UNIVERSITY OF CALIFORNIA SAN DIEGO

# A Class of kNN-Type Entropy Estimators: Algorithm, Convergence, and Application to Molecular Modeling

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

in

Mathematics

by

Chao Fan

Committee in charge:

Professor Bo Li, Chair
Professor Li-Tien Cheng
Professor Dimitris N. Politis
Professor Zhuowen Tu
Professor Tianyi Zheng

2021

The dissertation of Chao Fan is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2021

TABLE OF CONTENTS


Dissertation Approval Page . . . . . iii

Table of Contents . . . . . iv

List of Figures . . . . . vi

List of Tables . . . . . vii

Acknowledgements . . . . . viii

Vita . . . . . ix

Abstract of the Dissertation . . . . . x

Chapter 1 Introduction . . . . . 1
1.1 The Shannon entropy and its kNN-type estimators . . . . . 2
1.2 Entropy of a particle system . . . . . 4
1.3 Main results and organization of this thesis . . . . . 6

Chapter 2 The kNN, kpN and kp-kernel Entropy Estimators . . . . . 8
2.1 Entropy: Definition and example . . . . . 8
2.2 The $k$th nearest distance and entropy estimators . . . . . 10
2.2.1 The $k$th nearest distance . . . . . 10
2.2.2 A general framework of kNN-type entropy estimators . . 13
2.3 The kNN, kpN and kp-kernel estimators: Definition and algorithm 16
2.3.1 The kNN estimator . . . . . 16
2.3.2 The kpN estimator . . . . . 18
2.3.3 The kp-kernel estimator . . . . . 22
2.4 Simulation results . . . . . 23
2.4.1 A Gaussian distribution . . . . . 24
2.4.2 A beta distribution . . . . . 30
2.4.3 A parabola distribution . . . . . 32
2.4.4 Efficiency . . . . . 34
2.5 Convergence analysis . . . . . 36

Chapter 3 Entropy Estimation for a Particle System . . . . . 46
3.1 Entropy for a particle system . . . . . 46
3.2 Monte Carlo Simulations . . . . . 51
3.2.1 Simulation results of a Lennard-Jones system . . . . . 52
3.2.2 Simulation results of an ionic system . . . . . 53
3.3 Numerical entropy calculation of particle systems . . . . . 55

Chapter 4 Additional Results: Molecular Dynamics Simulations of Hydration of a Single Ion . . . . . 59
4.1 Molecular dynamics simulations . . . . . 60
4.2 Implicit-solvent modeling . . . . . 63
4.3 Determining effective radius of an ion in water . . . . . 68

Chapter 5 Conclusions and Discussions . . . . . 73

Bibliography . . . . . 76

LIST OF FIGURES


Figure 2.1: A depiction of k-nearest neighbour and $\epsilon$-ball: $\mathcal{B}(x_i, \epsilon_i)$. . . . . 11
Figure 2.2: Demonstration of the differences of integration over local region between uniform, Gaussian and Gaussian Kernel. . . . . 15
Figure 2.3: Error analysis for a 1-dimensional Gaussian distribution . . . . . 26
Figure 2.4: Error analysis for a 3-dimensional Gaussian distribution with varies of correlation $\alpha \in (10^{-6}, 1)$ . . . . . 27
Figure 2.5: Error analysis for a 4-dimensional Gaussian distribution with determinant of covariance close to 0 . . . . . 29
Figure 2.6: Error analysis for a 1-dimensional beta distribution . . . . . 31
Figure 2.7: Error analysis for a 1-dimensional parabola distribution . . . . . 33
Figure 2.8: Computational Time for kNN, kpN and kp-kernel entropy estimators for bivariate Gaussian distribution. The ratio of local sample in kpN method $pr_{kpN} = 0.02$. The ratio of local sample in kp-kernel method $pr_{kpk} = 0.02$. . . . . 34
Figure 2.9: Computational Time for kNN, kpN and kp-kernel entropy estimators for bivariate Gaussian distribution. The number of local sample in kpN method is fixed $p_{kpN} = 20$. The number of local sample in kp-kernel method $p_{kpk}$ varies. . . . . 35

Figure 3.1: One Monte Carlo Step in Particle System . . . . . 51
Figure 3.2: A Lennard-Jones system initial and steady state positions . . . . . 54
Figure 3.3: An ionic system initial and steady State Positions . . . . . 55

Figure 4.1: A schematic diagram of charged molecules immersed in an aqueous solvent. $\Gamma$ is the solute-solvent interface, for each $i$th particle in the system, it carries charge $Q_i$ with position $x_i$. . . . . 64
Figure 4.2: Schematic of an ion (big circle in center) surrounded by water molecules (small solid circles). . . . . 66
Figure 4.3: The probability distribution of the radius $R = R(t)$ obtained from the simulation of the time-independent Stokes equation (4.2.9). The vertical line in the plot shows the equilibrium radius defined by $F(R) = 0$ with $F(R)$ given in (3.3). . . . . 68
Figure 4.4: The radial distribution of water molecules around a fixed ion (in the center, red) is a scaled local density of water molecules. . . . . 69
Figure 4.5: The radial distribution of water molecules around an artificial ion carrying the point charge $Q$ . . . . . 71

LIST OF TABLES


Table 2.1: Uniform, Gaussian and Gaussian Kernel methods calculate the tail part of integration - 10000 Samples. . . . . 16
Table 2.2: Uniform, Gaussian and Gaussian Kernel methods calculate the tail part of integration - 5000. Samples . . . . . 16
Table 2.3: Absolute errors table of the kNN, kpN, kp-kernel methods for 3d Gaussian Distribution . . . . . 28
Table 2.4: Absolute errors table of the kNN, kpN, kp-kernel methods for 4-dimensional Gaussian Distribution . . . . . 30

Table 3.1: Parameters Setting for Lennard-Joines Fluids Monte Carlo Simulation . 53
Table 3.2: Parameters Setting for ionic Monte Carlo Simulation . . . . . 54
Table 3.3: kNN, kpN methods for Lennard-Jones System . . . . . 57
Table 3.4: kNN, kpN methods for Lennard-Jones System with $k = 7$ . . . . . 57
Table 3.5: kNN, kpN methods for ionic system with $k = 7$ . . . . . 58

Table 4.1: Parameters for MD simulations. . . . . 70
Table 4.2: Effective ionic radii determined by the radial distribution of water from MD simulations (First nonzero, Peak, Half-peak, and Bulk) and by the generalized RP equation for four artificially designed ions. . . . . 72

ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude to all who helped me during these tough years. In particular, a huge thank you to my advisor Professor Bo Li, for his always support and guide for many years. To me, he is always encouraging and patient.

I would like to sincerely thank all members of my PhD committee, Professor Li-Tien Cheng, Professor Dimitris N. Politis, Professor Zhuowen Tu and Professor Tianyi Zheng. Thank you for all nice suggestions and good conversations on my thesis. I would also like to thank all professors who ever taught me or shared good ideas with me, especially, Professor Jiawang Nie and Professor Philip Gill. I also want to thank all staff members at UCSD mathematics department, especially Terry Lee, Wilson Cheung and Kelly Guerriero. Thanks to Wilson for all technical support on the Gromacs Package.

I enjoyed my life at San Diego accampany with many graduate students at UCSD Mathematics department. I would like to thank my officemates as well as my PhD colleagues Ben Ciotti, Yi Luo and Zirui Zhang. Thank you for nice discussions! To all my friends at UCSD through past few years, your friendship made my life at San Diego a great experience.

My foremost and sincere thanks to my parents, who created me, raised me, supported me and loved me throughout my whole life.

Chapter 4, in part, has been submitted for publication of the material as it may appear in SIAM Journal of Applied Mathematics, 2020, Chao Fan; Bo Li; Michael White, 2020. All authors contributed essentially equally to the article.

## VITA

| | |
|---|---|
| 2014 | B.S. in Applied Mathematics, Tongji University, Shanghai, China |
| 2017 | C.Phil. in Mathematics, University of California San Diego |
| 2019 | M.S. in Computational Science, University of California San Diego |
| 2021 | Ph.D. in Mathematics, University of California San Diego |

## PUBLICATIONS

Chao Fan, Bo Li, and Michael White, A generalized Rayleigh–Plesset equation for ions with solvent fluctuations, *SIAM Journal on Applied Mathematics*, 2020 (submitted).

Zirui Zhang, Clarisse Ricci, Chao Fan, Li-Tien Cheng, Bo Li, and J. Andrew McCammon, Coupling Monte Carlo, Variational implicit solvation, and binary level-Set for simulations of biomolecular binding, *Journal of Chemical Theory and Computation*, 2021 (accepted).

ABSTRACT OF THE DISSERTATION

**A Class of kNN-Type Entropy Estimators: Algorithm, Convergence, and Application to Molecular Modeling**

by

Chao Fan

Doctor of Philosophy in Mathematics

University of California San Diego, 2021

Professor Bo Li, Chair

Entropy is a fundamental concept in science. It describes the disorder, randomness, and uncertainty of a physical, biological, or social system. While understanding entropy has far-reaching impact to advancing our knowledge in many scientific areas and our society, the development of rigorous theories and computational technologies for entropy is a rather challenging task due to the vast complexity of an underlying system. In the context of biological molecules such as proteins and DNAs, entropy as defined in statistical mechanics and thermodynamics is a critical part of the total free energy of such molecules in a chemical environment. Efficient and accurate calculations of such entropy is of particular

interest as the calculation of free energy, which is fundamental to physical and biological processes, is known to be notoriously difficult. The need and recent interest in advanced computational methods for entropy in biological molecular systems have motivated directly this dissertation work.

The basic mathematical and statistical definition of entropy, the Shannon entropy in information theory, for a random variable in an Euclidean space is the negative expectation of the natural logarithm of the probability density function (PDF) for the random variable. The entropy of a physical and biological system can be written in the form of, or approximated by, the Shannon entropy with a suitably defined PDF that has physical meanings. Practically, the dimension of an underlying random variable can be very high, and in addition, its PDF may not be known. The goal of my study is to develop efficient and accurate computational methods for the Shannon entropy with the application to the calculation of entropy of a particle system that may consist of many particles, forming a liquid.

In this dissertation work, I begin with a formal derivation of a class of nonparametric kNN-type estimators of the entropy, including the classical kNN estimator, the kpN estimator introduced recently by some physicists, and a new estimator called kp-kernel estimator that I have constructed. One of my objectives here is to understand if theses estimators can better capture some properties that are related to singular behaviors of an underlying PDF, such as the "tail" part of the PDF. My extensive numerical simulations using these estimators with several different PDFs show some of such advanced features. These include a better description of a strongly correlated system and more accurate sampling of the tail part of a given distribution. I will then present a convergence analysis to show that some of these estimators converge in expectation, under some realistic assumptions.

Subsequently, I apply these kNN-type entropy estimators to calculate the entropy of simple molecular systems. Here a statistical mechanics theory of simple liquids is invoked,

and the entropy is expressed in a series of terms, each is a Shannon entropy, where the first two terms are known to be the most important. I implement the Markov chain Monte Carlo method to sample an underlying molecular system, and then I use the kNN and kpN methods to estimate the entropy.

Finally, I present my related work on the molecular dynamics (MD) simulations of the solvation of an ion in water. Using the radial distribution function of water molecules surrounding an ion, obtained from the MD simulations, I find the effective radius of the ion. I also compare the results of the MD simulations with those of a stochastic ordinary differential equation (SODE) model to examine the validity of such an SODE approach. The work presented here is a first step toward combining statistical methods and computational analysis to tackle one of the very complex problems in mathematical modeling and computer simulations of biological molecules.

My detailed studies of a class of nonparametric entropy estimators and their application to molecular modeling demonstrate that these methods are promising. More work remains to improve the efficiency of some of these estimators, and to develop a complete theory of convergence. Further theories and more related methods are needed for better applications of these estimators in molecular modeling.

# Chapter 1

# Introduction

Entropy describes the disorder of a physical, biological, or social system. It is a fundamental concept and quantity in thermodynamics, statistical mechanics, information theory, and many other disciplines of science [6, 42]. In this dissertation work, I am primarily interested in understanding the entropy in biological molecular systems and developing computational methods for the calculation of such entropy. Specifically, my objectives are to construct and test new kNN-type entropy estimators, and combine them with simulation methods to estimate the entropy of biomolecular systems. I also provide some convergence analysis.

In this introductory chapter, I will first describe the background of my work. It has two parts. One is a class of kNN-type entropy estimators (cf. subsection 1.1), and the other is the derivation and approximation of the entropy of a particle system (cf. subsection 1.2). I will then present my main research results and describe their significance (cf. subsection 1.3).

## 1.1 The Shannon entropy and its kNN-type estimators

Let $X \in \mathbb{R}^d$ be a random variable with the probability density function (PDF) $f : \mathbb{R}^d \to \mathbb{R}$. The Shannon entropy of the PDF $f$ (or the random variable $X \in \mathbb{R}^d$) is defined to be

$$H(f) = \mathbb{E}[-\log f(X)] = -\int_{\mathbb{R}^d} f(x) \log f(x) dx,$$

where the expectation $\mathbb{E}[-\log f(X)]$ is assumed to exists and is finite. Throughout this dissertation, I use log to denote the natural logarithmic function and I use the convention that $0 \log 0 = 0$. I also only consider random variables in $\mathbb{R}^d$. A non-parametric estimator of entropy is obtained by replacing the PDF $f$ in the definition of entropy by its non-parametric kernel or histogram estimator [7].

Let $x_1, ..., x_n \in \mathbb{R}^d$ be data points sampled independently according to the PDF $f$. Let $k$ be an integer such that $1 \leq k \leq n$. For each $i$ $(1 \leq i \leq n)$, let $\epsilon_i$ be the $k$th nearest neighbor (kNN) distance of sample points $x_j$ $(1 \leq j \leq n, j \neq i)$ to $x_i$. The kNN entropy estimator $H_n$ of $H(f)$ is then defined as [39]

$$H_n(f) = \psi(n) - \psi(k) + \frac{d}{n} \sum_{i=1}^{n} \log \text{Vol}(\mathcal{B}(x_i, \epsilon_i)),$$

where

$$\psi(n) = \begin{cases} \sum_{i=1}^{n} \frac{1}{i} - \gamma \quad \text{if} \quad n \geq 2, \\ \psi(1) = -\gamma \quad \text{if} \quad n = 1, \end{cases}$$

$\gamma = 0.5722...$ is Euler's constant, and $\text{Vol}(\mathcal{B}(x_i, \epsilon_i))$ is the volume of the $d$-dimensional ball $\mathcal{B}(x_i, \epsilon_i)$ centered at $x_i$ with radius $\epsilon_i$. The idea is based on the following approximation

$$f(x_i)\text{Vol}(\mathcal{B}(x_i, \epsilon_i)) \approx \frac{k}{n}.$$

This leads to

$$f(x_i) \approx \frac{k}{n} \frac{1}{\mathrm{Vol}(\mathcal{B}(x_i, \epsilon_i))}.$$

Note that the special case $k = 1$ is the classical Kozachenko–Leonenko (KL) estimator [30]. This estimator has been used in various studies, especially in entropy estimation in molecular systems [22, 23, 24, 28, 48].

Based on the idea of the kNN entropy estimator, Lombardi and Pant [35] propose a new estimator, the kpN estimator of $H(f)$:

$$H_n^{(kpN)}(f) = \psi(n) - \psi(k) + \frac{1}{n} \sum_{i=1}^{n} \log \frac{G_{p,i}}{g_{p,i}(x_i)},$$

where $g_{p,i}(x)$ is a function proportional to a Gaussian function approximated by using the $p$-nearest neighbours of $x_i$, and $G_{p,i}$ is the integral of function $g_{p,i}(x)$ over the ball $\mathcal{B}(x_i, \epsilon_i)$. It has been demonstrated initially that the kpN estimator can better describe the "tail" property of the PDF $f$ than the kNN method.

In this dissertation work, I follow the framework of the kpN estimator and construct a new entropy estimator, the kp-kernel estimator:

$$\hat{H}_n^{(kpK)}(f) = \psi(n) - \psi(k) - \frac{1}{n} \sum_{i=1}^{n} \log(t_{p,i}(x_i)) + \frac{1}{n} \sum_{i=1}^{N} \log T_{p,i},$$

where $t_{p,i}(x)$ is kernel density function approximated by using the $p$-nearest neighbouring data points $x_i^1, ..., x_i^p$ of $x_i$,

$$t_{p,i}(x) = \frac{1}{ph^d} \sum_{j=1}^{p} K(\frac{x - x_i^j}{h}),$$

$$K(x) = \frac{\exp(-||x||^2/2)}{v_{1,d}},$$

$$v_{1,d} = \int_{\mathbb{R}^d} e^{-||x||^2/2} dx.$$

I conduct extensive numerical simulations to compare the kNN, kpN, and kp-kernel estimators for some distributions that have some special properties, such as the strong correlation and some degeneracies. I also estimate the efficiency of each of these estimators.

## 1.2 Entropy of a particle system

Consider a system of $N$ particles in a computational box of volume $V$. For the $i$th particle, let us denote the mass of the particle by $m_i$, the position of the particle by $\boldsymbol{r}_i$, the momentum of the particle by $p_i : p_i = m_i \dot{\boldsymbol{r}}_i$ $(i = 1, ..., N)$. Let $f_N$ be the $N$-body distribution in positions and momenta, $f_k$ the $k$th particle one-body distribution of momenta, $g_N$ the $N$ particle correlation function.

The total entropy of the underlying particle system is given by [5, 10, 26, 36, 43, 51]

$$S_{liquid} = -\frac{Rh^{3N}}{N!} \int \int f_N \log f_N d\boldsymbol{r}_N d\boldsymbol{p}_N,$$

with $R$ the gas constant and $h$ the Planck constant. The separability of the momentum can be exploited to divide the total entropy $S_{liquid}$ into two parts: the momentum entropy $S_{mom}$ and configuration entropy $S_{conf}$:

$$S_{liquid} = S_{mom} + S_{conf}.$$

The configurational entropy of a fluid is a function of the interatomic correlations and can

be expressed in terms of correlation functions.

$$\underbrace{\frac{R\rho^N}{N!}\int g_N(\boldsymbol{r})\log g_N(\boldsymbol{r})d\boldsymbol{r}}_{S_{conf}} = S_{1solute} - I_{2solute} + I_{3solute} - ...$$

In this work, I only consider the most severe truncation generates what is called the conditional one particle entropy (C1PE) $S_{1solute}$ defined by Irwin and Huggins [25].

Early work on the calculation of entropy for a molecular system included the work done by Karplus [27] and others [23, 28, 31]. Evaluating solvation entropies directly and combining with direct energy calculations is one way of calculating free energies of solvation [3, 27, 47]. A main approach developed in recent years is to calculate the entropy directly from a truncated series of integrals over the correlation functions. Many studies truncate all terms higher than the solvent-solute correlations.

Molecular Dynamics (MD) simulations and Monte Carlo (MC) simulations are very powerful tools to describe the motion of a particle system with many degrees of freedom. In such simulations, many different states of an underlying system are randomly sampled to provide statistical quantities, such as the averages or probability densities. MC simulations have many applications in computational physics, chemistry, and biology, particularly for strongly coupled solids, charged molecular systems, and cellular structures. In this dissertation, I use the Metropolis–Hastings algorithm to sample a particle system and combine it with the kNN and kpN estimators to calculate the entropy of such a system. The Metropolis–Hasting algorithm is a Markov chain Monte Carlo method. More details about the MC acceleration are discussed in Chapter 3. I present more additional MD simulations in Chapter 4 to investigate more properties of the solvation of an ion.

## 1.3 Main results and organization of this thesis

The main contributions of my dissertation work include:

(1) The construction of a new entropy estimator: the kp-kernel estimator, using Gaussian kernel functions.

(2) The design and implementation of the corresponding entropy estimation algorithms, and extensive simulations for several distributions. For a high-correlation 3-dimensional Gaussian distribution and a 4-dimensional Gaussian distribution with determinant of covariance matrix close to 0, it is found that the kp-kernel method and kpN method outperforms the kNN method.

(3) An alternative proof of the convergence of kNN-type entropy estimators in mean under a realistic assumption: all sample data lie in a bounded computational box.

(4) Extensive MC simulations and the related kNN and kpN entropy calculations for two particle systems.

(5) Additional simulation results: conduct MD simulations of the solvation of ions, and determine the radial distribution and effective radius of an ion.

It is hoped that the mathematical and numerical methods developed here can be used to solve other problems in other related scientific area.

The rest of this dissertation thesis is organized as follows. In chapter 2, I review a general framework of entropy estimators and construct a new entropy estimator, the kp-kernel estimator. I compare kNN, kpN, and kp-kernel estimators via numerical experiments for different known probability density functions. I then provide a convergence analysis for the kNN method.

In chapter 3, I investigate the entropy calculation of a molecular system. I implement the Metropolis–Hastings Monte Carlo algorithm for particle system simulations. I simulate

the Lennard-Jones potential model and an ionic particle model and apply the kNN and kpN entropy estimators for these models.

In Chapter 4, I report additional MD simulations results for the solvation of ions and investigate the ion radius by the MD simulation and a stochastic ordinary differential model.

Finally, in Chapter 5, I discuss briefly my results and draw conclusions. I also point out some future directions of further studies.

# Chapter 2

# The kNN, kpN and kp-kernel Entropy Estimators

## 2.1 Entropy: Definition and example

Let $(\Omega, \mathcal{A}, P)$ be a probability space. Let $X : \Omega \to \mathbb{R}^d$ be a random variable. Assume that the distribution of $X$, the Borel measure $\mu_x = P \circ X^{-1} : \mathcal{B}_{\mathbb{R}^d} \to \mathbb{R}$ (where $\mathcal{B}_{\mathbb{R}^d}$ is the Borel $\sigma$-algebra of $\mathbb{R}^d$) has an $L^1(\mathbb{R}^d)$–density $f : \mathbb{R}^d \to \mathbb{R}$ with respect to the Lebesgue measure on $\mathbb{R}^d$. The probability density function (PDF) $f$ is characterized by the following properties: $f \in L^1(\mathbb{R}^d)$, $f \geq 0$ in $\mathbb{R}^d$, and $\int_{\mathbb{R}^d} f dx = 1$.

**Definition 1.** *The Shannon entropy of the PDF $f$, or the random variable $X \in \mathbb{R}^d$ with its PDF $f$ is*

$$H(f) = \mathbb{E}[-\log f(X)] = -\int_{\mathbb{R}^d} f(x) \log f(x) dx. \tag{2.1.1}$$

Here and below I define $0 \log 0 = 0$. I assume $H(f)$ is well-defined and is finite. The Shannon entropy or differential entropy (2.1.1) depends only on the probability density of the random variable, and therefore is sometimes written as $H(f)$ rather than $H(X)$. The concept of differential entropy was introduced in Shannon's original paper [38]. Since then,

entropy has been applied in many scientific areas.

**Example 1.** *(Entropy of a multivariate Gaussian distribution) If $X \in \mathbb{R}^d$ is the multivariate Gaussian random variable with mean $\mu$ and covariance matrix $\Sigma$, then its PDF is*

$$f(x) = \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} \exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)).$$

*The entropy of the random variable $X$ is:*

$$\begin{aligned} H(f) &= -\int_{\mathbb{R}^d} f(x) \log f(x) dx \\ &= \int_{\mathbb{R}^d} f(x) \frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu) dx - \int_{\mathbb{R}^d} f(x) \frac{1}{2} \log((2\pi)^d det(\Sigma)) dx. \end{aligned}$$

*The second term equals $\frac{1}{2}\log((2\pi)^d det(\Sigma)$ since it is a constant. The first term equals*

$$\frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} \int_{\mathbb{R}^d} \exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)) \frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu) dx.$$

*By the change of variables $y = \Sigma^{-1/2}(x-\mu)$, the first term is*

$$\begin{aligned} &\int_{\mathbb{R}^d} f(x) \frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu) dx \\ &= \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} \int_{\mathbb{R}^d} \exp(-\frac{1}{2} y^T y) \frac{1}{2} y^T y dy \\ &= \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} 2^{\frac{d}{2}} \int_{\mathbb{R}^d} \exp(-|z|^2)|z|^2 dz \\ &= \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} 2^{\frac{d}{2}} \int_{\mathbb{R}^d} (z_1^2 + ... + z_d^2) \exp(-z_1^2 - ... - z_d^2) dz_1 ... dz_d \\ &= \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} 2^{\frac{d}{2}} d \int_{\mathbb{R}} z_1^2 e^{-z_1^2} dz_1 \int_{\mathbb{R}} e^{-z_2^2} dz_2 ... \int_{\mathbb{R}} e^{-z_d^2} dz_d \\ &= \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} 2^{\frac{d}{2}} d \frac{\sqrt{\pi}}{2} (\sqrt{\pi})^{d-1} \\ &= \frac{d}{2}. \end{aligned}$$

*In these calculations, I used*

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}.$$

*and*

$$\int_{-\infty}^{\infty} x^2 e^{-x^2} dx = \frac{\sqrt{\pi}}{2}.$$

*Hence,*

$$H(f) = \frac{d}{2} + \frac{1}{2}\log((2\pi)^d det(\Sigma) = \frac{1}{2}\log[(2\pi e)^d \det(\Sigma)].$$

## 2.2 The $k$th nearest distance and entropy estimators

### 2.2.1 The $k$th nearest distance

My aim is to estimate the entropy $H(f)$ from $n$ samples $x_1, ..., x_n$. Let us define the Monte Carlo estimator of the entropy as

$$H_n(f) = -\frac{1}{n}\sum_{i=1}^{n} \log(f(x_i)). \tag{2.2.1}$$

If $f$ is unknown, an estimate of $f$ must be substituted in equation (2.2.1). Singh *et al.* [39] provide a reasonable estimator $\hat{f}(x_i)$ of $f(x_i)$.

$$\hat{f}(x_i) = \frac{k}{n}\frac{1}{\text{Vol}(\mathcal{B}(x_i, \epsilon_i))},$$

where $\epsilon_i$ is the $k$th distance from sample $x_i$ to its $k$th nearest sample.

Instead of approximate the unknow function $f(x_i)$ directly, one can derive the kNN estimator through approximating the mass probability of the ball $\mathcal{B}(x_i, \epsilon_i) = \{x \in \mathbb{R}^d : ||x - x_i|| < \epsilon\}$ with respect to the distribution $f$.

This mass probability is defined by

$$P_i(\epsilon) = \int_{\mathcal{B}(x_i,\epsilon)} f(x)dx. \tag{2.2.2}$$

Consider the probability distribution $p_i(\epsilon)$ for the distance between $x_i$ and its $k$th nearest neighbour. The probability $p_i(\epsilon)d\epsilon$ is equal to the probability that exactly one point in $[\epsilon, \epsilon + d\epsilon]$, exactly $k-1$ points are at distance less than the $k$th nearest neighbor sample, and remaining $n-k-1$ points are farther than the $k$th nearest neighbor sample. Then it follows that

$$p_i(\epsilon)d\epsilon = \binom{k}{1}\binom{n-1}{k}\frac{dP_i(\epsilon)}{d\epsilon}d\epsilon(P_i(\epsilon))^{k-1}(1-P_i(\epsilon))^{n-k-1}.$$

Remarks: $\frac{dP_i(\epsilon)}{d\epsilon}d\epsilon$ represents the probability that there exists one sample point which

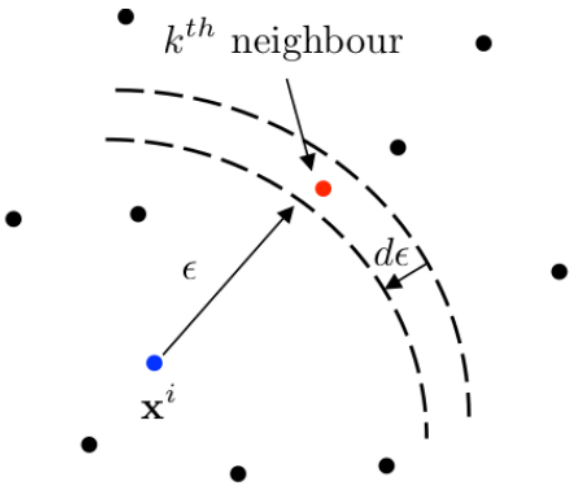


**Figure 2.1**: A depiction of k-nearest neighbour and $\epsilon$-ball: $\mathcal{B}(x_i, \epsilon_i)$.

the distance between sample $x_i$ is $\epsilon$, $P_i(\epsilon)^{k-1}$ represents, the probability that there exists $k-1$ sample points which the distance between sample $x_i$ is less than $\epsilon$, $(1-P_i(\epsilon))^{n-k-1}$ represents the probability that there exist $n-k-1$ points which the disstance between sample $x_i$ is greater than $\epsilon$.

The expected value of $\log(P_i)$ can be obtained from the definition of $P_i(\epsilon)$ and $p_i(\epsilon)$ [29, 53]:

$$\begin{aligned}
\mathbb{E}(\log P_i) &= \int_0^\infty \log P_i(\epsilon) p_i(\epsilon) d\epsilon \\
&= k\binom{n-1}{k} \int_0^\infty \frac{dP_i(\epsilon)}{d\epsilon} d\epsilon (P_i)^{k-1}(1-P_i)^{n-k-1} \log P_i \\
&= k\binom{n-1}{k} \int_0^1 dt (t)^{k-1}(1-t)^{n-k-1} \log t \qquad (consider\ t = P_i(\epsilon)) \\
&= \psi(k) - \psi(n),
\end{aligned} \tag{2.2.3}$$

where $\psi$ is the digamma function. See Lemma 1 for proof.

**Lemma 1.** *Proof of equation* (2.2.3)

$$k\binom{n-1}{k} \int_0^1 (p)^{k-1}(1-p)^{n-k-1} \log p \; dp = \psi(k) - \psi(n).$$

*Proof.* The Beta function is defined by

$$B(x,y) = \int_0^1 p^x (1-p)^y dp.$$

A key property of the beta function is its close relationship to the gamma function

$$B(x,y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}. \tag{2.2.4}$$

Take partial derivative on $B(x,y)$ with respect to $x$,

$$\frac{\partial B(x,y)}{\partial x} = \int_0^1 \frac{\partial}{\partial x} p^{x-1}(1-p)^{y-1} dp = \int_0^1 p^{x-1}(1-p)^y \log p \; dp.$$

Take logarithm on both sides of (2.2.4),

$$\log B(x,y) = \log\Gamma(x) + \log\Gamma(y) - \log\Gamma(x+y). \tag{2.2.5}$$

Take partial derivative on equation (2.2.5) with respect to $x$,

$$\frac{1}{B(x,y)}\frac{\partial}{\partial x}B(x,y) = \psi(x) - \psi(x+y), \tag{2.2.6}$$

where $\psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}$ is the digamma function. Setting $x = k$, $y = n-k$ in equation (2.2.6), we obtain

$$\frac{1}{B(k,n-k)}\int_0^1 p^{x-1}(1-p)^y \log p \; dp = \psi(k) - \psi(n).$$

Hence,

$$k\binom{n-1}{k}\int_0^1 (p)^{k-1}(1-p)^{n-k-1}\log p \; dp = \psi(k) - \psi(n).$$

This completes the proof. □

### 2.2.2 A general framework of kNN-type entropy estimators

Assume the probability mass $P_i(\epsilon_i)$ defined in (2.2.2) can be written as the following form

$$P_i(\epsilon_i) = \eta_i f(x_i). \tag{2.2.7}$$

Taking logarithm on equation (2.2.7), we obtain

$$\log f(x_i) = \log P_i(\epsilon_i) - \log\eta_i. \tag{2.2.8}$$

Combine equation (2.2.1) and equation (2.2.8). The entropy estimator can be written as

$$H_n(f) = -\frac{1}{n}\sum_{i=1}^{n}\log(f(x_i)) = \frac{1}{n}\sum_{i=1}^{n}\log\eta_i - \frac{1}{n}\sum_{i=1}^{n}\log P_i(\epsilon_i).$$

By property of equation (2.2.3), one can propose a general entropy estimator [35]

$$H_n(f) = \psi(n) - \psi(k) + \frac{1}{n}\sum_{i=1}^{n}\log\eta_i. \tag{2.2.9}$$

The main challenge is how to estimate the mass probability defined in equation (2.2.2). Generally, the more precise equation (2.2.2) estimate, the better for entropy estimator. Lombardi and Pant [35] proposed a kpN method in 2016. The central idea is to estimate the probability mass around each sample point by a local Gaussian approximation. The local approximation is obtained by looking at $p$ neighbors around the sample point. the tails of the true probability distribution are better captured. However, there still exists some error.

A graphical demonstration of the difference in the integrals of probability density considered by the kNN, kpN and kp-kernel estimators is shown in Figure 2.2. While near the mode of the distribution the approximations to the integral of the probability density are similar for the two estimators, in the tails the integral is better captured by the kpN estimator as a local Gaussian is constructed and is much better captured by the kp-kernel estimator. In each plot, the true distribution (Gaussian) is shown by a solid (black) line and the 50 samples are shown with "+" green symbols. For the two points (shown by solid vertical lines), the integration region $\mathcal{B}(x_i, \epsilon_i)$ with $k = 3$ is shown by dashed vertical lines, and the integrals are shown in shaded gray. In the upper-left panel, the true area of integration is shown. In the upper-right panel shows the uniform approximation to this area, the left-bottom panel shows the area approximations by the local Gaussian approximation with $p = 10$, and right-bottom panel shows the area approximations by the

local kernel Gaussian approximation with $p = 10$. Two different points – one near the tails and one near the mode – of a Gaussian distribution are shown.

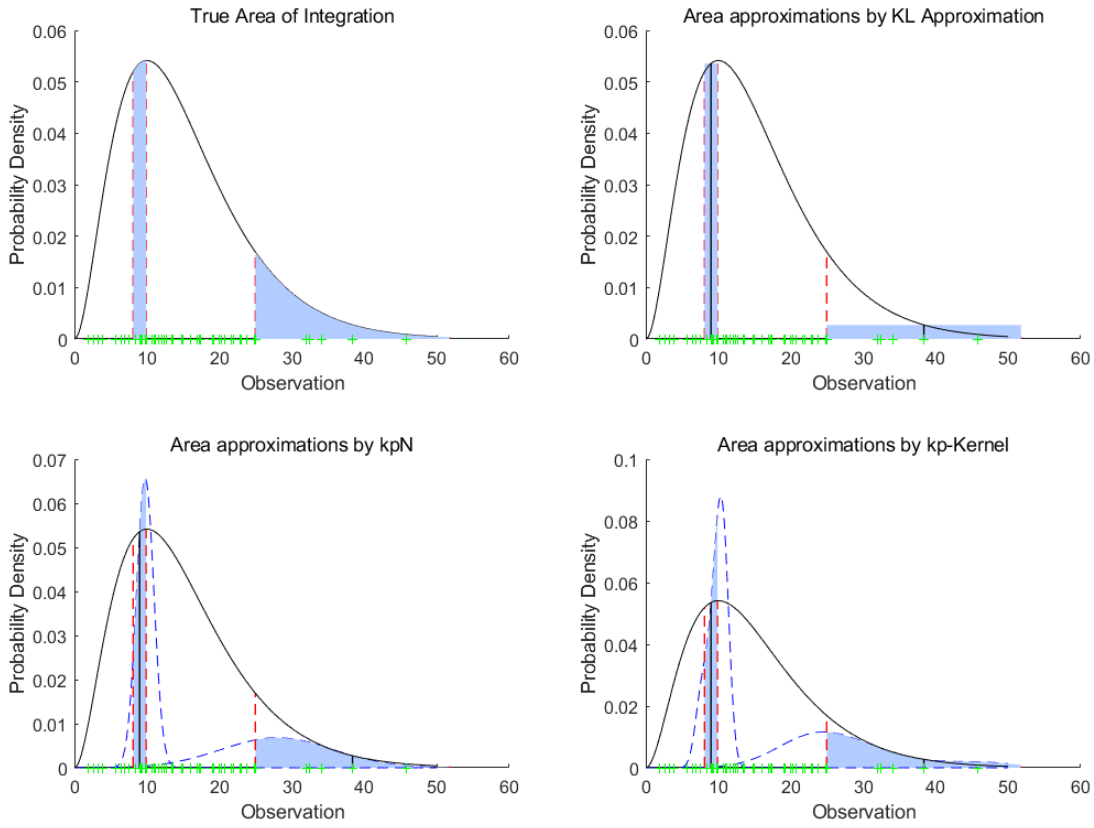


**Figure 2.2**: Demonstration of the differences of integration over local region between uniform, Gaussian and Gaussian Kernel.

In Table 2.1 and Table 2.2, I extract 10000 and 5000 samples from the Gamma distribution. I calculate the local integration inside $\mathcal{B}(x_i, \epsilon_i)$ approximated by uniform distribution, local Gaussian distribution and local Gaussian Kernel distribution. I present the 5 times average of relative errors in Table 2.1 and Table 2.2.

**Table 2.1**: Uniform, Gaussian and Gaussian Kernel methods calculate the tail part of integration - 10000 Samples.

| | Different index of sample from tail part | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Uniform | 4.4176% | 4.5223% | 4.4826% | 4.2809% | 4.4576% | 1.5299% |
| Gaussian | 2.0076% | 5.3505% | 5.2208% | 7.6908% | 8.2297% | 3.5533% |
| Gaussian Kernel | 2.6139% | 1.5459% | 1.4998% | 5.5317% | 5.6564% | 3.2706% |

**Table 2.2**: Uniform, Gaussian and Gaussian Kernel methods calculate the tail part of integration - 5000 Samples.

| | Different index of sample from tail part | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Uniform | 12.4301% | 8.1695% | 3.5782% | 0.9154% | 0.8921% | 0.4962% |
| Gaussian | 13.0452% | 8.5605% | 4.305% | 1.0897% | 1.0621% | 0.7158% |
| Gaussian Kernel | 4.9901% | 1.9007% | 7.1924% | 1.0734% | 1.0749% | 1.236% |

## 2.3 The kNN, kpN and kp-kernel estimators: Definition and algorithm

Let $x_1, ..., x_n$ be random and independent samples in $\mathbb{R}^d$ of a random variable $X$ with the PDF $f$.

### 2.3.1 The kNN estimator

The key problem is how to derive a reasonable estimator of $\eta_i$ in equation (2.2.2). The classical estimate by Kozachenko and Leonenko [30] and its extension by Singh *et al.* [39] assume that the probability density $f(x)$ is constant $f(x_i)$ inside $\mathcal{B}(x_i, \epsilon_i)$. So,

$$P^{(kNN)}(x) = \text{Vol}(\mathcal{B}(x_i, \epsilon_i))f(x_i) = c_d \epsilon_i^d f(x_i) = \eta_i f(x_i), \tag{2.3.1}$$

where $c_d$ is the volume of the d-dimensional unit-ball. The expression $c_d$ depends on the type of the norm I apply to calculate the distances. I take Euclidean norm ($L_2$), $c_d = \pi^{d/2}/\Gamma(1 + d/2)$, where $\Gamma$ is the Gamma function. I substitute equation (2.3.1) in equation (2.2.9),

$$\eta_i = c_d \epsilon_i^d.$$

I obtain the kNN estimator of the entropy $H(f)$ by:

$$H_n^{(kNN)} = \psi(n) - \psi(k) + \log(c_d) + \frac{d}{n} \sum_{i=1}^{n} \log(\epsilon_i).$$

where

$$\psi(n) = \begin{cases} \sum_{i=1}^{n} \frac{1}{i} - \gamma \quad \text{if} \quad n \geq 2, \\ \\ \psi(1) = -\gamma \quad \text{if} \quad n = 1, \end{cases}$$

and $\gamma = 0.5722...$ is Euler's constant. This estimator is referred as the kNN estimator in the remainder of this thesis.

**Algorithm 1:** Algoritm to estimate kNN entropy

**Input** : • $x_i \in \mathbb{R}^d$, $i = 1, 2, ..., n$: the samples
• $k$: the number of nearest neighbors for calculating $\epsilon_i$

**Output** : $H(f)$: the kNN entropy estimate

$c_d \leftarrow \pi^{d/2}/\Gamma(1 + d/2)$
$H(f) = \psi(n) - \psi(k) + c_d$
**for** $i \leftarrow 0$ **to** $n$ **do**
    $\epsilon_i \leftarrow L_2$ distance to the $k$th nearest neighbor of $x_i$.
    $H(f) \leftarrow H(f) + (d/n) \log(\epsilon_i)$
**end for**
**return** $H(f)$

### 2.3.2 The kpN estimator

The primary cause of high error in the kNN estimator is the assumption of constant density in each local region $\mathcal{B}(x_i, \epsilon_i)$. This may be unjustified when the sample $x_i$ lies in the tail part of the density function, see Figure 2.2. In such cases, a constant density assumption in $B(x_i, \epsilon_i)$ leads to an overestimation of the probability mass and hence the entropy estimate. To remedy this, an alternate formulation for $\eta_i$ in equation (2.2.7) is sought. Contrary to a constant density assumption in Kozachenko and Leonenko Estimator, I assume the probability density in $\mathcal{B}(x_i, \epsilon)$ is represented as

$$f^{(kpN)}(x) = f(x_i)\frac{g_{p,i}(x)}{g_{p,i}(x_i)}.$$

Essentially, the probability density is assumed to be proportional to a Gaussian function $g_{p,i}(x)$ approximated by using $p$-nearest neighbours of $x_i$.

$$g_{p,i}(x) = \exp\left(-\frac{1}{2}(x-\mu_i)^T S_i^{-1}(x-\mu_i)\right).$$

where $\mu_i$ and $S_i$ represent the empirical mean and covariance matrix of the $p$-nearest neighbouring data points $x_i^1, ..., x_i^p$ of $x_i$. In order to guarantee the inverse of empirical covariance matrix $S_i$ exist, I apply pseudo-inverse in implementation. Consequently, the probability mass in $\mathcal{B}(x_i, \epsilon_i)$ can be written as

$$P_i^{(kpN)}(\epsilon_i) = \int_{\mathcal{B}(x_i,\epsilon_i)} f^{(kpN)}(x)dx = f(x_i)\frac{1}{g_{p,i}(x_i)}G_{p,i}, \tag{2.3.2}$$

where

$$G_{p,i} = \int_{\mathcal{B}(x_i,\epsilon_i)} g_{p,i}(x)dx.$$

I substitute equation (2.3.2) in equation (2.2.8),

$$\eta_i = \frac{1}{g_{p,i}(x_i)} G_{p,i}.$$

I then obtain the kpN estimator of the entropy $H(f)$ by:

$$H_n^{(kpN)} = \psi(n) - \psi(k) - \frac{1}{n}\sum_{i=1}^{n} \log(g_{p,i}(x_i)) + \frac{1}{n}\sum_{i=1}^{n} \log G_{p,i},$$

where $\epsilon_i$ is the distance of the ith sample $x_i$ to its $k$th nearest neighbor.

**Algorithm 2:** Algoritm to estimate kpN entropy

**Input :**
- $x_i \in \mathbb{R}^d$, $i = 1, 2, ..., n$: the samples
- $k$: the number of nearest neighbors for calculating $\epsilon_i$
- $p$: the number of nearest neighbors for calculating the local Gaussian approximation ($p \geq k$)

**Output :** $H(f)$: the kpN entropy estimate

**for** $i \leftarrow 0$ **to** $n$ **do**
  $\{x_i\}^p \leftarrow$ set of $p$-nearest neighbors of $x_i$ ($L_\infty$ norm)
**end for**
**for** $i \leftarrow 0$ **to** $n$ **do**
  $\epsilon_i \leftarrow L_\infty$ distance to the $k$th nearest neighbor of $x_i$.
  $\mathcal{B}(x_i, \epsilon_i) \leftarrow x_i \pm \epsilon_i \mathbf{e}$; $\mathbf{e}$ being the canonical basis
  $\mu_i \leftarrow$ mean of $\{x_i\}^p$
  $S_i \leftarrow$ covariance of $\{x_i\}^p$
  $G_i \leftarrow$ integral of local Gaussian in $\mathcal{B}(x_i, \epsilon_i)$ using EMPGP
  $g_i \leftarrow$ local Gaussian density value at $x_i$
  $H(f) \leftarrow H(f) + n^{-1}[\log(G_i) - \log(g_i)]$
**end for**
$H(f) = \psi(n) - \psi(k)$

Remarks: To compute the integration $G_{p,i}$, a multivariate Gaussian definite integral inside $\mathcal{B}(x_i, \epsilon_i)$ has to be computed. Since I adopt the $L_\infty$ distance, this operation amounts to computing the integral of a multivariate Gaussian inside a box. The expectation propagation multivariate Gaussian probability (EPMGP) method, proposed in [14], is chosen. A brief description about expectation propagation multivariate Gaussian probability

(EPMGP) method is below.

I define the Gaussian distribution $p_0(x) = \mathcal{N}(x; m; K)$ as

$$p_0(x) = \frac{1}{(2\pi)^{n/2}|K|^{n/2}} \exp -\frac{1}{2}(x-m)^T K^{-1}(x-m),$$

where $x \in \mathbb{R}^d$ is a vector with $d$ real valued elements, $m \in \mathbb{R}^d$ is the mean vector, and $K \in \mathbb{R}^{n\times n}$ is the symmetric, positive semidefinite covariance matrix. I consider the probability that a draw from $p_0(x)$ falls in a region $\mathbb{A} \subseteq \mathbb{R}^d$, which we will denote as

$$Prob(x \in \mathbb{A}) = \int_{\mathbb{A}} p_0(x)dx = \int_{l_1}^{u_1} ... \int_{l_d}^{u_d} p_0(x)dx_1...dx_d,$$

where $l_1, ..., l_d$ and $u_1, ..., u_d \in \mathbb{R}$ and $\mathbb{A} \subseteq \prod_{i=1}^{d}[l_i, u_i]$. We consider an intractable distribution $p(x)$ is a product of a prior distribution $p_0(x)$ and one or more likelihood functions or factors $t_i(x)$:

$$p(x) = p_0(x)\prod_{i=1}^{d} t_i(x).$$

where $t_i(x), i = 1, ..., d$ is an indicator function defined in a particular direction, namely:

$$t_i(x) = \mathbb{1}\{l_i < c_i^T < u_i\} = \begin{cases} 1 & l_i < c_i^T x < u_i, \\ 0 & otherwise. \end{cases}$$

The unnormalised Gaussian approximation is

$$q(x) = p_0(x)\prod_{i=1}^{d} \tilde{t}_i(x),$$

We will require the cavity distribution for the derivation of the updates, which is defined as

$$q^{\setminus i}(x) = \frac{q(x)}{\tilde{t}_i(x)} = Z^{\setminus i}\mathcal{N}(x; u^{\setminus i}, V^{\setminus i}), \tag{2.3.3}$$

The above step is the cavity step of Expectation Propagation [14]. We now must do the projection operation, which involves moment matching the approximation $\tilde{t}_i(x)q^{\setminus i}(x)$ to the appropriate moments of $t_i(x)q^{\setminus i}(x)$. Here, we only consider $t_i(x)$ have rank one structure:

$$\hat{Z}_i = \int t_i(x)q^{\setminus i}(x)dx$$

$$\hat{u}_i = \hat{\mu}_i c_i$$

$$\hat{V}_i = \hat{\sigma}_i^2 c_i c_i^T,$$

where $\{\hat{Z}_i, \hat{\mu}_i, \hat{\sigma}_i^2\}$ depend on the factor $t_i(x)$ and $c_i$ is the rank one direction as in equation (2.3.3). See more projection step in [14].

Now, by the definition of the approximation, we can calculate the new approximation $q(x)$ as the product $q(x) = p_0(x)\prod_{i=1}^d \tilde{t}_i(x)$:

$$\begin{aligned} q(x) &= Z\mathcal{N}(\mu, \Sigma) \\ where \quad \mu &= \Sigma(K^{-1}m + \sum_{i=1}^m \frac{\tilde{\mu}_i}{\tilde{\sigma}_i^2}c_i), \quad \Sigma = (K^{-1} + \sum_{i=1}^m \frac{1}{\tilde{\sigma}_i^2}c_i c_i^T)^{-1}. \end{aligned}$$

Lastly, once the algorithm has converged, we can calculate the normalisation constant of $q(x)$. While this step is again general to Gaussian EP, we highlight

$$\begin{aligned} \log Z = &-\frac{1}{2}(m^T K^{-1} m + \log|K|) \\ &+ \sum_{i=1}^m (\log \tilde{Z}_i - \frac{1}{2}(\frac{\tilde{\mu}_i^2}{\tilde{\sigma}_i^2} + \log\tilde{\sigma}_i^2 + \log(2\pi))) \\ &+ \frac{1}{2}(\mu^T\Sigma^{-1}\mu + \log|\Sigma|). \end{aligned}$$

In the above we have broken this equation up into three lines to clarify that the normalisation term $\log Z$ has contribution from the prior $p_0(x)$ (first line), the approximate factors (the second line), and the full approximation $q(x)$ (third line).

### 2.3.3 The kp-kernel estimator

Similar to the idea of kpN estimator assumption, in order to improve the accuracy of the local probability density, I proposed a local kernel density function instead of a local Gaussian to approximate the local region mass probability. Essentially, the probability density is assumed to be proportional to a Gaussian function approximated by using $p$ nearest neighbors of $x_i$. The idea is that the $p$ neighbors kernel density function would capture the local nonuniformity of the true probability density inside $\mathcal{B}(x_i, \epsilon_i)$ more precisely. Assume the local probability density in $\mathcal{B}(x_i, \epsilon_i)$ has the form:

$$f^{(kpK)}(x) = f(x_i)\frac{t_{p,i}(x)}{t_{p,i}(x_i)},$$

where $t_{p,i}(x)$ is kernel density function approximated by using $p$-nearest neighbours $\{x_i^j\}, j = 1, 2, ..., p$ of $x_i$.

$$t_{p,i}(x) = \frac{1}{ph^d}\sum_{j=1}^{p} K(\frac{x - x_i^j}{h}),$$

where $K : \mathbb{R}^d \to \mathbb{R}$ is a smooth function called kernel function and $h > 0$ is the smoothing bandwidth that controls the amount of smoothing. I apply Silverman's rule of thumb, i.e. $h = \hat{\sigma}(\frac{4}{3n})^{-1/5}$, where $\hat{\sigma}$ is the $p$ local sample variance.

I select Gaussian kernel function:

$$K(x) = \frac{\exp(-||x||^2/2}{v_{1,d})},$$

and

$$v_{1,d} = \int \exp\{-||x||^2/2\}dx.$$

Consequently, the probability mass in $\mathcal{B}(x_i, \epsilon_i)$ can be written as

$$P_i^{(kpK)}(\epsilon) = \int_{\mathcal{B}(x_i,\epsilon)} f^{(kpK)}(x)dx = f(x_i)\frac{1}{t_{p,i}(x_i)}T_{p,i}, \tag{2.3.4}$$

where

$$T_{p,i} = \int_{\mathcal{B}(x_i,\epsilon_i)} t_{p,i}(x)dx.$$

I substitute equation (2.3.4) in equation (2.2.8),

$$\eta_i = \frac{1}{t_{p,i}(x_i)}T_{p,i}.$$

I then obtain the kp-kernel estimator of the entropy $H(f)$ by:

$$H_n^{(kpK)} = \psi(n) - \psi(k) - \frac{1}{n}\sum_{i=1}^{n}\log(t_{p,i}(x_i)) + \frac{1}{n}\sum_{i=1}^{N}\log T_{p,i},$$

where $\epsilon_i$ is the distance of the ith sample $x_i$ to its $k$th nearest neighbor. Remarks: I apply the EMPGP algorithm $p$ times to calculate the integration of the kernel density function.

## 2.4 Simulation results

In this section, I test the different estimators of mutual information and compare them against each other. I compare three different entropy estimators using three classes of numerical tests on simulated samples. The first class aims at studying the Gaussian distribution function with different cases validating the proposed kp-kernel estimator, and compare with existing kNN method and kpN method. Several relevant properties about Gaussian distribution are investigated in more complicated setting, including correlation, determinant of the Gaussian distribution and dimension increase effect. The second class is beta distribution. The third class study a constructed bounded parabola distribution

**Algorithm 3:** Algoritm to estimate kp-kernel entropy

**Input :**
- $x_i \in \mathbb{R}^d$, $i = 1, 2, ..., n$: the samples
- $k$: the number of nearest neighbors for calculating $\epsilon_i$
- $p$: the number of nearest neighbors for calculating the local Kernel approximation ($p \geq k$)

**Output :** $H(f)$: the kp-kernel entropy estimate

**for** $i \leftarrow 0$ **to** $n$ **do**
  $\{x_i\}^p \leftarrow$ set of $p$-nearest neighbors of $x_i$ ($L_\infty$ norm)
**end for**
$H(f) = \psi(n) - \psi(k)$
**for** $i \leftarrow 0$ **to** $n$ **do**
  $\epsilon_i \leftarrow L_\infty$ distance to the $k$th nearest neighbor of $x_i$.
  $\mathcal{B}(x_i, \epsilon_i) \leftarrow x_i \pm \epsilon_i \mathbf{e}$; $\mathbf{e}$ being the canonical basis
  $f_p^i(x) \leftarrow$ kernel density approximation using $\{x_i\}^p$
  $T_{p,i} \leftarrow$ integral of kernel density $f_p^i(x)$ in $\mathcal{B}(x_i, \epsilon_i)$ using EMPGP $p$ times
  $t_p(x_i) \leftarrow$ local kernel density $f_p^i(x)$ at $x_i$
  $H(f) \leftarrow H(f) + n^{-1}[\log(T_{p,i}) - \log(t_p(x_i))]$
**end for**
**return** $H(f)$

function.

### 2.4.1 A Gaussian distribution

The first family of distribution is a simple 1-dimensional Gaussian, the second family of distribution is 3-dimensional multivariate Gaussian with covariance vary geometrically from $[0, 1]$. Generally, I model the random variable $X \in \mathbb{R}^d$ as:

$$X \sim \mathcal{N}(0, \Sigma).$$

where $\Sigma$ is the covariance of the variable $X$. The multivariate normal distribution is:

$$f(x) = \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} \exp(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)).$$

The exact entropy for random variable $X$ is:

$$H(f) = \frac{1}{2}\log[(2\pi e)^d \det(\Sigma)].$$

I do three tests on the Gaussian distribution, including simple 1-dimensional Gaussian distribution, 3-dimensional Gaussian distribution with high-correlation variance, 4-dimensional Gaussian distribution with variance determinant close to 0.

**Test1:** 1-dimensional Gaussian distribution, I estimate the error of the estimator, plot against sample size $n$ for each estimators. I also plot the relative error against sample size $n$. The results is shown in Figure 2.3. Note that all three method perform closely well when number of samples goes to large.

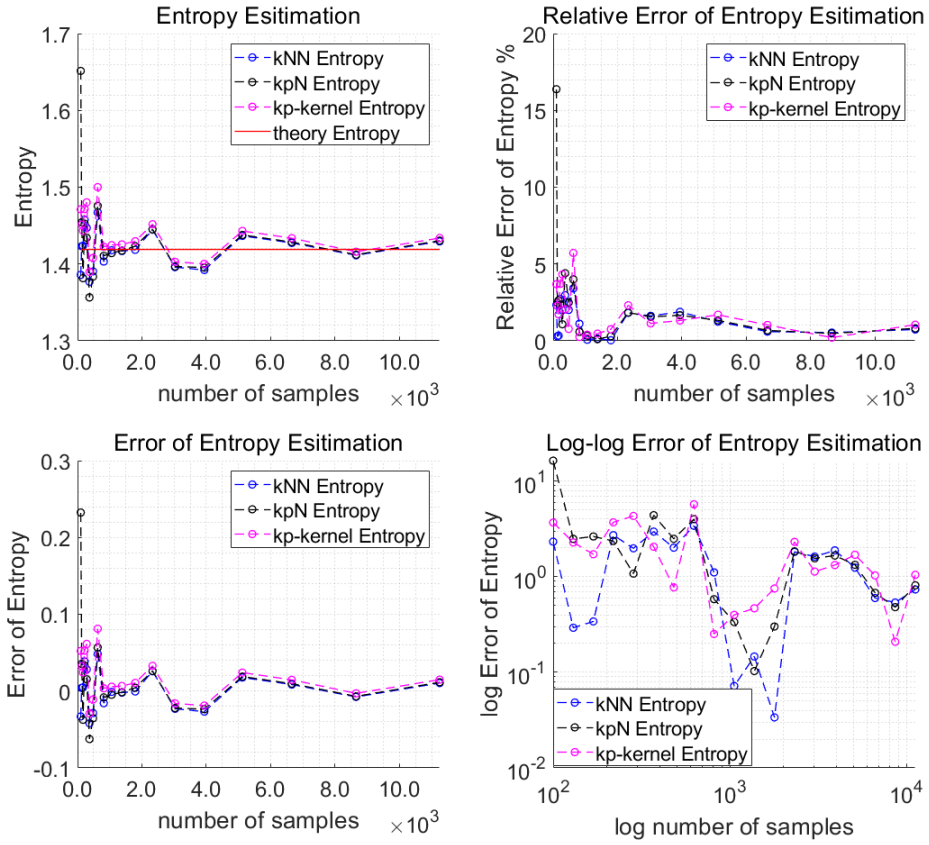


**Figure 2.3**: Error analysis for a 1-dimensional Gaussian distribution. The entropy estimation results for kNN (blue dot line), kpN (black dot line), kp-kernal (pink dot line) and theory (red solid line) results are presented on the left of the figure. Error of Entropy estimation is presented on the middle of figure. Relative error of entropy estimation is presented on the right of figure. Log-log error of entropy estimation is presented on the bottom right of figure.

**Test2:** I consider 3-d normally distributed variables with standard deviation $\sigma_{i,i} = 1$ and correlation $\sigma_{i,j} = 1 - \alpha, i \neq j$. For each $\alpha \in \{3^{-j} : j = 2, ..., 18\}$ a sample of size $n = 10000$ is drawn and the 3-d gaussian entropy estimated by kNN, kpN, kp-kernel and true entropy is drawn in Figure 2.4. Note that the error for both kNN and kp-kernel method increase up tp 30% when $\alpha < 10^{-4}$. Initially, kNN captured redundant local ball area, and kp-kernel method's multiple kernel function overfitting on the local region when the correlation is small. kp-kernel method perform better than kpN and kNN when $\alpha \in (10^{-3}, 10^{-1})$ since it

capture the local region better.

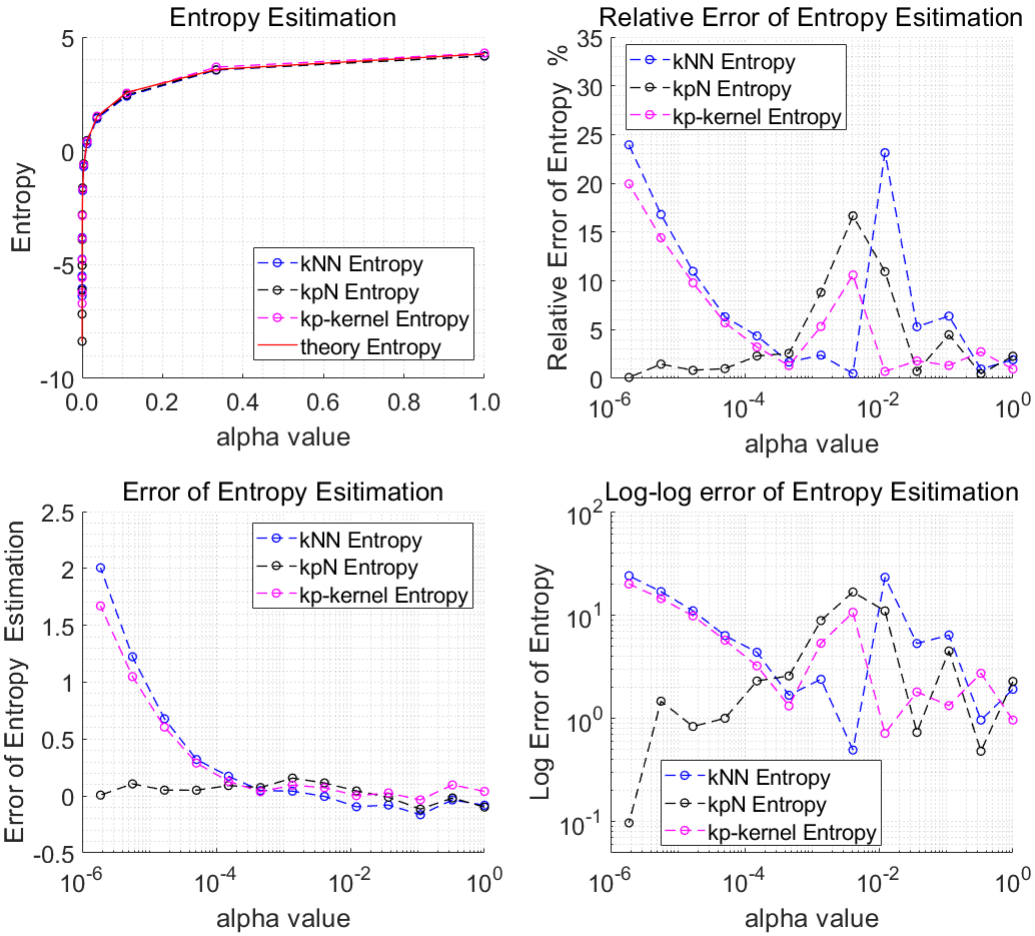


**Figure 2.4**: Error analysis for a 3-dimensional Gaussian distribution with varies of correlation $\alpha \in (10^{-6}, 1)$. The entropy estimation results for kNN (blue dot line), kpN (black dot line), kp-kernal (pink dot line) and theory (red solid line) results are presented on the top left of the figure. Relative Error of Entropy estimation is presented on the top right of figure. Error of entropy estimation is presented on the bottom left of figure. Log-log error of entropy estimation is presented on the bottom right of figure.

**Table 2.3**: Absolute errors of the kNN, kpN, kp-kernel methods for 3-dimensional Gaussian Distribution with value of $\alpha = 0.01$

| | | Different number of $k$ | | | | |
|---|---|---|---|---|---|---|
| | | 3 | 4 | 5 | 6 | 7 |
| kNN Method | | 0.0925 | 0.1081 | 0.0861 | 0.0465 | 0.0535 |
| kpN Method | $p = 50$ | 0.0557 | 0.1260 | 0.1773 | 0.2002 | 0.2443 |
| | $p = 100$ | 0.0293 | 0.0189 | 0.0537 | 0.0566 | 0.0900 |
| | $p = 150$ | 0.0633 | 0.0246 | 0.0021 | 0.0026 | 0.0249 |
| | $p = 200$ | 0.0807 | 0.0476 | 0.0263 | 0.0347 | 0.0104 |
| kp-kernel Method | $p = 50$ | 0.0728 | 0.0649 | 0.0457 | 0.0517 | 0.0282 |
| | $p = 100$ | 0.0791 | 0.0878 | 0.0754 | 0.0877 | 0.0723 |
| | $p = 150$ | 0.0666 | 0.0878 | 0.0764 | 0.0903 | 0.0765 |
| | $p = 200$ | 0.0603 | 0.0853 | 0.0738 | 0.0883 | 0.0753 |

**Test3:** The next example is 4-dimensional multivariate Gaussian. The covariance of 4-dimensional multivariate Gaussian in the simulation is:

$$\Sigma = \begin{bmatrix} 18 & -3 & 2 & -4 \\ -3 & 11 & 1 & -2 \\ 2 & 1 & 1 & -2 \\ -4 & -2 & -2 & 4+\alpha \end{bmatrix},$$

The determinant of $\Sigma$, $\det \Sigma = 115\alpha$, hence when $\alpha$ goes to 0, the determinant of $\Sigma$ goes to 0. I study the entropy of the 4-dimensional Gaussian when $\alpha$ is close to 0. I estimate $H(f)$ for variant $\alpha$ and I set number of samples $n = 10000$ in Figure 2.5. In this simulation, it can be observed from the results that the kp-kernel outperforms other estimators especially kNN method when $\alpha$ is around 0. The kpN estimator's performance still better than kNN and kp-kernel when the correlation is high (i.e. determinant of covariance is nearly equal to 0). I set the parameter $\alpha = 0.01$, number of samples $n = 10000$, $k$ varies from 3 and 7, and set $p$ vary from 10 to 25. I compare the kNN, kpN and kp-kernel methods.

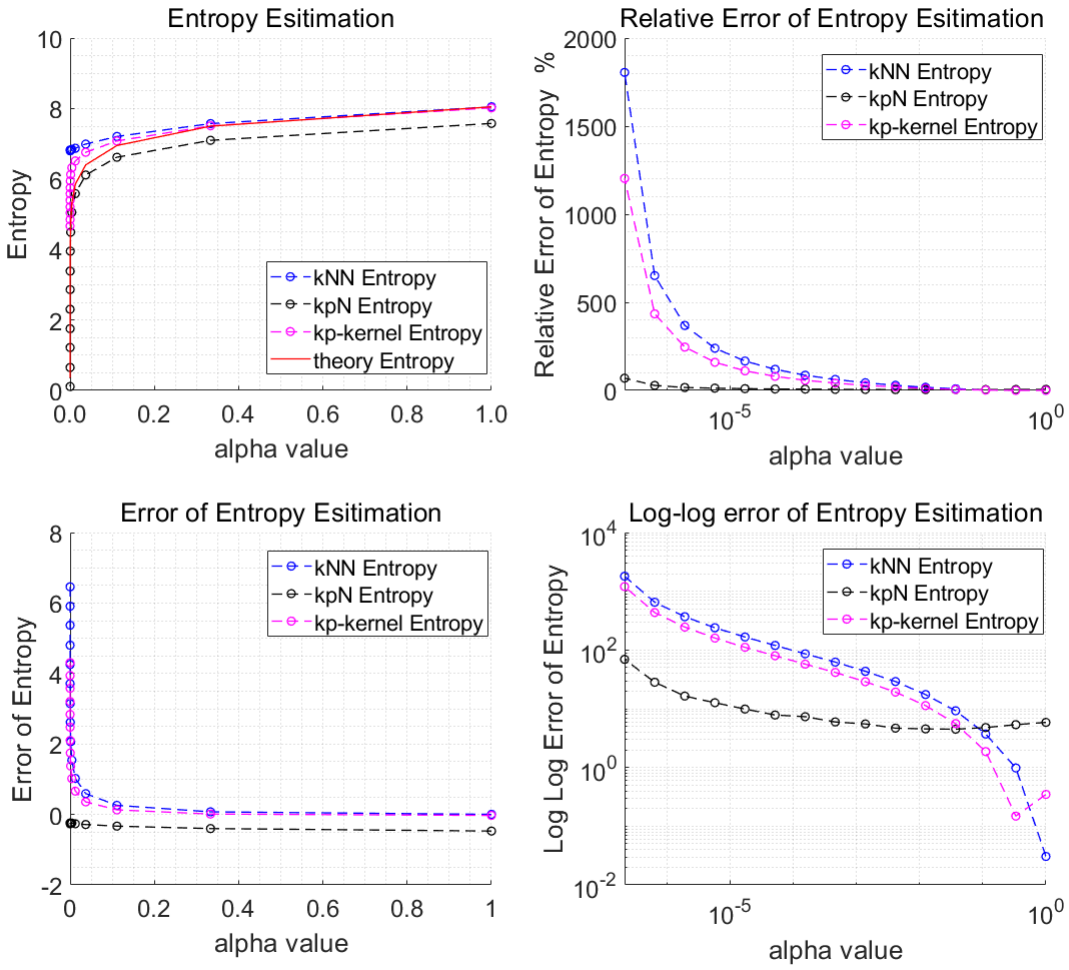


**Figure 2.5**: Error analysis for a 4-dimensional Gaussian distribution with determinant of covariance close to 0. The entropy estimation results for kNN (blue dot line), kpN (black dot line), kp-kernal (pink dot line) and theory (red solid line) results are presented on the top left of the figure. Relative Error of Entropy estimation is presented on the top right of figure. Error of entropy estimation is presented on the bottom left of figure. Log-log error of entropy estimation is presented on the bottom right of figure.

**Table 2.4**: Absolute errors table of the kNN, kpN, kp-kernel methods for 4-dimensional Gaussian Distribution with fixed value $\alpha = 0.01$.

| | | Different number of $k$ | | | | |
|---|---|---|---|---|---|---|
| | | 3 | 4 | 5 | 6 | 7 |
| kNN Method | | 1.1411 | 1.2116 | 1.3188 | 1.3341 | 1.4095 |
| kpN Method | $p = 50$ | 0.1026 | 0.1658 | 0.1998 | 0.2701 | 0.2788 |
| | $p = 100$ | 0.0220 | 0.0660 | 0.0803 | 0.1418 | 0.1348 |
| | $p = 150$ | 0.0147 | 0.0198 | 0.0262 | 0.0811 | 0.0676 |
| | $p = 200$ | 0.0380 | 0.0072 | 0.0039 | 0.0455 | 0.0269 |
| kp-kernel Method | $p = 50$ | 0.9362 | 0.9948 | 0.9332 | 0.9383 | 0.8881 |
| | $p = 100$ | 1.0208 | 1.0994 | 1.0545 | 1.0749 | 1.0384 |
| | $p = 150$ | 1.0725 | 1.1630 | 1.1285 | 1.1590 | 1.1314 |
| | $p = 200$ | 1.1069 | 1.2052 | 1.1785 | 1.2160 | 1.1948 |

### 2.4.2 A beta distribution

Consider the beta distribution function.

$$f(x) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} x^{\alpha-1}(1-x)^{\beta-1} x \in (0 \leq x \leq 1),$$

where $\Gamma$ is the Gamma function. The exact entropy for random variable $X$ is:

$$H(f) = \log(B(\alpha,\beta))) - (\alpha-1)\psi(\alpha) - (\beta-1)\psi(\beta) + (\alpha+\beta-2)\psi(\alpha+\beta),$$

where $\psi$ is the digamma function [32].

I set the beta distribution function parameter $\alpha = 2$ and $\beta = 8$. I estimate the error of the estimator, plot against sample size $N$ for each estimators. I also plot the relative error against sample size $N$. The results is shown in Figure 2.6. Note that all three method perform closely well when number of samples goes to large.

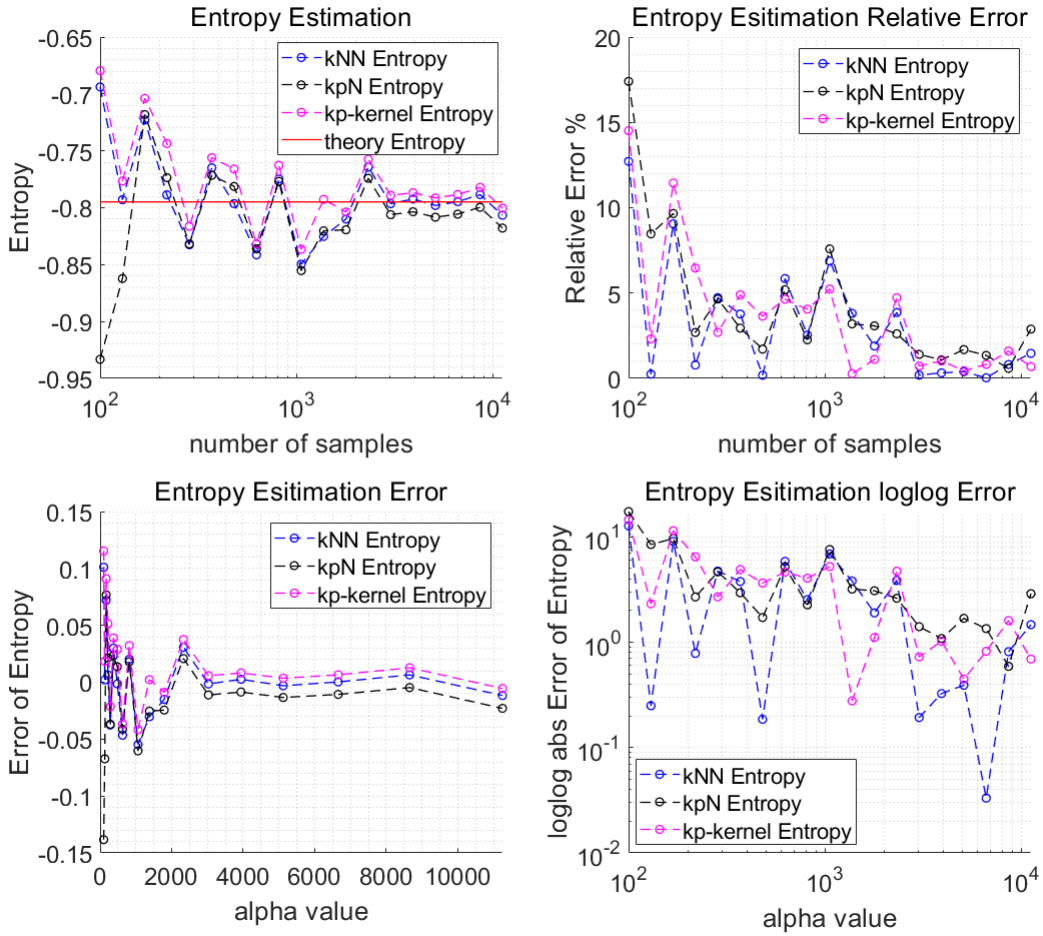


**Figure 2.6**: Error analysis for a 1-dimensional beta distribution with parameters $\alpha = 2$ and $\beta = 8$. The entropy estimation results for kNN (blue dot line), kpN (black dot line), kp-kernal (pink dot line) and theory (red solid line) results are presented on the top left of the figure. Relative Error of Entropy estimation is presented on the top right of figure. Error of entropy estimation is presented on the bottom left of figure. Log-log error of entropy estimation is presented on the bottom right of figure.

### 2.4.3 A parabola distribution

Consider a parabola distribution:

$$f(x) = \begin{cases} \frac{3}{4}(-x^2+1) & x \in [-1,1], \\ 0 & else. \end{cases}$$

The exact entropy for random variable $X$ is:

$$H(f) = \int_{[-1,1]} \log f(x) f(x) dx = 0.5681.$$

I use numerical method to calculate the integral of the parabola distribution. I estimate the error of the estimator, plot against sample size $n$ for each estimators. I also plot the relative error against sample size $n$. The results is shown in Figure 2.7.

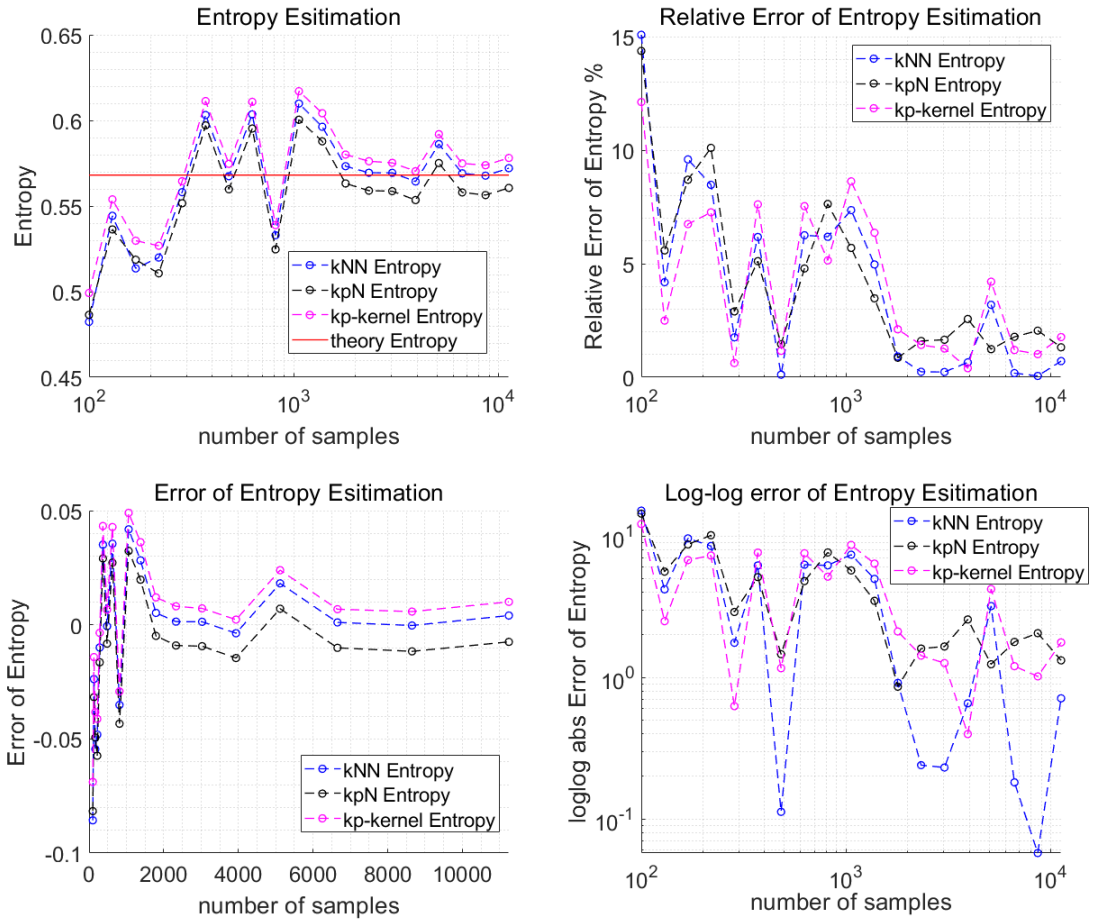


**Figure 2.7**: Error analysis for a 1-dimensional parabola distribution. The entropy estimation results for kNN (blue dot line), kpN (black dot line), kp-kernal (pink dot line) and theory (red solid line) results are presented on the top left of the figure. Relative Error of Entropy estimation is presented on the top right of figure. Error of entropy estimation is presented on the bottom left of figure. Log-log error of entropy estimation is presented on the bottom right of figure.

A kNN-based entropy estimator that is efficient in high dimensions and in the presence of large nonuniformity is proposed. The idea relies on the introduction of a Gaussian and Gaussian kernel interpolation, which in turn is based on an empirical evaluation of p nearest neighbors. By this introduction, the local nonuniformity of the underlying probability distribution is captured while retaining all the appealing computational advantages of classical kNN estimators. Across all the tests, the kpN estimator and proposed kp-kernel

estimator is shown consistently to outperform the classical kNN estimator. The main perspective of the current work is that the proposed estimator can be used to construct estimators for other quantities of interest such as mutual information, particularly in high dimensions.

### 2.4.4 Efficiency

In this section, I evaluate the efficiency of the estimators for different sample size $n$. I fix the ratio of local sample in the kpN method $pr_{kpN}$ and kp-kernel method $pr_{kpk}$ be the same 0.02 . In Figure 2.8, I plot the average computational time spent of each estimator variant sample size $n$ using the bivariate Gaussian model with correlation $\sigma = 0$. The computation is implemented in MATLAB R2019b, in a Laptop with h CPU 8th Generation Intel R CoreTM i7-8700 and RAM 16 GB.

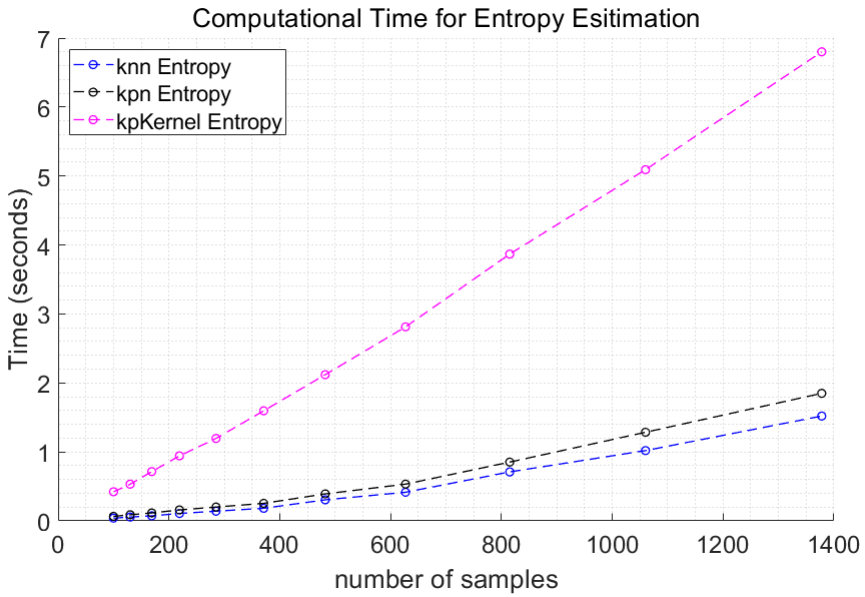


**Figure 2.8**: Computational Time for kNN, kpN and kp-kernel entropy estimators for bivariate Gaussian distribution. The ratio of local sample in kpN method $pr_{kpN} = 0.02$. The ratio of local sample in kp-kernel method $pr_{kpk} = 0.02$.

In Figure **2.11**, I plot the average time spent of each estimator variant number of local

sample in the kp-kernel method $p_{kpk}$ using the bivariate Gaussian model with correlation $\sigma = 0$. I can observe the time complexity for kp-kernel method is higher than kpN and kNN method since kp-kernel using calculate $p$ times integration when calculating the integration of local density function.

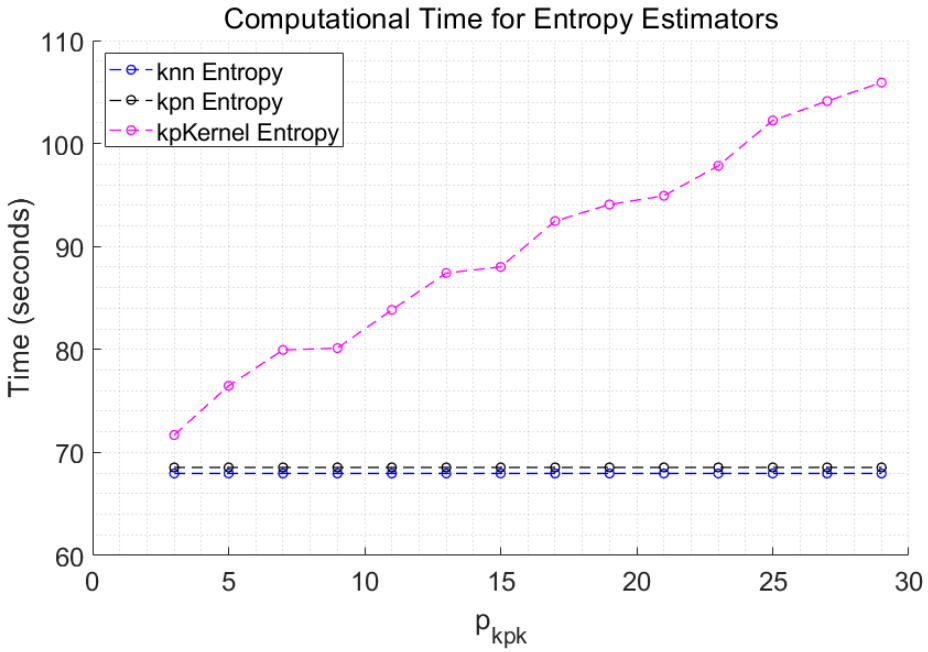


**Figure 2.9**: Computational Time for kNN, kpN and kp-kernel entropy estimators for bivariate Gaussian distribution. The number of local sample in kpN method is fixed $p_{kpN} = 20$. The number of local sample in kp-kernel method $p_{kpk}$ varies.

## 2.5 Convergence analysis

In this section, I will introduce a new framework for convergence analysis of the kNN-type estimators. It is comprised of two contributions: a statistical error related to the MC integration, and an analytical error resulting from the hypothesis of constant density in $\mathcal{B}(x_i, \epsilon_i)$. I apply a finite-sample analysis of a general framework derived by Singh and Poczos [41] for using the expectation of upper bound of the k-nearest neighbor statistics.

**Theorem 1.** *Let $x_1, ..., x_n$ be random samples from the distribution having PDF $f : \mathbb{R}^d \to \mathbb{R}$. The error between the Monte Carlo estimate of the entropy $H_n(f)$ and the entropy $H(f)$ converges to 0 in probability as the number of data $n \to \infty$, i.e.,*

$$\lim_{n\to\infty}[H_n(f) - H(f)] = 0 \quad \textit{in probability},$$

*the error converges to 0 in order of $n^{-1/2}$, regardless of the dimension $d$ of the integral. Furthermore, the error between the Monte Carlo estimate $H_n(f)$ and the entropy $H(f)$ in mean is 0, i.e.,*

$$\mathbb{E}[H_n(f) - H(f)] = 0.$$

*Proof.* Consider $\{x_i\}_{i=1}^n$ are independent random sample of random variable $X$ follows the probability density function $f$.

The weak law of large numbers (also called Khinchin's law) states that the sample average converges in probability towards the expected value. Hence,

$$\lim_{n\to\infty} H_n = \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} \log(\frac{1}{f(x_i)}) \to \mathbb{E}(-\log(f(X))) \quad \textit{in probability}.$$

This approximation converges, by the weak law of large numbers, as $n \to \infty$, to the analytical entropy $H(f)$. On the other hand, since $\{x_i\}_{i=1}^n$ are independent random sample

of random variable $X$,

$$\begin{aligned}\mathbb{E}[H_n(f)] &= \mathbb{E}[\frac{1}{n}\sum_{i=1}^{n}\log(\frac{1}{f(x_i)})] \\ &= -\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[\log f(x_i)] \\ &= -\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[\log f(X)] \\ &= H(f).\end{aligned}$$

□

**Theorem 2.** *Let $x_1, ..., x_n \in [-A, A]^d$ be random sample from the distribution having PDF $f : \mathbb{R}^d \to \mathbb{R}$, where $A > 0$ is a constant. $f \in C^2(\mathbb{R}^d)$ and $f \geq f_*$ ($f_*$ constant $> 0$) on $[-A, A]^d$. Then the error on the kNN entropy estimator converges to 0 in mean as the number of sample $n \to \infty$, i.e.,*

$$\lim_{n\to\infty}\mathbb{E}[H_n^{(kNN)} - H(f)] = 0.$$

*Proof.* The error between the kNN estimator $H_n^{(kNN)}$ and the exact entropy $H(f)$ can be divided into three parts.

$$H_n^{(kNN)} - H(f) = (H_n^{(kNN)} - H_n^{(P)}) + (H_n^{(P)} - H_n^{(MC)}) + (H_n^{(MC)} - H(f)), \qquad (2.5.1)$$

where

$$H_n^{(P)} = H_n^{(kNN)} + \frac{1}{n}\sum_{i=1}^{n}\log\frac{P_i}{P_i^{(kNN)}}.$$

For the first part of equation (2.5.1), the unbiasedness in mean for $\frac{1}{n}\sum_{i=1}^{n}\log\frac{P_i}{P_i^{(kNN)}}$ is presented in Lemma 2.

For the second part of equation (2.5.1), the kNN approximation of $P_i^{(kNN)}$ is

introduced, obtained:

$$\begin{aligned} H_n^{(P)} - H_n^{(MC)} &= H_n^{(kNN)} + \frac{1}{n}\sum_{i=1}^{n} \log \frac{P_i}{P_i^{(kNN)}} - H_n^{(MC)} \\ &= \frac{d}{n}\sum_{i=1}^{n} \log(\epsilon_i) + \log(c_d) + \psi(n) - \psi(k) + \frac{1}{n}\sum_{i=1}^{n} \log \frac{P_i}{P_i^{(kNN)}} - \frac{1}{n}\sum_{i=1}^{n} \log \frac{1}{f(x_i)}. \end{aligned} \tag{2.5.2}$$

Hence,

$$\begin{aligned} \sum_{i=1}^{n} \log \frac{P_i}{P_i^{(kNN)}} &= \sum_{i=1}^{n} \log P_i - \sum_{i=1}^{n} \log P_i^{(kNN)} \\ &= \sum_{i=1}^{n} \log P_i - \sum_{i=1}^{n} \log f(x_i) - \sum_{i=1}^{n} \log(\epsilon_i^d c_d). \end{aligned} \tag{2.5.3}$$

Combined equation (2.5.2) and equation (2.5.3) derive,

$$H_n^{(P)} - H_n^{(MC)} = \psi(n) - \psi(k) + \frac{1}{n}\sum_{i=1}^{n} \log P_i.$$

By equation (2.2.3) in section 2.2,

$$\mathbb{E}[\frac{1}{n}\sum_{i=1}^{n} \log P_i] = \psi(k) - \psi(n).$$

Therefore,

$$\mathbb{E}[H_n^P - H_n^{(MC)}] = 0.$$

For the third part of the error, by Theorem 1, $H_n^{(MC)}$ converges to $H(f)$ in mean. □

**Error in the approximation of the probability mass of kNN estimator**

The error of the kNN estimator is analysed. First, the error on the probability mass in a generic $\mathcal{B}(x_i, \epsilon)$ is computed, and the result is used to compute the error on the entropy. The analytical contribution to the error is due to the approximation of the probability

mass $P_i$. Consider a Taylor expansion of $f$ centered around $x_i$:

$$P_i = \int_{\mathcal{B}(x_i,\epsilon)} f(x_i) + (\xi - x_i) \cdot \nabla f(x_i) + \frac{1}{2}(\xi - x_i)^T \nabla^2 f(\eta_i)(\xi - x_i) d\xi, \tag{2.5.4}$$

where $\nabla^2 f(\eta_i)$ is the Hessian matrix of $f$ computed in $\eta_i$, $\eta_i \in \mathcal{B}(x_i, \epsilon)$. The first term of the series yields the kNN approximation $P_i^{(kNN)}$. Here, assume $P_i^{(kNN)} > 0$.

$$P_i^{(kNN)} = \int_{\mathcal{B}(x_i,\epsilon)} f(x_i) d\xi = \text{Vol}(\mathcal{B}(x_i, \epsilon)) f(x_i).$$

The second term in equation (2.5.4) vanishes since it is the integral of an even function over a symmetric interval.

$$P_i = P_i^{(kNN)} + \frac{1}{2} \int_{\mathcal{B}(x_i,\epsilon)} (\xi - x_i)^T \nabla^2 f(\eta_i)(\xi - x_i) d\xi.$$

Define

$$h_i = \frac{1}{2} \int_{\mathcal{B}(x_i,\epsilon)} (\xi - x_i) \nabla^2 f(\eta_i)(\xi - x_i) d\xi.$$

A standard estimation for quadratic form is derived:

$$\lambda^{\min} ||\xi - x_i||^2 \leq (\xi - x_i)^T \nabla^2 f(\eta_i)(\xi - x_i) \leq \lambda^{\max} ||\xi - x_i||^2, \tag{2.5.5}$$

where $\lambda^{\min} = -||\nabla^2 f||_\infty$ and $\lambda^{\max} = ||\nabla^2 f||_\infty$.

Then, the integral of the quadratic form over $\mathcal{B}(x_i, \epsilon)$ is,

$$\int_{\mathcal{B}(x_i,\epsilon)} ||\xi - x_i||^2 d\xi = \sum_{i=1}^{d} \int_{\mathcal{B}(x_i,\epsilon)} (\xi_j - x_{i,j})^2 d\xi. \tag{2.5.6}$$

Let $\mathcal{B}(x_i, \epsilon) = [x_{i,j} - \epsilon_i, x_{i,j} + \epsilon_i] \times [x_{i,k} - \epsilon_i, x_{i,k} + \epsilon_i]^{d-1}$, $k \neq j$. This integral can be computed for just one $j$ and then multiplied by $d$.

$$\int_{\mathcal{B}(x_i,\epsilon)} (\xi_j - x_{i,j})^2 d\xi = (2\epsilon)^{d-1} \int_{-\epsilon}^{\epsilon} \eta^2 d\eta = (2\epsilon)^{d-1} \frac{2}{3}\epsilon^3. \tag{2.5.7}$$

By equation (2.5.6) and equation (2.5.7). We obtained the bounds for $h_i$:

$$\int_{\mathcal{B}(x_i,\epsilon)} ||\xi - x_i||^2 d\xi = \frac{2}{3} 2^{d-1} d\epsilon^{d+2}. \tag{2.5.8}$$

Combine equation (2.5.5) and equation (2.5.8), the bound of $h_i$:

$$\frac{\lambda^{\min}}{3} d 2^{d-1} \epsilon^{d+2} \leq h_i \leq \frac{\lambda^{\max}}{3} d 2^{d-1} \epsilon^{d+2},$$

where $\lambda^{\min} = -||\nabla^2 f||_\infty$ and $\lambda^{\max} = ||\nabla^2 f||_\infty$.

**Lemma 2.** *Let $x_1$, $x_2$, ... $x_n \in [-A, A]^d$ be random sample from the distribution having PDF $f : \mathbb{R}^d \to \mathbb{R}$, where $A > 0$ is a constant. Assume $f \geq f_* = constant > 0$ on $[-A, A]^d$, $f \in C^2(\mathbb{R}^d)$.*

$$\mathbb{E}[\frac{1}{n} \sum_{i=1}^{n} \log(1 + \frac{h_i}{P_i^{(kNN)}})] \to 0 \text{ as } n \to \infty.$$

*Proof.* Divide the $\sum_{i=1}^n \log(1 + \frac{h_i}{P_i^{(kNN)}})$ into two parts,

$$\sum_{i=1}^{n} \log(1 + \frac{h_i}{P_i^{(kNN)}}) = \sum_{+} \log(1 + \frac{h_i}{P_i^{(kNN)}}) + \sum_{-} \log(1 + \frac{h_i}{P_i^{(kNN)}}). \tag{2.5.9}$$

where

$$\sum_{+} \log(1 + \frac{h_i}{P_i^{(kNN)}}) = \sum_{f(x_i) \geq f_* \ \frac{h_i}{P_i^{(kNN)}} > -1/2} \log(1 + \frac{h_i}{P_i^{(kNN)}}),$$

and

$$\sum_{-} \log(1+\frac{h_i}{P_i^{(kNN)}}) = \sum_{f(x_i)\geq f_* \ \frac{h_i}{P_i^{(kNN)}} \leq -1/2} \log(1+\frac{h_i}{P_i^{(kNN)}}).$$

By Lemma 3 below, for $x > -\frac{1}{2}$,

$$\frac{x}{1+x} \leq \log(1+x) \leq x. \tag{2.5.10}$$

After setting $x = \frac{h_i}{P_i^{(kNN)}}$ in equation (2.5.10), we get

$$\frac{h_i}{h_i + P_i^{(kNN)}} \leq \log(1+\frac{h_i}{P_i^{(kNN)}}) \leq \frac{h_i}{P_i^{(kNN)}}.$$

To derive the lower bound of $\log(1+\frac{h_i}{P_i^{(kNN)}})$, I consider the left hand side,

$$\frac{h_i}{h_i + P_i^{(kNN)}} \geq \frac{\min(h_i)}{\max(h_i) + P_i^{(kNN)}} \geq \frac{\lambda^{\min} d 2^{d-1} \epsilon_i^{d+2}}{\lambda^{\max} d 2^{d-1} \epsilon_i^{d+2} + 3P_i^{(kNN)}}. \tag{2.5.11}$$

To derive the upper bound of $\log(1+\frac{h_i}{P_i^{(kNN)}})$, let us consider the right hand side,

$$\frac{h_i}{P_i^{(kNN)}} \leq \frac{\max(h_i)}{P_i^{(kNN)}} \leq \frac{\lambda^{\max} d 2^{d-1} \epsilon_i^{d+2}}{3P_i^{(kNN)}}. \tag{2.5.12}$$

By using equation (2.5.11) and (2.5.12), we can derive the error bound

$$\begin{aligned} \frac{d2^{d-1}}{3n} \sum_{+} \frac{(\lambda^{\min})\epsilon_i^{d+2}}{(\lambda^{\max}) d 2^{d-1} \epsilon_i^{d+2} + 3P_i^{(kNN)}} &\leq \frac{1}{n} \sum_{+} \log(1+\frac{h_i}{P_i^{(kNN)}}) \\ &\leq \frac{d2^{d-1}}{3n} \sum_{+} \frac{\lambda^{\max}}{P_i^{(kNN)}} \epsilon_i^{d+2}. \end{aligned}$$

By Lemma 5 below with $\alpha = 2$, we obtain

$$\mathbb{E}(\frac{d2^{d-1}}{3n}\sum_{+}\frac{\lambda^{\min}\epsilon_i^{d+2}}{\lambda^{\max}d2^{d-1}\epsilon_i^{d+2}+3P_i^{(kNN)}}) =$$
$$\mathbb{E}(\frac{d2^{d-1}}{3n}\sum_{+}\frac{\lambda^{\min}\epsilon_i^{2}}{\lambda^{\max}d2^{d-1}\epsilon_i^{2}+3f(x_i)}) \to 0 \ as \ n \to \infty.$$

and

$$\mathbb{E}(\frac{d2^{d-1}}{3n}\sum_{i=1}^{n}\frac{\lambda^{\max}}{P_i^{(kNN)}}\epsilon_i^{d+2}) \to 0 \ as \ n \to \infty.$$

Hence,

$$\mathbb{E}(\frac{1}{n}\sum_{+}\log(1+\frac{h_i}{P_i^{(kNN)}})) \to 0 \ as \ n \to \infty$$

Secondly, consider $\frac{h_i}{P_i^{(kNN)}} \leq -\frac{1}{2}$, summuation terms $\sum_{-}\log(1+\frac{h_i}{P_i^{(kNN)}})$.
Let $\eta_n = \max_{i,f(x_i)\geq f_*,h_i\leq 0}\frac{|h_i|}{f(x_i)c_d\epsilon_i^d}$, Hence, $\forall i$

$$\frac{|h_i|}{f(x_i)c_d\epsilon_i^d} \leq \frac{\frac{1}{2}||\nabla^2 f||_\infty\epsilon_i^2}{f_*c_d} = \frac{C}{f_*}\epsilon_i^2,$$

where $C = \frac{\frac{1}{2}||\nabla^2 f||_\infty}{c_d}$. Hence,

$$\log\frac{1}{2} \leq \log(1-\eta_n) \leq \eta_n \leq \max_{i,f(x_i)\geq f_*,h_i\leq 0}\frac{|h_i|}{f(x_i)c_d\epsilon_i^d} \leq \frac{C}{f_*}\epsilon_i^2. \tag{2.5.13}$$

By Lemma 5, take $\alpha = 2$, then

$$\mathbb{E}[\epsilon_i^2] \leq C_2(\frac{k}{f_*n})^{\frac{2}{d}}. \tag{2.5.14}$$

Combined by equation (2.5.13) and (2.5.14), derive

$$\mathbb{E}[\eta_n] \leq \frac{CC_2}{f_*}(\frac{k}{f_*n})^{\frac{2}{d}} \to 0 \ as \ n \ \to \ \infty. \tag{2.5.15}$$

Since

$$\frac{1}{n}\sum_{-}\log(1-\eta_n) \leq \frac{1}{n}\sum_{-}\log(1+\frac{h_i}{P_i^{(kNN)}}) \leq 0.$$

To show $\lim_{n\to\infty}\sum_{-}\frac{1}{n}\log(1+\frac{h_i}{P_i^{(kNN)}}) = 0$, only need to show $\lim_{n\to\infty} E(\frac{1}{n}\sum_{-}\log(1-\eta_n)) = 0$.

Proof by contradiction: Assume there exists a constant $\beta < 0$, s.t.

$$\lim_{n\to\infty}\mathbb{E}[\log(1-\eta_n)] = \beta < 0. \tag{2.5.16}$$

By equation (2.5.15), Since $\eta_n$ converges to 0 in mean, by theorem 2.30 in Folland [17], there exists a subsequence $\eta_{n_k}$, s.t. $\{\eta_{n_k}\}$ converges to 0. Since by equation (2.5.16),

$$\lim_{k\to\infty}\mathbb{E}[\log(1-\eta_{n_k})] = \beta < 0.$$

One the other hand, since $\eta_{n_k}$ is bounded by

$$\max(\frac{1}{2}, \max_{i,h_i\leq 0}\frac{|h_i|}{f(x_i)c_d\epsilon_i^d}) \in L^1.$$

By the Lebesgue Dominated Convergence Theorem [17]

$$\lim_{k\to\infty}\int\log(1-\eta_{n_k})dp = \int\lim_{k\to\infty}\log(1-\eta_{n_k})dp = 0.$$

This leads to a contradiction. Then $\lim_{n\to\infty} E(\log(1-\eta_n)) \geq 0$, and $\lim_{n\to\infty} E(\log(1-\eta_n)) = 0$. Hence,

$$\lim_{n\to\infty}\mathbb{E}[\frac{1}{n}\sum_{-}\log(1+\frac{h_i}{P_i^{(kNN)}})] = 0.$$

Therefore,

$$\mathbb{E}[\frac{1}{n}\sum_{i=1}^{n}\log(1+\frac{h_i}{P_i^{(kNN)}})] \to 0 \; as \; n\to\infty.$$

□

**Lemma 3.** *For all $x > -1$,*

$$\frac{x}{1+x} \leq \log(1+x) \leq x.$$

*Proof.* Let $t = x + 1 > 0$, we consider the function $f(t) = \log(t) - t + 1$,

$$f'(t) = \frac{1}{t} - 1.$$

We have $f'(1) = 0$. When $t > 1$, $f'(1) < 0$. When $t < 1$, $f'(1) > 0$. Hence, $f(t) < 0$ for all $t > 0$. For the lower bound, consider the inequality,

$$\log y \leq y - 1,$$

for $y > 0$, Let $t = \frac{1}{y} > 0$. We have

$$\log(\frac{1}{t}) \leq \frac{1}{t} - 1 \Leftrightarrow \log(t) \geq \frac{t-1}{t},$$

Let $x = t - 1$. We then derive

$$\frac{x}{1+x} \leq \log(1+x).$$

□

**Lemma 4.** *Let $x_1, ...x_n \in [-A, A]^d$ be random sample from the distribution having PDF $f : \mathbb{R}^d \to \mathbb{R}$, where $A > 0$ is a constant. Suppose density function $f : [-A, A]^d$ satisfy the tail condition.*

$$\mathbb{E}[\int_{\rho}^{\infty} [1 - P(B(X, f^{-1}(r)))]^n] \leq \frac{C_T}{n}$$

*for some constant $C_T > 0$. Suppose $F : (0, \rho) \to \mathbb{R}$ is continuously differentiable, with*

$F' > 0$. *Then for any* $x \in [-A, A]^d$, *we have the upper bound*

$$\mathbb{E}[F(\epsilon_k(x))] \leq F((\frac{k}{f_* n})^{\frac{1}{d}}) + \frac{(e/k)^k}{d(nf_*)^{\frac{1}{d}}} \int_k^{nf_*\rho^d} e^{-y} y^{\frac{dK+1-d}{d}} F'((\frac{y}{nf_*})^{\frac{1}{d}}) dy,$$

*where* $F$ *is a nonnegative function.*

*Proof.* See Singh and Poczos [40]. □

**Lemma 5.** *Consider* $F(x) = x^\alpha, \alpha > 0$ *in the setting of Lemma 4, we have*

$$\mathbb{E}[\epsilon_k^\alpha(x)] \leq C_2 (\frac{k}{f_* n})^{\frac{\alpha}{d}},$$

*where* $C_2 = 1 + \frac{\alpha}{d}$.

*Proof.* See Singh and Poczos [40]. □

# Chapter 3

# Entropy Estimation for a Particle System

## 3.1 Entropy for a particle system

Consider a canonical ensemble of mechanical system. Each system contains $N$ particles in a volume $V$ computational box. For each $i$th particle, the mass of the particle is $m_i$, the position of the particle is $\boldsymbol{r}_i$, the momentum of the particle is $p_i = m_i\dot{\boldsymbol{r}}_i$ $(i = 1, ..., N)$. The phase space $\Gamma = \{(\boldsymbol{p}, \boldsymbol{r}) : \boldsymbol{p} = (\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_N), \boldsymbol{r} = (\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_N) \in \mathbb{R}^{3N}\}$. Define Hamiltonian $H : \Gamma \to \mathbb{R}$ by

$$H(\boldsymbol{p}, \boldsymbol{r}) = \sum_{k=1}^{N} \frac{1}{2m}|p_i|^2 + V(\boldsymbol{r_1}, ..., \boldsymbol{r_N}) = K(\boldsymbol{p}) + V(\boldsymbol{r}),$$

where $K$ is the momentum potential, and $V$ is the internal potential. We have the $N$ body distribution in positions and momenta

$$f_N = f_N(\{\boldsymbol{r}\}_N, \{\boldsymbol{p}\}_N) = g_N(\{\boldsymbol{r}\}_N) \prod_{k=1}^{N} f_k(\boldsymbol{p}_k),$$

where $f_k$ is the $k$th particle one-body distribution of momenta, $g_N$ is the $N$ particle correlation function. The total entropy of a bulk fluid is given by

$$S_{liquid} = -\frac{Rh^{3N}}{N!}\int\int f_N \log f_N d\boldsymbol{r}_N d\boldsymbol{p}_N,$$

with $R$ the gas constant and $h$ the Planck constant. Then, the separability of the momentum can be exploited to give

$$S_{liquid} = S_{mom} + S_{conf},$$

which is written explicitly as

$$S_{liquid} = \underbrace{-\frac{NR}{\rho}\int f_1(\boldsymbol{p}_1)\log f_1(\boldsymbol{p}_1)d\boldsymbol{p}_1}_{S_{mom}} - \underbrace{\frac{R\rho^N}{N!}\int g_N(\boldsymbol{r})\log g_N(\boldsymbol{r})d\boldsymbol{r}}_{S_{conf}}.$$

The momentum terms are the same as those of an ideal gas where the one-body distribution of momenta is given by

$$f_1(\boldsymbol{p}) = \rho(2\pi mkT)^{-3/2}\exp\Big(\frac{\boldsymbol{p}^2}{2mkT}\Big),$$

where $k$ is the Boltzmann constant, $\rho$ is the number density of the equivalent ideal gas, and $m$ is the mass of the atom. Then,

$$S_{mom} = -\frac{NR}{\rho}\int f_1(\boldsymbol{p}_1)\log f_1(\boldsymbol{p}_1)d\boldsymbol{p}_1 = \frac{3NR}{2} - NR\log(\rho\lambda^3),$$

where $\lambda$ the thermal wavelength of an atom in the liquid. By the theory of conditional entropy [19],

$$\underbrace{\frac{R\rho^N}{N!}\int g_N(\boldsymbol{r})\log g_N(\boldsymbol{r})d\boldsymbol{r}}_{S_{conf}} = S_2 - I_3 + I4 - ...$$

For a system with a solute,

$$\underbrace{\frac{R\rho^N}{N!}\int g_N(\boldsymbol{r})\log g_N(\boldsymbol{r})d\boldsymbol{r}}_{S_{conf}} = S_{1solute} - I_{2solute} + I_{3solute} - ...$$

It is at this point I choose a truncation of $S_{configuration}$. The most severe truncation generates what I will call the conditional one particle entropy (C1PE),

$$\Delta S_{1|S} = S_{1solute} = -R\rho_0 \int g_N^{(1)}(\boldsymbol{r_1}|s)\log g_N^{(1)}(\boldsymbol{r_1}|s)d\boldsymbol{r}_1.$$

$g_N^{(1)}$ is the unitless ratio of the number density of molecular at r in the presence of the solute to the bulk density, $g_N^{(1)} = \frac{\rho(r)}{\rho_0}$, $\rho_0$ is the number of bulk density, $V\rho_0 = N$. $\rho(r)$ is the one-point number density of molecular at locations $r$. This is achieved by removing all integrals with a subscript greater than 1.

It is convenient to construct estimator to measure the quantity $S_{1solute}$

$$S_{1solute} = -R\rho_0 \int g_N^{(1)}(\boldsymbol{r_1}|s)\log g_N^{(1)}(\boldsymbol{r_1}|s)d\boldsymbol{r}_1. \tag{3.1.1}$$

Replace the $g_N^{(1)}$ by $\frac{\rho(r)}{\rho}$ in equation (3.1.1),

$$\begin{aligned} S_{1solute} &= -R\rho_0 \int g_N^{(1)}(\boldsymbol{r_1}|s)\log g_N^{(1)}(\boldsymbol{r_1}|s)d\boldsymbol{r}_1 \\ &= -R\int \rho_0 g_N^{(1)}(\boldsymbol{r_1}|s)\log[\rho_0 g_N^{(1)}(\boldsymbol{r_1}|s)/\rho_0]d\boldsymbol{r}_1 \\ &= -R\int \rho(\boldsymbol{r_1})[\log\rho(\boldsymbol{r_1}) - \log\rho_0]d\boldsymbol{r}_1 \\ &= -R\int \rho(\boldsymbol{r_1})\log\rho(\boldsymbol{r_1})d\boldsymbol{r}_1 + RN\log\rho_0. \end{aligned} \tag{3.1.2}$$

Use the normalized density function $f = \rho/N$ in equation (3.1.2),

$$\begin{aligned} S_{1solute} &= -R \int \rho(r) \log \rho(r) d\boldsymbol{r}_1 + RN \log \rho_0 \\ &= -R \int fN \log fN d\boldsymbol{r}_1 + RN \log \rho_0 \\ &= -RN \int [f \log f + \log N] d\boldsymbol{r}_1 + RN \log \rho_0 \\ &= -RN \int [f \log f] d\boldsymbol{r}_1 + RN \log N - RN \log \rho_0 = -RN \int [f \log f] d\boldsymbol{r}_1 + RN \log V. \end{aligned} \tag{3.1.3}$$

In latter section, I will focus on calculating the C1PE entropy $S_{1solute}$ to represents the entropy I interested in the particle systems. In the following, I mainly consider two particle system: Lennard-Jones system and Ionic system.

**A Lennard-Jones system.** Consider a molecular system with $N$ neutral particles. For a given configuration of the system, the Hamiltonian is defined to be the work needed to bring all the molecular from infinity to their current positions. It is the sum of all pairwise interaction energies between all the particles, including the fixed particle in the center of the computational box. I only consider the hard-sphere contribution, which defined simply as impenetrable spheres that cannot overlap in space and the Lennard-Jones potential. Therefore, define the total potential energy of the system to be

$$U = \sum_{1 \le i < j \le N} u_{ij},$$

where

$$u_{ij} = 4\epsilon[\Big(\frac{\sigma}{r}\Big)^{12} - \Big(\frac{\sigma}{r}\Big)^{6}].$$

where $r$ is the distance between particle $i$th and $j$th particle.

**An Ionic system.** Consider an electrolyte with $m$ species of ions. For each $i$ $(1 \le i \le n)$, denote by $z_i$ the valence and $v_i$ the volume of an ion of the $i$th species.

Denote $N_i$ the total number of ions of the $i$th species. The total number of all ions is $N = \sum_{i=1}^{m} N_i$. Assume that there is a spherical colloidal particle - a charged macroion - of radius $R$ inside the electrolyte solution and that its charge effect is described effectively by a constant surface charge density, denoted $\sigma$. Assume the system charge neutrality

$$ze + \sum_{i=1}^{M} N_i z_i e = 0,$$

where $z = 4\pi R^2 \sigma / e$ is the valence of the macroion and $e$ is the elementary charge.

For a given configuration of the system, the Hamiltonian is defined to be the work needed to bring all the ions from infinity to their current positions. It is the sum of all pairwise interaction energies between all the ions, including the macroion. Only consider the hard-sphere contribution and the Coulomb interaction. Therefore, define the total potential energy of the system to be

$$U = \sum_{0 \le j < k \le N} u_{jk},$$

where

$$\beta u_{jk} = \begin{cases} \dfrac{l_B \hat{z}_j \hat{z}_k}{r_{jk}} & if \quad r_{jk} \ge \hat{R}_j + \hat{R}_k, \\ \infty & else. \end{cases}$$

Here $\beta = (k_B T)^{-1}$, $l_B = e^2 \beta / (4\pi\epsilon\epsilon_0)$ is the Bjerrum length, and $r_{jk}$ is the center-center distance between $i$th and $k$th ions. Consider the water solvent at room-temperature and thus take $l_B = 7$Å.

For both Lennard-Jones and ionic system, I will assume to use periodic boundary condition.

## 3.2 Monte Carlo Simulations

In this section, I use an unrestricted primitive model for our underlying particle system and apply the canonical ensemble MC simulations with the Metropolis criterion [2, 18, 49]. Consider $n$ particles in a computational box, I will examine a simple Monte Carlo simulation, whose energy function is given by:

$$U = \sum_{1 \le i < j \le N} U_{i,j},$$

where $U_{i,j}$ is the internal potential between $i$th particle and $j$th particle. The simulation will be performed at fixed reduced temperature $T$. Our simulation progresses through iterations of the following basic Monte Carlo step:

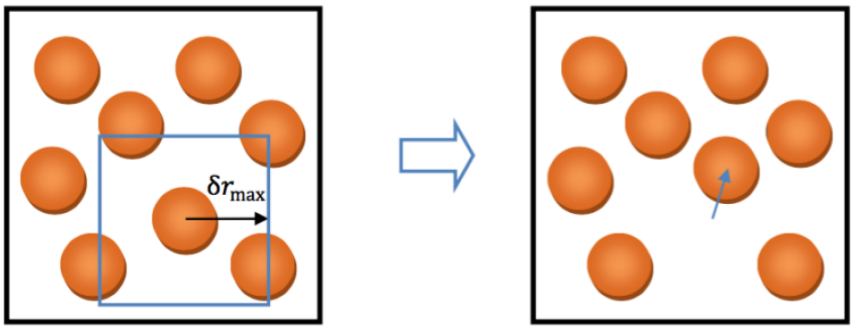


**Figure 3.1**: One Monte Carlo Step in Particle System

- Randomly pick one of $N$ particles.
- Perturb each of the $x, y, z$ coordinates separately by three random values taken from the uniform distribution on the interval $[-r_{max}, r_{max}]$. Here, $r_{max}$ is the maximum displacement.
- Compute the change in potential energy due to the particle move, $\Delta U = U_{new} - U_{old}$.

- Use the Metropolis criterion to decide whether or not to accept the move, i.e. If $\Delta U < 0$, accept the move. If $\Delta U > 0$, compute $p^{acc} = e^{-\Delta U \beta}$. Draw a random number $r$ on the interval $[0.0, 1.0]$ and accept the move if and only if $p^{acc} > r$.

- If the move is accepted, keep the new configuration and update any running averages with it (e.g., the potential energy). If the move is rejected, discard the new configuration and update any running averages with the original state.

In canonical Metropolis MC simulations, one MC move is accepted if

$$p^{acc} = e^{-\Delta U \beta} > r.$$

and rejected otherwise. Here, $\Delta U$ is the change of internal energies of the N-particle simulation ensemble before and after the one possible MC move, respectively. $r$ denotes a random number which is uniformly distributed over the interval $[0, 1]$. After a sufficient number of such displacements, the potential energies of the molecules will conform to a Boltzmann distribution.

### 3.2.1 Simulation results of a Lennard-Jones system

In Lennard-Jones system, I divided the entire MC simulation into two parts: acceleration part and equilibration. I dynamically change the value of $\beta$ in the first part of moves, to speed up the thermal equilibration of the crowded system of particles. I generate a geometrical sequence of $10^5 N$ terms with the first and last terms being $\beta = 1$ and $\beta = 7$, respectively. In the $m$th MC move with $m \leq 10^5 N$, the parameter $\beta$ is taken to be the $m$th term in the geometrical sequence. After the first $10^5 N$ moves, I fix $\beta = 7$ for all of the rest MC moves. I run another $9 \times 10^5 N$ moves so that the system can reach an equilibrium. I start to record the equilibrium state after $6 \times 10^5 N$.

**Table 3.1**: Parameters Setting for Lennard-Joines Fluids Monte Carlo Simulation

| | |
|---|---|
| Number of particles $n$ | 900 |
| Computational Box Size $L$ | 27.376Å |
| Total Monte Carlo Steps | $10^6 n$ |
| initial maximum length $\Delta_{max}$ | 1Å |
| LJ parameter $\epsilon$ | 0.215 $[kcal/mol]$ |
| LJ parameter $\sigma$ | 3.06 Å |

See Table 3.1 for key parameters setting for Lennard-Jones Monte Carlo Simulation. Throughout the entire simulation, I keep the percentage of acceptance of MC moves by adaptively adjusting the value of the maximum length $\Delta_{max}$. Initially, I set $\Delta_{max} = 1$Å. Then change it after every 100 moves. If the acceptance rate is larger than 50 in current 100 moves, I increase max by multiplying it by 1.05 but always keep the new value of max to be less than or equal to 2Å. If the acceptance rate is smaller than 20 in current 100 moves, I decease max by multiplying it by 0.95, and I keep the new max to be greater. In Figure 3.2, I show the initial of the positions of the particles in initial stage and steady state. In the left panel of Figure 3.2, the points colored by blue are the initial positions of particles. In the right panel of Figure 3.2, the points colored by blue are the final positions of particles.

### 3.2.2 Simulation results of an ionic system

In ionic system, the entire MC simulation is divided into two parts: acceleration part and equilibration part. I dynamically change the value of $\beta$ in the first part of moves, a total of $10^5 N$ of them, to speed up the thermal equilibration of the crowded system of particles. Generate a geometrical sequence of $10^5 N$ terms with the first and last terms being $\beta = 1$ and $\beta = 7$, respectively. In the $m$th MC move with $m \leq 10^5 N$, the parameter $\beta$ is taken to be the $m$th term in the geometrical sequence. After the first $10^5 N$ moves, fix $\beta = 7$ for all of the rest MC moves. I run another $9 \times 10^5 N$ moves so that the system can


**Figure 3.2**: A Lennard-Jones system initial and steady state positions

reach an equilibrium.

**Table 3.2**: Parameters Setting for ionic Monte Carlo Simulation

| | |
|---|---|
| Number of particles $n$ | 200 |
| charge of ion $n$ | $+1$ |
| radius of ion $n$ | 1Å |
| Computational Box Size $L$ | 50Å |
| Total Monte Carlo Step | $10^6 n = 10^8$ |
| value of the maximum length $\Delta_{max}$ | 0.5Å |

See Table 3.2 for key parameters setting for ionic Monte Carlo Simulation. Throughout the entire simulation, I keep the percentage of acceptance of MC moves by adaptively adjusting the value of the maximum length $\Delta_{max}$. Initially, I set $\Delta_{max} = 0.5$Å. I then change it after every 100 moves. If the acceptance rate is larger than 50 in current 100 moves, I increase max by multiplying it by 1.05 but always keep the new value of max to be less than or equal to 2Å. If the acceptance rate is smaller than 20 in current 100 moves, I decease max by multiplying it by 0.95, and I keep the new max to be greater. The total accepted rate is: 35.67%. In Figure 3.3, I show the initial of the positions of the particles

in initial stage and steady state. In the left panel of Figure 3.3, the points colored by blue are the initial positions of particles. In the right panel of Figure 3.3, the points colored by blue are the final positions of particles.

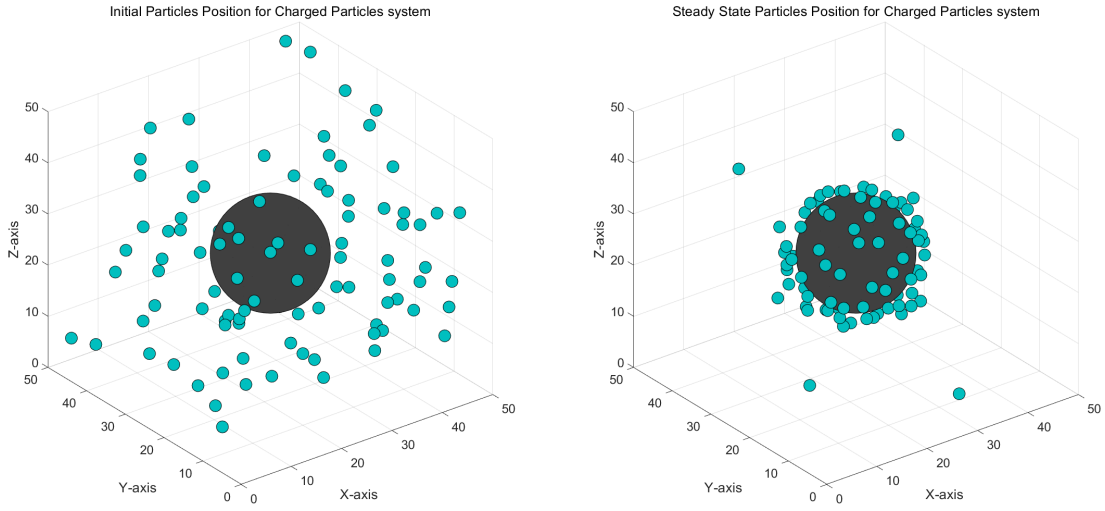


**Figure 3.3**: An ionic system initial and steady State Positions

## 3.3 Numerical entropy calculation of particle systems

In this section, I compare three different entropy estimation methods: KL method, kpN method and kp-kernel methods under two particle systems: Lennard-Jones system and ionic system. In Lennard-Jones system, I compare the entropy result with Huggins [25]. In ionic system, I apply histogram method as benchmark to compare with the kNN and kpN method.

The histogram method uses MC sampling data to estimate the various densities in the equation (3.1.3) by counting the instances of molecules in each voxels:

$$f(l) = \frac{n_l}{n_f V_{vox}}.$$

Here, $n_l$ is the number of MC frames for which a water is found in voxels $l$, $n_f$ is the number of MC frames, $V_{vox} = R_{vox}^3$ is the volume of cubic voxel, where $R_{vox}$ is the length of the voxel. Thus, the histogram method estimates equation (3.1.3) is

$$S_{hist} = -R[\sum_{l \in L}[\frac{n_l}{n_f} V_{vox} \log(\frac{n_l}{n_f} V_{vox}) - 3\log(R_{vox}) - \log(V)R]. \tag{3.3.1}$$

Here, the contribution of the entropy in region $L$ is estimated in terms of the quantities summed over voxels $l \in L$.

The kNN estimator for an C1PE term, given $F$ sufficiently uncorrelated MD frames containing $N$ particles, is given by the expression

$$S_{kNN} = \frac{1}{NF} \sum_{i=1}^{N} \sum_{j=1}^{F} \log(\frac{4NF\pi\epsilon_{ij}^3}{3V}) - \psi(k).$$

The kpN estimator for an C1PE term, is given by the expression

$$S_{kpN} = \frac{1}{NF} \sum_{i=1}^{N} \sum_{j=1}^{F} \log(\frac{NF\pi G_{p,ij}}{g_{p,ij} V}) - \psi(k).$$

**A Lennard-Jones system**

Under the Monte Carlo Computation Setting in section 3.2.3, I extract the MC samples when the system reach equilibrium, i.e. the samples after $1 \times 10^5 N$ MC steps. I record the particles positions for every 200 MC cycles after the system reach equilibrium for getting uncorrelated sample data in the computational box. I apply kNN, kpN and histogram method to compute the entropy of Lennard-Jones system and compared with the benchmark (C1PE) entropy value is: $-0.541$ [25]. For each simulation, I set the local number of samples parameter $p = 0.07 \times n$, where $n$ represents the number of samples. For histogram method, I set the length of the voxel $R_{vox} = 0.07$Å. I set the length of the voxel

$R_{vox} = 0.07$Å in histogram method.

**Table 3.3**: kNN, kpN methods for Lennard-Jones System

| | k = 1 | | k = 3 | | k = 5 | |
|---|---|---|---|---|---|---|
| | kNN | kpN | kNN | kpN | kNN | kpN |
| num of frames = 100 | -1.0524 | -0.9855 | -0.9347 | -0.8858 | -0.6718 | -0.7498 |
| num of frames = 200 | -0.8987 | -0.8599 | -0.8595 | -0.8382 | -0.652 | -0.7116 |
| num of frames = 400 | -0.8217 | -0.7833 | -0.8114 | -0.7784 | -0.648 | -0.6868 |

In Table 3.3, I present the entropy calculation results for cases $k = 1$, $k = 3$ and $k = 5$. It clearly state that when number of frames increase, the estimated value is getting closer to the exact experiment result $-0.541$. The parameter $k = 5$ is better than the other two cases.

**Table 3.4**: kNN, kpN methods for Lennard-Jones System with $k = 7$

| | Data Set 1 | | Data set 2 | | Data Set 3 | |
|---|---|---|---|---|---|---|
| | kNN | kpN | kNN | kpN | kNN | kpN |
| num of frames = 100 | -0.6020 | -0.5961 | -0.5986 | -0.5972 | -0.6086 | -0.6068 |
| num of frames = 200 | -0.5722 | -0.5719 | -0.5714 | -0.5684 | -0.5760 | -0.5678 |
| num of frames = 400 | -0.5369 | -0.5383 | -0.5432 | -0.5436 | -0.5511 | -0.5540 |

In Table 3.4, I present the entropy calculation results for cases $k = 7$ with three different data set. In each cases, the kpN method is generally better than the kNN method.

### A ionic system

Under the Monte Carlo Computation Setting in section 3.2.4, I extract the MC samples when the system reach equilibrium, i.e. the samples after $1 \times 10^5 n$ MC steps. Then I apply histogram method as the benchmark to compute the entropy of ionic system. I set the length of the voxel $R_{vox} = 0.07$Å in histogram method. I apply kNN and kpN methods to compute the entropy of the system.

In Table 3.5, I present the entropy calculation results for cases $k = 7$ with three different data set. I use histogram method as the benchmark. For the histogram method,

**Table 3.5**: kNN, kpN methods for ionic system with $k = 7$

| | Data Set 1 | | Data set 2 | | Data Set 3 | |
|---|---|---|---|---|---|---|
| | kNN | kpN | kNN | kpN | kNN | kpN |
| num of frames = 100 | -18.1912 | -19.5026 | -18.3377 | -19.5569 | -18.5162 | -19.6400 |
| num of frames = 200 | -18.7926 | -19.7559 | -18.7499 | -19.7242 | -18.9509 | -19.9232 |
| num of frames = 400 | -19.1326 | -20.0629 | -19.2012 | -20.1921 | -19.2517 | -20.4478 |

I set the length of the voxel $R_{vox}$ in formula (3.3.1) as 0.9 Å. I applied 2000 frames and then apply the histogram method to calculate the entropy, the benchmark entropy value is $-20.4864$.

# Chapter 4

# Additional Results: Molecular Dynamics Simulations of Hydration of a Single Ion

Molecular dynamics (MD) simulations are a popular class of methods to study the dynamic properties of charged molecules (such as an ion, a protein, a membrane, etc.) in water. Many large–scale simulation program have been developed for MD simulations [8, 9, 37]. In MD simulations, all the molecular atoms and water molecules are treated as particles, and their motions are governed by Newton's law. Mathematically, one tracks all the particle motions by solving a system of second-order ordinary differential equations (ODEs) for the coordinates of all these particles in a computational box. A different class of approach, called implicit-solvent models, treat water as a continuum instead of individual particles. These models are more efficient and less accurate, compared with MD simulations. A key quantity in implicit-solvent modeling is an interface (called solute-solvent interface) that separates a solute molecule (e.g., an ion or protein) from the water environment (called solvent). A basic question here is how to determine an effective solute-solvent interface. In

this chapter, I intend to answer this question for an ion immersed in water.

I will first describe briefly MD simulations. I will then describe a variational implicit-solvent model (VISM) [11, 15, 33] and also a stochastic Ordinary differential equation (SODE) model for the effective radius of an ion. Then I will report MD simulations for the system of ion in water. I use the resulting radial distribution function (which is a scaled density of water molecules around the ion) and design different methods to determine an effective radius of an underlying ion. Finally, I compare my MD simulations results with those obtained from the SODE model.

## 4.1 Molecular dynamics simulations

Consider $N$ particles in a computational box $\Omega = (-L, L)^3$ for some $L > 0$. A particle here can be an atom of a protein, or a water molecule, or an ion in water. I denote by $m_i$ and $x_i = x_i(t)$ the mass and the position vector at time $t$ of the $i$th particle. The velocity and acceleration vectors of the $i$th particle of time $t$ are $v_i = \dot{x}_i$ and $a_i = \ddot{x}_i$, respectively, where a dot denotes the time derivative. Suppose $F_i$ is the force exerted on the $i$th particle. Then, the Newton's law of motion states that

$$F_i = m_i \ddot{x}_i(t), \qquad i = 1, 2, ..., N.$$

I shall assume that the force is governed by a potential $U : \mathbb{R}^{3N} \to \mathbb{R}$, i.e., the force, $F_i$, exerted on the $i$th particle is given by

$$F_i = -\nabla_{x_i} U(\boldsymbol{x}),$$

where $\boldsymbol{x} = (x_1, x_2, ..., x_N)$.

The potential of the system is given by

$$U(\boldsymbol{x}) = U_{vdW}(\boldsymbol{x}) + U_{elec}(\boldsymbol{x}) + U_{mech}(\boldsymbol{x}).$$

The first term $U_{vdW}(\boldsymbol{x})$ is the van der Waals potential, given by

$$U_{vdW}(\boldsymbol{x}) = \sum_{i,j} U_{LJ}(|x_i - x_j|),$$

where the sum is taken over pairs of non-bonded solute atoms $(x_i, x_j)$ with $i < j$ and

$$U_{LJ}(|x_i - x_j|) = 4\epsilon_{ij}\Big[\Big(\frac{\sigma_{ij}}{|x_i - x_j|}\Big)^{12} - \Big(\frac{\sigma_{ij}}{|x_i - x_j|}\Big)^{6}\Big].$$

is a Lennard-Jones (LJ) potential. The parameters $\epsilon_{ij}$ and $\sigma_{ij}$ can vary with particles. The second term $U_{ele}(\boldsymbol{x})$ is the electrostatic interaction, the Coulomb interaction potential, given by

$$U_{ele}(\boldsymbol{x}) = \sum_{i,j} \frac{Q_i Q_j}{4\pi\epsilon\epsilon_0|x_i - x_j|},$$

where the sum is taken over all pairs of particles $(x_i, x_j)$ with $i < j$. The parameters $\epsilon_0$ is the vacuum permittivity and $\epsilon$ is the relative permittivity, and $Q_i, Q_j$ are the charges of the particles $x_i$ and $x_j$, respectively. The last term $G_{mech}$ is the energy of the molecular mechanical interactions among all the particles $x_1, ..., x_N$. This includes the usual bonding, bending, and torsion energies. Specifically, I use [12, 13, 21, 50]

$$U_{mech}(\boldsymbol{x}) = \sum_{i,j} W_{bond}(x_i, x_j) + \sum_{i,j,k} W_{bend}(x_i, x_j, x_k) + \sum_{i,j,k,l} W_{torsion}(x_i, x_j, x_k, x_l).$$

Here the term $\sum_{i,j} W_{bond}(x_i, x_j)$ accounts for the bonding energy of particles. The term $\sum_{i,j,k} W_{bend}(x_i, x_j, x_k)$ accounts for the bending energy of solute atoms. The term $\sum_{i,j,k,l} W_{torsion}(x_i, x_j, x_k, x_l)$ accounts for the torsion energy. Specific formulas of $W_{bond}$,

$W_{bend}$ and $W_{torsion}$ can be found in [13].

The kinetic energy of the system is defined as

$$K(\boldsymbol{p}) = \frac{1}{2}\sum_{i=1}^{N} m_i v_i^2 = \frac{1}{2}\sum_{i=1}^{N} \frac{p_i^2}{m_i}.$$

Then the total energy of the particle system at a given time is

$$H(\boldsymbol{x}, \boldsymbol{p}) = U(\boldsymbol{x}) + K(\boldsymbol{p}).$$

The Hamiltonian equations that describe the motion of these $N$ particles are then given by

$$\begin{cases} \dot{x}_i = \nabla_{\boldsymbol{p}_i} H(\boldsymbol{x}, \boldsymbol{p}), \\ \dot{p}_i = -\nabla_{\boldsymbol{x}_i} H(\boldsymbol{x}, \boldsymbol{p}), \end{cases} \qquad i = 1, \ldots, N. \tag{4.1.1}$$

A widely used algorithm for integrating the equations of motion (4.1.1) is the so-called Verlet algorithm [45]. For a chosen small time increment $\Delta t$, I have the position update: For each index $i$,

$$\begin{cases} x_i^{n+1} = 2x_i^n - x_i^{n-1} + \dfrac{1}{m_i} F_i^n (\Delta t)^2, \\ p_i^{n+1} = m_i \dfrac{x_i^{n+1} - x_i^{n-1}}{2\Delta t}, \end{cases} \qquad i = 1, \ldots, N.$$

where $x_i^n, F_i^n$, and $p_i^n$ approximate $x_i(t_n), F_i(t_n)$ and $p_i(t_n)$, respectively, and $t_n = t_0 + n\Delta t$.

In a typical MD simulation study, one first sets up the quantitative system (model) of interest under a given condition (e.g., fixed number of particles and constant total energy). Then, successive configurations of the system, as a function of time, are generated by following Newton's laws of motion. After a period of time for "equilibration" one can start to collect "data" from this computer experiment the data consist of a sequence of snapshots that record the positions and velocities of the particles in the system during a

period of time. Based on trajectories, one can estimate "typical characteristic", which can often be expressed as the time average of a function of the realized configurations, of the simulated physical system [34].

If an underlying system is ergodic, then the average over a period of time of a function of the system configuration, as the time period goes to infinity, is equal to the average of that function over all configurations weighted by the Boltzmann factor $\exp\{-\beta U(x)\}$, where $U(x)$ is the potential energy of the system and $\beta = \frac{1}{k_B T}$, $k_B$ is the Boltzmann constant, and $T$ is the temperature; that is,

$$\lim_{t\to\infty} \frac{1}{t} \int_0^t h(\boldsymbol{x}(s), \boldsymbol{p}(s))ds = Z^{-1} \int\int h(\boldsymbol{x}) \exp\{-\beta H(\boldsymbol{x}, \boldsymbol{p})\} d\boldsymbol{x} d\boldsymbol{p},$$

where $Z$ is a normalizing constant, called the partition function, defined by

$$Z = \int\int \exp\{-\beta H(\boldsymbol{x}, \boldsymbol{p})\} d\boldsymbol{x} d\boldsymbol{p}.$$

One can derive many interesting ensemble averaged physical quantities, e.g. the ensemble free energy by using MD simulations. In this chapter, I will apply a popular package: GROMACS for MD simulation.

## 4.2 Implicit-solvent modeling

I consider the solvation of a charged solute molecule (such as a protein or an ion) in an aqueous solvent (i.e. water or salted water) that is treated implicitly as a continuum [4, 44]. I assume that the solute consists of $N$ atoms that are located at $x_1, ..., x_N$ and carry partial charges $Q_1, ..., Q_N$, respectively. The region of solvation $\Omega$ is divided into the solvent region $\Omega_w$ ($w$ stands for water), the solute region $\Omega_p = \cup_{i=1}^m \Omega_p^{(i)}$ ($p$ stands for protein), and the solute-solvent interface (i.e., the dielectric boundary) $\Gamma = \cup_{i=1}^m \Gamma^{(i)}$, where

$\Omega_p^{(i)}$ is a connected component of $\Omega_p$ and $\Gamma^{(i)} = \partial\Omega_p^{(i)}$ is the boundary of $\Omega_p^{(i)}$. The solute region $\Omega_p$ contains all the solute atoms $x_j$ carrying partial charges $Q_j$ $(j = 1, \ldots, N)$, with each component $\Omega_p^{(i)}$ containing $N_i$ atoms. See Figure 4.2 for illustration.

In the variational implicit-solvent model (VISM) (cf. figure 4.1) one minimizes the solvation free energy function [11, 15]

$$G[\Gamma] = Pvol(\Omega_p) + \int_\Gamma \gamma dS + \rho_w \sum_{i=1}^{N} \int_{\Omega_w} U_i(|\boldsymbol{x} - \boldsymbol{x}_i|)dV + G_{elec}[\Gamma]. \tag{4.2.1}$$

among all possible solute-solvent interfaces $\Gamma$.

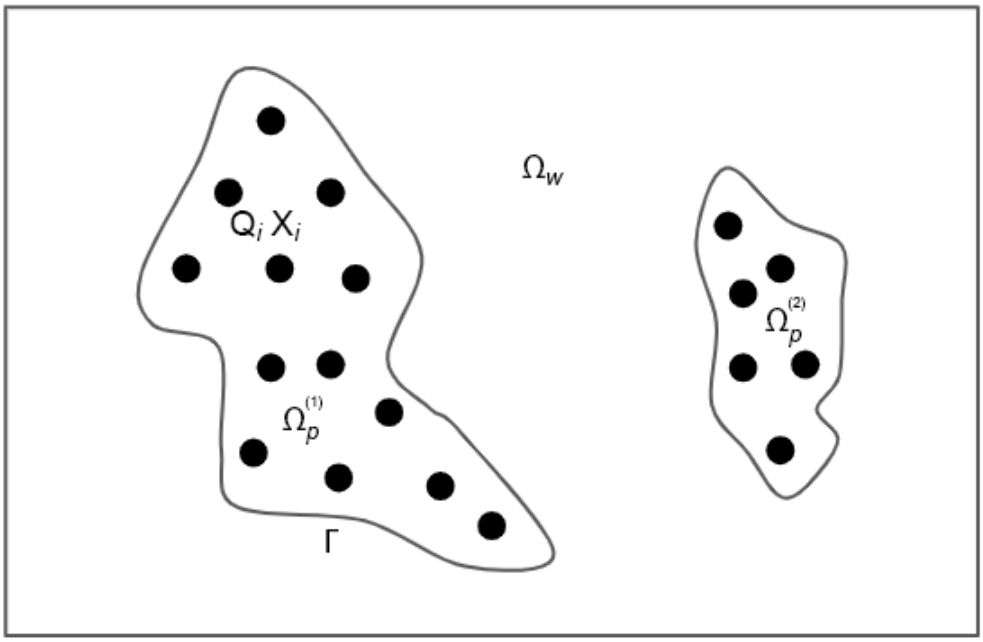


**Figure 4.1**: A schematic diagram of charged molecules immersed in an aqueous solvent. $\Gamma$ is the solute-solvent interface, for each $i$th particle in the system, it carries charge $Q_i$ with position $x_i$.

The first term in equation (4.2.1), proportional to the volume of solute region $\Omega_p$, describes the work it takes to create a solute cavity in the solvent, $P$ is the pressure difference between the solvent liquid and solute vapor. The second term is the surface

energy, where $\gamma$ is the surface tension. It is known that at the molecular scale the surface tension depends on local geometry of the surface. Here, I use $\gamma = \gamma_0(1 - 2\tau H)$, where $\gamma_0$ is the surface tension for a planar interface, $\tau$ is the curvature correction coefficient or the Tolman length, and $H$ is the mean curvature defined as the average of the two principal curvatures [54]. I call the sum of the first two terms in (4.2.1) the geometrical part of the solvation free energy.

For each $i$ $(1 \leq i \leq N)$, $U_i(|x - x_i|)$ in equation (4.2.1) is the van der Waals (vdW) type interaction potential between the solute particle at $x_i$ and a solvent molecule at $x$ that is coarse grained. The summation term represents the vdW interaction between the solute and solvent, where $\rho_w$ is the bulk density of the solvent. Here, I use $U_i$ to be the Lennard-Jones (LJ) potential

$$U_i(r) = 4\epsilon_i\Big[\big(\frac{\sigma_i}{r}\big)^{12} - \big(\frac{\sigma_i}{r}\big)^{6}\Big],$$

where the parameters $\epsilon_i$ of energy and $\sigma_i$ of length can vary with different solute atoms.

The last term $G_{elec}[\Gamma]$ in equation (4.2.1) is the electrostatic part of the solvation free energy. The first representation of last term in equation (4.2.1) is by Poisson-Boltzmann Theory [1, 46]. It is given by

$$G_{elec}[\Gamma] = \frac{1}{2}\sum_{i=1}^{N} Q_i\psi_{reac}(x_i) - \frac{1}{2}\int_{\Omega_w}\sum_{j=1}^{M} q_j c_j^{\infty}\psi e^{-\beta q_j\psi}dx - \beta^{-1}\int_{\Omega_w}\sum_{i=1}^{M} c_j^{\infty}(e^{-\beta q_j\psi} - 1)dx.$$

Here, $\psi = \psi(x)$ is the electrostatic potential, $\psi_{reac} = \psi - \psi_{ref}$ is the reaction field, and $\psi_{ref}$ is the potential for the reference state

$$\psi_{ref} = \sum_{i=1}^{N} \frac{Q}{4\pi\epsilon_0\epsilon_m|x - x_i|},$$

with $\epsilon_0$ being the vacuum permittivity and $\epsilon_m$ the dielectric coefficient of solutes. I have

assumed here that there are $M$ ionic species in the solvent, with $c_j^\infty$ and $q_j$ being the bulk concentration and charge for the $j$th species. In equation (4.2.3), $\beta^{-1} = k_B T$ with $k_B$ the Boltzmann constant and T the absolute temperature.

A different form of the last term in equation (4.2.1) is the Coulomb-field approximation (CFA):

$$G_{elec}[\Gamma] = \frac{1}{32\pi^2\epsilon_0}\left(\frac{1}{\epsilon_w} - \frac{1}{\epsilon_m}\right)\int_{\Gamma_w}\left|\sum_{i=1}^{L}\frac{Q_i(x-r_i)}{|x-r_i|^3}\right|^2 dV_x,$$

where $\epsilon_0$ being the vacuum permittivity and $\epsilon_m$ the dielectric coefficient of solutes.

I now consider a spherical solute of radius $R = R(t)$ at time $t$, carrying a single point charge $Q$ at its center that is assumed to be the origin of $\mathbb{R}^3$. See Figure 4.2.

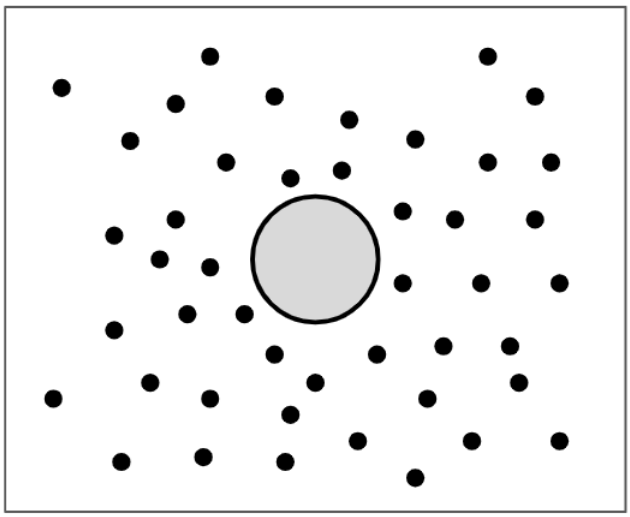

**Figure 4.2**: Schematic of an ion (big circle in center) surrounded by water molecules (small solid circles).

The region of solvation is $\Omega = \mathbb{R}^3$. In the spherical coordinates, the dielectric boundary $\Gamma(t)$, solute region $\Omega_p(t)$, and solvent region $\Omega_w(t)$ are defined by $r = R(t)$, $r < R(t)$, and $r > R(t)$, respectively, where $r = |\boldsymbol{r}|$ and $\boldsymbol{r} = (x, y, z) \in \mathbb{R}^3$. Since there is

$N = 1$ solute atom, I denote $U_{LJ} = U_{LJ}^{(1)}$, $\epsilon = \epsilon_1$, and $\sigma = \sigma_1$ i.e., I set

$$U_{\mathrm{LJ}}(r) = 4\varepsilon \left[ \left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right].$$

I shall assume that the external force in stochastic Navier–Stokes arises from a potential $U_{\mathrm{ext}}$, i.e.,

$$F_{\mathrm{ext}} = -n_{\mathrm{w}} \nabla U_{\mathrm{ext}} \qquad \text{with} \quad U_{\mathrm{ext}}(\infty) = 0.$$

I have the total surface force density $F(R)$, the derivation see detail in [16, 52].

$$F(R) = P_p(R) - P_\infty - 2\gamma_0 \left(\frac{1}{R} - \frac{\tau}{R^2}\right) + n_w \left[U_{LJ}(R) + U_{ext}(R)\right] + f_{elec}(R), \tag{4.2.2}$$

$$P_p(R) = \frac{3k_B T}{4\pi R^3},$$

$$f_{elec}(R) = \frac{Q^2}{32\pi^2 \epsilon_0} \left[ \left( \frac{1}{\epsilon_{\mathrm{w}}} - \frac{1}{\epsilon_{\mathrm{p}}} \right) \frac{1}{R^4} - \frac{\kappa^2}{\epsilon_{\mathrm{w}}(1+\kappa R)^2 R^2} \right].$$

Practically, the stochastic time-independent Stokes equation is preferred, as for a system at molecular scale the the inertia is weak and can be neglected. Then the radius $R = R(t)$ of the spherical charged molecule with a point charge $Q$ at its center satisfies the time-dependent Stokes equation:

$$\frac{4\mu_{\mathrm{w}} \dot{R}}{R} = F(R) + \xi, \tag{4.2.3}$$

where $\xi = \xi(t)$ is a Gaussian white noise with

$$\langle \xi(t)\xi(t') \rangle = \frac{4}{3} \mu_{\mathrm{w}} k_{\mathrm{B}} T \delta(t - t').$$

In my simulations for the time-independent Stokes equation (4.2.3), I choose $\Delta t = 10^{-1}$ picoseconds and run $10^6$ steps. I collect the histogram data of the simulated radius

R(t) and use the Gaussian convolution to filter the histogram. In Figure 4.3, I plot the probability distribution of the simulated radius for equation. Note that the peak of the distribution corresponds to the optimal radius, which is very close to but not exactly the same as the equilibrium radius $R_{eq}$ defined by $F(R_{eq}) = 0$ with $F(R)$ given in equation (4.2.8).

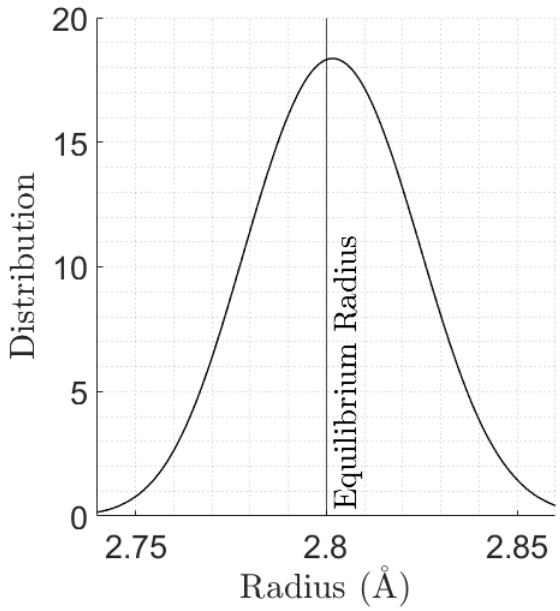


**Figure 4.3**: The probability distribution of the radius $R = R(t)$ obtained from the simulation of the time-independent Stokes equation (4.2.9). The vertical line in the plot shows the equilibrium radius defined by $F(R) = 0$ with $F(R)$ given in (3.3).

## 4.3 Determining effective radius of an ion in water

In this section, I use the radial distribution function of water molecules surrounding an ion, obtained from the MD simulation, I find the effective radius of the ion. I compare the results of effective radius of the ion with the solution of the Rayleigh-Plesset equation.

I now consider a typical charged ion in water solvent as illustrated by Figure 4.3. To determine the radius of the single ion, I consider two different methods: (1) Using the

radial distribution function of water molecules surrounding an ion, obtained from the MD simulation, find the effective radius of the ion. (2) Apply the dynamic implicit-solvent modeling framework to derive the radius of the ion.

The radial distribution function (RDF), denoted by $g(r)$ is the probability of finding a particle (here water molecule) at a distance $r$ from the fixed ion. It is given by

$$g(r) = \frac{dn_r}{4\pi r^2 dr \rho},$$

where $dn_r$ is the number of particles within a shell of thickness $dr$, $\rho = N/V$ is the average number density of water molecules. See figure 4.3 from Wikipedia for illustration of the radial distribution function.

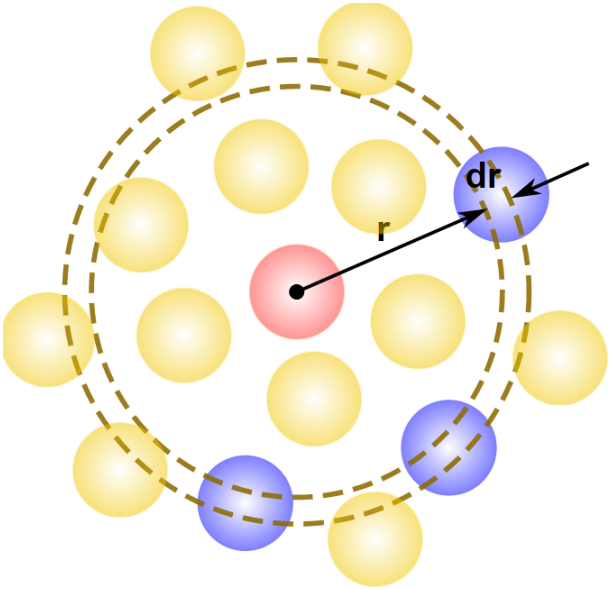


**Figure 4.4**: The radial distribution of water molecules around a fixed ion (in the center, red) is a scaled local density of water molecules.

I use the GROMACS MD simulations package with the SPC/E water model and OPLS/AA force field. I also use Particle Mesh Ewald (PME) summation for the calculation of electrostatic interactions. The temperature is set to be $T = 298\,\mathrm{K}$, and the simulation

box is a cube of size around 50 Å × 50 Å × 50 Å. At the center of this box, I place an artificial ion of point charge $Q$ with $Q = +1$, $-1$, $+2$, or $-2$. I also place around $4,050$ water molecules, and a few $Na^+$ and $Cl^-$ ions in the simulation box. The ion-water Lennard-Jones (LJ) parameters are fixed to be $\sigma = 3.5$ Å and $\epsilon = 0.3\ k_BT$. See Table 4.1 for more parameters. In each simulation, I run for $1,000,000$–$10,000,000$ time steps with each step of 1 femtosecond.

**Table 4.1**: Parameters for MD simulations.

| Parameter | Symbol | Value | Unit |
|---|---|---|---|
| temperature | $T$ | 298 | K |
| solvent dynamic viscosity | $\mu_w$ | 0.2 | $k_BT \cdot ps/Å^3$ |
| solvent number density | $n_w$ | 0.0333 | $Å^{-3}$ |
| solvent mass density | $\rho_w$ | $2.42 \times 10^{-3}$ | $k_BT \cdot ps^2/Å^5$ |
| bulk solvent pressure | $P_\infty$ | $2.46 \times 10^{-5}$ | $k_BT/Å^3$ |
| surface tension | $\gamma_0$ | 0.175 | $k_BT/Å^2$ |
| Tolman length | $\tau$ | 1 | Å |
| LJ length parameter | $\sigma$ | 3.5 | Å |
| LJ energy parameter | $\varepsilon$ | 0.3 | $k_BT$ |
| vacuum permittivity | $\varepsilon_0$ | $1.4372 \times 10^{-4}$ | $e^2/(k_BT \cdot Å)$ |
| solute dielectric constant | $\varepsilon_p$ | 1 | |
| solvent dielectric constant | $\varepsilon_w$ | 78 | |
| inverse Debye length | $\kappa$ | 0.025 | $Å^{-1}$ |
| point charge | Q | 1 | e |

From such a distribution, I can extract four different radii of the ion as illustrated in the subplot (a). They are:

(1) The "First nonzero" radius, marked by × on the horizontal axis in subplot (a), defined to be the first distance to the center of the ion at which the distribution is nonzero;

(2) The "Peak" radius, defined to be the distance to the center of the ion at which the distribution reaches its first maximum;

(3) The "Half peak" radius, defined to be the distance to the center of the ion at which the distribution reaches half of its first peak value; and

(4) The “Bulk" radius, defined to be the distance at which the distribution reaches the bulk value for the first time.

We also solve the generalized RP equation (4.2.3) with the same parameters as those in the MD simulations and others in Table 4.1. I record in Table 4.2 all the four radii defined from the radial distribution of water around the ion and the radius determined by the generalized RP equation for all the four ions (defined by the four Q-values).

I observe from Table 4.1 that the SODE radius for each of the anions (with negative $Q$ value) is larger than any of those radii defined from the water distribution, while the

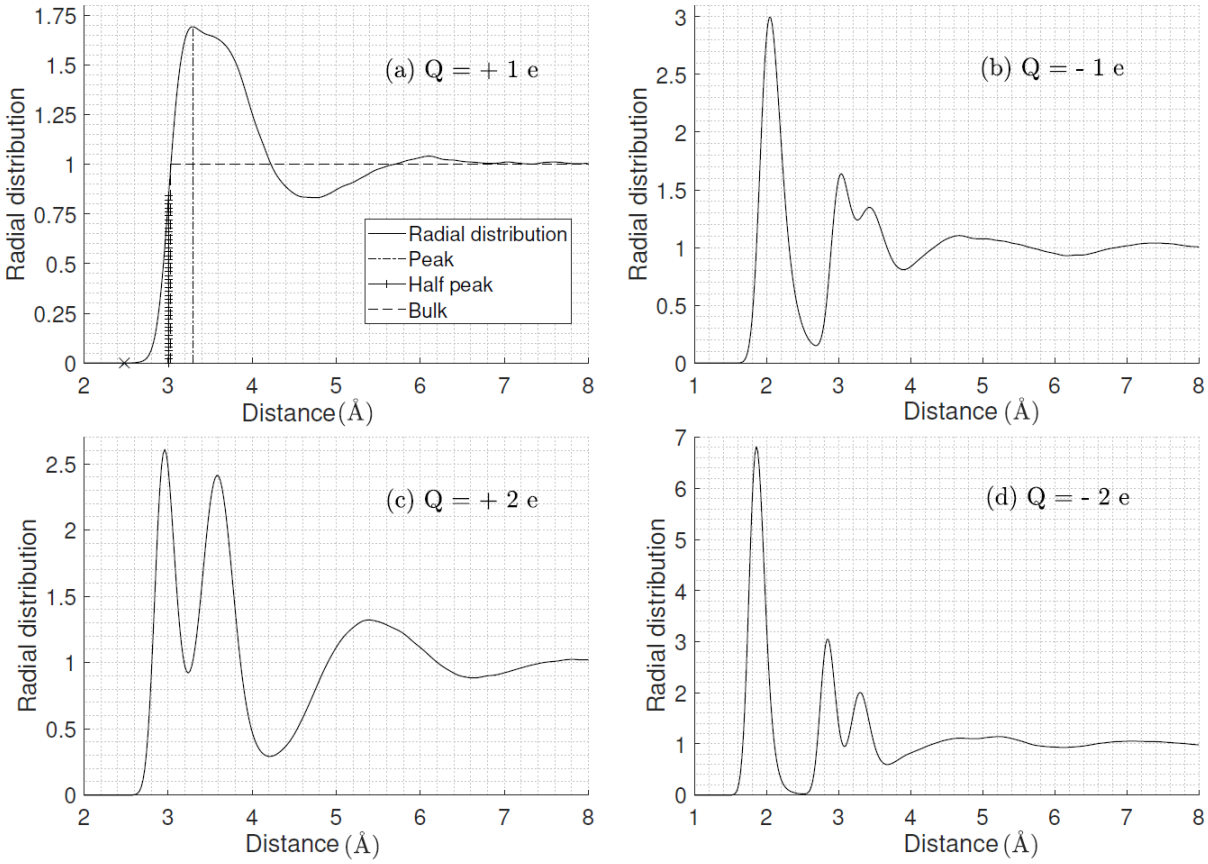


**Figure 4.5**: The radial distribution of water molecules around an artificial ion carrying the point charge $Q$: In (a), the dash-dotted line, broken line, and the line with symbol + are used to define the peak, half-peak, and bulk values of water distribution. The cross sign × between 2 and 3 on the horizontal axis of (a) defines the first point of distance to the center of the ion with a nonzero distribution.

**Table 4.2**: Effective ionic radii determined by the radial distribution of water from MD simulations (First nonzero, Peak, Half-peak, and Bulk) and by the generalized RP equation for four artificially designed ions.

| Ion | $Q$ (e) | First nonzero (Å) | Peak (Å) | Half-Peak (Å) | Bulk (Å) | RP (Å) |
|---|---|---|---|---|---|---|
| 1 | 1 | 2.48 | 3.32 | 3.00 | 3.03 | 2.80 |
| 2 | $-1$ | 1.56 | 2.04 | 1.90 | 1.86 | 2.80 |
| 3 | 2 | 2.32 | 2.96 | 2.83 | 2.81 | 2.46 |
| 4 | $-2$ | 1.46 | 1.86 | 1.74 | 1.67 | 2.46 |

SODE radius is always smaller than those defined from such distributions for cations (with positive $Q$ value). The SODE does not distinguish the sign of charge $Q$. This is known to be an issue of the continuum electrostatic model that is unable to capture the charge asymmetry [20]. However, the SODE radius approximates very well the averaged peak radius over those for the two ions with the same amount of charge (same absolute value of $Q$) i.e., average over $Q = +1$ and $Q = -1$ or over $Q = +2$ and $Q = -2$. In general, the SODE radius approximates those of cation better than an anion. A good value of the effective radius for an anion will be the SODE radius minus 0.5 Å.

MD simulations usually take much longer time. For an experiment above in Table 4.2, it took around 16 hour to finish one experiment. Our numerical results indicate that the fluctuation-dissipation balance is reached for the stochastic time-independent Stokes equation. Although this stochastic differential equation is only for a spherical molecule, it can be used to test theories and methods for more complex systems, and also used for further mathematical analysis.

Chapter 4, in part, has been submitted for publication of the material as it may appear in SIAM Journal of Applied Mathematics, 2020, Chao Fan; Bo Li; Michael White, 2020. All authors contributed essentially equally to the article.

# Chapter 5

# Conclusions and Discussions

In this dissertation, I study a class of kNN-type entropy estimators with application to particle systems. I implemented the kNN, kpN and tested these methods under different cases. In addition to kNN and kpN methods, I introduced the new method: kp-kernel method. I also couple Monte Carlo simulation and these entropy estimators for calculating the entropy of the particle systems. Finally, I studied a modeling framework combined of a static variational model and a stochastic fluid mechanics approach. I have presented my related work on the molecular dynamics simulations of the solvation of a single ion in water.

In Chapter 2, I have studied a formal derivation of class of kNN estimators of Shannon entropy, including the kNN entropy estimator, the kpN estimator introduced by some physicists, and a new kNN-type estimator kp-kernel entropy estimator. The most important object is to improve the accuracy to capture the probability density function in the "tail" parts. I have completed several numerical experiments and compared with theses three different entropy estimators. In implementation part, we apply a fast local integration method called: expectation propagation multivariate Gaussian probability (EPMGP) method to calculate the integration of a local approximated Gaussian density

function and Gaussian kernel density function in kpN and kp-kernel methods. I have provided an alternative proof of the convergence of kNN in mean. The main assumption is the all the sample we extract is within the computational box, i.e. there's a lower bound of the distribution, and this is the $f_*$, $\exists f_*, \forall x_i, f(x_i) > f_*, i = 1, 2, ...n$ has been made, where $x_i$ are samples under a unkonwn distribution function $f$. Under this assumption, we studied the convergence in expectation for the kNN distance. And we apply this result to our kNN, kpN and kp-kernel method for convergence analysis. By the proof of Singh *et al.* [39], we know the fact the assumption is not necessary for kNN method, a reasonable guess is that we may try to proof the kpN and kp-kernel method without bounded assumption. The correctness is to be verified. However, if it is true, kpN and kp-kernel method can be applied in wildly scientific area related to entropy estimation.

In chapter 3, I have studied a very important and challenging subject: entropy calculation in particle system. In this thesis, I focus on calculating the most contributed entropy in particle system: conditional one particle entropy (C1EP). I implemented the Monte Carlo Markov Chain method, specifically the metropolis hasting algorithm for particle simulation. I applied the kNN class entropy estimator in some simple particle system to observe the accuracy of these estimators. I apply histogram method, kNN and kpN method with varies of k increase and number of frames increase on both Lennard-Jones system and charged ions system.

In chapter 4, I have presented my related work on the molecular dynamics simulations of the solvation of a single ion in water. A brief introduction of molecular dynamics simulation has been presented and I have applied a popular MD package: Gromacs to simulate the particles moves in a fixed size computational box. I have compared developed several mathematical algorithms to calculate the interface (i.e. radius) of a single ion in water solvent determined by the radial distribution function. Then I compared these results with the radius by derived RP equation.

In the future, two main parts can be improved in Chapter 2. The first one is more interesting estimators followed by the formal derivation of class of kNN-type estimators may be developed to fit different complex cases. The second one is in implementation part, a more efficient algorithm is hoped to constructed to calculate the Gaussian kernel density in a local region. In Chapter 3, two parts can be improved. The first part is we could also consider the condition two particle entropy (C2EP). Even the contributed is not that large compared to C1EP, it can improve the accuracy of the whole particle system if the formula of the C2EP can be constructed. The second part is we can apply our estimator into more complicated system, i.e. protein binding (p53-MDM2) to observe the efficiency and accuracy of kNN-type estimators. In Chapter 4, a better way to compare the interface (radius) is to calculate the free energy of the single ion in water solvent. Both variational implicit solvation method(VISM) or lambda-method developed in Groamcs can be applied.

# Bibliography


[1] Dmytro Antypov, Marcia C. Barbosa, and Christian Holm. Incorporation of excluded-volume correlations into poisson-boltzmann theory. *Physical Review E*, 71, 2005.

[2] Carlos Avendano and Alejandro Gil-Villegas. Monte Carlo simulations of primitive models for ionic systems using the wolf method. *Molecular Physics*, 104:1475–1486, 2006.

[3] Andras Baranyai and Denis James Evans. Direct entropy calculation from computer simulation of liquids. *Physical Review A*, 40(7):3817–3822, 1989.

[4] Alain Barrat, Marc Barthelemy, and Alessandro Vespignani. *Dynamical Processes on Complex Networks*. Cambridge University Press, 2008.

[5] Paul D. Beale. *Statistical Mechanics*. Academic Press; 3rd edition, 2011.

[6] Berthold Bei. Entropy. *Best Practice and Research Clinical Anaesthesiology*, 20:101–109, 2006.

[7] Jan Beirlant, Edward Dudewicz, Laszlo Gyorfi, and E. C. van der Meulen. Nonparametric entropy estimation: An overview. *Resonance*, 6:17–39, 1997.

[8] Herman Johan Christiaan Berendsen, David van der Spoel, and David VanDrunen. GROMACS: A message-passing parallel molecular dynamics implementation. *Computer Physics Communications*, 91:43–56, 1994.

[9] Bernard R. Brooks, Robert E. Bruccoleri, Barry D. Olafson, David J. States, Sambasivan Swaminathan, and Martin Karplus. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *Journal of Computational Chemistry*, 4:187–217, 1983.

[10] David Chandler. *Introduction to Modern Statistical Mechanics*. Oxford University Press; 1st edition, 1987.

[11] Li-Tien Cheng, Joachim Dzubiellab, J. Andrew McCammonc, and Bo Li. Application of the level-set method to the implicit solvation of nonpolar molecules. *Journal of Chemical Physics*, 127:084503, 2007.

[12] Li-Tien Cheng, Zhongming Wang, Piotr Setny, Joachim Dzubiella, Bo Li, and J. Andrew McCammon. Interfaces and hydrophobic interactions in receptor-ligand systems: A level-set variational implicit solvent approach. *Journal of Chemical Physics*, 131:144102, 2009.

[13] Li-Tien Cheng, Yang Xie, Joachim Dzubiella, J. Andrew McCammon, Jianwei Che, and Bo Li. Coupling the level-set method with molecular mechanics for variational implicit solvation of nonpolar molecules. *Journal of Chemical Theory and Computation*, 5:257–266, 2009.

[14] John Cunningham, Philipp Hennig, and Simon Lacoste-Julien. Gaussian probabilities and expectation propagation. *arXiv:1111.6832*, 2013.

[15] Joe Dzubiella, M. Jessica Swanson, and J. Andrew McCammon. Coupling hydrophobicity, dispersion, and electrostatics in continuum solvent models. *Physical Review Letters*, 96:087802, 2006.

[16] Chao Fan, Bo Li, and Michael White. A generalized rayleigh-plesset equation for ions with solvent fluctuations. *SIAM Journal on Applied Mathematics*, 2020 (submitted).

[17] Gerald B. Folland. *Real Analysis: Modern Techniques and Their Applications*. Wiley 2nd edition, 2007.

[18] Daan Frenkel and Berend Smit. Understanding Molecular Simulation: From Algorithms to Applications. *Academic Press*, 2002.

[19] Ehud Friedgut. Hypergraphs, entropy, and inequalities. *The American Mathematical Monthly*, 111:749–760, 2004.

[20] Alan Grossfield. Dependence of ion hydration on the sign of the ion's charge. *Journal of Chemical Physics*, 122:024506, 2005.

[21] Zuojun Guo, Bo Li, Joachim Dzubiella, Li-Tien Cheng, J. Andrew McCammon, and Jianwei Che. Evaluation of hydration free energy by level-set variational implicit-solvent model with Coulomb-field approximation. *Journal of Chemical Theory and Computation*, 9:1778–1787, 2013.

[22] Vladimir Hnizdo, Eva Darian, Adam Fedorowicz, Eugene Demchuk, Shengqiao Li, and Harshinder Singh. Nearest-neighbor nonparametric method for estimating the configurational entropy of complex molecules. *Journal of Physical Chemistry B*, 28:655–668, 2006.

[23] Vladimir Hnizdo, Jun Tan, Benjamin Killian, and Michael Gilson. Efficient calculation of configurational entropy from molecular simulations by combining the mutual-information expansion and nearest-neighbor methods. *Journal of Chemical Physics*, 29:1605–1614, 2008.

[24] David Huggins and Mike Payne. Assessing the accuracy of inhomogeneous fluid solvation theory in predicting hydration free energies of simple solutes. *Journal of Physical Chemistry*, 117:8232–8244, 2013.

[25] David J. Huggins and Mike C. Payne. On the accuracy of one- and two-particle solvation entropies. *Journal of Chemical Physics*, 146:1–8, 2017.

[26] Mehran Kardar. *Statistical Physics of Particles*. Cambridge University Press; 1st edition, 2007.

[27] Martin Karplus and Joseph N. Kushick. Method for estimating the configurational entropy of macromolecules. *Macromolecules*, 14:325–332, 1981.

[28] Benjamin J. Killian, Joslyn Yundenfreund Kravitz, and Michael K. Gilson. Extraction of configurational entropy from molecular simulations via an expansion approximation. *Journal of Chemical Physics*, 127:024107, 2007.

[29] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E*, 69(066138), 2004.

[30] Nikolai N. Leonenko L. F. Kozachenko. Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii*, 23:9–16, 1987.

[31] Frank Lambert. Entropy is simple, qualitatively. *Journal of Chemical Physics*, 79:1241–1246, 2002.

[32] Cornelius Lanczos. A precision approximation of the gamma function. *SIAM Journal on Numerical Analysis*, 1:86–96, 1964.

[33] Bo Li and Yanxiang Zhao. Variational implicit solvation with solute molecular mechanics: From diffuse–interface to sharp-interface models. *SIAM Journal on Applied Mathematics*, 73:1–13, 2013.

[34] Jun Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2004.

[35] Damiano Lombardi and Sanjay Pant. Nonparametric k nearest neighbor entropy estimator. *Physical Review E*, 93:1–12, 2016.

[36] Nathaniel F. G. Martin and James W. England. *Mathematical theory of entropy*. Cambridge University Press; 1st edition, 2011.

[37] Romelia Salomon-Ferrer, David A. Case, and Ross C. Walker. An overview of the Amber biomolecular simulation package. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 3:198–210, 2013.

[38] Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 1948.

[39] Harshinder Singh, Neeraj Misra, Vladimir Hnizdo, Adam Fedorowicz, and Eugene Demchuk. Nearest neighbor estimates of entropy. *American Journal of Mathematical and Management Sciences*, 23:301–321, 2013.

[40] Shashank Singh and Barnabas Poczos. Analysis of k-nearest neighbor distances with application to entropy estimation. *arXiv:1603.08578*, 2016.

[41] Shashank Singh and Barnabas Poczos. Finite-sample analysis of fixed-k nearest neighbor density functional estimators. *arXiv:1603.08578*, 2016.

[42] Mark Srednicki. Entropy and area. *Physical Review Letters*, 71:666–669, 1993.

[43] Mark E. Tuckerman. *Statistical Mechanics: Theory and Molecular Simulation*. Oxford University Press, 2010.

[44] John J. Tyson and Bela Novak. A dynamical paradigm for molecular cell biology. *Trends in Cell Biology*, 30:504–515, 2020.

[45] Loup Verlet. Computer experiments on classical fluids. I. thermodynamical properties of lennard-jones molecules. *Physical Review*, 159:98–103, 1967.

[46] Vojko Vlachy. Electrostatic properties of membranes: the poisson-boltzmann theory. *Annual Review of Physical Chemistry*, 50:145–165, 1999.

[47] Duane C. Wallace. On the role of density fluctuations in the entropy of a fluid. *Journal of Chemical Physics*, 87:2282–2284, 1987.

[48] Lingle Wang, Robert Abel, Richard A. Friesner, and Bruce J. Berne. Thermodynamic properties of liquid water: An application of a nonparametric approach to computing the entropy of a neat fluid. *Journal of Chemical Theory and Computation*, 5:1462–1473, 2009.

[49] Zhiyong Wang and Yuqiang Ma. Monte Carlo determination of mixed electrolytes next to a planar dielectric interface with different surface charge distributions. *Journal of Chemical Physics*, 131:244715, 2009.

[50] Zhongming Wang, Jianwei Che, Li-Tien Cheng, Joachim Dzubiella, Bo Li, and J. Andrew McCammon. Level-set variational implicit-solvent modeling of biomolecules with the Coulomb-field approximation. *Journal of Chemical Theory and Computation*, 9:1778–1787, 2012.

[51] Alfred Wehrl. General properties of entropy. *Reviews of Modern Physics*, 50:221–260, 1964.

[52] Michael Robert White. Mathematical theory and numerical methods for biomolecular modeling. *University of California, San Diego*, 2013.

[53] Puning Zhao and Lifeng Lai. Analysis of kNN information estimators for smooth distributions. *IEEE Transactions on Information Theory*, 66:3798–3826, 2019.

[54] Shenggao Zhou, Li-Tien Cheng, Joachim Dzubiella, Bo Li, and J. Andrew McCammon. Variational implicit solvation with poisson–boltzmann theory. *Journal of Chemical Theory and Computation*, 10:1454–1467, 2014.