} \left\| \sum_{q\in\mathbf{Q}} w_q(x) \left(\bar{Y}_q - \mathop{\mathbb{E}}_{\mathbf{Y}|\mathbf{X}} \bar{Y}_q \right) \right\|^2 \\
&\leq \sum_{q\in\mathbf{Q}} w_q^2(x) \mathop{\mathbb{E}}_{\mathbf{Y}|\mathbf{X}} \left\| \bar{Y}_q - \mathop{\mathbb{E}}_{\mathbf{Y}|\mathbf{X}} \bar{Y}_q \right\|^2 \\
&= \sum_{q\in\mathbf{Q}} w_q^2(x) \mathop{\mathbb{E}}_{\mathbf{Y}|\mathbf{X}} \left\| \sum_{i:X_i\in\mathbf{X}(q)} \frac{1}{n_q} \left(Y_i - \mathop{\mathbb{E}}_{\mathbf{Y}|\mathbf{X}} Y_i \right) \right\|^2 \\
&\leq \sum_{q\in\mathbf{Q}} w_q^2(x) \sum_{i:X_i\in\mathbf{X}(q)} \frac{1}{n_q} \mathop{\mathbb{E}}_{\mathbf{Y}|\mathbf{X}} \left\| Y_i - \mathop{\mathbb{E}}_{\mathbf{Y}|\mathbf{X}} Y_i \right\|^2 \\
&= \sum_{q\in\mathbf{Q}} w_q^2(x) \frac{\sigma_Y^2}{n_q} \\
&\leq \left(\max_{q\in\mathbf{Q}} \left\{ w_q(x) \frac{\sigma_Y^2}{n_q} \right\} \right) \sum_{q\in\mathbf{Q}} w_q \\
&= \max_{q\in\mathbf{Q}} \left\{ w_q(x) \frac{\sigma_Y^2}{n_q} \right\} \\
&= \max_{q\in\mathbf{Q}} \frac{(K(x,q,h)+\epsilon)\sigma_Y^2}{\sum_{q'\in\mathbf{Q}} n_{q'}(K(x,q',h)+\epsilon)} \\
&\leq \frac{2K(0)\sigma_Y^2}{\sum_{q\in\mathbf{Q}} n_q K(x,q,h)}. \qquad (6.3)
\end{aligned}$$

To bound the fraction in (6.3), we lower-bound the denominator as:

$$\begin{aligned}
\sum_{q\in\mathbf{Q}} n_q K(x,q,h) &\geq \sum_{q:\|x-q\|\leq h/2} n_q K(x,q,h) \\
&\geq \sum_{q:\|x-q\|\leq h/2} n_q K\left(1/2\right) \geq K\left(1/2\right) n\mu_n(B(x,h/4)).
\end{aligned}$$

The last inequality follows by remarking that, since $\mathbf{Q}$ is an $\frac{h}{4}$-cover of $\mathbf{X}$, the ball $B(x,h/4)$ can only contain points from $\cup_{q:\|x-q\|\leq h/2}\mathbf{X}(q)$. Plug this last inequality into (6.3) and conclude. □

Lemma 47 (Bias at x)**.** *As before, fix $\mathbf{X}$, let $\mathbf{Q}$ be an $\frac{h}{4}$-cover of $\mathbf{X}$, and $0 < h < \Delta_{\mathcal{X}}$. Consider $x \in \mathcal{X}$ such that $\mathbf{X} \cap (B(x,h/4)) \neq \emptyset$. We have*

$$\left\| \widetilde{f}_{n,\mathbf{Q}}(x) - f(x) \right\|^2 \leq 2\lambda^2 h^2 + \frac{\Delta_{\mathcal{Y}}^2}{n}.$$

Proof. We have

$$\begin{aligned}\left\|\widetilde{f}_{n,\mathbf{Q}}(x) - f(x)\right\|^2 &= \left\|\sum_{q\in\mathbf{Q}} \frac{w_q(x)}{n_q} \sum_{X_i\in\mathbf{X}(q)} (f(X_i) - f(x))\right\|^2 \\ &\leq \sum_{q\in\mathbf{Q}} \frac{w_q(x)}{n_q} \sum_{X_i\in\mathbf{X}(q)} \|f(X_i) - f(x)\|^2,\end{aligned}$$

where we just applied Jensen's inequality on the norm square. We bound the r.h.s by breaking the summation over two subsets of $\mathbf{Q}$ as follows.

$$\begin{aligned}\sum_{q:\|x-q\|<h} \frac{w_q(x)}{n_q} \sum_{X_i\in\mathbf{X}(q)} \|f(X_i) - f(x)\|^2 &\leq \sum_{q:\|x-q\|<h} \frac{w_q(x)}{n_q} \sum_{X_i\in\mathbf{X}(q)} \lambda^2 \|X_i - x\|^2 \\ &\leq \sum_{q:\|x-q\|<h} \frac{w_q(x)}{n_q} \sum_{X_i\in\mathbf{X}(q)} \lambda^2 \left(\|x-q\| + \|q - X_i\|\right)^2 \\ &\leq \sum_{q:\|x-q\|<h} \frac{w_q(x)}{n_q} \sum_{X_i\in\mathbf{X}(q)} \frac{25}{16}\lambda^2 h^2 \leq 2\lambda^2 h^2.\end{aligned}$$

Next, we have

$$\begin{aligned}\sum_{q:\|x-q\|\geq h} \frac{w_q(x)}{n_q} \sum_{X_i\in\mathbf{X}(q)} \|f(X_i) - f(x)\|^2 &\leq \sum_{q:\|x-q\|\geq h} w_q(x)\Delta_{\mathcal{Y}}^2 \\ &= \frac{\Delta_{\mathcal{Y}}^2 \sum_{q:\|x-q\|\geq h} n_q\epsilon}{\displaystyle\sum_{q:\|x-q\|\geq h} n_q\epsilon + \sum_{q:\|x-q\|<h} n_q\left(K(x,q,h) + \epsilon\right)} \\ &= \Delta_{\mathcal{Y}}^2 \left(1 + \frac{\sum_{q:\|x-q\|<h} n_q\left(K(x,q,h)+\epsilon\right)}{\displaystyle\sum_{q:\|x-q\|\geq h} n_q\epsilon}\right)^{-1} \\ &\leq \Delta_{\mathcal{Y}}^2 \left(1 + \frac{K(1/2)}{\sum_{q:\|x-q\|\geq h} n_q\epsilon}\right)^{-1} \leq \Delta_{\mathcal{Y}}^2 \left(1 + \frac{K(1/2)}{n\epsilon}\right)^{-1} \leq \frac{\Delta_{\mathcal{Y}}^2}{1+n},\end{aligned}$$

where the second inequality is due to the fact that, since $\mu_n(B(x,h/4)) > 0$, the set $B(x,h/2) \cap \mathbf{Q}$ cannot be empty (remember that $\mathbf{Q}$ is an $\frac{h}{4}$-cover of $\mathbf{X}$). This concludes the argument. □

Proof of Lemma 44

Applying Fubini's theorem, the expected excess risk, $\mathbb{E}_{(\mathbf{X},\mathbf{Y})} \|f_{n,\mathbf{Q}} - f\|^2$, can be written as

$$\mathop{\mathbb{E}}_{X} \mathop{\mathbb{E}}_{(\mathbf{X},\mathbf{Y})} \|f_{n,\mathbf{Q}}(X) - f(X)\|^2 \left(\mathbb{1}\left[\mu_n(B(X,h/4)) > 0\right] + \mathbb{1}\left[\mu_n(B(X,h/4)) = 0\right]\right).$$

By lemmas 46 and 47 we have for $X = x$ fixed,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(\mathbf{X},\mathbf{Y})} \|f_{n,\mathbf{Q}}(x) - f(x)\|^2 \mathbb{1}\left[\mu_n(B(x,h/4)) > 0\right] \\
&\quad \leq C_1 \mathop{\mathbb{E}}_{\mathbf{X}} \left[\frac{\Delta_{\mathcal{Y}}^2 \mathbb{1}\left[\mu_n(B(x,h/4)) > 0\right]}{n\mu_n(B(x,h/4))}\right] + 2\lambda^2 h^2 + \frac{\Delta_{\mathcal{Y}}^2}{n} \\
&\quad \leq C_1 \left(\frac{2\Delta_{\mathcal{Y}}^2}{n\mu(B(x,h/4))}\right) + 2\lambda^2 h^2 + \frac{\Delta_{\mathcal{Y}}^2}{n}, \qquad (6.4)
\end{aligned}$$

where for the last inequality we used the fact that (see lemma 4.1 of [GKKW02]) for a binomial $b(n,p)$,

$$\mathbb{E}\left[\frac{\mathbb{1}\left[b(n,p) > 0\right]}{b(n,p)}\right] \leq \frac{2}{np}.$$

For the case where $B(x,h/4)$ is empty, we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(\mathbf{X},\mathbf{Y})} \|f_{n,\mathbf{Q}}(x) - f(x)\|^2 \mathbb{1}\left[\mu_n(B(x,h/4)) = 0\right] \leq \Delta_{\mathcal{Y}}^2 \mathop{\mathbb{E}}_{\mathbf{X}} \mathbb{1}\left[\mu_n(B(x,h/4)) = 0\right] \\
&\quad = \Delta_{\mathcal{Y}}^2 \left(1 - \mu(B(x,h/4)\right)^n \leq \Delta_{\mathcal{Y}}^2 e^{-n\mu(B(x,h/4))} \leq \frac{\Delta_{\mathcal{Y}}^2}{n\mu(B(x,h/4))}. \qquad (6.5)
\end{aligned}$$

Combining (6.5) and (6.4) into the excess risk as in equation (1.2), we get

$$\mathop{\mathbb{E}}_{(\mathbf{X},\mathbf{Y})} \|f_{n,\mathbf{Q}} - f\|^2 \leq \frac{3C_1\Delta_{\mathcal{Y}}^2}{n} \mathop{\mathbb{E}}_{X}\left[\frac{1}{\mu(B(X,h/4))}\right] + 2\lambda^2 h^2 + \frac{\Delta_{\mathcal{Y}}^2}{n}. \qquad (6.6)$$

The expectation on the r.h.s is bounded using a standard covering argument (see e.g. [GKKW02]). Let $\{z_i\}_1^N$ be an $\frac{h}{8}$-cover of $\mathcal{X}$. Notice that for any z_i, $x \in B(z_i, h/8)$ implies $B(x,h/4) \supset B(z_i,h/8)$. We therefore have

$$\begin{aligned}
\mathop{\mathbb{E}}_{X}\left[\frac{1}{\mu(B(X,h/4))}\right] &\leq \sum_{i=1}^N \mathop{\mathbb{E}}_{X}\left[\frac{\mathbb{1}\left[X \in B(z_i,h/8)\right]}{\mu(B(X,h/4))}\right] \\
&\leq \sum_{i=1}^N \mathop{\mathbb{E}}_{X}\left[\frac{\mathbb{1}\left[X \in B(z_i,h/8)\right]}{\mu(B(z_i,h/8))}\right] \\
&= N \leq C_2\left(\frac{\Delta_{\mathcal{X}}}{h}\right)^d, \text{ where } C_2 \text{ depends just on } d.
\end{aligned}$$

We conclude by combining the above with (6.6) to obtain

$$\mathop{\mathbb{E}}_{(\mathbf{X},\mathbf{Y})} \|f_{n,\mathbf{Q}} - f\|^2 \leq \frac{3C_1C_2\Delta_{\mathcal{Y}}^2}{n(h/\Delta_{\mathcal{X}})^d} + 2\lambda^2h^2 + \frac{\Delta_{\mathcal{Y}}^2}{n}.$$

□

6.4.2 Choosing a good h by empirical risk minimization

In this section, we analyze the following simple procedure for choosing a good h:

> Define $H \doteq \{\Delta_{\mathcal{X}}/2^i\}_0^{\lceil \log n \rceil}$. For every $h \in H$, pick the r-net $\mathbf{Q}_{h/4}$ over the sample $(\mathbf{X},\mathbf{Y})$, and let $f_{n,\mathbf{Q}_{h/4}}$ be as previously defined (equation 6.1). Draw a new sample $(\mathbf{X}',\mathbf{Y}')$ of size n. For every $h \in H$, test $f_{n,\mathbf{Q}_{h/4}}$ on $(\mathbf{X}',\mathbf{Y}')$; let the empirical risk be minimized at h_o, i.e. $h_o \doteq \text{argmin}_{h\in H} \frac{1}{n}\sum_{i=1}^n \left\| f_{n,\mathbf{Q}_{h/4}}(X_i') - Y_i' \right\|^2$.
> Return $f_{n,\mathbf{Q}_{h_o/4}}$ as the final regressor.

Corollary 48 (Follows from Lemma 44)**.** *Let* $n \geq \max\left(9, \left(\frac{\Delta_{\mathcal{Y}}}{\lambda\Delta_{\mathcal{X}}}\right)^2, \left(\frac{\lambda\Delta_{\mathcal{X}}}{\Delta_{\mathcal{Y}}}\right)^2\right)$. *The final regressor selected satisfies*

$$\mathbb{E}\left\| f_{n,\mathbf{Q}_{h_o/4}} - f\right\|^2 \leq C\left(\lambda\Delta_{\mathcal{X}}\right)^{2d/(2+d)}\left(\frac{\Delta_{\mathcal{Y}}^2}{n}\right)^{2/(2+d)} + 3\Delta_{\mathcal{Y}}^2\sqrt{\frac{\ln(n\log n)}{n}},$$

where C *depends on the Assouad dimension* d *and on* $K(0)/K(1/2)$.

Proof. Let $\tilde{h} = C_3\left(\Delta_{\mathcal{X}}^{d/(2+d)}\left(\frac{\Delta_{\mathcal{Y}}^2}{\lambda^2 n}\right)^{1/(2+d)}\right) \in H$. We note that n is lower bounded so that such an $\tilde{h}$ is in H. We have by Lemma 44 that for $\tilde{h}$,

$$\mathop{\mathbb{E}}_{\mathbf{X},\mathbf{Y}}\left\| f_{n,\mathbf{Q}_{\tilde{h}/4}} - f\right\|^2 \leq C_0\left(\lambda\Delta_{\mathcal{X}}\right)^{2d/(2+d)}\left(\frac{\Delta_{\mathcal{Y}}^2}{n}\right)^{2/(2+d)}.$$

Applying McDiarmid's to the empirical risk followed by a union bound over H, we have that, with probability at least $1 - 1/\sqrt{n}$ over the choice of $(\mathbf{X}',\mathbf{Y}')$, for all $h \in H$

$$\left| \mathop{\mathbb{E}}_{X,Y}\left\| f_{n,\mathbf{Q}_{h/4}}(X) - Y\right\|^2 - \frac{1}{n}\sum_{i=0}^n \left\| f_{n,\mathbf{Q}_{h/4}}(X_i') - Y_i'\right\| \right| \leq \Delta_{\mathcal{Y}}^2\sqrt{\frac{\ln(|H|\sqrt{n})}{n}}.$$

It follows that

$$\mathop{\mathbb{E}}_{X,Y}\left\|f_{n,\mathbf{Q}_{h_o/4}}(X)-Y\right\|^2 \leq \mathop{\mathbb{E}}_{X,Y}\left\|f_{n,\mathbf{Q}_{\tilde{h}/4}}(X)-Y\right\|^2 + 2\Delta_{\mathcal{Y}}^2\sqrt{\frac{\ln(|H|\sqrt{n})}{n}},$$

which by (1.1) implies

$$\left\|f_{n,\mathbf{Q}_{h_o/4}}-f\right\|^2 \leq \left\|f_{n,\mathbf{Q}_{\tilde{h}/4}}-f\right\|^2 + 2\Delta_{\mathcal{Y}}^2\sqrt{\frac{\ln(|H|\sqrt{n})}{n}}.$$

Take the expectation (given the randomness in the two samples) over this last inequality and conclude. □

6.5 A fast implementation of r-nets-hybrids

In this section we show how to modify the cover-tree procedure of [BKL06] to enable fast evaluation of $f_{n,\mathbf{Q}_{h/4}}$ for any $h \in H \doteq \{\Delta_{\mathcal{X}}/2^i\}_1^I$, $I = \lceil \log n \rceil$.

The cover-tree performs proximity search by navigating a hierarchy of nested r-nets of $\mathbf{X}$. The navigating-nets of [KL04] implement the same basic idea. They require additional book-keeping to enable range queries of the form $\mathbf{X}\cap B(x,h)$, for a query point x. Here we need to perform range searches of the form $\mathbf{Q}_{h/4}\cap B(x,h)$ and our book-keeping will therefore be different from [KL04]. Note that, for each h and $\mathbf{Q}_{h/4}$, one could use a generic range search procedure such as [KL04] with the data in $\mathbf{Q}_{h/4}$ as input, but this requires building a separate data structure for each h, which is expensive. We use a single data structure.

6.5.1 The hierarchy of nets

Consider an ordering $\left\{X_{(i)}\right\}_1^n$ of the data points obtained as follows: $X_{(1)}$ and $X_{(2)}$ are the farthest points in $\mathbf{X}$; inductively for $2 < i < n$, $X_{(i)}$ in $\mathbf{X}$ is the farthest point from $\left\{X_{(1)},\ldots,X_{(i-1)}\right\}$, where the distance to a set is defined as the minimum distance to a point in the set. defined as the minimum distance between points in the sets. In other words, $\left\{X_{(i)}\right\}_1^n$ is built by starting with the farthest two points in $\mathbf{X}$, and inductively picking the farthest point from the current sequence.

For $r \in \{\Delta_{\mathcal{X}}/2^i\}_0^{I+2}$, define $\mathbf{Q}_r = \left\{X_{(1)},\ldots,X_{(i)}\right\}$ where $i \geq 1$ is the highest index such that $\left\|X_{(i)} - \left\{X_{(1)},\ldots,X_{(i-1)}\right\}\right\| \geq r$. Notice that, by construction, $\mathbf{Q}_r$ is an r-net of $\mathbf{X}$.

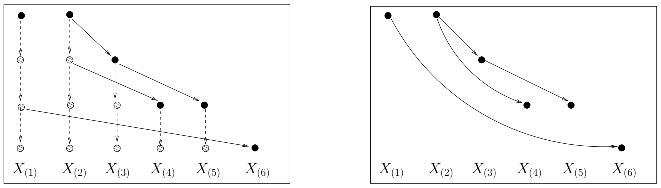


Figure 6.5: The r-nets (rows of left subfigure) are implicit to an ordering of the data. They define a parent-child relationship implemented by the *neighborhood graph* (right), the structure traversed for fast evaluation.

6.5.2 Data structure

The data structure consists of an acyclic directed graph, and *range* sets defined below.

Neighborhood graph: The nodes of the graph are the $\left\{X_{(i)}\right\}_1^n$, and the edges are given by the following parent-child relationship: starting at $r = \Delta_{\mathcal{X}}/2$, the parent of each node in $\mathbf{Q}_r \setminus \mathbf{Q}_{2r}$ is the point it is closest to in $\mathbf{Q}_{2r}$. The graph is implemented by maintaining an ordered list of children for each node, where the order is given by the children's appearance in the sequence $\left\{X_{(i)}\right\}_1^n$. These relationships are depicted in Figure 6.5.

These ordered lists of children are used to implement the `nextChildren` operation defined iteratively as follows. Given $\mathbf{Q} \subset \left\{X_{(i)}\right\}_1^n$, let *visited* children denote any child of $q \in \mathbf{Q}$ that a previous call to `nextChildren` has already returned. The call `nextChildren` ($\mathbf{Q}$) returns children of $q \in \mathbf{Q}$ that have not yet been visited, starting with the unvisited child with lowest index in $\left\{X_{(i)}\right\}_1^n$, say $X_{(i)}$, and returning all unvisited children in $\mathbf{Q}_r$, the first net containing $X_{(i)}$, i.e. $X_{(i)} \in \mathbf{Q}_r \setminus \mathbf{Q}_{2r}$; r is also returned. The children returned are then marked off as visited. The time complexity of this routine is just the number of children returned.

Range sets: For each node $X_{(i)}$ and each $r \in \{\Delta_{\mathcal{X}}/2^i\}_0^\infty$, we maintain a set of neighbors of $X_{(i)}$ in $\mathbf{Q}_r$ defined as $\mathbf{R}_{(i),r} \doteq \left\{q \in \mathbf{Q}_r : \left\|X_{(i)} - q\right\| \leq 8r\right\}$.

```
Procedure evaluate($x$, $h$)
$\mathbf{Q} \leftarrow \mathbf{Q}_{\Delta_{\mathcal{X}}}$.
repeat
    $(\mathbf{Q}', r) \leftarrow$ nextChildren $(\mathbf{Q})$.
    $\mathbf{Q}'' \leftarrow \mathbf{Q} \cup \mathbf{Q}'$.
    if $r < h/4$ or $\mathbf{Q}' = \emptyset$ then // We reached past $\mathbf{Q}_{h/4}$.
        $X_{(i)} \leftarrow \operatorname{argmin}_{q \in \mathbf{Q}} \|x - q\|$; // Closest point to $x$ in $\mathbf{Q}_{h/4}$.
        $\mathbf{Q} \leftarrow \mathbf{R}_{(i),h/4} \cap B(x, h)$; // Search in a range of $2h$ around $X_{(i)}$.
        Break loop.
    if $\|x - \mathbf{Q}''\| \geq h + 2r$ then // The set $\mathbf{Q}_{h/4} \cap B(x, h)$ is empty.
        $\mathbf{Q} \leftarrow \emptyset$.
        Break loop.
    $\mathbf{Q} \leftarrow \{q \in \mathbf{Q}'', \|x - q\| < \|x - \mathbf{Q}''\| + 2r\}$.
until ... ;
//At this point $\mathbf{Q} = \mathbf{Q}_{h/4} \cap B(x, h)$.
Define $\bar{Y} \doteq \frac{1}{n} \sum_{y_i \in \mathbf{Y}} y_i$.
return
    $f_{n,\mathbf{Q}_{h/4}}(x) \leftarrow \frac{\sum_{q \in \mathbf{Q}} n_q(K(x, q, h))\bar{Y}_q + \epsilon n \bar{Y}}{\sum_{q \in \mathbf{Q}} n_q(K(x, q, h)) + \epsilon n}$;
```

6.5.3 Evaluation

The evaluation procedure consists of quickly identifying the closest point $X_{(i)}$ to x in $\mathbf{Q}_{h/4}$ and then searching in the range of $X_{(i)}$ for the points in $\mathbf{Q}_{h/4} \cap B(x,h)$. The identification of $X_{(i)}$ is done by going down the levels of nested nets, and discarding those points (and their descendants) that are certain to be farther to x than $X_{(i)}$ (we will argue that "$\|x - \mathbf{Q}''\| + 2r$" is an upper-bound on $\left\|x - X_{(i)}\right\|$). Also, if x is far enough from all points at the current level (second if-clause), we can safely stop early because $B(x,h)$ is unlikely to contain points from $\mathbf{Q}_{h/4}$ (we'll see that points in $\mathbf{Q}_{h/4}$ are all within $2r$ of their ancestor at the current level).

Lemma 49. *The call to procedure* `evaluate` *(x,h) correctly evaluates* $f_{n,\mathbf{Q}_{h/4}}(x)$ *and has time complexity* $C \log\left(\Delta_{\mathcal{X}}/h\right) + \log n$ *where* C *is at most* 2^{8d}*.*

Proof. We first show that the algorithm correctly returns $f_{n,\mathbf{Q}_{\alpha h}}(x)$, and we then argue its run time.

Correctness of `evaluate`*.* The procedure works by first finding the closest point to x in $\mathbf{Q}_{h/4}$, say $X_{(i)}$, followed by the identification of all nodes in $\left(\mathbf{R}_{(i),h/4} \cap B(x,h)\right) = \left(\mathbf{Q}_{h/4} \cap B(x,h)\right)$ (see the first if-clause). We just have to show that this closest point $X_{(i)}$ is correctly identified.

We'll argue the following loop invariant $\mathcal{I}$: at the beginning of the loop, $X_{(i)}$ is either in $\mathbf{Q}'' = \mathbf{Q} \cup \mathbf{Q}'$ or is a descendant of a node in $\mathbf{Q}'$. Let's consider some iteration where $\mathcal{I}$ holds (it certainly does in the first iteration).

If the first if-clause is entered, then $\mathbf{Q}$ is contained in $\mathbf{Q}_{h/4}$ but $\mathbf{Q}'$ is not, so $X_{(i)}$ must be in $\mathbf{Q}$ and we correctly return.

Suppose the first if-clause is not entered. Now let $X_{(j)}$ be the ancestor in $\mathbf{Q}'$ of $X_{(i)}$ or let it be $X_{(i)}$ itself if it's in $\mathbf{Q}''$. Let r be as defined in `evaluate`, we have $\left\|X_{(i)} - X_{(j)}\right\| < \sum_{k=0}^{\infty} r/2^k = 2r$ by going down the parent-child relations. It follows that

$$\|x - \mathbf{Q}''\| \leq \left\|x - X_{(j)}\right\| \leq \left\|x - X_{(i)}\right\| + \left\|X_{(i)} - X_{(j)}\right\| < \left\|x - X_{(i)}\right\| + 2r.$$

In other words, we have $\left\|x - X_{(i)}\right\| > \|x - \mathbf{Q}''\| - 2r$. Thus, if the second if-clause is entered, we necessarily have $\left\|x - X_{(i)}\right\| > h$, i.e. $B(x, h) \cap \mathbf{Q}_{h/4} = \emptyset$ and we correctly return.

Now assume none of the if-clauses is entered. Let $X_{(j)} \in \mathbf{Q}''$ be any of the points removed from $\mathbf{Q}''$ to obtain the next $\mathbf{Q}$. Let $X_{(k)}$ be a child of $X_{(j)}$ that has not yet been visited, or a descendant of such a child. If neither such $X_{(j)}$ or $X_{(k)}$ is $X_{(i)}$ then, by definition, $\mathcal{I}$ must hold at the next iteration. We sure have $X_{(j)} \neq X_{(i)}$ since $\left\|x - X_{(j)}\right\| \geq \|x - \mathbf{Q}''\| + 2r \geq \left\|x - X_{(i)}\right\| + 2r$. Now notice that, by the same argument as above, $\left\|X_{(j)} - X_{(k)}\right\| < \sum_{k=0}^{\infty} r/2^k = 2r$. We thus have $\left\|x - X_{(k)}\right\| > \left\|x - X_{(j)}\right\| - 2r \geq \left\|x - X_{(i)}\right\|$ so we know $X_{(j)} \neq X_{(i)}$.

Runtime of `evaluate`. Starting from $\mathbf{Q}_{\Delta_{\mathcal{X}}}$, a different net $\mathbf{Q}_r$ is reached at every iteration, and the loop stops when we reach past $\mathbf{Q}_{h/4}$. Therefore the loop is entered at most $\log\left(\Delta_{\mathcal{X}}/h/4\right)$ times. In each iteration, most of the work is done parsing through $\mathbf{Q}''$, besides time spent for the range search in the last iteration. So the total runtime is $O\left(\log\left(\Delta_{\mathcal{X}}/h/4\right) \cdot \max|\mathbf{Q}''|\right) +$ range search time. We just need to bound $\max|\mathbf{Q}''| \leq \max|\mathbf{Q}| + \max|\mathbf{Q}'|$ and the range search time.

The following fact (see e.g. Lemma 4.1 of [Cla05]) will come in handy: consider r_1 and r_2 such that r_1/r_2 is a power of 2, and let $B \subset \mathcal{X}$ be a ball of radius r_1; since $\mathcal{X}$ has Assouad dimension d, the smallest r_2-cover of B is of size at most $(r_1/r_2)^d$, and the largest r_2-packing of B is of size at most $(r_1/r_2)^{2d}$. This is true for any metric space, and therefore holds for $\mathbf{X}$ which is of Assouad dimension at most d by inclusion in $\mathcal{X}$.

Let $\mathbf{Q}' \subset \mathbf{Q}_r$ so that $\mathbf{Q} \subset \mathbf{Q}_{2r}$ at the beginning of some iteration. Let $q \in \mathbf{Q}$, the children of q in $\mathbf{Q}'$ are not in $\mathbf{Q}_{2r}$ and therefore are all within $2r$ of $\mathbf{Q}$; since these children an r-packing of $B(q, 2r)$, there are at most 2^{2d} of them. Thus, $\max|\mathbf{Q}'| \leq 2^{2d} \max|\mathbf{Q}|$.

Initially $\mathbf{Q} = \mathbf{Q}_{\Delta_{\mathcal{X}}}$ so we have $|\mathbf{Q}| \leq 2^{2d}$ since $\mathbf{Q}_{\Delta_{\mathcal{X}}}$ is a $\Delta_{\mathcal{X}}$-packing of $\mathbf{X} \subset B\left(X_{(1)}, 2\Delta_{\mathcal{X}}\right)$. At the end of each iteration we have $\mathbf{Q} \subset B(x, \|x - \mathbf{Q}''\| + 2r)$. Now $\|x - \mathbf{Q}''\| \leq h + 2r \leq 4r + 2r$ since the if-clauses were not entered if we got to the end of the iteration. Thus, $\mathbf{Q}$ is an r-packing of $B(x, 8r)$, and therefore $\max|\mathbf{Q}| \leq 2^{8d}$.

To finish, the range search around $X_{(i)}$ takes time $\left|\mathbf{R}_{(i),h/4}\right| \leq 2^{8d}$ since

$\mathbf{R}_{(i),h/4}$ is an $\frac{h}{4}$-packing of $B\left(X_{(i)}, 2h\right)$. □

Appendix A

A.1 On the adaptivity of an axis-parallel splitting rule

In this section we show that if the input space $\mathcal{X}$ is a subset of $[-1,1]^D$ of Assouad dimension d, then a dyadic tree regressor (Figure 4.1(a)) achieves a convergence rate of the form $O(n^{-2/(2+d)})$, but with a leading constant that is exponential in D.

The dyadic tree starts with a single cell corresponding to all of $[-1,1]^D$, and then grows one level at a time. In each such expansion, a particular coordinate direction is chosen and every current leaf cell is bisected at its midpoint along that coordinate. There is flexibility in how the coordinate direction is chosen; a common choice is to simply cycle through the D coordinates. The final level of the tree defines a partition $\mathcal{A}$ of $[-1,1]^D$, and a regressor $f_{n,\mathcal{A}}$ is obtained by averaging the Y values in each cell $A \in \mathcal{A}$.

Unlike an RP tree, the dyadic tree is not data-dependent. In such cases, a generic risk bound applies. If the cells of $\mathcal{A}$ have diameter $\leq \zeta$, and if $\mathcal{A}_{\mathcal{X}}$ is the subset of cells intersecting $\mathcal{X}$, then it is implicit, for instance, in the proof of Theorem 4.3 of [GKKW02], that

$$\mathbb{E}\,\|f_{n,\mathcal{A}} - f\|^2 \leq C\left(\Delta_{\mathcal{Y}}^2 \frac{|\mathcal{A}_{\mathcal{X}}|}{n} + \lambda^2\zeta^2\right). \tag{A.1}$$

The result in this section is obtained by noticing that most cells of $\mathcal{A}$ will be empty if $\mathcal{X}$ has Assouad dimension much smaller than D. Think for instance of $\mathcal{X}$ as a line curving slowly through the cube $[-1,1]^D$.

Theorem 50. *There are absolute constants C_1, C_2, and C_3 for which the following holds. Consider an input space $\mathcal{X} \subset [-1,1]^D$ of diameter 1 and Assouad dimension d. Let $\mathcal{A}$ be a dyadic partition where each cell has diameter $\zeta < 1$, that is, cells have side lengths $\zeta/\sqrt{D}$. If $\zeta = C_1 \left(\Delta_{\mathcal{Y}}^2 \cdot 2^{C_3 D \log D}/(\lambda^2 n)\right)^{1/(2+d)}$, we have*

$$\mathbb{E}\,\|f_{n,\mathcal{A}} - f\|^2 \leq C_2 \lambda^{2d/(2+d)} \left(\frac{\Delta_{\mathcal{Y}}^2 \cdot 2^{C_3 D \log D}}{n}\right)^{2/(2+d)}.$$

Proof. Let $\mathcal{A}_{\mathcal{X}} \subset \mathcal{A}$ be the cells of $\mathcal{A}$ that intersect $\mathcal{X}$. We'll first show that $|\mathcal{A}_{\mathcal{X}}| \leq 2^{O(D \log D)}(1/\zeta)^d$. By the Assouad assumption, $\mathcal{X}$ has a $(\zeta/2)$-cover of size $N \leq (2/\zeta)^d$; call it $\{z_i\}_1^N \subset \mathcal{X}$. Now consider the (closed) balls $B(z_i, \zeta)$. By a triangle inequality, the center of each hypercube $A \in \mathcal{A}_{\mathcal{X}}$ is contained in some ball $B(z_i, \zeta)$ (the center of each A is within $\zeta/2$ of all $x \in A \cap \mathcal{X}$ and each such x is within $\zeta/2$ of some z_i). Therefore, if M is the maximum number of such centers in a single ball $B(z_i, \zeta)$, then $|\mathcal{A}_{\mathcal{X}}| \leq M \cdot N$.

To bound M, notice that the centers of the hypercubes $A \in \mathcal{A}_{\mathcal{X}}$ are at least $\zeta/\sqrt{D}$ away from each other. In other words, the centers contained in any $B(z_i, \zeta)$ form a $(\zeta/\sqrt{D})$-packing of it. By a standard duality, any r-packing of a space is of size at most that of the minimum $(r/2)$-cover of the space. In this case the ball $B(z_i, \zeta) \subset \mathbb{R}^D$ has a minimum $(\zeta/2\sqrt{D})$-cover of size at most $(2\sqrt{D})^{c_o D}$ (recall discussion of Chapter 2) that $\mathbb{R}^D$ has Assouad dimension $\leq c_o D$ for some constant $c_o < 3$).

Thus $|\mathcal{A}_{\mathcal{X}}| \leq M \cdot N \leq 2^{C_3 D \log D}(1/\zeta)^d$ (for some constant C_3) and we conclude by plugging this value into (A.1). □

A.2 A more general setting

Finally, we consider a more general setting where the space $\mathcal{X} \subset \mathbb{R}^D$ has low Assouad dimension $d \ll D$ only in sufficiently small neighborhoods (as in Figure 5.2). In this case, an RP tree might initially decrease diameter slowly; but when its cells are small enough, further splits will rapidly decrease diameter. We will show that the higher dimensionality of large regions of space do not tremendously affect the final excess risk, provided n is large enough for the tree to arrive at well populated regions of sufficiently small diameter.

A.2.1 Result for the general case

The next definition of decrease rate is made more general by allowing for a good rate k to be attained only later down the tree; in other words we allow for speedups to occur only in smaller regions of $\mathcal{X}$, of diameter at most $2r < \Delta_{\mathcal{X}}$. The algorithm remains unchanged except that we now need $\alpha(n) \geq \left(\log^2 n\right) \log\log(n/\delta) + \log(\mathcal{N}_r/\delta)$, where $\mathcal{N}_r$ is the size of a minimal r-cover of $\mathcal{X}$. Note that $\mathcal{N}_r \leq (\Delta_{\mathcal{X}}/r)^{O(D)}$.

Definition 51. *Given a sample* $\mathbf{X}$*, we say that* `adaptiveRPtree` *attains a* ***diameter decrease rate*** *of* (k, γ) *on* $\mathbf{X}$*, for* $k \geq d$ *and* $\gamma \leq \frac{n}{\alpha(n)}$*, if the following holds:* `adaptiveRPtree` *arrives at an intermediate partition* $\mathcal{A}^{i_\gamma}$*,* $|\mathcal{A}^{i_\gamma}| = \gamma$*, such that any subsequent call to* `coreRPtree`$(A, \Delta_n(A)/2, \delta)$ *over cells* A *with ancestor in* $\mathcal{A}^{i_\gamma}$*, returns a tree rooted at* A *of height at most* k*.*

Theorem 52. *Assume that for every ball* $B \in \mathbb{R}^D$ *of radius* r*,* $B \cap \mathcal{X}$ *has Assouad dimension* d*. There exist constants* C*,* C' *independent of* d *and* $\mu(\mathcal{X})$*, and* $C'' = C''(\mu(\mathcal{X}), r)$ *such that the following holds.*

Suppose the algorithm uses the cross-validation option with a setting of $\alpha(n) \geq \left(\log^2 n\right)\log\log(n/\delta) + \log(\mathcal{N}_r/\delta)$*. Assume* $n \geq \max\left\{(\lambda\Delta_{\mathcal{X}}/\Delta_{\mathcal{Y}})^2, C''\alpha(n)\right\}$*. With probability at least* $1-\delta$*, the algorithm attains a diameter decrease rate of* (k, γ) *where* $k \leq C' d \log d$ *and* $\gamma \leq C''$*, and the excess risk of the regressor satisfies*

$$\|f_n - f\|^2 \leq C \cdot (\lambda\Delta_{\mathcal{X}})^{2k/(2+k)} \left(\frac{\Delta_{\mathcal{Y}}^2 \cdot \gamma \cdot \alpha(n)}{n}\right)^{2/(2+k)} + 2\Delta_{\mathcal{Y}}^2 \sqrt{\frac{\ln\log n^6 + \ln 1/\delta}{2n}}.$$

A.2.2 Proof of theorem 52

The proof of theorem 52 closely mirrors that of theorem 32. We'll therefore only show the key lemmas whose statement change. We assume in what follows that the cross-validation option is used.

The proof proceeds also by first bounding the risks in terms of the observed diameter decrease rate (lemma 56 of section A.2.2), and then bounding the worst case decrease rates (lemma 58 of section A.2.2).

Risk bound in terms of observed diameter decrease rate

Lemma 53 (Mass of cells of $\mathcal{A}'$)**.** *With probability at least $1-\delta'$ over $\mathcal{X}$ and the randomness in the algorithm, we have for all partitions $\mathcal{A} = \mathcal{A}^0, \mathcal{A}^1, \ldots$ found by* `adaptiveRPtree`*, for all $A' \in \mathcal{A}'$ that*

$$\begin{aligned} \mu(A') &\leq \mu_n(A') + 2\sqrt{\mu_n(A')\frac{\mathcal{V}+\ln(4/\delta')}{n}} + 4\frac{\mathcal{V}+\ln(4/\delta')}{n}, \quad \textit{where} \\ \mathcal{V} &\leq O\left(\log n(\log n + \operatorname{loglog}(1/\delta)) + \log \mathcal{N}_r\right). \end{aligned} \tag{A.2}$$

Proof. Follow the outline of lemma 38, the only difference being that the bound on $|\mathcal{B}|$ introduces the term $\mathcal{N}_r$. □

Lemma 54 (Excess risk)**.** *There exists a constant C_1 independent of d and $\mu(\mathcal{X})$ such that the following holds with probability at least $1-\delta/3$ over the choice of $(\mathbf{X}, \mathbf{Y})$ and the randomness in the algorithm.*

Let $\alpha(n) \geq \left(\log^2 n\right) \log\log(n/\delta) + \log(\mathcal{N}_r/\delta)$. Let $\mathcal{A}^i$ be the final partition reached by `adaptiveRPtree`*. For all partitions $\mathcal{A} \in \{\mathcal{A}^j\}_{j=0}^{i}$, we have*

$$\left\|f_{n,\mathcal{A}} - f\right\|^2 \leq C_1 \left(\Delta_{\mathcal{Y}}^2 \left|\mathcal{A}\right| \frac{\alpha(n)}{n} + \lambda^2 \left(\Delta_n^2\left(\mathcal{A}\right) + n^{-4/(2+d)}\Delta_{\mathcal{X}}^2\right)\right).$$

Proof. The proof is identical to that of lemma 39, using lemma 53 in place of lemma 38. □

Lemma 55 (Existence of a good pruning)**.** *Suppose the cross-validation option is used, and* `adaptiveRPtree` *attains a diameter decrease rate of (k, γ) on $\mathbf{X}$. Let $\alpha(n) \geq \left(\log^2 n\right)\log\log(n/\delta) + \log(\mathcal{N}_r/\delta)$, and $\zeta \doteq \left(\frac{\Delta_{\mathcal{Y}}^2\cdot\gamma\cdot\alpha(n)}{\lambda^2\Delta_{\mathcal{X}}^2\cdot n}\right)^{1/(2+k)}$. Finally, assume*

$n \geq \max\left\{\left(\lambda\Delta_{\mathcal{X}}/\Delta_{\mathcal{Y}}\right)^2, \gamma\cdot\alpha(n)\right\}$. Then there exists an RPtree partition $\mathcal{A}$ such that $|\mathcal{A}| \leq \gamma\cdot\zeta^{-k}$ and $\Delta_n\left(\mathcal{A}\right) \leq 2\zeta\cdot\Delta_n\left(\mathcal{X}\right)$.

Proof. Follow the outline of lemma 40, while noticing that now we have for all $i \geq 1$, $\text{level}\left(\mathcal{A}^i\right) \leq ki + \log\gamma$ and $\Delta_n\left(A^i\right) \leq 2^{-i}\Delta_n\left(\mathcal{X}\right)$. □

Lemma 56. *There exists a constant C independent of d and $\mu(\mathcal{X})$ such that the following holds with probability at least $1 - 2\delta/3$ over $(\mathbf{X}, \mathbf{Y})$ and the randomness in the algorithm.*

Suppose the cross-validation option is used, and `adaptiveRPtree` *attains a diameter decrease rate of (k, γ) on $\mathbf{X}$. Let $\alpha(n) \geq \left(\log^2 n\right)\log\log(n/\delta) + \log(\mathcal{N}_r/\delta)$, and assume $n \geq \max\left\{\left(\lambda\Delta_{\mathcal{X}}/\Delta_{\mathcal{Y}}\right)^2, \gamma \cdot \alpha(n)\right\}$. The excess risk of the regressor is then bounded as*

$$\|f_n - f\|^2 \leq C \cdot \left(\lambda\Delta_{\mathcal{X}}\right)^{2k/(2+k)} \left(\frac{\Delta_{\mathcal{Y}}^2 \cdot \gamma \cdot \alpha(n)}{n}\right)^{2/(2+k)} + 2\Delta_{\mathcal{Y}}^2\sqrt{\frac{\ln\log n^6 + \ln 1/\delta}{2n}}.$$

Proof. Follow the outline of lemma 41. □

Worst case decrease rates

Lemma 57. *Assume that for every ball $B \in \mathbb{R}^D$ of radius r, $B \cap \mathcal{X}$ has Assouad dimension d and consider the tree built by* `adaptiveRPtree`*. There exists a constant $C'' = C''(\mu(\mathcal{X}), r)$, such that with probability at least $1 - \delta/3$ over the randomness in the algorithm, we have $\Delta_n(A) \leq r$ for all cells A of the tree at level at least $\log C''$.*

Proof outline. This is a consequence of the fact that $\mathcal{X}$ has finite Assouad dimension at most $O(D)$. By theorem 28 and the fact that `basicRPtree` is called multiple times to boost the probability of obtaining a small tree (see proof of Lemma 31) we have the following: with probability at least $1 - \delta/3$, and independently of the distribution, it takes at most a constant number of levels to get the data diameter within the cells below r.

The number of levels needed for each particular distribution is therefore just a constant. □

Lemma 58. *Assume that for every ball $B \in \mathbb{R}^D$ of radius r, $B \cap \mathcal{X}$ has Assouad dimension d. There exist constants C independent of $\mathcal{X}$ and d, and $C'' =$*

$C''(\mu(\mathcal{X}), r)$, such that with probability at least $1 - \delta/3$, the algorithm attains a diameter decrease rate of (k, γ) where $k \leq C' d \log d$ and $\gamma \leq C''$.

Proof. This results from Lemma 57 and Theorem 28. □

Most of the appendix appear in:

– S. Kpotufe, S. Dasgupta, "A tree-based regressor that adapts to intrinsic dimension", Journal of Computer and System Sciences, Special Issue on Learning Theory, (Invited Submission).

Bibliography

[AB98] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *Symposium on Computational Geometry*, 1998.

[AMS97] C. Atkeson, A. Moore, and S. Schaal. Locally weighted learning. *AI Review*, 1997.

[BKL06] A. Beygelzimer, S. Kakade, and J. Langford. Cover trees for nearest neighbors. *ICML*, 2006.

[BL06] P. Bickel and B. Li. Local polynomial regression on unknown manifolds. *Tech. Re. Dep. of Stats. UC Berkley*, 2006.

[BN03] M. Belkin and N. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2003.

[BPV$^+$03] Y. Bengio, J. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In *NIPS*, 2003.

[Bre96] Leo Breiman. Heuristics of instability and stabilization in model selection. *The Annals of Statistics*, 24(6):2350–2383, 1996.

[BW07] R. Baraniuk and M. Wakin. Random projections of smooth manifolds. *Foundations of Computational Mathematics*, 2007.

[CA91] R. Cao-Abad. Rate of convergence for the wild bootstrap in nonparametric regression. *Annals of Statistics*, 19:22262231, 1991.

[Cla05] K. Clarkson. Nearest-neighbor searching and metric space dimensions. *Nearest-Neighbor Methods for Learning and Vision: Theory and Practice*, 2005.

[Cla07] K. Clarkson. Tighter bounds for random projections of manifolds. *Comp. Geometry*, 2007.

[DF08a] S. Dasgupta and Y. Freund. Random projection trees and low dimensional manifolds. *STOC*, 2008.

[DF08b] S. Dasgupta and Y. Freund. Random projection trees and low dimensional manifolds. *ACM Symposium on Theory of Computing*, 2008.

[DGL96a] L. Devroye, L. Gyorfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

[DGL96b] L. Devroye, L. Gyorfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

[DHS01] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley and Sons, 2001.

[DvdL05] S. Dudoit and M. van der Laan. Asymptotics of cross-validated risk estimation in estimator selection and performance assessment. *Statistical Methodologies*, 2:131154, 2005.

[GKKW02] L. Gyorfi, M. Kohler, A. Krzyzak, and H. Walk. *A Distribution Free Theory of Nonparametric Regression*. Springer, New York, NY, 2002.

[GLZ08] A. B. Goldberg, M. Li, and X. Zhu. Online manifold regularization: a new learning setting and empirical study. *ECML PKDD*, 2008.

[GN05] S. Gey and E. Nedelec. Model selection for cart regression trees. *IEEE Transactions on Information Theory*, 51, 2005.

[IN07] P. Indyk and A. Naor. Nearest neighbor preserving embeddings. *ACM Transactions on Algorithms*, 3(3), 2007.

[Kad02] M.W. Kadous. *Temporal Classification: Extending the Classification Paradigm to Multivariate Time Series*. PhD thesis, School of Comp. Sci. and Engg., University of New South Wales, 2002.

[KL04] R. Krauthgamer and J. Lee. Navigating nets: Simple algorithms for proximity search. *SODA*, 2004.

[KP95] S. Kulkarni and S. Posner. Rates of convergence of nearest neighbor estimation under arbitrary sampling. *IEEE Transactions on Information Theory*, 41, 1995.

[Kpo09] S. Kpotufe. Escaping the curse of dimensionality with a tree-based regressor. *COLT*, 2009.

[LG08] D. Lee and A. Grey. Fast high-dimensional kernel summations using the monte carlo multipole method. *NIPS*, 2008.

[LW07] J. Lafferty and L. Wasserman. Statistical analysis of semi-supervised regression. *NIPS*, 2007.

[ML09] B. McFee and G. Lanckriet. UCSD multiple kernel repository, 2009.

[NSW06] P. Niyogi, S. Smale, and S. Weinberger. Finding the homology of submanifolds with high confidence from random samples. *Disc. Computational Geometry*, 2006.

[RS00] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[RW06] C. Rasmussen and C. Williams. Gaussian processes for machine learning - SARCOS dataset, 2006.

[SN06a] C. Scott and R.D. Nowak. Minimax-optimal classification with dyadic decision trees. *IEEE Transactions on Information Theory*, 52, 2006.

[SN06b] C. Scott and R.D. Nowark. Minimax-optimal classification with dyadic decision trees. *IEEE Transactions on Information Theory*, 52, 2006.

[Sta89] J. G. Staniswalis. Local bandwidth selection for kernel estimates. *Journal of the American Statistical Association*, 84:284–288, 1989.

[Sto80] C. J. Stone. Optimal rates of convergence for non-parametric estimators. *Ann. Statist.*, 8:1348–1360, 1980.

[Sto82] C. J. Stone. Optimal global rates of convergence for non-parametric estimators. *Ann. Statist.*, 10:1340–1353, 1982.

[Tor97] Luís Torgo. Kernel regression trees. In *ECML*, 1997.

[TSL00] J.B. TenenBaum, V. De Silva, and J. Langford. A global geometric framework for non-linear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[VC71] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their expectation. *Theory of probability and its applications*, 16:264–280, 1971.

[VKD09] N.A. Verma, S. Kpotufe, and S. Dasgupta. Which spatial partition trees are adaptive to intrinsic dimension? *Uncertainty in Artificial Intelligence*, 2009.

[VS00] S. Vijayakumar and S. Schaal. Locally weighted projection regression: An O(n) algorithm for incremental real time learning in high dimensional space. In *ICML*, 2000.