Estimation of Information Measures and its Applications in Machine Learning

by

Mortaza Noushad Iranzad

A dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
University of Michigan
2019

Doctoral Committee:

Professor Alfred O. Hero III, Chair
Assistant Professor Hessam Mahdavifar
Assistant Professor Danai Koutra
Associate Professor Emily Mower Provost

Dedicated to my beloved parents,
my lovely wife,
my brother and sister
for their endless love, support and encouragement

ACKNOWLEDGEMENTS

First, I want to thank my advisor Professor Alfred Hero for all the support and encouragement he provided during my PhD studies. I am extremely grateful for the opportunity I have had to work with him. I appreciate all his contributions of time, ideas, and funding to make my Ph.D. experience productive and stimulating.

Besides my advisor, I would like to thank the members of my dissertation committee, Professor Danai Koutra, who also helped me as an academic advisor, Professor Emily Mower Provost, and Professor Hessam Mahdavifar, for their encouragement, insightful comments and suggestions.

I would also like to sincerely thank professor Ivo Dinov who introduced me to the world of bioinformatics by giving me the opportunity to collaborate with his group on a healthcare-related project. Without his continuous support and patience it would not be possible to conduct this research.

I thank my fellow labmates at Hero Group for the stimulating discussions: Kevin Moon, Brandon Oselio, Yu Zeng, Li Xu, Salimeh Yasaei Sekeh, Zeyu Sun, Haonan Zhu, Lin Zhou, Yaya Zhai, Mohammadreza Tavassoli, Yun Wei, Robert Malinas, Byoungwook Jang, Neophytos Charalambides, Mayank Baranwal, Joel LeBlanc, Elyas Sabeti, Elizabeth Hou, and my other collaborators at other departments and institutes, Yuming Sun, Jerome Choi and Atta Jodeiri.

To all my Ann Arbor friends, thank you for your understanding and encouragement. Your friendship makes my life a wonderful experience.

Finally, I must express my very profound gratitude to my parents, my wife, my brother and sister for providing me with unfailing support and continuous encouragement throughout my years of study. This accomplishment would not have been possible without them. Thank you.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

Abstract

Information theoretic measures such as Shannon entropy, mutual information, and the Kullback-Leibler (KL) divergence have a broad range of applications in information and coding theory, statistics, machine learning, and neuroscience. KL-divergence is a measure of difference between two distributions, while mutual information captures the dependencies between two random variables. Furthermore, the binary Bayes classification error rate specifies the best achievable classifier performance and is directly related to an information divergence measure.

In most practical applications the underlying probability distributions are not known and empirical estimation of information measures must be performed based on data. In this thesis, we propose scalable and time-efficient estimators of information measures that can achieve the parametric mean square error (MSE) rate of $O(1/N)$. Our approaches are based on different methods such as k-Nearest Neighbor (k-NN) graphs, Locality Sensitive Hashing (LSH), and Dependence Graphs. The core idea in all of these estimation methods is a unique plug-in estimator of the density ratio of the samples. We prove that the average of an appropriate function of density ratio estimates over all of the points converges to the divergence or mutual information measures. We apply our methods to several machine learning problems such as structure learning, feature selection, and information bottleneck (IB) in deep neural networks.

CHAPTER I

Introduction

1.1 Background and Related Work

In this section we provide the necessary background and the related work of the contributions of this thesis.

1.1.1 Estimation of Information Measures

Shannon entropy, mutual information, and the Kullback-Leibler (KL) divergence are the most well-known information theoretic measures [105, 27]. Shannon entropy can measure diversity or uncertainty of samples, while KL-divergence is a measure of dissimilarity, and mutual information is a measure of dependency between two random variables or vectors [28]. Rényi proposed a divergence measure which generalizes KL-divergence [102] and is related to source coding with nonlinear code length. f-divergence is another general family of information measrues, which has been well studied, and comprises many important divergence measures such as KL-divergence, total variation distance, and α-divergence [3, 26]. The divergences belong to the f-divergence family have a wide range of applications in information and coding theory, statistics and machine learning [28, 79, 81].

We introduce the two well known Rényi and f-divergence measures. Consider two density functions f_X and f_Y with support $\mathcal{M} \subseteq \mathbb{R}^d$. The Rényi divergence between

f_X and f_Y is

$$\begin{aligned} D_\alpha\left(f_X(x)||f_Y(x)\right) &:= \frac{1}{\alpha-1}\log\int f_X(x)^\alpha f_Y(x)^{1-\alpha}dx \\ &= \frac{1}{\alpha-1}\log J_\alpha(f_X,f_Y), \end{aligned} \tag{1.1}$$

where in the second line, $J_\alpha(f_X, f_Y)$ denotes:

$$J_\alpha(f_X,f_Y) := \mathbb{E}_{f_Y}\left[\left(\frac{f_X(x)}{f_Y(x)}\right)^\alpha\right].$$

The f-divergence family, is defined as follows [3]:

$$\begin{aligned} D_g\left(f_X(x)||f_Y(x)\right) &:= \int g\left(\frac{f_X(x)}{f_Y(x)}\right)f_Y(x)dx \\ &= \mathbb{E}_{f_Y}\left[g\left(\frac{f_X(x)}{f_Y(x)}\right)\right], \end{aligned} \tag{1.2}$$

where g is a smooth and convex function such that $g(1) = 0$. KL-divergence, Hellinger distance and total variation distance are particular cases of this family.

We define mutual information functions in terms of the divergence between the joint density of X and Y and and their marginals. Let $f_X(x)$, $f_Y(y)$ and $f_{XY}(x,y)$ be marginal and joint densities of (X,Y). The Rényi mutual information is defined as

$$I_\alpha(X,Y) = D_\alpha\left(f_X(x)f_Y(y)\|f_{XY}(x,y)\right). \tag{1.3}$$

Also the general mutual information function based on the f-divergence measure with a function g is defined as

$$I_g(X,Y) = D_g\left(f_X(x)f_Y(y)\|f_{XY}(x,y)\right), \tag{1.4}$$

which includes Shannon mutual information function for the choice of $g(x) = -\log(x)$.

A popular class of estimators for information measures are nonparametric estimators, for which minimal assumptions on the density functions are required. This is in contrast to parametric estimators that assume a parametric model for the underlying density. An approach used for non-parametric estimators is plug-in estimation, in which we find an estimate of a distribution function and then plug it into the measure function. The k-Nearest Neighbor (K-NN) and Kernel Density Estimator (KDE) estimation methods are examples of this approach.

Another approach is graphical estimation, in which we find a relationship between the measure function and a graph-related functional in Euclidean space. In a seminal work in 1959, Beardwood et al derived the asymptotic behavior of the weighted functional of minimal graphs such as K-NN and travelling sales person (TSP) graph of N i.i.d random points [10]. They showed that the sum of weighted edges of these graphs converges to the integral of a weighted density function, which can be interpreted as Rényi entropy. Since then, this work has been of great interest in the signal processing and machine learning communities [50]. More recent studies of direct graph theoretical approaches include the estimation of Rényi entropy using minimal graphs [49], as well as the estimation of Henze-Penrose divergence using minimal spanning tree (MST) [38, 14]. Yet the extension to information theoretic divergence and correlation measures such as Rényi and f-divergences as well as Rényi and Shannon mutual information functions has remained an open question. Moreover, among various estimators of information measures, developing accurate and computationally tractable approaches has often been a challenge. Therefore, for practical and computational reasons, direct graphical algorithms have received much attention more in the literature lately.

Several previous works have investigated an estimator for a particular type of

divergence and mutual information. k-NN [96], KDE [108], and histogram [121] estimators are among the studied plug-in estimators for divergence and mutual information functions. In general, most of these estimators suffer from restrictive conditions such as lack of analytic convergence rates, or high computational complexity.

Recent works have focused on deriving the MSE convergence rates for plug-in estimators, such as KDE. Singh and Póczos proposed estimators for general density functionals, Rényi divergence and mutual information, based on the kernel density plug-in estimator [108, 110]. Their approach achieves the convergence rate of $O(1/N)$ when the densities are at least d times differentiable. In a similar approach, Kandasamy et al proposed another KDE-based estimator for general density functionals and divergence measures, which can achieve the convergence rate of $O(1/N)$ when the densities are at least $d/2$ differentiable [57]. Moon et al proposed simple kernel density plug-in estimators using weighted ensemble methods to improve the MSE rate [82, 78]. The proposed estimator can achieve the optimal convergence rate when the densities are at least $(d+1)/2$ times differentiable. The main drawback of these plug-in estimators is handling the bias at the support set boundary. For example, use of the estimators proposed in [108, 57] requires knowledge of the densities' support set, boundary smoothness assumptions, and numerous computations at the support boundary, which become complicated when the dimension increases. To circumvent this issue, Moon et al [82] assumed smoothness conditions at the support set boundary for the ensemble estimator, which may not always be satisfied in practice. Our basic estimator does not require any smoothness assumptions on the support set boundary although our ensemble estimator does. Regarding the algorithm time complexities, our KNN and hash-based estimators respectively spend $O(kN \log N)$ and $O(N)$ time versus the time complexity of KDE based estimators which spend

$O(N^2)$ time.

A rather different method for estimating f-divergences is suggested by Nguyen et al [89], which is based on a variational representation of f-divergences that connects the estimation problem to a convex risk minimization problem. This approach achieves the parametric rate of $O(1/N)$ when the likelihood ratio is at least $d/2$ times differentiable. However, the algorithm's time complexity can be worse than $O(N^2)$.

Finally, in an independent work, Wisler et al proposed a graph based estimator of density functionals which resembles our approach in some aspects such as using the k-NN graph of joint data set and direct estimation of the density ratio based on the type of neighbor nodes [122]. However, they do not provide any convergence rate for their estimator.

1.1.2 Estimation of Bayes Error Rate

In any classification problem, the error rate of a classifier might be better than the human error, but it is always greater than the Bayes error rate. The Bayes error rate is the best achievable misclassification error rate, and provides a lower bound on the error rate of any practical classifier. Note that when the labels are assigned by human based on the generated samples, it doesn't make sense to observe a better performance than the human level. However, in cases which the samples are generated based on the predefined labels, the machine learning performance could overcome the human level performance.

Consider an observation-label pair (X, T) takes values in $\mathbb{R}^d \times \{1, 2, \ldots, \lambda\}$. For class i, the prior probability is $\Pr\{T = i\} = p_i$ and f_i is the conditional distribution function of X given that $T = i$. Let $\mathbf{p} = (p_1, p_2, \ldots, p_\lambda)$. A classifier $C : \mathbb{R}^d \to \{1, 2, \ldots, \lambda\}$ maps each d-dimensional observation vector X into one of λ classes. The misclassification error rate of C is defined as

$$\epsilon_C = \Pr(C(X) \neq T), \tag{1.5}$$

which is the probability of classification associated with classifier function C. Among all possible classifiers, the Bayes classifier achieves minimal misclassification rate and has the form

$$C^{\text{Bayes}}(x) =_{1 \leq i \leq \lambda} \Pr(T = i | X = x), \tag{1.6}$$

The Bayes misclassification error rate is

$$\mathcal{E}_{\mathbf{p}}^{\text{Bayes}}(f_1, f_2, \ldots, f_\lambda) = \Pr(C^{\text{Bayes}}(X) \neq T). \tag{1.7}$$

The Bayes error rate is the best achievable misclassification error rate, and provides a lower bound on the error rate of any practical classifier. The problem of learning to bound the Bayes error rate has been a topic of recent interest. Lower and upper bounds on Bayes error rate are typically estimated by estimating an f-divergence that measures dissimilarity between the class distributions [15, 91, 77]. The f-divergence or Ali-Silvey distance, first introduced in [4], is a useful measure of the distribution distance, which is a key notion in the field of information theory and machine learning [29]. The f-divergence generalizes several measures like Kullback-Leibler divergences [68], Lin's divergences [74], and Rényi divergences [100]. For example, the problem of learning the Bhattacharya (BC) divergence [56] was addressed in [15, 120, 77]. This divergence is a special form of the Chernoff α-divergence [19] with $\alpha = 1/2$ and arises in a number of applications. The BC divergence is defined as

$$I_{\frac{1}{2}}(f_1, f_2) = \int \sqrt{p_1 p_2 f_1(x) f_2(x)} dx, \tag{1.8}$$

and the lower and upper bounds on Bayes error rate are given by

$$\frac{1}{2} - \sqrt{\frac{1}{4} - I_{\frac{1}{2}}(f_1, f_2)^2} \leq \mathcal{E}_{\mathbf{p}}^{\text{Bayes}}(f_1, f_2) \leq I_{\frac{1}{2}}(f_1, f_2). \tag{1.9}$$

The Henze-Penrose (HP) divergence, first introduced in [47], is another divergence measure, defined as

$$D_{\mathbf{p}}(f_1, f_2) := \frac{1}{4p_1p_2}\left[\int \frac{(p_1f_1(x) - p_2f_2(x))^2}{p_1f_1(x) + p_2f_2(x)}dx - (p_1 - p_2)^2\right]. \tag{1.10}$$

In [15], Berisha *et. al.* provided bounds on the Bayes misclassification error probability based on the HP divergence:

$$\frac{1}{2} - \sqrt{4p_1p_2D_{\mathbf{p}}(f_1, f_2) + (p_1 - p_2)^2} \leq \mathcal{E}_{\mathbf{p}}^{\text{Bayes}}(f_1, f_2) \leq 2p_1p_2(1 - D_{\mathbf{p}}(f_1, f_2)). \tag{1.11}$$

It was demonstrated that (1.11) is tighter than (1.9) when $p_1 = p_2 = 1/2$ [15].

1.1.3 Feature Selection

Feature selection is a data processing technique which is widely used in various areas such as signal processing, machine learning and pattern recognition. In many machine learning applications, feature selection is mainly considered as a preprocessing stage and has certain computational and performance benefits. By making the data less redundant, feature selection helps with reducing overfitting in training algorithms. In addition, it could improve the accuracy by removing the misleading data. Computationally, feature selection can improve the training runtime of the algorithms by reducing the dimension of the data [42, 72]. Interpretability is another advantage that feature selection can bring in to the machine learning approaches. Feature selection methods can be categorized into one of the following types: wrapper methods [63, 43], filter methods [61, 106] and embedded methods [69].

Wrapper methods consider different subsets of features and evaluate the performance of the selected features based on the resulting performance of the applied

model. Therefore, any learning model can be used to evaluate the performance of the selected features. There are three major approaches for the wrapper methods; subset selection [6], forward selection [71, 43] and recursive feature elimination [43]. Subset selection method performs a brute-force or a greedy search on all possible subsets of features and chooses the subset with the best performance on the learning model. Forward selection method starts with an empty set of features, and iteratively selects new features based on the performance of the selected set of features. In contrast, recursive feature elimination method starts with the set of all features and at each step removes a feature from the selected set so that the resulting performance is best. Since the search space in the subset selection method is extremely huge it is not used in practice for large datasets. For most of the cases where the dimension of the data is large and the final feature size is small, the forward selection method provides a better computational complexity and therefore it is commonly used in practice.

In contrast to the wrapper methods, filter methods are independent of any learning models. In filter methods the features are usually ranked according to a statistical importance score and then the final set of features are selected based on the feature ranking. Some common importance scores used in filter methods are chi-square test, fisher score, correlation coefficient, etc [72]. The third type of feature selection methods is called embedded approach [69]. This approach makes a trade-off between the wrapper and filter methods by embedding the feature selection into algorithm learning. In general these methods are computationally more efficient than the wrapper methods.

1.1.4 Applications in Structure Learning

Mutual information has been used for structure learning in graphical models (GM) [21], which are factorizable multivariate distributions that are Markovian according their conditional distributions, representable as a graph [70]. GMs have been used in fields such as bioinformatics, image processing, control theory, social science, and marketing analysis. However, structure learning for GMs remains an open challenge since the most general case requires a combinatorial search over the space of all possible structures [76, 124]. Furthermore, most nonparametric approaches have poor convergence rates as the number of samples increases. This has prevented reliable application of nonparametric structure learning except for impractically large sample sizes.

Several structure learning algorithms have been proposed for parametric GMs including discrete Markov random fields [60], Gaussian GMs [32], and Bayesian networks [95]. Recently, the authors of [5] proposed learning latent variable models from observed samples by estimating dependencies between observed and hidden variables. Numerous other works have demonstrated that latent tree models can be learned efficiently in high dimensions (e.g. [20, 88]).

1.2 Contributions

1.2.1 Estimation of Information Measures Based on KNN

In Chapter II we propose a KNN-based method for direct estimations of divergence and mutual information. Given two sample sets $\mathbf{X}$ and $\mathbf{Y}$ with respective densities of f_1 and f_2, we consider the sets of k-nearest neighbor (k-NN) points among the joint sample set for each point in $\mathbf{Y}$. We show that the average exponentiated ratio of the number of points with $\mathbf{X}$ type (from $\mathbf{X}$ set) to the number of points with $\mathbf{Y}$

type (from $\mathbf{Y}$ set) among all k-NN sets converges to the Rényi divergence. Using this fact, we design a consistent estimator for the Rényi and f-divergences. Also, based on the representation of mutual information functions in terms of a divergence measure between joint and marginal densities, we propose a direct estimator for Rényi and general mutual information functions.

Unlike most distance-based divergence estimators, our proposed estimator can use non-Euclidean metrics, which makes this estimator appealing in many information theoretic and machine learning applications. Our estimator requires no boundary correction, and surprisingly, the boundary issues do not show up. This is because the proposed estimator automatically cancels the extra bias of the boundary points in the ratio of nearest neighbor points. Our approach is also more computationally tractable than other estimators, with a time complexity of $O(kN \log N)$, required to construct the k-NN graph [118]. For example, for $k = O\left(N^{1/(d+1)}\right)$ the proposed approach requires computaional complexity of $O(N^{(d+2)/(d+1)} \log N)$. We derive the convergence rates of the proposed estimator for the Hölder smoothness class which include densities that are not so smooth, as well as for the differentiable densities. We show that for the class of γ-Hölder smooth functions with $0 < \gamma \leq 1$, the estimator achieves the MSE rate of $O(N^{-2\gamma/(\gamma+d)})$. for the optimal choice of $k = O\left(N^{\frac{\gamma}{d+\gamma}}\right)$. With the aim of MSE rate improvement, a randomized estimator is also proposed which can achieve the optimal MSE rates when $k = O(\log N)$. Furthermore, by using the theory of optimally weighted ensemble estimation [82, 81], for density functions with continuous and bounded derivatives of up to the order d, and some extra conditions at the support set boundary, we derive an ensemble estimator that achieves the optimal MSE rate of $O(1/N)$, which is independent of the dimension. The current work in this thesis is an important step towards extending the direct estimation

method studied in [113, 126] to more general information theoretic measures.

1.2.2 Estimation of Bayes Error Based on Minimal Graphs

In chapter III we introduce the Henze-Penrose (HP) divergence bound the Bayes error rate, and investigate the convergence rate of the FR-test statistics. The HP divergence was defined by Henze [48, 46] as the almost sure limit of the Friedman-Rafsky (FR) multi-variate two sample test statistic. Thus the FR two sample test statistic can be interpreted as an asymptotically consistent estimator of the HP divergence. The FR procedure is as follows. Assume that we have two data-sets X and Y. The FR test statistic is formed by counting the edges of MST graph of the joint data set $Z := X \cup Y$, which connect dichotomous points, i.e., a point in X to a point in Y. Later in [46], Henze proposed another similar graph based estimator that considers k-NN graph instead of the MST graph. However, the main FR test statistics using MST graph has received more attention than the k-NN variant. The authors of [46] proved the asymptotic consistency of FR statistics based on type coincidence, but the convergence rates of these estimators have been elusive. In Chapter III we investigate the convergence rate of the FR test statistics, providing an upper bound on the convergence rate.

In the second section of III we propose a new direct estimator of the HP divergence based on a weighted k-NN graph. We first derive the convergence rates of the k-NN based FR test statistics, defined as the number of edges in the k-NN graph over the joint data set $Z := X \cup Y$, which connect dichotomous points. We prove that the bias rate of this estimator is upper bounded by $O\left((k/N)^{\gamma/d}\right) + e^{-ck}$, where N and d respectively are the number and dimension of the samples, γ is the Hölder smoothness parameter of the densities and c is a constant. Note that the convergence rate of this estimator worsens in higher dimensions and does not achieve the optimal

parametric rate of $O(1/N)$. Therefore, we propose a direct estimation method based on a weighted k-NN graph. We refer to this method as the weighted nearest neighbor (WNN) estimator. The graph includes a weighted, directed edge between any pair of nodes R and S only if the types of R and S are different (i.e. $R \in X$ and $S \in Y$) and S belongs to the set of kth nearest neighbors of R. We prove that if the edge weights are obtained from the solution of a certain optimization problem, we can construct a rate-optimal HP divergence estimator based on the sum of the weights of the dichotomous edges. The convergence rate of this estimator is established to be $O(1/N)$, which is both optimal and independent of d. Finally, we emphasize that the proposed WNN estimator is completely different from the weighted matching estimator.

1.2.3 Bayes Error Learning to Analyze Classifiers' Performance

In Chapter IV we proposes a framework for empirical estimation of minimal achievable classification error, i.e., Bayes error rate, directly from training data, a framework we call *learning to benchmark*. Chapter IV presents a method for learning much tighter bounds on the Bayes error than ever before. In particular, for binary classification the proposed estimator asymptotically converges to the exact Bayes classification error. Specifically, the contributions of this chapter are as follows:

- A simple base learner of the Bayes error is proposed for general binary classification, its MSE convergence rate is derived, and it is shown to converge to the exact Bayes error probability (see Theorem IV.4). Furthermore, expressions for the rate of convergence are specified and we prove a central limit theorem for the proposed estimator (Theorem IV.5).

- An ensemble estimation technique based on Chebyshev function approximation is proposed. Using this method a weighted ensemble of benchmark base learners is proposed having optimal (parametric) MSE convergence rates (see Theorem IV.8). As contrasted to the ensemble estimation technique discussed in [77], our method provides closed form solutions for the optimal weights based on Chebyshev polynomials (Theorem IV.9).

- An extension of the ensemble benchmark learner is obtained for estimating the multiclass Bayes classification error rate and its MSE convergence rate is shown to achieve the optimal rate (see Theorem IV.10).

1.2.4 Bayes Error Based Feature Selection

In chapter V we propose a feature selection method that resembles the forward selection wrapper methods, yet it is independent of any learning model. The proposed feature selection method uses the Bayes error as a measure of quality of the features. The Bayes error rates in this method are estimated using the ϵ-ball estimator proposed in Chapter IV. A set of features have more importance if their Bayes error rate is smaller. Bayes error rate is defined as the misclassification error of the optimum Bayes classifier. Similar to the feature quality measures used in the filter methods, Bayes error is independent of any learning algorithms. However, unlike most of the filter methods, Bayes error is directly related to the error of the classification. The proposed Bayes error based feature selection (BEFS) method consists of sequential feature selection steps. The method starts with an empty set and at each step the feature that decreases the Bayes error of the selected feature set the most is selected and added to the list. Similar to the filter methods, BEFS is computationally efficient. BEFS only involves estimation of the Bayes error rate instead of the

computationally expensive training process performed at each step in the wrapper methods.

1.2.5 Hash-based Estimation of Divergence Measure

In Chapter VI we propose a low complexity divergence estimator that can achieve the optimal MSE rate of $O(1/N)$ for the densities with bounded derivatives of up to d. Our estimator has optimal runtime complexity of $O(N)$, which makes it an appropriate tool for large scale applications. Also in contrast to other competing estimators, our estimator does not require stringent smoothness assumptions on the support set boundary.

The structure of the proposed estimator borrows ideas from hash based methods for KNN search and graph constructions problems [128, 75], as well as from the NNR estimator proposed in [94]. The advantage of hash based methods is that they can be used to find the approximate nearest neighbor points with lower complexity as compared to the exact k-NN search methods. This suggests that fast and accurate algorithms for divergence estimation may be derived from hashing approximations of k-NN search. Hashing approaches are also used in other problems such as graph classification and summarization [44], kernel based image classification [41], hierarchical clustering [62], and genome-wide association study [17].

Noshad et al [94] consider the k-NN graph of Y in the joint data set (X, Y), and show that the average exponentiated ratio of the number of X points to the number of Y points among all k-NN points is proportional to the Rényi divergence between the X and Y densities. It turns out that for estimation of the density ratio around each point we really do not need to find the exact k-NN points, but only need sufficient local samples from X and Y around each point. By using a randomized locality sensitive hashing (LSH), we find the closest points in Euclidean space. In this

manner, applying ideas from the NNR estimation and hashing techniques to KNN search problem, we obtain a more efficient divergence estimator. Consider two sample sets X and Y with a bounded density support. We use a particular two-level locality sensitive random hashing, and consider the ratio of samples in each bin with a number of Y samples. We prove that the weighted average of these ratios over all of the bins can be made to converge almost surely to f-divergences between the two samples populations. We also argue that using the ensemble estimation technique provided in [79], we can achieve the optimal parametric rate of $O(1/N)$. Furthermore, using a simple algorithm for online estimation method has $O(N)$ complexity and $O(1/N)$ convergence rate, which is the first optimal online estimator of this type.

1.2.6 Hash-based Estimation of Mutual Information

In Chapter VII we propose a reduced complexity MI estimator called the ensemble dependency graph estimator (EDGE). The estimator combines randomized locality sensitive hashing (LSH), dependency graphs, and ensemble bias-reduction methods. A dependence graph is a bipartite directed graph consisting of two sets of nodes V and U. The data points are mapped to the sets V and U using a randomized LSH function H that depends on a hash parameter ϵ. Each node is assigned a weight that is proportional to the number of hash collisions. Likewise, each edge between the vertices v_i and u_j has a weight proportional to the number of (X_k, Y_k) pairs mapped to the node pairs (v_i, u_j). For a given value of the hash parameter ϵ, a base estimator of MI is proposed as a weighted average of non-linearly transformed of the edge weights. The proposed EDGE estimator of MI is obtained by applying the method of weighted ensemble bias reduction [82, 85] to a set of base estimators with different hash parameters. This estimator is a non-trivial extension of the LSH divergence estimator defined in [92]. LSH-based methods have previously been used

for KNN search and graph constructions problems [128, 75], and they result in fast and low complexity algorithms.

As an application of the optimum estimator of mutual information, we study the information bottleneck theory of deep learning. Recently, Shwartz-Ziv and Tishby utilized MI to study the training process in Deep Neural Networks (DNN) [107]. Let X, T and Y respectively denote the input, hidden and output layers. The authors of [107] introduced the information bottleneck (IB) that represents the tradeoff between two mutual information measures: $I(X,T)$ and $I(T,Y)$. They observed that the training process of a DNN consists of two distinct phases; 1) an initial fitting phase in which $I(T,Y)$ increases, and 2) a subsequent compression phase in which $I(X,T)$ decreases. Saxe *et al* in [103] countered the claim of [107], asserting that this compression property is not universal, rather it depends on the specific activation function. Specifically, they claimed that the compression property does not hold for ReLu activation functions. The authors of [107] challenged these claims, arguing that the authors of [103] had not observed compression due to poor estimates of the MI. We use our proposed rate-optimal ensemble MI estimator to explore this controversy, observing that our estimator of MI does exhibit the compression phenomenon in the ReLU network studied by [103].

Our contributions are as follows:

- To the best of our knowledge the proposed MI estimator is the first estimator to have linear complexity and can achieve the optimal MSE rate of $O(1/N)$.
- The proposed MI estimator provides a simplified and unified treatment of mixed continuous-discrete variables. This is due to the hash function approach that is adopted.

- The proposed dependence graph provides an intuitive way of understanding interdependencies in the data; e.g. sparsity of the graph implies a strong dependency between the covariates, while an equally weighted dense graph implies that the covariates are close to independent.
- EDGE is applied to IB theory of deep learning, and provides evidence that the compression property does indeed occur in ReLu DNNs, contrary to the claims of [103].

1.2.7 Information Theoretic Structure Learning

Chapter VIII proposes a nonparametric MI-based ensemble estimator for structure learning that achieves the parametric mean squared error (MSE) rate when the densities are sufficiently smooth and admits a central limit theorem (CLT) which enables us to perform hypothesis testing.

We focus on two methods of nonparametric structure learning based on ensemble MI estimation. The first method is the Chow-Liu (CL) algorithm which constructs a first order tree from the MI of all pairs of RVs to approximate the joint pdf [21]. Since structure learning approaches can suffer from performance degradation when the model does not match the true distribution, we propose hypothesis testing via MI estimation to determine how well the tree structure imposed by the CL algorithm approximates the joint distribution. The second method learns the structure by performing hypothesis testing on the MI of all pairs of RVs. An edge is assigned between two RVs if the MI is statistically different from zero. We demonstrate this estimator in multiple structure learning experiments.

1.2.8 Publications

1. M. Noshad, X. Li and A. Hero. "Learning to Benchmark: Estimating Best Achievable Misclassification Error from Training Data", Submissted.

2. S. Yasaei Sekeh, M. Noshad, K.R. Moon, A.O. Hero, "Convergence Rates for Empirical Estimation ofBinary Classification Bounds", Submitted.

3. M. Noshad, Y. Zeng, A. Hero, Scalable Mutual Information Estimation using Dependence Graphs. IEEE Intl. Conf. on Acoust., Speech, and Sig. Proc. (ICASSP 2019), arXiv preprint arXiv:1801.09125

4. M. Noshad and A. Hero, "Rate-optimal meta learning of classification error" IEEE Intl. Conf. on Acoust., Speech, and Sig. Proc. (ICASSP 2018)

5. M. Noshad, A. Hero, "Scalable Hash-Based Estimation of Divergence Measures", International Conference on Artificial Intelligence and Statistics (AISTATS 2018)

6. Morteza Noshad, Kevin R.Moon, Salimeh Yasaei Sekeh, Alfred O.Hero "Direct Divergence Estimation Using Nearest Neighbor Ratios", ISIT (International Symposium on Information Theory) 2017.

7. Kevin R Moon, Morteza Noshad, Salimeh Yasaei Sekeh, Alfred O Hero III, "Information Theoretic Structure Learning with Confidence" ICASSP (IEEE International Conference on Acoustics, Speech and Signal Processing), 2017.

1.3 Definitions and Notations

In this section we list the necessary definitions that we need in the statement of the theorems, and define some notations.

1.3.1 Definitions

Definition I.1 (Hölder Continuity)**.** Given a support $\mathcal{X} \subseteq \mathbb{R}^d$, a function $f : \mathcal{X} \to \mathbb{R}$ is called Hölder continuous with smoothness parameter $0 < \gamma \leq 1$, if there exists a positive constant G_f, depending on f, such that

$$|f(y) - f(x)| \leq G_f \|y - x\|^\gamma, \tag{1.12}$$

for every $x \neq y \in \mathcal{X}$.

Remark I.2. The γ-Hölder smoothness family comprises a large class of continuous functions including continuously differentiable and Lipschitz continuous functions. Also note that for $\gamma > 1$, any $\gamma-$Hölder continuous function on any bounded and continuous support is constant.

Definition I.3 (Lipschitz Continuity)**.** Given a support $\mathcal{X} \subseteq \mathbb{R}^d$, a function $f : \mathcal{X} \to \mathbb{R}$ is called Lipschitz continuous if there exists a constant $H_f > 0$ such that

$$|f(y) - f(x)| \leq H_f \|y - x\|, \tag{1.13}$$

for every $x \neq y \in \mathcal{X}$.

Definition I.4. (Hölder Class). Let $\mathcal{X} \subset \mathbb{R}^d$ be a compact space. For a vector $\alpha = (\alpha_1, ..., \alpha_d), \alpha_i \in \mathbb{N}$ define $D^\alpha = \frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} ... \partial x_d^{\alpha_d}}$ where $|\alpha| = \sum_{i=1}^d \alpha_i$. The Hölder Class $\Sigma(\gamma, H)$ consists of the functions f that satisfy

$$|D^\alpha f(x) - D^\alpha f(y)| \leq H \|x - y\|^{\gamma - \alpha}, \tag{1.14}$$

for all $x, y \in \mathcal{X}$ and for all α such that $|\alpha| \leq \lfloor \gamma \rfloor$. Also H is a constant depending on f.

1.3.2 Notations

Let T be a random variable, and $\mathbb{E}[T]$ denote the expectation of T. We define the notations $\mathbb{B}[\hat{T}] = \mathbb{E}[\hat{T}] - T$ for bias and $\mathbb{V}[\hat{T}] = \mathbb{E}[\hat{T}^2] - \mathbb{E}[\hat{T}]^2$ for variance of $\hat{T}$.

1.4 Assumptions

We list the required assumptions on the density functions which are required for the convergence results of the estimators of divergence, mutual information and Bayes error rate. Note that assumptions ($\mathcal{A}$.4) and ($\mathcal{A}$.5) which assume further smoother densities, are not used in base estimators on divergence, mutual information and Bayes error rate, and are only required for the ensemble estimators which can achieve the optimal parametric MSE rate of $O(1/N)$.

- ($\mathcal{A}$.1) The densities' support set is bounded.
- ($\mathcal{A}$.2) The densities are lower bounded by $C_L > 0$ and upper bounded by C_U.
- ($\mathcal{A}$.3) The densities are γ-Hölder continuous.
- ($\mathcal{A}$.4) The densities are in the Hölder class $\Sigma(\gamma, L)$, where $d \leq \gamma$.
- ($\mathcal{A}$.5) The density derivatives up to order d vanish at the boundary.

CHAPTER II

Estimation of Information Measures Based on KNN

In this chapter, we propose an estimation method for general divergence and mutual information measures based on a direct graph estimation method. Given two sample sets $\mathbf{X}$ and $\mathbf{Y}$ with respective densities of f_1 and f_2, we consider the sets of k-nearest neighbor (k-NN) points among the joint sample set for each point in $\mathbf{Y}$. We show that the average exponentiated ratio of the number of points with $\mathbf{X}$ type (from $\mathbf{X}$ set) to the number of points with $\mathbf{Y}$ type (from $\mathbf{Y}$ set) among all k-NN sets converges to the Rényi divergence. Using this fact, we design a consistent estimator for the Rényi and f-divergences. Also, based on the representation of mutual information functions in terms of a divergence measure between joint and marginal densities, we propose a direct estimator for Rényi and general mutual information functions.

The rest of this chapter is organized as follows: In section 2.1 we discuss estimation of divergence measures such as Rényi and f-divergences. In section 2.2 we propose an estimation method for mutual information measures. In section 2.3, we propose a randomized estimator which can perform slightly better in certain cases, than the original estimator discussed in the first two parts. In section 2.4, we use the ensemble estimation technique to improve the convergence rates. Finally in section 2.5 we

provide the numerical results. Proofs and relevant lemmas are given in Appendix. A.

2.1 Nearest Neighbor Ratio (NNR) Divergence Estimator

Definition II.1 (NNR Estimator)**.** Consider the i.i.d samples $\mathbf{X} = \{X_1, ..., X_N\}$ drawn from f_X and $\mathbf{Y} = \{Y_1, ..., Y_M\}$ drawn from f_Y. We define the set $\mathbf{Z} := \mathbf{X} \cup \mathbf{Y}$. Let $Q_k(Y_i)$ be the k-NN points for each of the points Y_i in the set $\mathbf{Z}$. Let N_i and M_i be the number of points of from the sets $\mathbf{X}$ and $\mathbf{Y}$ in $\{Q_k(Y_i)\}_{i=1}^k$, respectively. Then we define the NNR estimator for Rényi divergence as

$$\widetilde{D}_\alpha(\mathbf{X}, \mathbf{Y}) := \frac{1}{(\alpha - 1)} \log \left(\frac{\eta^\alpha}{M} \sum_{i=1}^{M} \left(\frac{N_i}{M_i + 1} \right)^\alpha \right), \tag{2.1}$$

where $\eta := M/N$. Similarly, using the alternative form in (1.1), we have

$$\widehat{J}_\alpha(\mathbf{X}, \mathbf{Y}) := \frac{\eta^\alpha}{M} \sum_{i=1}^{M} \left(\frac{N_i}{M_i + 1} \right)^\alpha. \tag{2.2}$$

Figure **??** represents a simple example of how we consider k-nearest neighbors for a point and how N_i and M_i are computed.

Note that the estimator defined in (2.1) can be negative and unstable in extreme cases. To correct this, we propose the NNR estimator for Rényi divergence denoted by $\widehat{D}_\alpha(\mathbf{X}, \mathbf{Y})$:

$$\min \left\{ \max \left\{ \widetilde{D}_\alpha(X, Y), 0 \right\}, \frac{1}{|1 - \alpha|} \log \left(\frac{C_U}{C_L} \right) \right\}. \tag{2.3}$$

The NNR estimator of f-divergence is defined as

$$\widehat{D}_g(\mathbf{X}, \mathbf{Y}) := \max \left\{ \frac{1}{M} \sum_{i=1}^{M} \widetilde{g} \left(\frac{\eta N_i}{M_i + 1} \right), 0 \right\}, \tag{2.4}$$

where $\widetilde{g}(x) := \max\{g(x), g\left(C_L/C_U\right)\}$. Note that we only need the function $g(x)$ to be Lipschitz continuous; i.e. g is Hölder continuous with $\gamma = 1$. Note that none of

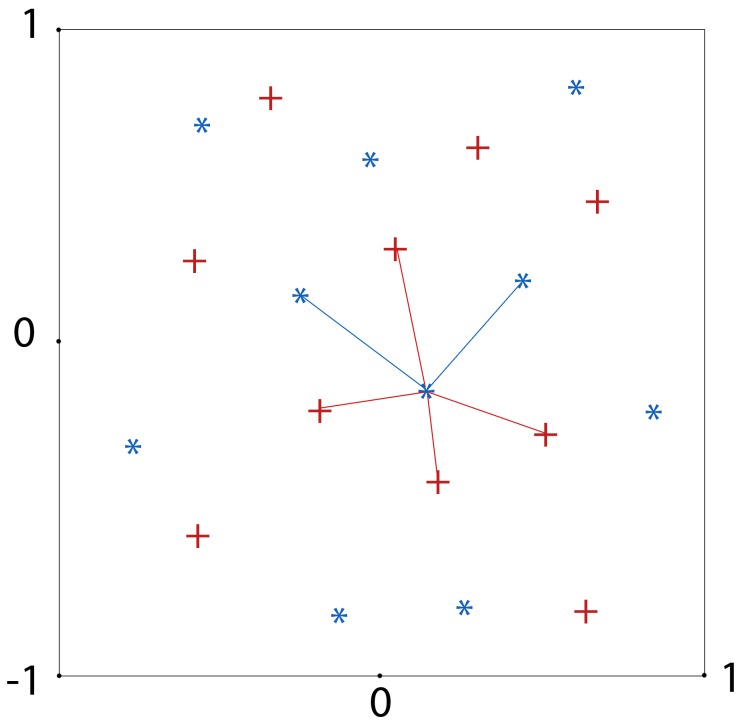


Figure 2.1: $k = 6$, $M_i = 2$ and $N_i = 4$

the other conditions from the f-divergence definition such as convexity nor $g(1) = 0$ are required for the convergence proof.

The intuition behind the proposed estimators is that, the ratio $\frac{N_i}{M_i+1}$ can be considered an estimate of density ratios at Y_i. Note that if the densities f_X and f_Y are almost equal, then for each point Y_i, $N_i \approx M_i + 1$, and therefore both $\widehat{D}_\alpha(\mathbf{X}, \mathbf{Y})$ and $\widehat{D}_g(\mathbf{X}, \mathbf{Y})$ tend to zero and g(1), respectively. Algorithm. **??** provides a pseudocode for the NNR divergence estimator.

Algorithm 1: NNR Estimator of Rényi Divergence

Input : Data sets $\mathbf{X} = \{X_1, ..., X_N\}$, $\mathbf{Y} = \{Y_1, ..., Y_M\}$

1 $\mathbf{Z} \leftarrow \mathbf{X} \cup \mathbf{Y}$
2 **for** *each point* Y_i *in* Y **do**
  `/* Set of k-NN points of` Y_i `in` $\mathbf{Z}$ `*/`
3 $\mathbf{S}_i \leftarrow \{Q_1(Y_i), ..., Q_k(Y_i)\}$
4 $R_i \leftarrow |\mathbf{S}_i \cap \mathbf{X}| / |\mathbf{S}_i \cap \mathbf{Y}|$
5 $\widehat{D}_\alpha \leftarrow 1/(\alpha - 1) \log \left[(\eta^\alpha \sum_i R_i^\alpha) / M \right]$

Output: $\widehat{D}_\alpha$

In the following we provide the convergence analysis of the NNR estimator.

Theorem II.2. *Under the assumptions (X1), (X2) and (X3) (defined in Chapter. I) for the densities, the bias of NNR estimator for Rényi divergence, defined in* (2.3)*, can be bounded as*

$$\widehat{D}_\alpha(\mathbf{X},\mathbf{Y}) = O\left(\left(\frac{k}{N}\right)^{\gamma/d}\right) + O\left(\frac{1}{k}\right). \tag{2.5}$$

Here γ is the Hölder smoothness parameter.

Theorem II.3. *Under the assumptions (X1) and (X2) for the densities, the variance of the NNR estimator is*

$$\mathbb{V}\left[\widehat{D}_\alpha(\mathbf{X},\mathbf{Y})\right] \leq O\left(\frac{1}{N}\right) + O\left(\frac{1}{M}\right). \tag{2.6}$$

Remark II.4. The same variance bound holds true for the RV $\widehat{J}_\alpha(X,Y)$. Also bias and variance results easily extend to the NNR estimator of f-divergence with a Lipschitz continuous function g.

Remark II.5. Note that in most cases, the $1/k$ term in (6.5) is the dominant error term. To have an asymptotically unbiased NNR estimator, k should be a growing function of N. The term $1/k$ comes from the error of the Poissonization technique used in the proof. By equating the terms $O\left((k/N)^{\gamma/d}\right)$ and $O(1/k)$, it turns out that for $k_{opt} = O\left(N^{\frac{\gamma}{d+\gamma}}\right)$, we get the optimal MSE rate of $O\left(N^{\frac{-2\gamma}{d+\gamma}}\right)$. The optimal choice for k can be compared to the optimum value $k = O\left(\sqrt{N}\right)$ in [79], where a plug-in KNN estimator is used. Also considering the average and worst-case computational complexity of $O(kN\log N)$ to construct the k-NN graph [118], we see that there is a trade-off between MSE rate and complexity for different values of k. In the particular case of optimal MSE, the computational complexity of this method is $O\left(N^{\frac{d+2\gamma}{d+\gamma}}\log N\right)$.

2.2 NNR Mutual Information Estimator

To estimate mutual information measures, we use a similar method as used to estimate divergence measures. Let $\mathbf{Z} := (\mathbf{X}, \mathbf{Y})$ be the $d_X + d_Y$ dimensional joint dataset with N samples. We also define $Z^{\otimes} := (\mathbf{X}, \mathcal{P}\mathbf{Y})$, where P is a random permutation operator applied on $\mathbf{Y}$. In other words, we shuffle the $\mathbf{Y}$ dataset and consider the rearranged pairs as the samples of $\mathbf{Z}^{\otimes}$. Note that the density of the points in $\mathbf{Z}$ is $f_{XY}(x, y)$, and the density of the points $\mathbf{Z}^{\otimes}$ are approximately equal to $f_X(x)f_Y(y)$. So according to the definitions in (1.3) and (1.4) we can apply the divergence estimator between $\mathbf{Z}$ and $\mathbf{Z}^{\otimes}$ to estimate the mutual information measure. This can be formulated as follows:

$$\hat{I}(\mathbf{X}, \mathbf{Y}) := \hat{D}(\mathbf{Z}^{\otimes} \| \mathbf{Z}), \tag{2.7}$$

where $\hat{I}$ can be the estimated Rényi or general mutual information functions. Algorithm **??** represents a pseudocode for the NNR mutual information estimator.

Algorithm 2: NNR Estimator of Rényi Mutual Information

Input : Data sets $\mathbf{X} = \{X_1, ..., X_N\}$, $\mathbf{Y} = \{Y_1, ..., Y_N\}$

$\mathbf{Z} \leftarrow (\mathbf{X}, \mathbf{Y})$ `/* Z is a` N `by` $d_X + d_Y$ `dataset */`

$\mathbf{Z}^{\otimes} \leftarrow (\mathbf{X}, \mathcal{P}\mathbf{Y})$ `/* Random permutation on Y */`

$\mathbf{W} \leftarrow \mathbf{Z} \cup \mathbf{Z}^{\otimes}$

for *each point* Z_i *in* $\mathbf{Z}$ **do**

  `/* Set of` k`-NN points of` Z_i `in W */`

  $\mathbf{S}_i \leftarrow \{Q_1(Z_i), ..., Q_k(Z_i)\}$

  $R_i \leftarrow |\mathbf{S}_i \cap \mathbf{Z}^{\otimes}| / |\mathbf{S}_i \cap \mathbf{Z}|$

$\widehat{I} \leftarrow 1/(\alpha - 1) \log \left[\left(\eta^{\alpha} \sum_i R_i^{\alpha} \right) / M \right]$

Output: $\widehat{I}$

The convergence results for this estimator are stated in the following theorem, and are proved in Appendix. A.

Theorem II.6. *Under the assumptions (X1), (X2) and (X3) on the densities* f_X,

f_Y, and f_XY, the bias and variance of the NNR estimator of Rényi and general mutual information measure, defined in (2.7)*, can be bounded as*

$$\widehat{I}(X,Y) = O\left(\left(\frac{k}{N}\right)^{\gamma/2d}\right) + O\left(\frac{1}{k}\right). \tag{2.8}$$

$$\mathbb{V}\left[\widehat{I}(X,Y)\right] = O\left(\frac{1}{N}\right) \tag{2.9}$$

2.3 Randomized NNR (RNNR) Estimator

The error terms in theorems VII.2 and II.6 suggest that for obtaining optimal rates, one has to set $k = O\left(N^{\frac{\gamma}{d+\gamma}}\right)$, and for smaller values of k the convergence rate is slower. This problem arises from the $O(1/k)$ error term in the bias term, which is due to de-Poissonization in the proof. In a randomized estimator we can improve the complexity by assuming that k for each node is randomly chosen from a Poisson distribution, which gives rise to elimination of the $1/k$ term. Therefore, the advantage of randomized estimator is that we can obtain the optimal MSE rates with lower average computational complexity. The RNNR estimator is defined as follows.

Definition II.7 (RNNR Estimator)**.** Let N_i and M_i respectively be the number of points from $\mathbf{X}$ and $\mathbf{Y}$ among the K_i nearest neighbors of Y_i, where K_i is randomly chosen from Poisson distribution with mean k. Then the RNNR estimator for Rényi divergence is defined as

$$\widetilde{D}_\alpha(\mathbf{X},\mathbf{Y}) := \frac{1}{(\alpha-1)}\log\left[\frac{\eta^\alpha}{M}\sum_{i=1}^{M}\left(\frac{N_i}{M_i+1}\right)^\alpha\right]. \tag{2.10}$$

The following theorem states the bias bound for the RNNR estimator of Rényi

divergence. This bound can also be extended for f-divergence as well as Rényi and general mutual information measures, in the same way.

Theorem II.8. *Under the assumptions (X1), (X2) and (X3) for the densities, bias and variance of the RNNR estimator for Rényi divergence, proposed in* (2.3)*, can be bounded as*

$$\mathbb{B}\left[\widehat{D}_\alpha(X,Y)\right] = O\left(\left(\frac{k}{N}\right)^{\gamma/d}\right) + O\left(e^{-vk}\right),$$
$$\mathbb{V}\left[\widehat{D}_\alpha(X,Y)\right] = O\left(\frac{1}{N}\right), \tag{2.11}$$

where v is a positive constant, and k is the expectation of the Poisson random parameter.

Note that the optimal bias rate of $O\left(\left(\frac{logN}{N}\right)^{\gamma/d}\right)$ can be obtained for $k = O(\log N)$. Compared to the regular NNR approach, the RNNR estimator has the advantage of requiring smaller average k, and obtaining better convergence rates for smaller choices of k. Also using the algorithm in [118], it can easily be shown that the average complexity is $O(kN \log N)$, which is the same as the complexity of NNR. However, in the worst case scenario, RNNR's complexity is $O(N^2)$, which happens when $K_i = N$ for all $0 \leq i \leq N$. Note that the probability that this occurs tends to zero as N increases. The algorithm for this estimator is shown in Algorithm **??**.

Algorithm 3: RNNR Estimator of Rényi Divergence

Input : Data sets $\mathbf{X} = \{X_1, ..., X_N\}$, $\mathbf{Y} = \{Y_1, ..., Y_M\}$

$\mathbf{Z} \leftarrow \mathbf{X} \cup \mathbf{Y}$

for *each point* Y_i *in* $\mathbf{Y}$ **do**

  `/* Set of` K_i`-NN points of` Y_i `in` $\mathbf{Z}$ `*/`

  $K_i \sim Poisson(k)$

  $\mathbf{S}_i \leftarrow \{Q_1(Y_i), ..., Q_{K_i}(Y_i)\}$

  $R_i \leftarrow |\mathbf{S}_i \cap \mathbf{X}| / |\mathbf{S}_i \cap \mathbf{Y}|$

$\widehat{D} \leftarrow 1/(\alpha - 1) \log\left[(\eta^\alpha \sum_i R_i^\alpha)/M\right]$

Output: $\widehat{D}$

2.4 Ensemble NNR (ENNR) Estimator

Under extra conditions on the densities and support set boundary, we can improve the bias rate by applying the ensemble theory in [82, 81]. We assume that the density functions are in the Hölder class $\Sigma(\gamma, L)$ defined in Definition. I.4.

Let $\mathcal{L} := \{l_1, ..., l_L\}$ be a set of index values with $l_i < c_e$ for some constant c_e. Let $k(l) := \left\lfloor l\sqrt{N} \right\rfloor$. The weighted ensemble estimator is defined as

$$\widehat{D}_w := \sum_{l\in\mathcal{L}} w(l)\widehat{D}_{k(l)}, \tag{2.12}$$

where $\widehat{D}_{k(l)}$ is the NNR estimator of Rényi or f-divergence, using the $k(l)$-NN graph (defined in (2.3) and (6.4)).

Theorem II.9. *Let $L > d$ and w_0 be the solution to the following optimization problem:*

$$\begin{aligned} \min_{w} \quad & \|w\|_2 \\ \textit{subject to} \quad & \sum_{l\in\mathcal{L}} w(l) = 1, \\ & \sum_{l\in\mathcal{L}} w(l) l^{i/d} = 0, i \in \mathbb{N}, i \leq d. \end{aligned} \tag{2.13}$$

Then under the assumptions (X1), (X2), (X4) and (X5) on the densities f_X and f_Y, the MSE rate of the ensemble estimator $\widehat{D}_{w_0}$ is $O(1/N)$. The proof is provided in Appendix D.

Remark II.10. The weighted ensemble mutual information estimator is defined as $\widehat{I}_{w_0} := \sum_{l\in\mathcal{L}} w_0(l)\widehat{I}_{k(l)}$, where $\widehat{I}_{k(l)}$ is the NNR mutual information estimator (defined in (1.3) and (1.4)), and the weight vector $w_0(l)$ is the solution to optimization problem in (2.13). Then under the assumptions (X1), (X2), (X4) and (X5) with the dimension $d_x + d_Y$, the MSE rate of the ensemble estimator $\widehat{I}_{w_0}$ is $O(1/N)$.

2.5 Numerical Results

In this section we provide numerical results to show the consistency of the proposed estimator and compare the estimation quality in terms of different parameters such as sample size N and parameter k. In our experiments, we choose i.i.d samples for X and Y from different independent distributions such as Gaussian, truncated Gaussian and uniform functions. We perform a variety of experiments for different types of divergence and information measures with various density functions. Although the convergence results of the estimators are based on the assumption that the densities have bounded support, by doing some experiments we show that the proposed estimators have good performance even when the densities are unbounded.

Figure 6.2, shows the mean estimated KL-divergence as N grows for k equal to $20, 40, 60$, using the NNR estimator. The divergence measure is between a 2D truncated normal RV with mean $[0, 0]$, variance of $2I_2$, where I_d is the identity matrix of size d, and the support in $x, y \in [-5, 5]$, and a uniform distribution with $x, y \in [-5, 5]$. For each case we repeat the experiment 100 times, and compute the mean of the estimated value and the standard deviation error bars. For small sample sizes, smaller k results in smaller bias error, which is due to the $\left(\frac{k}{N}\right)^{\gamma/d}$ bias term. As N grows, we get larger bias for small values of k, which is due to the fact that the $(1/k)$ term dominates. If we compare the standard deviations for different values of k at $N = 4000$, they are almost equal, which verifies the fact that variance is independent of k.

Figure 2.3 shows the MSE of the NNR estimator of the Renyi divergence with $\alpha = 0.5$ in terms of k, for two independent, truncated normal RVs. The RVs are 2D with means $\mu_1 = \mu_2 = [0, 0]$ and covariance matrices $\sigma_1 = I_2$ and $\sigma_2 = 3I_2$. Both of

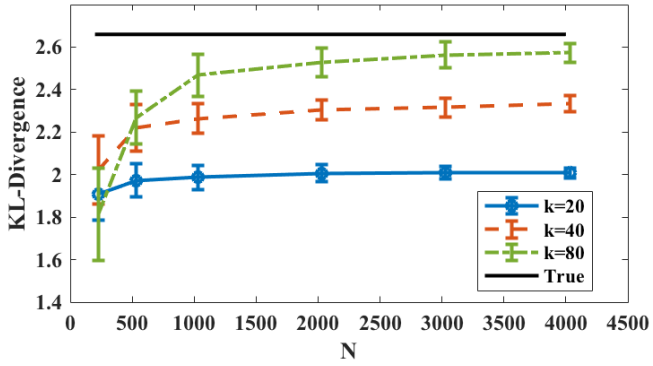


Figure 2.2: The estimated value for various values of $k = 20, 40$, and 80 is compared to the true value for KL-divergence between a truncated normal and a uniform distribution, in terms of the number of samples.

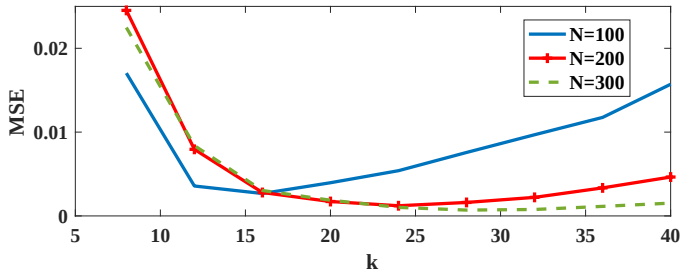


Figure 2.3: The MSE of the NNR estimator of Rényi divergence with $\alpha = 0.5$ for two independent, truncated normal RVs, as a function of k.

the RVs are truncated with the range $x \in [-2, 2]$ and $y \in [-2, 2]$. The experiment has been run for three different sample size $N = 100, 200$ and 300. As k increases initially, MSE decreases due to the $O(1/k)$ bias term. After reaching an optimal point, MSE increases as k increases, indicating that the other bias terms begin to dominate. The optimal k increases with the sample size which validates our theory (Remark II.5).

Figure 2.4 compares the theoretical and experimental MSE of the NNR estimator of Rényi divergence ($\alpha = 2$) versus N, for two i.i.d. Normal RVs with the same mean and $\sigma_1 = I_d, \sigma_2 = 6I_d$ for two different dimension sizes $d = 2$ and $d = 4$. The parameter $k = 90$ is also fixed so that the $O\left(1/k\right)$ term in the bias can be ignored in comparison to the $O\left((k/N)^{\gamma/d}\right)$ term. As dimension grows, the MSE decreases

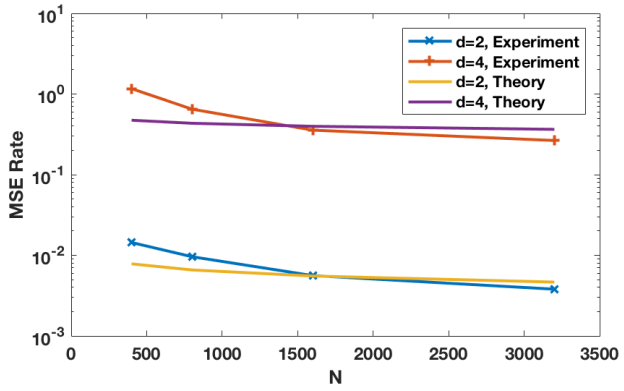


Figure 2.4: The MSE rate of the NNR estimator of the Renyi divergence with $\alpha = 2$ versus N, for two i.i.d. Normal RVs truncated at $[-10, 10]$ along each axis.

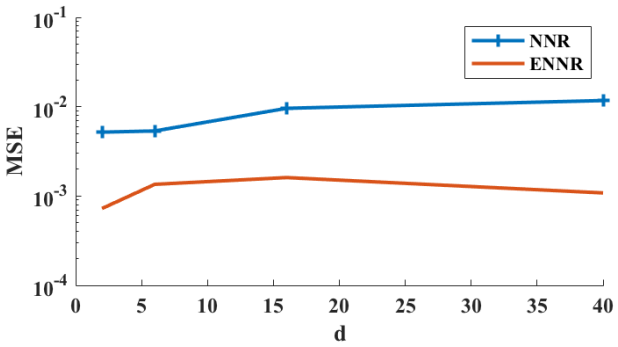


Figure 2.5: Comparison of the MSE of the NNR and ENNR estimators of Rényi divergence with $\alpha = 3$ for the constant number of samples, $N = 1000$, when we increase the dimension of densities. The densities are normal, with the same mean and variance of $\sigma_1 = \sigma_2 = I_2$, truncated within $[-10, 10]$ along each dimension.

almost linearly in the logarithmic scale, which verifies the bias term.

In Figure 2.5, we compare the performance of the NNR and ENNR estimators of Rényi divergence ($\alpha = 3$) for the constant number of samples, $N = 1000$, when we increase the dimension of the densities. The densities are normal, with the same mean and variance of $\sigma_1 = \sigma_2 = I_2$, truncated within $[-10, 10]$ along each dimension. For the NNR estimator, we have set the optimal choice for k. The MSE rate of the NNR estimator increases as dimension increases. However, the error of the ENNR estimator has little change as the dimension increases.

In Figure 2.6, we compare the performance of NNR and RNNR estimators of

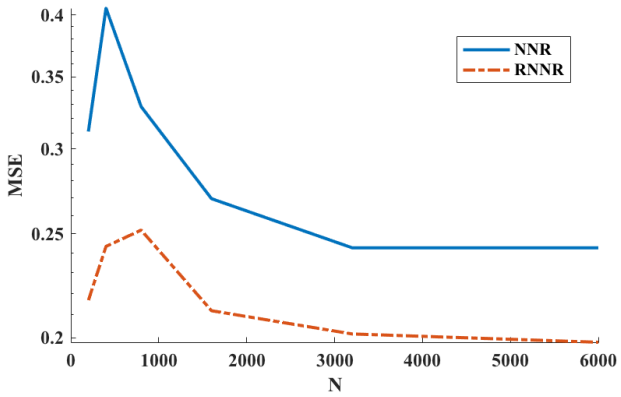


Figure 2.6: Comparison of the MSE of the NNR and RNNR estimators of Rényi divergence with $\alpha = 2$, when when increase the number of samples and $k = O(\log N)$. The densities are Normal, with the same mean and variance of $\sigma_1 = \sigma_2 = I_2$.

Rényi divergence with $\alpha = 2$, when we increase the number of samples. The densities are Normal, with the same mean and variance of $\sigma_1 = \sigma_2 = I_2$. Also we have set $k = O(\log N)$. As discussed in the previous sections, we expect the RNNR estimator to perform better than NNR for smaller values of k. However, in the next experiments we will see that if we set an optimal k for the NNR estimator, the performances of NNR and RNNR are similar.

In Figure 2.7, we compare the performance of the NNR, RNNR and ENNR estimators of Rényi divergence with $\alpha = 2$ with two of the standard optimal estimators Ensemble KDE [82] and Mirror KDE [110], which are theoretically proven to achieve the optimal parametric MSE rate. Also note that since the type of the density kernel estimator, and the method of boundary correction of the estimators discussed in [57] and [66] are similar to the one used in [110], we don't include them in the numerical comparisons. The densities are Normal, with the means $\mu_1 = [0, 0], \mu_2 = [0, 1]$, and variances of $2\sigma_1 = \sigma_2 = 2I_2$, truncated within $[-2, 2]$ along each dimension. Also we have set $k = O(\sqrt{N})$. As shown in the figure, the ENNR has the best MSE rate among others. Also as discussed before, for the choice of $k = O(\sqrt{N})$, the two

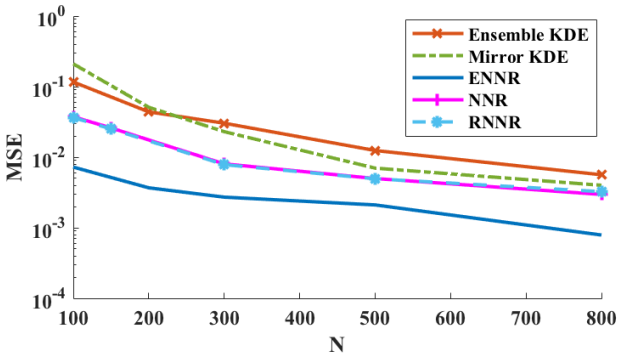


Figure 2.7: Comparison of the MSE rates of the NNR, RNNR and ENNR estimators of Rényi divergence with $\alpha = 2$ with two standard optimal estimators Ensemble KDE [82] and Mirror KDE [110]. The densities are Normal, with the means $\mu_1 = [0, 0], \mu_2 = [0, 1]$, and variances of $2\sigma_1 = \sigma_2 = 2I_2$, truncated within $[-2, 2]$ along each dimension.

estimators NNR and RNNR should show almost the same convergence rate, and the experiment verifies this fact.

One of our main contributions in this chapter is in fact proposing the low computational complexity estimators. We have discussed this by implementing the NNR and RNNR estimators and emphasize that they improve the runtime compared to the previous estimators. We verify our analytical claims above by running an experiment shown in Figure 2.8. In this experiment, we compare the runtime of the proposed estimators as well as Ensembled KDE and Miror KDE. As shown, the NNR estimator has the best runtime, while RNNR has a slightly greater runtime compared to NNR estimator. ENNR has higher practical runtime since it includes a constant time optimization procedure. Ensemble KDE has longer runtime due to its complexity of $O(N^2)$. Finally Mirror KDE has the worst runtime among these estimators. Note that for $d = 2$, Mirror KDE method assumes $8N$ extra points in addition to the original points, to consider the mirror effect at the boundaries, which makes the total complexity 81 times slower than the standard KDE method. In general in a d dimensional system it is $O(9^d)$ times slower. Therefore, we do not include Mirror KDE in our experiments in higher dimensions due to its slow performance.

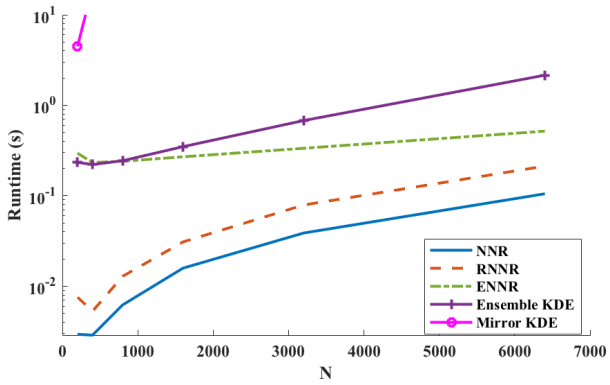


Figure 2.8: Comparison of runtime of of NNR, RNNR and ENNR estimators of Rényi divergence with $\alpha = 2$ with two of the standard optimal estimators Ensemble KDE [82] and Mirror KDE [110]. The densities are Normal, with the means $\mu_1 = [0, 0], \mu_2 = [0, 1]$, and variances of $2\sigma_1 = \sigma_2 = 2I_2$, truncated within $[-2, 2]$ along each dimension.

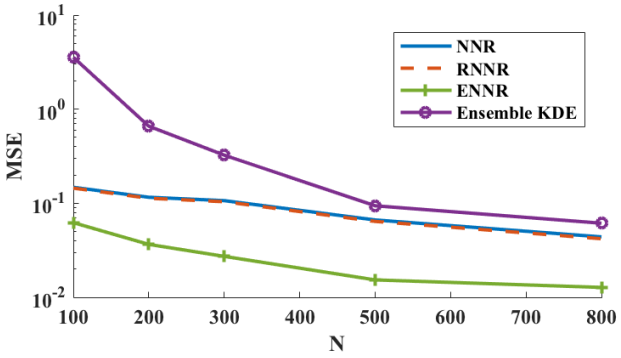


Figure 2.9: Comparison of MSE in higher dimension. Estimators of Rényi divergence with $\alpha = 3/2$ of two Normal densities, with the means $\mu_1 = [0, 0, 0, 0], \mu_2 = [0, 0, 0, 1]$, variances of $2\sigma_1 = \sigma_2 = 2I_4$, and truncated at $[-5, 5]$ at each axis.

In Figure 2.9, we compare the performance of these estimators in a relatively higher dimension, $d = 4$. The densities are Normal, with the means $\mu_1 = [0, 0, 0, 0], \mu_2 = [0, 0, 0, 1]$, variances of $2\sigma_1 = \sigma_2 = 2I_4$, and truncated at $[-5, 5]$ at each axis. Also we set $k = O(\sqrt{N})$. Again, the ENNR has the best MSE rate among others.

In Figures 2.10 and 2.11 we compare the performance of the Rényi mutual information estimators with $\alpha = 1/2$ and $\alpha = 3$ respectively. In the first experiment, the mutual information function is non-zero, while in the second experiment the mutual information function is zero. In Figure 2.10, the densities are jointly Normal, with

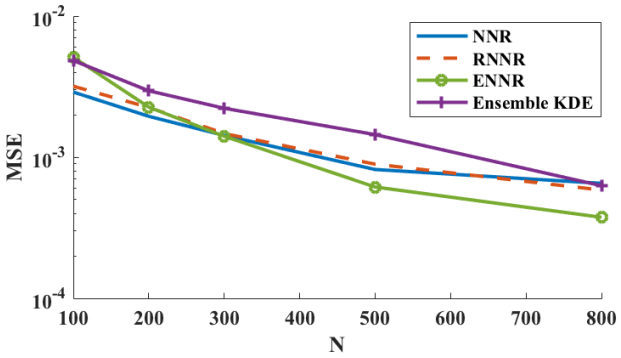


Figure 2.10: Comparison of MSE of NNR, RNNR, ENNR and Ensemble KDE estimators of Rényi mutual information with $\alpha = 1/2$. The densities are jointly Normal, with non-zero mutual information.

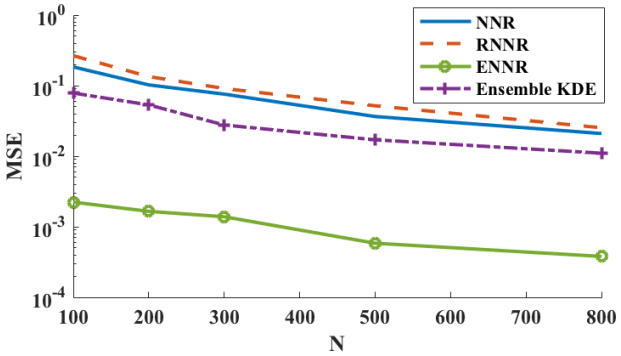


Figure 2.11: Comparison of MSE of NNR, RNNR, ENNR and Ensemble KDE estimators of Rényi mutual information with $\alpha = 3$. The densities have independent Normal distribution, with the means $\mu_1 = [0, 0]$ and $\mu_2 = [0, 1]$, and variances of $2\sigma_1 = \sigma_2 = 2I_2$, truncated at $[-5, 5]$ along each axis.

the mean $\mu = [0, 0]$, covariance of $\Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$, truncated at $[-5, 5]$ at each axis. For large data sets, the ENNR has the best MSE rate among the others, which is not surprising, and as before, NNR and RNNR have similar performance.

In Figure 2.11, the densities have independent Normal distribution, with the means $\mu_1 = [0, 0]$ and $\mu_2 = [0, 1]$, and variances of $2\sigma_1 = \sigma_2 = 2I_2$, truncated at $[-5, 5]$ at each axis. ENNR performs better than the other estimators, and Ensemble KDE which is also an optimal estimator, has better convergence rate than NNR and RNNR estimators of mutual information.

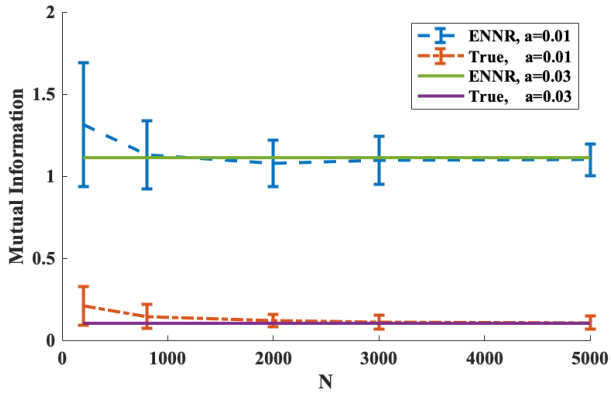


Figure 2.12: Renyi mutual information with $\alpha = 2$ between X and Y, where $Y = X + aN$, versus the number of samples. X samples are drawn from a 100-dimensional Dirichlet distribution with the parameter $\alpha = [1, 1, ..., 1]$. a is constant which controls the level of the noise and N is a multivariate Normal noise with mean 0 and covariance matrix $\sigma = I_{100}$. The error bars correspond to 0.95 confidence intervals

In Figure 2.12, we perform an experiment on a high-dimensional simulated dataset with Dirichlet distribution. X samples are drawn from a 100-dimensional Dirichlet distribution with the parameter $\alpha = [1, 1, ..., 1]$. Y samples are obtained by adding a Normal noise to the X samples. we have $Y = X + aN$, where a is constant which controls the level of the noise and N is a multivariate Normal noise with mean 0 and covariance matrix $\sigma = I_100$. Figure 2.12 shows the Renyi mutual information with $\alpha = 2$ between X and Y for two different noise levels.

2.6 Conclusion

In this chapter we proposed a direct approach for estimating general divergence and mutual information measures, based on the k-NN graph and density ratio estimates. We derived the bias and variance of the estimator, and showed that for the class of γ-Hölder smooth functions, the estimator achieves the MSE rate of $O\left(N^{-2\gamma/(\gamma+d)}\right)$. A randomized estimator was also proposed which can achieve the optimal mean square error (MSE) rates when the average k is $O(\log N)$. Furthermore, we considered continuous and bounded derivatives up to order d assumption

along with extra smoothness conditions at the set boundary. This led us to use a weighted ensemble estimation technique and derive an ensemble estimator with MSE rate of $O(1/N)$. Finally we applied the proposed estimators on the simulated datasets and compared the numerical results with the convergence theorems.

CHAPTER III

Estimation of Bounds on the Bayes Error Based on Minimal Graphs

Bounding the best achievable error probability for binary classification problems is relevant to many applications including machine learning, signal processing, and information theory. Many bounds on the Bayes binary classification error rate depend on information divergences between the pair of class distributions. Recently, the Henze-Penrose (HP) divergence has been proposed for bounding classification error probability. In this chapter we consider the problem of empirically estimating the HP-divergence from random samples. We first introduce the HP-divergence and Friedman-Rafsky (FR) estimator of the HP-divergence, which is related to a multivariate runs statistic for testing between two distributions. We then provide the bias and variance rates of the FR-based estimator of HP-divergence. Further, we give several simulations that validate the theory. In the second part of this chapter we introduce a KNN based estimator of HP-divergence which can achieve the optimal MSE convergence rate of $O(1/N)$. Proofs and relevant lemmas are given in Appendix. B.

3.1 The Henze-Penrose Divergence Measure

Consider parameters $p \in (0,1)$ and $q = 1-p$. We focus on estimating the HP-divergence measure between distributions f_0 and f_1 with domain $\mathbb{R}^d$ defined by [45]:

$$D_p(f_0, f_1) = \frac{1}{4pq}\left[\int \frac{\left(pf_0(x) - qf_1(x)\right)^2}{pf_0(x) + qf_1(x)}\, dx - (p-q)^2\right]. \tag{3.1}$$

It can be verified that this measure is bounded between 0 and 1 and if $f_0(x) = f_1(x)$, then $D_p = 0$. In contrast with some other divergences such as the Kullback-Liebler [67] and Rényi divergences [99], the HP-divergence is symmetrical, i.e., $D_p(f_0, f_1) = D_q(f_1, f_0)$. By invoking (3) in [13], one can rewrite D_p in the alternative form:

$$D_p(f_0, f_1) = 1 - A_p(f_0, f_1) = \frac{u_p(f_0, f_1)}{4pq} - \frac{(p-q)^2}{4pq}, \tag{3.2}$$

where

$$A_p\left(f_0, f_1\right) := \int \frac{f_0(\mathbf{x}) f_1(\mathbf{x})}{pf_0(\mathbf{x}) + qf_1(\mathbf{x})} \mathrm{d}\mathbf{x} \tag{3.3}$$

$$= \mathbb{E}_{f_0}\left[\left(p\frac{f_0(\mathbf{X})}{f_1(\mathbf{X})} + q\right)^{-1}\right] \tag{3.4}$$

$$u_p\left(f_0, f_1\right) = 1 - 4pqA_p\left(f_0, f_1\right) \tag{3.5}$$

The term $A_p(f_0, f_1)$ is refered to as the HP-integral [45]. The HP-divergence measure belongs to the class of ϕ-divergences [2]. For the special case $p = 0.5$, the divergence (3.1) becomes the symmetric χ^2-divergence and is similar to the Rukhin f-divergence [18], [101].

3.2 The Multivariate Runs Test Statistic

The MST is a graph of minimum weight among all graphs $\mathcal{E}$ that span n vertices. The MST has many applications including pattern recognition [116], clustering [127],

nonparametric regression [7], and testing of randomness [53]. In this section we focus on the FR multivariate two sample test statistic constructed from the MST.

Assume that sample realizations from f_0 and f_1, denoted by $\mathbf{X} \in \mathbb{R}^{m\times d}$ and $\mathbf{Y} \in \mathbb{R}^{n\times d}$, respectively, are available. Construct an MST spanning the samples from both f_0 and f_1 and color the edges in the MST that connect dichotomous samples green and color the remaining edges black. The FR test statistic $R_{m,n} := R_{m,n}(\mathbf{X}_m, \mathbf{Y}_n)$ is the number of green edges in the MST. Note that the test assumes a unique MST, therefore all inter point distances between data points must be distinct. We recall the following theorem from [12] and [13]:

Theorem III.1. *As $m \to \infty$ and $n \to \infty$ such that $\frac{m}{n+m} \to p$ and $\frac{n}{n+m} \to q$,*

$$1 - \mathfrak{R}_{m,n}(\mathbf{X}, \mathbf{Y}) \, \frac{m+n}{2mn} \to D_p(f_0, f_1), \quad a.s. \tag{3.6}$$

3.2.1 Convergence Rates

We obtain the mean convergence rate bounds for general non-uniform Lebesgue densities f_0 and f_1 belonging to the Hölder class $\Sigma(\eta, K)$:

Theorem III.2. *(Bias Bound) Let $d \geq 2$, and $R_{m,n}$ be the FR statistic for samples drawn from strong Hölder continuous and bounded density functions f_0 and f_1 in $\Sigma(\eta, K)$. Then for $d \geq 2$,*

$$\left| \frac{\mathbb{E}[R_{m,n}]}{m+n} - 2pq \int \frac{f_0(\mathbf{x}) f_1(\mathbf{x})}{p f_0(\mathbf{x}) + q f_1(\mathbf{x})} \mathrm{d}\mathbf{x} \right| \leq O\left((m+n)^{-\eta_1(\mathbf{x})} \right) , \tag{3.7}$$

where the $O()$ notation may contain dependencies to the density functions as well as their dimensions.

The following variance bound uses the Efron-Stein inequality [33]. Note that in Theorem III.3, unlike Theorem III.2, we only assume that the density functions are absolutely continuous and bounded with support on the unit cube $[0, 1]^d$.

Theorem III.3. *The variance of the HP-integral estimator based on the FR statistic,* $\mathfrak{R}_{m,n}/(m+n)$ *is bounded by*

$$\text{Var}\left(\frac{R_{m,n}(\mathbf{X},\mathbf{Y})}{m+n}\right) \leq \frac{32c_d^2 q}{(m+n)} \tag{3.8}$$

where the constant c_d *depends only on* d*.*

By combining Theorem III.2 and Theorem III.3 we obtain the MSE rate of the form $O\left(m+n\right)^{-\eta^2/(d(\eta+1))}\Big) + O\left((m+n)^{-1}\right)$.

Fig. 3.1 indicates a heat map showing the MSE rate as a function of d and $N = m = n$. The heat map shows that the MSE rate of the FR test statistic-based estimator given in (3.6) is small for large sample size N.

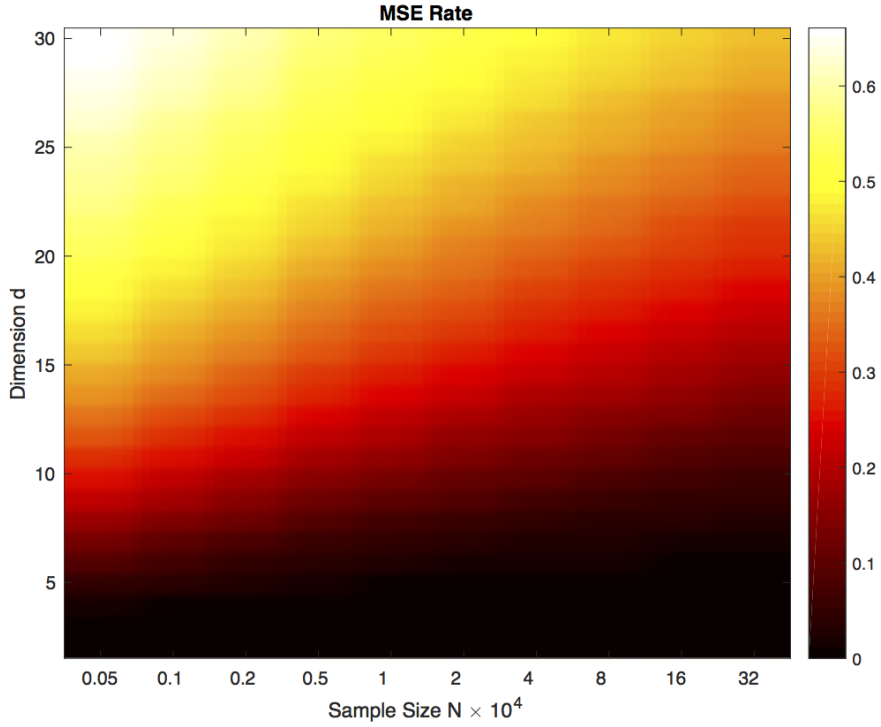


Figure 3.1: Heat map of the theoretical MSE rate of the FR estimator of the HP-divergence based on Theorems III.2 and III.3 as a function of dimension and sample size when $N = m = n$. Note the color transition (MSE) as sample size increases for high dimension. For fixed sample size N the MSE rate degrades in higher dimensions.

3.2.2 Bias Proof

In this subsection, we first establish subadditivity and superadditivity properties of the FR statistic which will be employed to derive the MSE convergence rate bound. This will establish that the mean of the FR test statistic is a quasi-additive functional:

Theorem III.4. *Let $\mathfrak{R}_{m,n}(\mathfrak{X}_m, \mathfrak{Y}_n)$ be the number of edges that link nodes from differently labeled samples $\mathfrak{X}_m = \{\mathbf{X}_1, \ldots, \mathbf{X}_m\}$ and $\mathfrak{Y}_n = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$ in $[0,1]^d$. Partition $[0,1]^d$ into l^d equal volume subcubes Q_i such that m_i and n_i are the number of samples from $\{\mathbf{X}_1, \ldots, \mathbf{X}_m\}$ and $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$, respectively, that fall into the partition Q_i. Then there exists a constant c_1 such that*

$$\mathbb{E}\Big[\mathfrak{R}_{m,n}(\mathfrak{X}_m, \mathfrak{Y}_n)\Big] \leq \sum_{i=1}^{l^d} \mathbb{E}\Big[\mathfrak{R}_{m_i,n_i}\big((\mathfrak{X}_m, \mathfrak{Y}_n) \cap Q_i\big)\Big] + 2\, c_1\, l^{d-1}\, (m+n)^{1/d}. \tag{3.9}$$

Here $\mathfrak{R}_{m_i,n_i}$ is the number of dichotomous edges in partition Q_i. Conversely, for the same conditions as above on partitions Q_i, there exists a constant c_2 such that

$$\mathbb{E}\Big[\mathfrak{R}_{m,n}(\mathfrak{X}_m, \mathfrak{Y}_n)\Big] \geq \sum_{i=1}^{l^d} \mathbb{E}\Big[\mathfrak{R}_{m_i,n_i}\big((\mathfrak{X}_m, \mathfrak{Y}_n) \cap Q_i\big)\Big] - 2\, c_2\, l^{d-1}\, (m+n)^{1/d}. \tag{3.10}$$

The inequalities (3.9) and (3.10) are inspired by the theory of subadditive functionals in [52] and [51]. The full proof is given in Appendix A. The key result in the proof is the inequality:

$$\mathfrak{R}_{m,n}(\mathfrak{X}_m, \mathfrak{Y}_n) \leq \sum_{i=1}^{l^d} \mathfrak{R}_{m_i,n_i}\big((\mathfrak{X}_m, \mathfrak{Y}_n) \cap Q_i\big) + 2|D|,$$

where $|D|$ indicates the number of all edges of the MST which intersect two different partitions.

Furthermore, we adapt the theory developed in [125, 52] to derive the MSE convergence rate of the FR statistic-based estimator by defining a dual MST and dual FR statistic, denoted by MST* and $\mathfrak{R}^*_{m,n}$ respectively (see Fig. 3.2):

Definition III.5. (Dual MST, MST* and dual FR statistic $\mathfrak{R}^*_{m,n}$) Let $\mathbb{F}_i$ be the set of corner points of the subsection Q_i for $1 \leq i \leq l^d$. Then we define MST$^*(\mathfrak{X}_m \cup \mathfrak{Y}_n \cap Q_i)$ as the boundary MST graph of partition Q_i [125], which contains $\mathfrak{X}_m$ and $\mathfrak{Y}_n$ points falling inside the section Q_i and those corner points in $\mathbb{F}_i$ which minimize total MST length. Notice it is allowed to connect the MSTs in Q_i and Q_j through points strictly contained in Q_i and Q_j and corner points are taking into account under the condition of minimizing total MST length. In other words, the dual MST can connect the points in $Q_i \cup Q_j$ by direct edges to pair to another point in $Q_i \cup Q_j$ or the corner the corner points (we assume that all corner points are connected) in order to minimize the total length. To clarify this, assume that there are two points in $Q_i \cup Q_j$, then the dual MST consists of the two edges connecting these points to the corner if they are closed to a corner point otherwise dual MST consists of an edge connecting one to another. Further, we define $\mathfrak{R}^*_{m,n}(\mathfrak{X}_m, \mathfrak{Y}_n \cap Q_i)$ as the number of edges in MST* graph connecting nodes from different samples and number of edges connecting to the corner points. Note that the edges connected to the corner nodes (regardless of the type of points) are always counted in dual FR test statistic $\mathfrak{R}^*_{m,n}$.

In Appendix B, we show that the dual FR test statistic is a quasi-additive functional in mean and $\mathfrak{R}^*_{m,n}(\mathfrak{X}_m, \mathfrak{Y}_n) \geq \mathfrak{R}_{m,n}(\mathfrak{X}_m, \mathfrak{Y}_n)$. This property holds true since MST$(\mathfrak{X}_m, \mathfrak{Y}_n)$ and MST$^*(\mathfrak{X}_m, \mathfrak{Y}_n)$ graphs can only be different in their edges connected to the corner nodes, and in $\mathfrak{R}^*(\mathfrak{X}_m, \mathfrak{Y}_n)$ we take all of the edges between these nodes and corner nodes into account.

To prove Theorem III.2, we partition $[0,1]^d$ into l^d subcubes. Then by applying Theorem III.4 and the dual MST we derive the bias rate in terms of partition parameter l (see (B.16) in Theorem B.7). See Appendix B and Supplementary Materials for details. According to (B.16), for $d \geq 2$, and $l = 1, 2, \ldots,$ the slowest rates as a

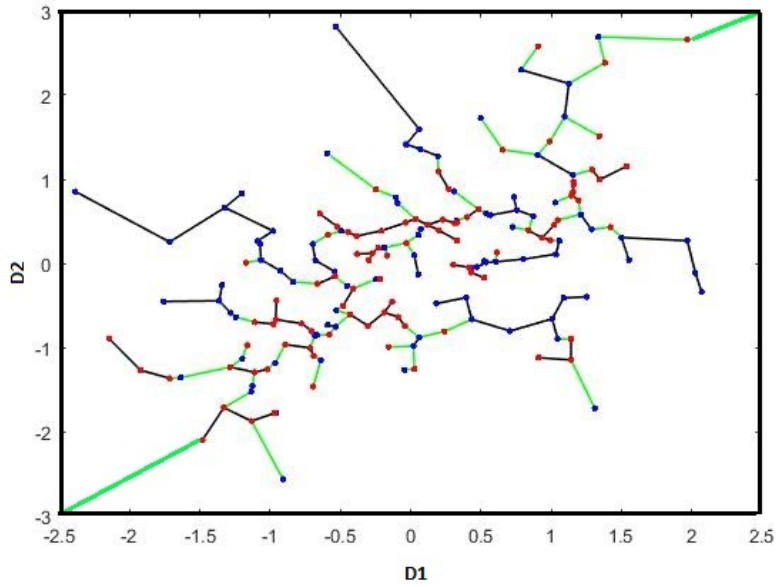


Figure 3.2: The dual MST spanning the merged set $\mathfrak{X}_m$ (blue points) and $\mathfrak{Y}_n$ (red points) drawn from two Gaussian distributions. The dual FR statistic ($\mathfrak{R}^*_{m,n}$) is the number of edges in the MST* (contains nodes in $\mathfrak{X}_m \cup \mathfrak{Y}_n \cup \{2 \text{ corner points}\}$) that connect samples from different color nodes and corners (denoted in green). Black edges are the non-dichotomous edges in the MST*.

function of l are $l^d(m+n)^{\eta/d}$ and $l^{-\eta d}$. Therefore we obtain an l-independent bound by letting l be a function of $m+n$ that minimizes the maximum of these rates i.e.

$$l(m+n) = arg \min_{l} \max \left\{ l^d (m+n)^{-\eta/d}, l^{-\eta d} \right\}.$$

The full proof of the bound in (III.2) is given in Appendix B.

3.2.3 Numerical Experiments

In this section, we apply the FR statistic estimate of the HP-divergence to both simulated and real data sets. We present results of a simulation study that evaluates the proposed bound on the MSE. We numerically validate the theory stated in Subsection 3.2.1 using multiple simulations. In the first set of simulations, We consider two multivariate Normal random vectors $\mathbf{X}$, $\mathbf{Y}$ and perform three experiments $d = 2, 4, 8$, to analyze the FR test statistic-based estimator performance as the sample sizes m, n increase. For the three dimensions $d = 2, 4, 8$ we generate

samples from two normal distributions with identity covariance and shifted means: $\mu_1 = [0, 0]$, $\mu_2 = [1, 0]$ and $\mu_1 = [0, 0, 0, 0]$, $\mu_2 = [1, 0, 0, 0]$ and $\mu_1 = [0, 0, ..., 0]$, $\mu_2 = [1, 0, ..., 0]$ when $d = 2$, $d = 4$ and $d = 8$ respectively. For all of the following experiments the sample sizes for each class are equal ($m = n$). We vary $N = m = n$

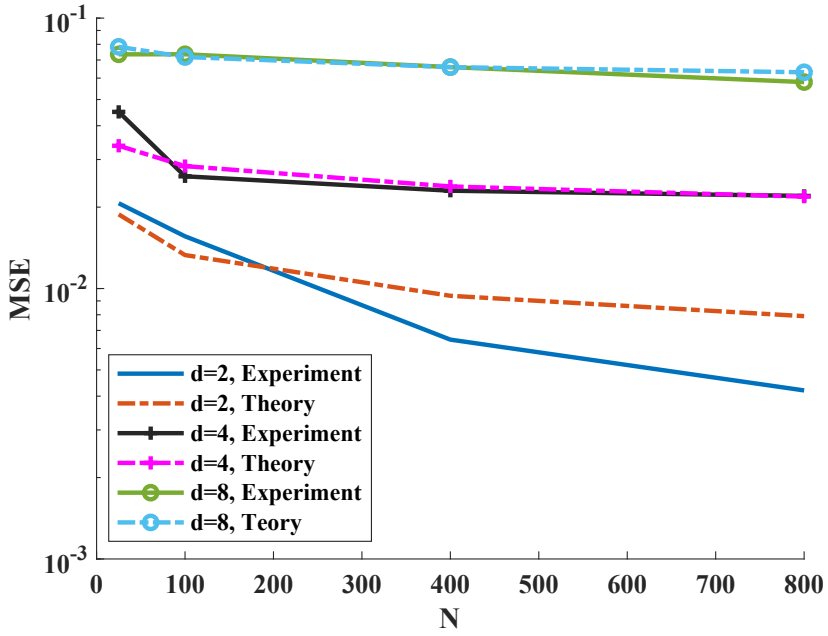


Figure 3.3: Comparison of the bound on the MSE theory and experiments for $d = 2, 4, 8$ standard Gaussian random vectors versus sample size from 100 trials.

up to 800. From Fig. 3.3 we deduce that when the sample size increases the MSE decreases such that for higher dimensions the rate is slower. Furthermore we compare the experiments with the theory in Fig. 3.3. Our theory generally matches the experimental results. However, the MSE for the experiments tends to decrease to zero faster than the theoretical bound. Since the Gaussian distribution has a smooth density, this suggests that a tighter bound on the MSE may be possible by imposing stricter assumptions on the density smoothness as in [87].

In our next simulation we compare three bivariate cases: First, we generate samples from a standard Normal distribution. Second, we consider a distinct smooth

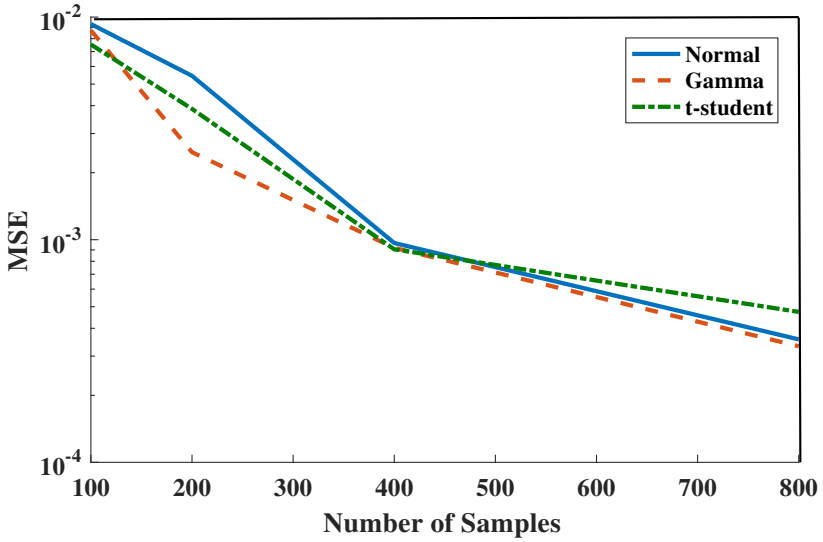


Figure 3.4: Comparison of experimentally predicted MSE of the FR-statistic as a function of sample size $m = n$ in various distributions Standard Normal, Gamma ($\alpha_1 = \alpha_2 = 1,\ \beta_1 = \beta_2 = 1,\ \rho = 0.5$) and Standard t-Student.

class of distributions i.e. binomial Gamma density with standard parameters and dependency coefficient $\rho = 0.5$. Third, we generate samples from Standard t-student distributions [104]. Our goal in this experiment is to compare the MSE of the HP-divergence estimator between two identical distributions, $f_0 = f_1$, when f_0 is one of the Gamma, Normal, and t-student density function. In Fig. 3.4, we observe that the MSE decreases as N increases for all three distributions.

We now show the results of applying the FR test statistic to estimate the HP-divergence using three different real datasets, [73]:

- Human Activity Recognition (HAR), Wearable Computing, Classification of Body Postures and Movements (PUC-Rio): This dataset contains 5 classes (sitting-down, standing-up, standing, walking, and sitting) collected on 8 hours of activities of 4 healthy subjects [117].

- Skin Segmentation dataset (SKIN): The skin dataset is collected by randomly sampling B,G,R values from face images of various age groups (young, middle,

and old), race groups (white, black, and asian), and genders obtained from the FERET and PAL databases [16].

- Sensorless Drive Diagnosis (ENGIN) dataset [9]: In this dataset features are extracted from electric current drive signals. The drive has intact and defective components. The dataset contains 11 different classes with different conditions. Each condition has been measured several times under 12 different operating conditions, e.g. different speeds, load moments and load forces.

We focus on two classes from each of the HAR, SKIN, and ENGIN datasets.

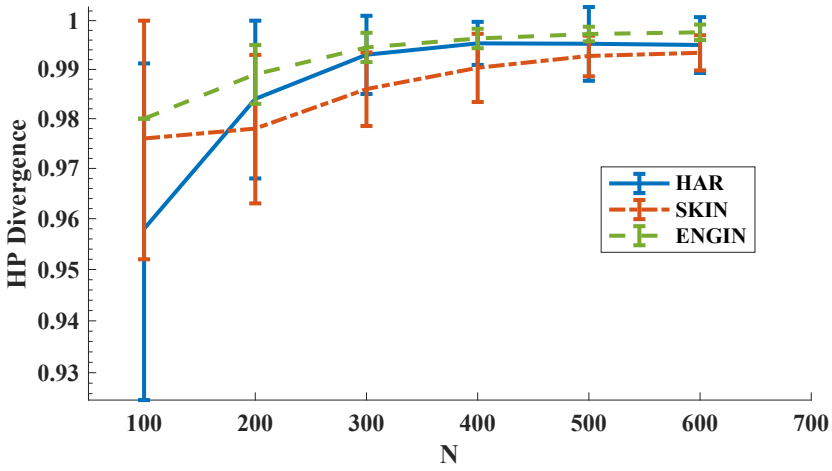


Figure 3.5: HP-divergence vs. sample size for three real datasets HAR, SKIN, and ENGIN.

In the first experiment, we computed the HP-divergence and the MSE for the FR test statistic estimator as the sample size $N = m = n$ increases. We observe in Fig. 3.5 that the estimated HP-divergence ranges in $[0, 1]$, which is one of the HP-divergence properties [13]. Interestingly, when N increases the HP-divergence tends to 1 for all HAR, SKIN, and ENGIN datasets, indicating perfect separation of the classes. Note that in this set of experiments we have repeated the experiments on independent parts of the datasets to obtain the error bars. Fig. 3.6 shows that the MSE expectedly decreases as the sample size grows for all three datasets. Here

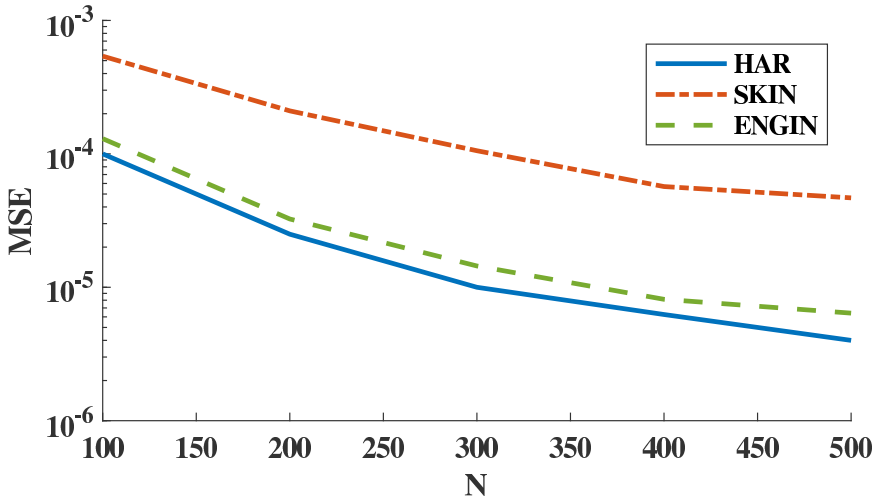


Figure 3.6: The empirical MSE vs. sample size. The empirical MSE of the FR estimator for all three datasets HAR, SKIN, and ENGIN decreases for larger sample size N.

we have used KDE plug-in estimator [87], implemented on all available samples, to determine the true HP-divergence. Furthermore, according to Fig. 3.6 the FR test statistic-based estimator suggests that the Bayes error rate is larger for the SKIN dataset compared to the HAR and ENGIN datasets.

In our next experiment, we explain the incremental value of adding features 1 to 6 features and evaluate the FR test statistic's accuracy as an HP-divergence estimator. Surprisingly, the estimated HP-divergence doesn't appear to increase for the HAR sample, however big increases are observed for the SKIN and ENGIN samples, (see Fig. 3.7).

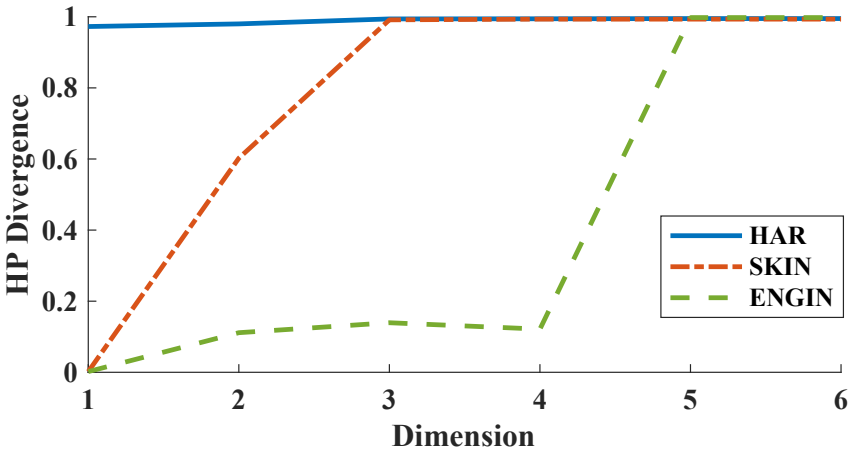


Figure 3.7: HP-divergence vs. dimension for three datasets HAR, SKIN, and ENGIN. This figure shows the incremental value of adding features 1 to 6 features and evaluate the FR test statistic's accuracy as an HP-divergence estimator. Surprisingly, the estimated HP-divergence doesn't appear to increase for the HAR sample, however big increases are observed for the SKIN and ENGIN samples.

3.3 Direct k-NN Estimator of HP Divergence

Let $X = \{X_1, ..., X_N\}$ and $Y = \{Y_1, ..., Y_M\}$ respectively denote i.i.d samples with densities f_1 and f_2, such that $M = \left\lfloor \frac{Nq}{p} \right\rfloor$. Let $G_k(X, Y)$ be the graph of k nearest neighbors of the joint set $X \cup Y$. In other words, edges of $G_k(X, Y)$ are those which connect the points $x \in Z$ to their kth nearest neighbors. Assume that $\mathcal{E}(X, Y)$ is the set of edges of $G_k(X, Y)$ connecting different types. Then the K-NN estimator of HP-divergence, $\widehat{D_p}$, is defined as

$$\widehat{D_p}(X, Y) = 1 - |\mathcal{E}(X, Y)| \frac{N + M}{2NM}. \tag{3.11}$$

The idea behind this estimator is similar to the idea of the MST estimator of HP-divergence proposed by Friedman and Rafsky (FR) [37], in which we count the number of edges connecting different node types in the minimal spanning tree of the merged multitype data. If $N = M$ and the densities are almost equal, then with probability of almost 1/2 every kth nearest neighbor edge belongs to E(X,Y). So

$|\mathcal{E}(X,Y)| \approx N$, and $\widehat{D_p} \approx 0$.

Algorithm 4: k-NN Estimator of HP Divergence

Input : Data sets $X = \{X_1, ..., X_N\}$, $Y = \{Y_1, ..., Y_M\}$

$Z \leftarrow X \cup Y$
for *each point* Z_i *in* Z **do**
  If ($Z_i \in X$ and $Q_k(Z_i) \in Y$)
    or ($Z_i \in Y$ and $Q_k(Z_i) \in X$)
      then $S \leftarrow S + 1$

Output: $1 - S\frac{N+M}{2NM}$

Theorem III.6. *The bias of the k-NN estimator of HP-divergence can be bounded as*

$$\mathbb{B}\left[\widehat{D_p}(X,Y)\right] = O\left((k/N)^{\gamma/d}\right) + O\left(\mathcal{C}(k)\right), \tag{3.12}$$

where $\mathcal{C}(k) := exp(-3k^{1-\delta})$ *for a fixed* $\delta \in (2/3, 1)$. *Here* γ *is the Hölder smoothness parameter.*

Remark III.7. Note that in order to have a asymptotically unbiased estimator, k needs to grow with N. The optimum bias rate of $O\left(\frac{\log N}{N}\right)^{\gamma/d}$ can be achieved for $k = O(\log N)$.

Theorem III.8. *The variance of the k-NN estimator of HP divergence is*

$$\mathbb{V}\left[\widehat{D_p}(X,Y)\right] \leq O\left(\frac{1}{N}\right). \tag{3.13}$$

3.3.1 WNN Estimator

Note that the bias term in Theorem VII.2 depends on d. This fact implies that for higher dimensions the convergence rate is slower. We propose an estimator that achieves the optimum convergence rate in any dimension using an ensemble estimator introduced in [112]. Assume that the density functions are in the Hölder space $\Sigma(\gamma, B)$, which consists of functions on $\mathcal{X}$ continuous derivatives up to order $q =$

$\lfloor\gamma\rfloor \geq d$ and the qth partial derivatives are Hölder continuous with exponent $\gamma' =: \gamma - q$ and constant parameter of B. Further, we need to assume that the density derivatives up to order d vanish at the boundary. Fix a constant L where $L \geq d$. Let $\mathcal{L} := \{l_1, ..., l_L\}$ be a set of index values with $l_i < c$, where $\kappa = \left\lfloor c\sqrt{N} \right\rfloor$. We further define $K(l) := \left\lfloor l\sqrt{N} \right\rfloor$.

Definition III.9. Let $X = \{X_1, ..., X_N\}$ and $Y = \{Y_1, ..., Y_M\}$ respectively denote i.i.d samples with densities f_1 and f_2, such that $M = \left\lfloor \frac{Nq}{p} \right\rfloor$. Let the weight vector $W := [W(l_1), W(l_2), ..., W(l_L)]$ be the solution to the following optimization problem:

$$\begin{aligned} \min_{w} \quad & \|w\|_2 \\ \textit{subject to} \quad & \sum_{l\in\mathcal{L}} w(l) = 1, \\ & \sum_{l\in\mathcal{L}} w(l) l^{i/d} = 0, i \in \mathbb{N}, i \leq d. \end{aligned} \tag{3.14}$$

Now define $G_K^W(X, Y)$ as a weighted directed graph with the vertices of the joint set $X \cup Y$. There is a directed edge with the weight $W(l)$ between any pair of nodes R and S, only if the types of R and S are different (i.e. $R \in X$ and $S \in Y$), and S is the $K(l)$th nearest neighbor of R for some $l \in \mathcal{L}$. We represent the set of edges of $G_K^W(X, Y)$ by $\mathcal{E}_K^W(X, Y)$.

Then the WNN estimator $\widehat{D}_p^W$, is defined as

$$\widehat{D}_p^W(X, Y) = 1 - |\mathcal{E}_K^W(X, Y)|\frac{N + M}{2NM}. \tag{3.15}$$

In the following theorem we prove that WNN estimator defined above achieves the optimal MSE rate of $O(1/N)$:

Theorem III.10. *Mean square error of WNN estimator can be bounded by* $O(1/N)$.

Algorithm 5: WNN Estimator of HP Divergence

Input : Data sets $X = \{X_1, ..., X_N\}$, $Y = \{Y_1, ..., Y_M\}$

$Z \leftarrow X \cup Y$
for $l \in \mathcal{L}$ **do**
  for *each point* Z_i *in* Z **do**
    If ($Z_i \in X$ and $Q_l(Z_i) \in Y$)
      or ($Z_i \in Y$ and $Q_l(Z_i) \in X$)
        then $S \leftarrow S + W(l)$

Output: $1 - S\frac{N+M}{2NM}$

3.4 Numerical Results

In this section we investigate the behavior of the proposed estimator using a few numerical experiments and compare them with the theoretical bounds.

The first experiment, in Fig. 3.8, shows the mean estimated HP divergence of two truncated Normal RVs with the mean vectors $[0, 0]$ and $[0, 1]$ and variance of $\sigma_1^2 = \sigma_2^2 = I_2$, as a function of number of samples, N, where I_d is the identity matrix with size d. Three different values of k are investigated. For each case we repeat the experiment 100 times, and compute the expectation of the estimated value and the standard deviation error bars. As N increases, the expected value of the estimated divergence measures for any k tend to the true value. The experiments show that as we increase k the bias also increases, which is due to the $\left(\frac{k}{N}\right)^{\gamma/d}$ bias term (the other term is ignorable). However, according to this experiment, variance is almost independent of k, which verifies the theoretical bound on variance.

Fig. 3.9 shows the MSE of the k-NN estimator for HP divergence between two identical, independent and truncated Normal RVs. The RVs have the same covariance matrix of I_d and are truncated with the range $x \in [-5, 5]$ and $y \in [-5, 5]$. The experiment is repeated for three different dimensions of $d = 2, 10, 20$ for a fixed $k = 5$. In agreement with the theoretical bias bound, as d increases, the experiment shows that MSE rate increases.

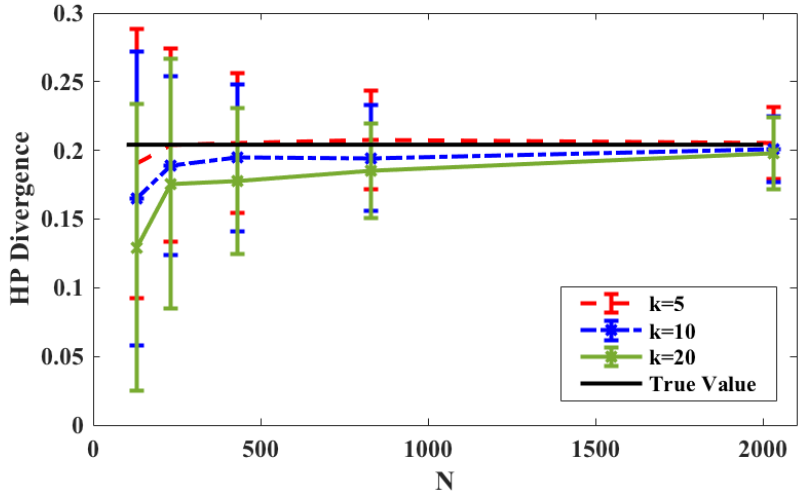


Figure 3.8: Comparison of the estimated values of k-NN estimator with $k = 5, 10, 20$ for HP divergence between two truncated Normal RVs with the mean vectors $[0, 0]$ and $[0, 1]$ and variances of $\sigma_1^2 = \sigma_2^2 = I_2$, versus N, the number of samples.

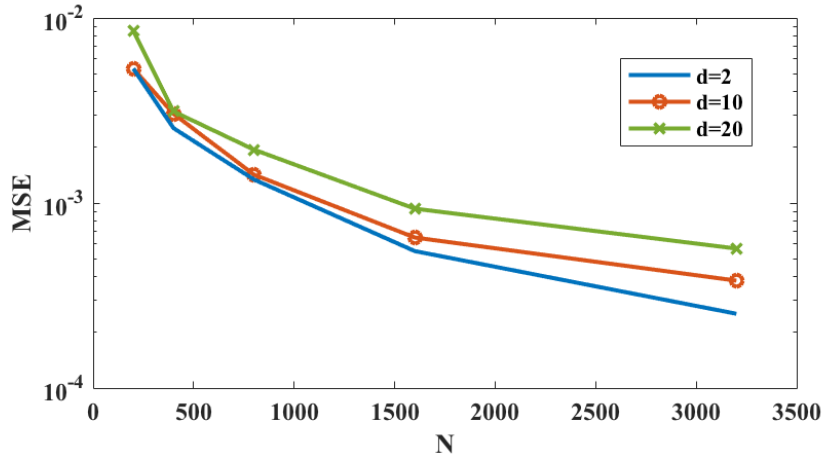


Figure 3.9: MSE of the k-NN estimator for HP divergence between two identical, independent and truncated Normal RVs, as a function of N.

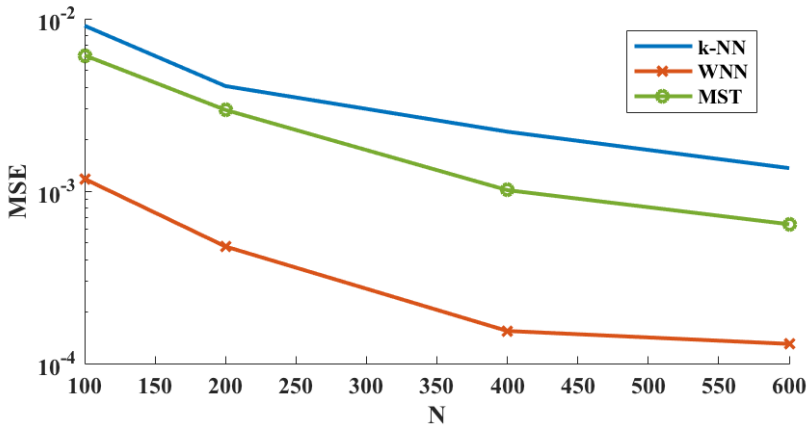


Figure 3.10: MSE comparison of the three graph theoretical estimators of HP divergence; MST, k-NN, and WNN estimators.

In Fig. 3.10 we compare the MSE rates of the three graph theoretical estimators of HP divergence; MST, k-NN, and WNN estimators. The divergence is considered between two truncated Normal random variables with $d = 2$, means of $\mu_1 = [0, 0]$, $\mu_2 = [1, 0]$, and covariance matrices of $\sigma_1 = I_2$ and $\sigma_2 = 2I_2$. This experiment verifies the advantage of WNN estimator over k-NN and MST estimators, in terms of their convergence rates. Also the performance of MST estimator is slightly better than the k-NN estimator. Note that in this experiment we have $k = 5$.

Fig. 3.11 shows the comparison of the estimators of HP divergence between a truncated Normal RV with mean $[0, 0]$ and covariance matrix of I_2, and uniform RV within $[-5, 5] \times [-5, 5]$, in terms of their mean value and %95 confidence band. The confidence band is narrower for greater values of N, and WNN estimator has the narrowest confidence band.

Finally in Fig. 3.12, we compare performance of WNN to k-NN estimators with $k = 5$ and $k = 10$, for a real data set [36, 35]. The data are measurement outcomes of a set of ultrasound sensors arranged circularly around a robot, which navigates through the room following the wall in a clockwise direction. There are total number

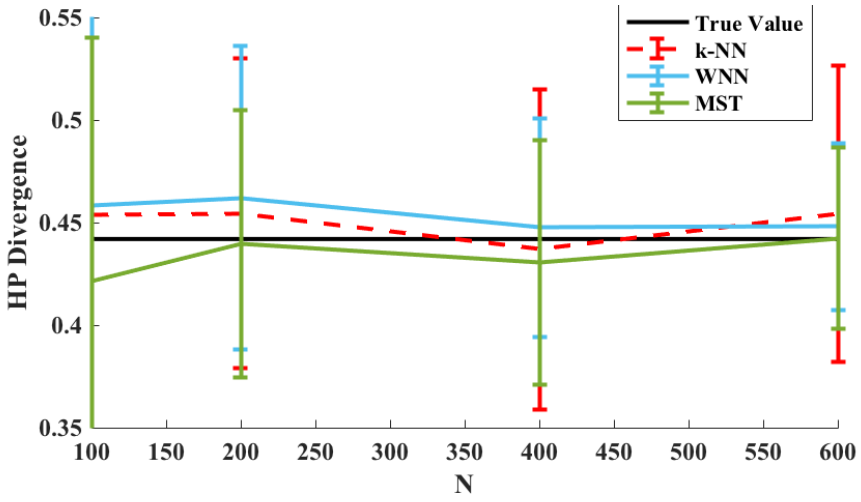


Figure 3.11: Comparison k-NN, MST and WNN estimators of HP divergence between a truncated Normal RV and a uniform RV, in terms of their mean value and %95 confidence band.

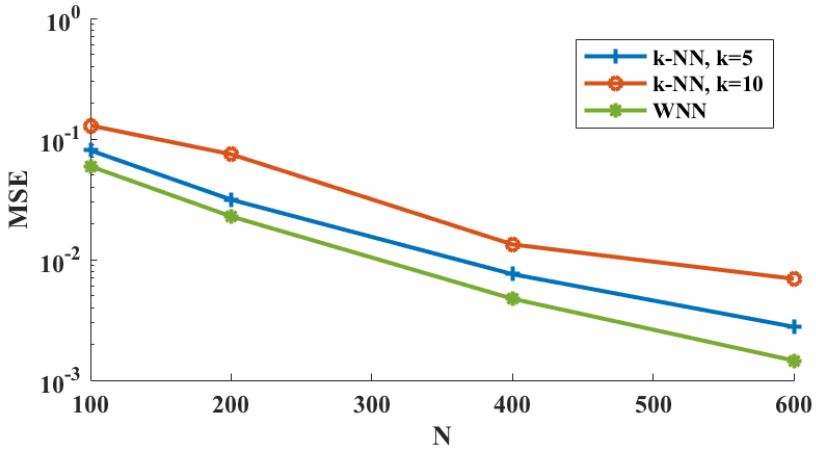


Figure 3.12: MSE Comparison of the WNN and k-NN estimator with two different parameters $k = 5$ and $k = 10$ for the robot navigation dataset

of 5456 instances (corresponding to different timestamps), and we use the information of four main sensors as the feature space. The instances are associated to four different classes of actions: move-forward, sharp-right-turn, slight-right-turn and turn-left. In Fig. 3.12 we consider the divergence between the sensor measurement for sharp-right-turn and move-forward classes. In general WNN estimator performs better than k-NN estimator.

3.5 Conclusion

In this chapter we first derived a bound on the MSE convergence rate for the Friedman-Rafsky estimator of the Henze-Penrose divergence assuming the densities are sufficiently smooth. We employed a partitioning strategy to derive the bias rate which depends on the number of partitions, the sample size $m + n$, the Hölder smoothness parameter η, and the dimension d. We validated our proposed MSE convergence rate using simulations and illustrated the approach for the meta-learning problem of estimating the HP-divergence for three real-world data sets.

In the second part of this chapter we proposed the k-NN version of the FR test statistic. We established convergence and proposed an optimum direct estimation method for HP divergence, based on the ensemble method. We proved that WNN estimator can achieve the optimum MSE rate of $O(1/N)$, and validated our results on simulated and real data sets. For future work, one interesting direction would be to investigate the convergence rate for the k-NN estimator, using fixed k independent of N.

CHAPTER IV

Learning to Benchmark: Optimum Estimation of Bayes Error

In this chapter we address the problem of learning to benchmark the best achievable classifier performance. In this problem, the objective is to establish statistically consistent estimates of the Bayes misclassification error rate without having to learn a Bayes-optimal classifier. This chapter's approach improves the previous chapter's work on learning bounds on Bayes misclassification rate since it learns the *exact* Bayes error rate instead of a bound on error rate. We propose a benchmark learner based on an ensemble of ε-ball estimators and Chebyshev approximation. Under a smoothness assumption on the class densities we show that our estimator achieves an optimal (parametric) mean squared error (MSE) rate of $O(N^{-1})$, where N is the number of samples. Experiments on both simulated and real datasets establish that our proposed benchmark learning algorithm produces estimates of the Bayes error that are more accurate than previous approaches for learning bounds on Bayes error probability.

In Section 4.1, we introduce our proposed Bayes error rate estimators for the binary classification problem. In Section 4.2 we use the ensemble estimation method to improve the convergence rate of the base estimator. We then address the multi-class classification problem in Section 4.3. In Section 4.4, we conduct numerical

experiments to illustrate the performance of the estimators. Finally, we discuss the future work in Section 4.5.

4.1 Benchmark learning for Binary Classification

Our proposed learning to benchmark framework is based on an exact f-divergence representation (not a bound) for the minimum achievable binary misclassification error probability. First, in section 4.1.1 we propose an accurate estimator of the density ratio (ε-ball estimator), and then in section 4.1.2, based on the optimal estimation for the density ratio, we propose a base estimator of Bayes error rate.

4.1.1 ε-Ball Density Ratio Estimator

Consider the independent and identically distributed (i.i.d) sample realizations $\mathbf{X}_1 = \left\{X_{1,1}, X_{1,2}, \ldots, X_{1,N_1}\right\} \in \mathbb{R}^{N_1 \times d}$ from f_1 and $\mathbf{X}_2 = \left\{X_{2,1}, X_{2,2}, \ldots, X_{2,N_2}\right\} \in \mathbb{R}^{N_2 \times d}$ from f_2. Let $\eta := N_2/N_1$ be the ratio of two sample sizes. The problem is to estimate the density ratio $U(x) := \frac{f_1(x)}{f_2(x)}$ at each of the points of the set $\mathbf{X}_2$. In this chapter similar to the method of [93] we use the ratio of counts of nearest neighbor samples from different classes to estimate the density ratio at each point. However, instead of considering the k-nearest neighbor points, we use the ϵ-neighborhood (in terms of euclidean distance) of the points. This allows us to remove the extra bias due to the discontinuity of the parameter k when using an ensemble estimation technique. As shown in Figure. 4.1, ε-ball density ratio estimator for each point Y_i in $\mathbf{Y}$ (shown by blue points) is constructed by the ratio of the counts of samples in $\mathbf{X}$ and $\mathbf{Y}$ which fall within ε-distance of Y_i.

Definition IV.1. For each point $X_{2,i} \in \mathbf{X}_2$, let $N_{1,i}^{(\varepsilon)}$ (*resp.* $N_{2,i}^{(\varepsilon)}$) be the number of points belonging to $\mathbf{X}_1$ (*resp.* $\mathbf{X}_2$) within the ε-neighborhood (ε-ball) of $X_{2,i}$. Then

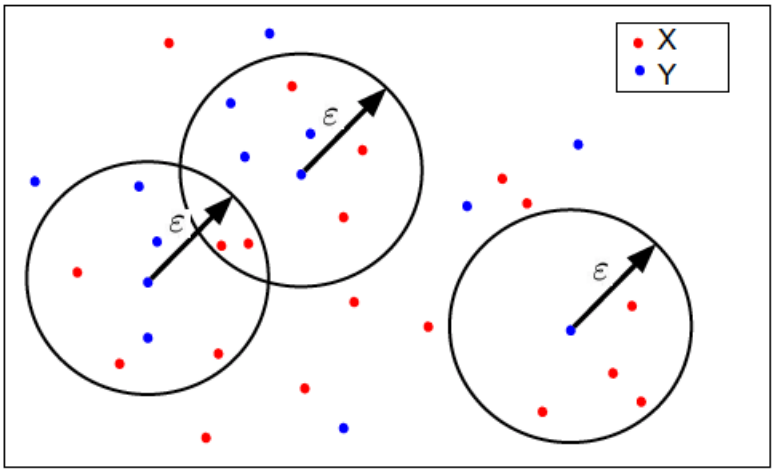


Figure 4.1: ε-ball density ratio estimator for each point Y_i in $\mathbf{Y}$ (shown by blue points) is constructed by the ratio of the counts of samples in $\mathbf{X}$ and $\mathbf{Y}$ which fall within ε-distance of Y_i.

the density ratio estimate is given by

$$\widehat{U}^{(\varepsilon)}(X_{2,i}) := \eta N_{1,i}^{(\varepsilon)} \Big/ N_{2,i}^{(\varepsilon)} \ . \tag{4.1}$$

Sometimes in this chapter we abbreviate $\widehat{U}(X_{2,i})$ as $\widehat{U}_i^{(\varepsilon)}$.

4.1.2 Base learner of Bayes error

The Bayes error rate corresponding to class densities f_1, f_2, and the class probabilities vector $\mathbf{p} = (p_1, p_2)$ is

$$\begin{aligned} &\mathcal{E}_{\mathbf{p}}^{\text{Bayes}}(f_1, f_2) = \Pr(C^{\text{Bayes}}(X) \neq T) \\ &= \int_{p_1 f_1(x) \leq p_2 f_2(x)} p_1 f_1(x) dx + \int_{p_1 f_1(x) \geq p_2 f_2(x)} p_2 f_2(x) dx, \end{aligned} \tag{4.2}$$

where $C^{\text{Bayes}}(X)$ is the classifier mapping $C^{\text{Bayes}} : \mathcal{X} \to \{1, 2\}$. The Bayes error (4.2) can be expressed as

$$\begin{aligned}
\mathcal{E}_{\mathbf{p}}^{\text{Bayes}}(f_1, f_2) &= \frac{1}{2}\int p_1 f_1(x) + p_2 f_2(x) - |p_1 f_1(x) - p_2 f_2(x)|dx \\
&= p_2 + \frac{1}{2}\int (p_1 f_1(x) - p_2 f_2(x)) - |p_1 f_1(x) - p_2 f_2(x)|dx \\
&= \min(p_1, p_2) - \int f_2(x) t\left(\frac{f_1(x)}{f_2(x)}\right) dx \\
&= \min(p_1, p_2) - \mathbb{E}_{f_2}\left[t\left(\frac{f_1(X)}{f_2(X)}\right)\right], \qquad (4.3)
\end{aligned}$$

where

$$t(x) := \max(p_2 - p_1 x, 0) - \max(p_2 - p_1, 0)$$

is a convex function. The expectation $\mathbb{E}_{f_2}\left[t\left(\frac{f_1(X)}{f_2(X)}\right)\right]$ is an f-divergence between density functions f_1 and f_2. The f-divergence or Ali-Silvey distance, introduced in [4], is a measure of the dissimilarity between a pair of distributions. Several estimators of f-divergences have been introduced [15, 120, 90, 97]. Expressions for the bias and variance of these estimators are derived under assumptions that the function t is differentiable, which is not true here. In what follows we will only need to assume that the divergence function t is Lipschitz continuous.

We make the following assumption on the densities. Note that these are similar to the assumptions made in the previous work [109, 91, 77].

Assumptions:

A.1. The densities functions f_1 and f_2 are both lower bounded by C_L and upper bounded by C_U with $C_U \geq C_L > 0$;

A.2. The densities f_1 and f_2 are Hölder continuous with parameter $0 < \gamma \leq 1$, that is there exists constants $H_1, H_2 > 0$ such that

$$|f_i(x_1) - f_i(x_2)| \leq H_i ||x_1 - x_2||^{\gamma}, \qquad (4.4)$$

for $i = 1, 2$ and $x_1, x_2 \in \mathbb{R}$.

Explicit upper and lower bounds C_U and C_L must be specified for the implementation of the base estimator below. However, the lower and upper bounds do not need to be tight and only affect the convergence rate of the estimator. We conjecture that this assumption can be relaxed, but this is left for future work.

Define the base estimator of the Bayes error

$$\widehat{\mathcal{E}}_\varepsilon(\mathbf{X}_1, \mathbf{X}_2) := \min(\hat{p}_1, \hat{p}_2) - \frac{1}{N_2} \sum_{i=1}^{N_2} \tilde{t}\left(\widehat{U}_i\right), \tag{4.5}$$

where $\tilde{t}(x) := \max(t(x), t(C_L/C_U))$, and empirical estimates vector $\hat{\mathbf{p}} = (\hat{p}_1, \hat{p}_2)$ is obtained from the relative frequencies of the class labels in the training set. $\widehat{U}_i$ is the estimation of the density ratio at point $\mathbf{X}_{2,i}$, which can be computed based on ε-ball estimates.

Remark IV.2. The definition of Bayes error in (4.2) is symmetric, however, the definition of Bayes error estimator in (4.5) is asymmetric with respect to $\mathbf{X}_1$ and $\mathbf{X}_2$. Therefore, we might get different estimations from $\widehat{\mathcal{E}}_\varepsilon(\mathbf{X}_1, \mathbf{X}_2)$ and $\widehat{\mathcal{E}}_\epsilon(\mathbf{X}_2, \mathbf{X}_1)$, while both of these estimations asymptotically converge to the true Bayes error. It is obvious that any convex combination of $\widehat{\mathcal{E}}_\varepsilon(\mathbf{X}_1, \mathbf{X}_2)$ and $\widehat{\mathcal{E}}_\epsilon(\mathbf{X}_2, \mathbf{X}_1)$ defined is also an estimator of the Bayes error (with the same convergence rate). In particular, we define the following symmetrized Bayes error estimator:

$$\begin{aligned} \mathcal{E}^*_\epsilon(\mathbf{X}_2, \mathbf{X}_1) &:= \frac{N_2}{N}\widehat{\mathcal{E}}_\varepsilon(\mathbf{X}_1, \mathbf{X}_2) + \frac{N_1}{N}\widehat{\mathcal{E}}_\epsilon(\mathbf{X}_2, \mathbf{X}_1) \\ &= \min(\hat{p}_1, \hat{p}_2) - \frac{1}{N} \sum_{i=1}^{N} \tilde{t}\left(\widehat{U}_i\right), \end{aligned} \tag{4.6}$$

where consistent with the definition in (4.1), for the points in $\mathbf{X}_1$, $\widehat{U}_i^{(\varepsilon)}$ is defined as the ratio of the ε-neighbor points in $\mathbf{X}_2$ to the number of points in $\mathbf{X}_1$, while for the points in $\mathbf{X}_2$ is defined as the ratio of the points in $\mathbf{X}_1$ to the number of points in

$\mathbf{X}_2$:

$$\widehat{U}_i^{(\varepsilon)} := \begin{cases} \eta N_{1,i}^{(\varepsilon)} \Big/ N_{2,i}^{(\varepsilon)} & 1 \le i \le N_2 \\ N_{2,i}^{(\varepsilon)} \Big/ \eta N_{1,i}^{(\varepsilon)} & N_2 \le i \le N. \end{cases} \tag{4.7}$$

Algorithm 6: Base Learner of Bayes Error

Input : Data sets $\mathbf{X} = \{X_1, ..., X_{N_1}\}$, $\mathbf{Y} = \{Y_1, ..., Y_{N_2}\}$

$\mathbf{Z} \leftarrow \mathbf{X} \cup \mathbf{Y}$

for *each point* Y_i *in* Y **do**

  $\mathbf{S}_i$: Set of ε-ball points of Y_i in $\mathbf{Z}$

  $\widehat{U}_i \leftarrow |\mathbf{S}_i \cap \mathbf{X}| / |\mathbf{S}_i \cap \mathbf{Y}|$

$\mathcal{E}_\epsilon^*(\mathbf{X}_2, \mathbf{X}_1) \leftarrow \min(N_1, N_2)/(N_1 + N_2) - \frac{1}{N} \sum_{i=1}^{N} \tilde{t}\left(\widehat{U}_i\right)$,

Output: $\mathcal{E}_\epsilon^*(\mathbf{X}_2, \mathbf{X}_1)$

Remark IV.3. The ε-ball density ratio estimator is equivalent to the ratio of plug-in kernel density estimators with a top-hat filter and bandwidth ε.

4.1.3 Convergence Analysis

The following theorem states that this estimator asymptotically converges in L^2 norm to the exact Bayes error as N_1 and N_2 go to infinity in a manner $N_2/N_1 \to \eta$, with an MSE rate of $O(N^{-\frac{2\gamma}{\gamma+d}})$.

Theorem IV.4. *Under the Assumptions on f_1 and f_2 stated above, as $N_1, N_2 \to \infty$ with $N_2/N_1 \to \eta$,*

$$\widehat{\mathcal{E}}_\varepsilon(\mathbf{X}_1, \mathbf{X}_2) \xrightarrow{L^2} \mathcal{E}_{\mathbf{p}}^{Bayes}(f_1, f_2), \tag{4.8}$$

where $\xrightarrow{L^2}$ denotes "convergence in L^2 norm". Further, the bias of $\mathcal{E}(\mathbf{X}_1, \mathbf{X}_2)$ is

$$\mathbb{B}\left[\widehat{\mathcal{E}}_\varepsilon(\mathbf{X}_1, \mathbf{X}_2)\right] = O\left(\epsilon^\gamma\right) + O\left(\epsilon^{-d} N_1^{-1}\right), \tag{4.9}$$

where ε is the radius of the neighborhood ball.

In addition, the variance of $\widehat{\mathcal{E}}_\varepsilon(\mathbf{X}_1, \mathbf{X}_2)$ is

$$\mathbb{V}\left[\widehat{\mathcal{E}}_\varepsilon(\mathbf{X}_1, \mathbf{X}_2)\right] = O\left(1/\min(N_1, N_2)\right). \tag{4.10}$$

Proof. Since according to (4.3) the Bayes error rate $\mathcal{E}^{\text{Bayes}}$ can be written as an f-divergence, it suffice to derive the bias and variance of the ε-ball estimator of the divergence. The details are given in Appendix. C.1. □

In the following we give a theorem that establishes the Gaussian convergence of the estimator proposed in equation (4.5).

Theorem IV.5. *Let $\varepsilon \to 0$ and $\frac{1}{\varepsilon^d N} \to 0$. If S be a standard normal random variable with mean 0 and variance 1, then,*

$$Pr\left(\frac{\widehat{\mathcal{E}}_\varepsilon(\mathbf{X}_1,\mathbf{X}_2) - \mathbb{E}\left[\widehat{\mathcal{E}}_\varepsilon(\mathbf{X}_1,\mathbf{X}_2)\right]}{\sqrt{\mathbb{V}\left[\widehat{\mathcal{E}}_\varepsilon(\mathbf{X}_1,\mathbf{X}_2)\right]}} \leq t\right) \to Pr(S \leq t) \tag{4.11}$$

Proof: The proof is based on the Slutsky's Theorem and Efron-Stein inequality and is discussed in details in Appendix. C.2.

4.2 Ensemble of Base Learners

It has long been known that ensemble averaging of base learners can improve the accuracy and stability of learning algorithms [30]. In this work in order to achieve the optimal parametric MSE rate of $O(1/N)$, we propose to use an ensemble estimation technique. The ensemble estimation technique has previously used in estimation of f-divergence and mutual information measures [77, 83, 90]. However, the method used by these articles depends on the assumption that the function f of the divergence (or general mutual information) measure is differentiable everywhere within its the domain. As contrasted to this assumption, function $t(x)$ defined in equation (4.3) is not differentiable at $x = p_1/p_2$, and as a result, using the ensemble estimation technique considered in the previous work is difficult. A simpler construction of the ensemble Bayes error estimation is discussed in section 4.2.1. Next, in section 4.2.2 we propose an optimal weight assigning method based on Chebyshev polynomials.

4.2.1 Construction of the Ensemble Estimator

Our proposed ensemble benchmark learner constructs a weighted average of L density ratio estimates defined in (4.1), where each density ratio estimator uses a different value of ϵ.

Definition IV.6. Let $\widehat{U}_i^{(\varepsilon_j)}$ for $j \in \{1, ..., L\}$ be L density ratio estimates with different parameters (ε_j) at point Y_i. For a fixed weight vector $\mathbf{w} := (w_1, w_2, \ldots, w_L)^T$, the ensemble estimator is defined as

$$\mathcal{F}(\mathbf{X}_1, \mathbf{X}_2) = \min(\hat{p}_1, \hat{p}_2) - \frac{1}{N_2} \sum_{i=1}^{N_2} \left[\max(\hat{p}_2 - \hat{p}_1 \widehat{U}_i^{\mathbf{w}}, 0) - \max(\hat{p}_2 - \hat{p}_1, 0) \right], \tag{4.12}$$

where for the weighted density ratio estimator, $\widehat{U}_i^{\mathbf{w}}$ is defined as

$$\widehat{U}_i^{\mathbf{w}} := \sum_{l=1}^{L} w_l \widehat{U}_i^{(\varepsilon_l)}. \tag{4.13}$$

Remark IV.7. The construction of this ensemble estimator is fundamentally different from standard ensembles of base estimators proposed before and, in particular, different from the methods proposed in [77, 90]. These standard methods average the base learners whereas the ensemble estimator (4.12) averages over the argument (estimated likelihood ratio f_1/f_2) of the base learners.

Under additional conditions on the density functions, we can find the weights w_l such that the ensemble estimator in (4.12) achieves the optimal parametric MSE rate $O(1/N)$. Specifically, assume that 1) the density functions f_1 and f_2 are both Hölder continuous with parameter γ and continuously differentiable of order $q = \lfloor \gamma \rfloor \geq d$,and 2) the q-th derivatives are Hölder continuous with exponent $\gamma' := \gamma - q$. These are similar to assumptions that have been made in the previous work [77, 109, 90]. We prove that if the weight vector $\mathbf{w}$ is chosen according to an optimization problem, the ensemble estimator can achieve the optimal parametric MSE rate $O(1/N)$.

Theorem IV.8. *Let $N_1, N_2 \to \infty$ with $N_2/N_1 \to \eta$. Also let $\widehat{U}_i^{(\varepsilon_j)}$ for $j \in \{1, ..., L\}$ be L ($L > d$) density ratio estimates with bandwidths $\varepsilon_j := \xi_j N_1^{-1/2d}$ at the points Y_i. Define the weight vector $\mathbf{w} = (w_1, w_2, \ldots, w_L)^T$ as the solution to the following optimization problem:*

$$\min_{\mathbf{w}} \quad ||\mathbf{w}||_2 \tag{4.14}$$

$$\text{subject to} \quad \sum_{l=1}^{L} w_l = 1 \quad \text{and} \quad \sum_{l=1}^{L} w_l \cdot \xi_l^i = 0, \qquad \forall i = 1, \ldots, d.$$

Then, under the assumptions stated above the ensemble estimator defined in (4.12) *satisfies,*

$$\mathcal{F}(\mathbf{X}_1, \mathbf{X}_2) \xrightarrow{L^2} \mathcal{E}_{\mathbf{p}}^{Bayes}(f_1, f_2), \tag{4.15}$$

with the MSE rate $O(1/N_1)$.

Proof. See Appendix C.3. □

One simple choice for ξ_l is an arithmetic sequence as $\xi_l := l$. With this setting the optimization problem in the following optimization problem:

$$\min_{\mathbf{w}} \quad ||\mathbf{w}||_2 \tag{4.16}$$

$$\text{subject to} \quad \sum_{l=1}^{L} w_l = 1 \quad \text{and} \quad \sum_{l=1}^{L} w_l \cdot l^i = 0, \qquad \forall i = 1, \ldots, d. \tag{4.17}$$

Note that the optimization problem in (4.16) does not depend on the data sample distribution and only depends on its dimension. Thus, it can be solved offline. In larger dimensions, however, solving the optimization problem can be computationally difficult. In the following we provide an optimal weight assigning approach based on Chebyshev polynomials that reduces computational complexity and leads to improved stability. We use the orthogonality properties of the Chebyshev polynomials to derive closed form solutions for the optimal weights in (4.14).

4.2.2 Chebyshev Polynomial Approximation Method for Ensemble Estimation

Chebyshev polynomials are frequently used in function approximation theory [59]. We denote the Chebyshev polynomials of the first kind defind in interval $[-1, 1]$ by T_n, where n is the degree of the polynomial. An important feature of Chebyshev polynomials is that the roots of these polynomials are used as polynomial interpolation points. We define the shifted Chebyshev polynomials with a parameter α as $T_n^\alpha(x) : [0, \alpha] \to \mathbb{R}$ in terms of the standard Chebyshev polynomials as

$$T_n^\alpha(x) = T_n(\frac{2x}{\alpha} - 1). \tag{4.18}$$

We denote the roots of $T_n^\alpha(x)$ by $s_i, i \in \{1, ..., n\}$. In this section we formulate the ensemble estimation optimization in equation (4.14) in the Chebyshev polynomials basis and we propose a simple closed form solution to this optimization problem. This is possible by setting the parameters of the base density estimators ε_l proportional to the Chebyshev nodes s_l. Precisely, in equation (4.14) we set

$$\xi_l := s_l. \tag{4.19}$$

Theorem IV.9. *For $L > d$, the solutions of the optimization problem in* (4.14) *for $\xi_l := s_l$ are given as:*

$$w_i = \frac{2}{L} \sum_{k=0}^{d} T_k^\alpha(0) T_k^\alpha(s_i) - \frac{1}{L} \qquad \forall i \in \{0, ..., L-1\}. \tag{4.20}$$

where $s_i, i \in \{0, ..., L-1\}$ are roots of $T_L^\alpha(x)$ given by

$$s_k = \frac{\alpha}{2} \cos\left(\left(k + \frac{1}{2}\right)\frac{\pi}{L}\right) + \frac{\alpha}{2}, \quad k = 0, \ldots, L-1 \tag{4.21}$$

Proof. The proof of Theorems IV.9 can be found in Appendix C.4. □

4.3 Benchmark Learning for Multi-class Classification

Consider a multi-class classification problem with λ classes having respective density functions $f_1, f_2, \ldots, f_\lambda$. The Bayes error rate for the multi-class classification is

$$\begin{aligned}
&\mathcal{E}_{\mathbf{p}}^{\text{Bayes}}(f_1, f_2, \ldots, f_\lambda) \\
&= 1 - \int \left[\max_{1\le i\le \lambda} p_i f_i(x)\right] dx \\
&= 1 - p_1 - \sum_{k=2}^{\lambda} \int \left[\max_{1\le i\le k} p_i f_i(x) - \max_{1\le i\le k-1} p_i f_i(x)\right] dx \\
&= 1 - p_1 - \sum_{k=2}^{\lambda} \int \max\left(0, p_k - \max_{1\le i\le k-1} p_i f_i(x)/f_k(x)\right) f_k(x) dx \\
&= 1 - p_1 - \sum_{k=2}^{\lambda} \int t_k\left(\frac{f_1(x)}{f_k(x)}, \frac{f_2(x)}{f_k(x)}, \ldots, \frac{f_{k-1}(x)}{f_k(x)}\right) f_k(x) dx, \qquad (4.22)
\end{aligned}$$

where

$$t_k(x_1, x_2, \ldots, x_{k-1}) := \max\left(0, p_k - \max_{1\le i\le k-1} p_i x_i\right).$$

We denote the density fractions $\frac{f_i(x)}{f_j(x)}$ in the above equation by $U_{(i/j)}(x)$. Let $\widehat{U}^{\mathbf{w}}_{(i/j)}(x)$ denote the ensemble estimates of $U_{(i/j)}(x)$ using the ε-ball method, similar to the estimator defined in (C.18). Thus, we propose the following direct estimator of $\mathcal{E}_{\mathbf{p}}^{\text{Bayes}}(f_1, f_2, \ldots, f_\lambda)$ as follows:

$$\begin{aligned}
\mathcal{H}(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_\lambda) &:= 1 - p_1 - \\
&\sum_{l=2}^{\lambda} \frac{1}{N_l} \sum_{i=1}^{N_l} \tilde{t}\left(\widehat{U}^{\mathbf{w}}_{(1/l)}(X_{l,i}), \widehat{U}^{\mathbf{w}}_{(2/l)}(X_{l,i}), \ldots, \widehat{U}^{\mathbf{w}}_{(l-1/l)}(X_{l,i})\right), \qquad (4.23)
\end{aligned}$$

where

$$\tilde{t}_k(x_1, x_2, \ldots, x_{k-1}) := \max\left\{t_k(x_1, x_2, \ldots, x_{k-1}), t_k(C_L/C_U, \ldots, C_L/C_U)\right\}.$$

Since t is elementwise Lipschitz continuous, we can easily generalize the argument used in the proof of Theorem IV.4 to obtain the convergence rates for the multiclass

case. Similar to the assumptions of the ensemble estimator for the binary case in section 4.2.1, we assume that 1) the density functions $f_1, f_2, ..., f_\lambda$ are both Hölder continuous with parameter γ and continuously differentiable of order $q = \lfloor \gamma \rfloor \geq d$ and 2) the q-th derivatives are Hölder continuous with exponent $\gamma' := \gamma - q$.

Theorem IV.10. *As $N_1, N_2, \ldots, N_\lambda \to \infty$ with $N_l/N_j \to \eta_{j,l}$ for $1 \leq j < l \leq \lambda$ and $N^* = \max(N_1, N_2, \ldots, N_\lambda)$,*

$$\mathcal{H}_k(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_\lambda) \xrightarrow{L^2} \mathcal{E}_{\mathbf{p}}^{Bayes}(f_1, f_2, \ldots, f_\lambda). \tag{4.24}$$

The bias and variance of $\mathcal{H}_k(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_\lambda)$ are

$$\mathbb{B}\left[\mathcal{H}_k(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_\lambda)\right] = O\left(\lambda/\sqrt{N^*}\right), \tag{4.25}$$

$$\mathbb{V}\left[\mathcal{H}_k(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_\lambda)\right] = O\left(\lambda^2/N^*\right). \tag{4.26}$$

Proof. See Appendix C.5. □

Remark IV.11. Note that the estimator $\mathcal{H}_k$ (4.23) depends on the ordering of the classes, which is arbitrary. However the asymptotic MSE rates do not depend on the particular class ordering.

Remark IV.12. In fact, (4.22) can be transformed into

$$\mathcal{E}_{\mathbf{p}}^{\text{Bayes}}(f_1, f_2, \ldots, f_\lambda) = 1 - p_1 - \sum_{k=2}^{\lambda} p_k \int \max\left(0, 1 - h_k(x)/f_k(x)\right) f_k(x) dx, \tag{4.27}$$

where $h_k(x) := \max_{1 \leq i \leq k-1} p_i f_i(x)/p_k$. That shows that the Bayes error rate is actually a linear combination of $(\lambda - 1)$ f-divergences.

Remark IV.13. The function t_k is not a properly defined generalized f-divergence [31], since $t_k\left(\frac{p_k}{p_1}, \frac{p_k}{p_2}, \ldots, \frac{p_k}{p_{k-1}}\right) = 0$, while $t_k(1, 1, \ldots, 1)$ is not necessarily equal to 0.

4.4 Numerical Results

We apply the proposed benchmark learner on several numerical experiments for binary and multi-class classification problems. We perform experiments on different simulated datasets with dimensions of up to $d = 100$. We compare the benchmark learner to previous lower and upper bounds on the Bayes error based on HP-divergence (1.11), as well as to a few powerful classifiers on different classification problem. The proposed benchmark learner is applied on the MNIST dataset with $70k$ samples and 784 features, learning theoretically the best achievable classification error rate. This is compared to reported performances of state of the art deep learning models applied on this dataset. Extensive experiments regarding the sensitivity with respect to the estimator parameter, the difference between the arithmetic and Chebyshev optimal weights and comparison of the corresponding ensemble benchmark learner performances, and comparison to the previous bounds on the Bayes error and classifiers on various simulated datasets with Gaussian, beta, Rayleigh and concentric distributions are provided in Appendix C.6.

Figure 4.2 compares the optimal benchmark learner with the Bayes error lower and upper bounds using HP-divergence, for a binary classification problems with 10-dimensional isotropic normal distributions with identity covariance matrix, where the means are shifted by 5 units in the first dimension. While the HP-divergence bounds have a large bias, the proposed benchmark learner converges to the true value by increasing sample size.

In Figure 4.3 we compare the optimal benchmark learner (Chebyshev method) with XGBoost, Random Forest and deep neural network (DNN) classifiers, for a 4-class classification problem 20-dimensional concentric distributions. Note that as

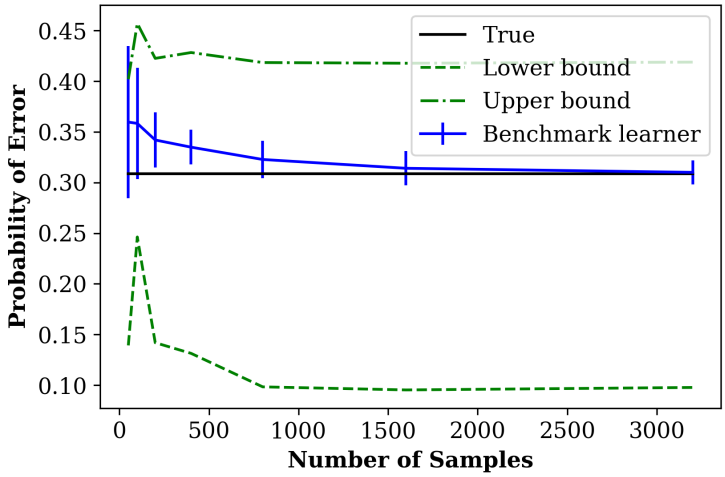
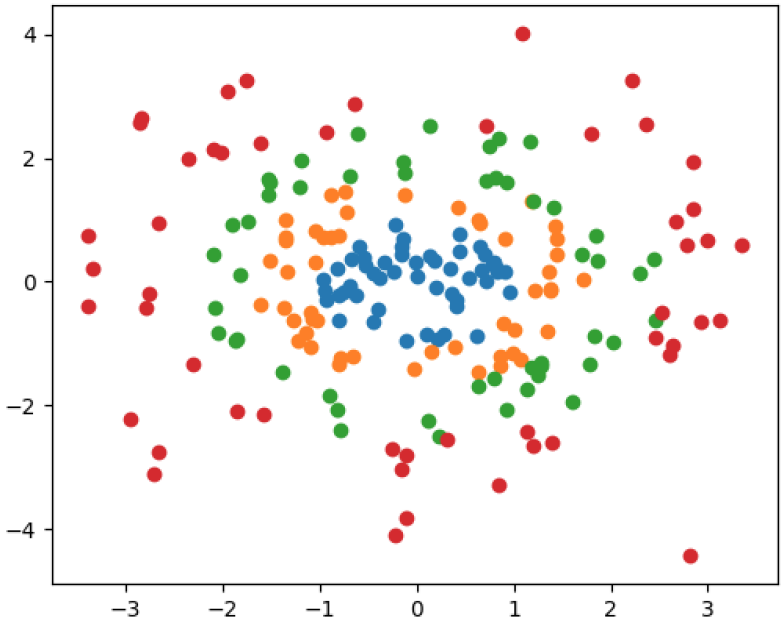


Figure 4.2: Comparison of the optimal benchmark learner (Chebyshev method) with the Bayes error lower and upper bounds using HP-divergence, for a binary classification problems with 10-dimensional isotropic normal distributions with identity covariance matrix, where the means are shifted by 5 units in the first dimension. While the HP-divergence bounds have a large bias, the proposed benchmark learner converges to the true value by increasing sample size.

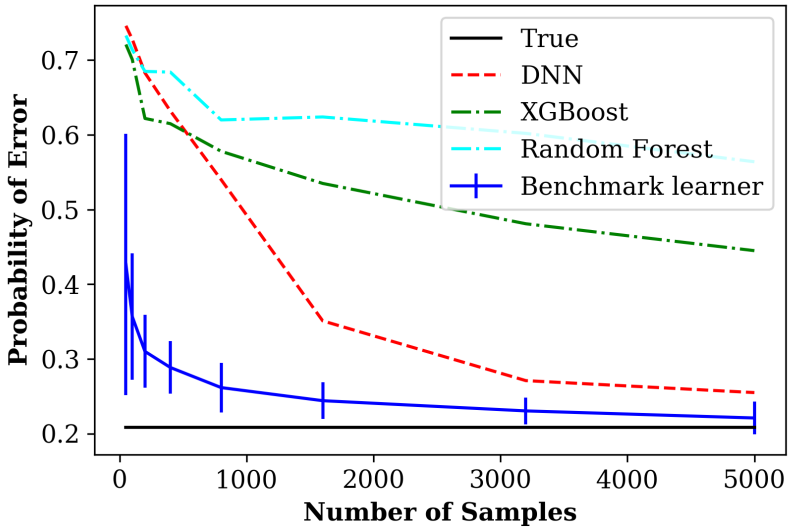
shown in (b) the concentric distributions are resulted by dividing a Gaussian distribution with identity covariance matrix into four quantiles such that each class has the same number of samples. The DNN classifier consists of 5 hidden layers with $[20, 64, 64, 10, 4]$ neurons and ReLU activations. Also in each layer a dropout with rate 0.1 is applied to diminish the overfitting. The network is trained using Adam optimizer and is trained for 150 epochs.

Further, we compute the benchmark learner for the MNIST dataset with 784 dimensions and 60,000 samples. In Table 4.1 we compare the estimated benchmark learner with the reported state of the art convolutional neural network classifiers with 60,000 training samples. Note that according to the online report [11] the listed models achieve the best reported classification performances.

The benchmark learner can also be used as a stopping rule for deep learning models. This is demonstrated in figures 4.4 and 4.5. In both of these figures we



(a) Four classes with concentric distributions



(b) Benchmark learner compared to a 5-layer DNN, XGBoost and Random Forest classifiers for the concentric distributions

Figure 4.3: Comparison of the optimal benchmark learner (Chebyshev method) with a 5-layer DNN, XGBoost and Random Forest classifiers, for a 4-class classification problem 20-dimensional concentric distributions. Note that as shown in (b), the concentric distributions are resulted by dividing a Gaussian distribution with identity covariance matrix into four quantiles such that each class has the same number of samples. The DNN classifier consists of 5 hidden layers with $[20, 64, 64, 10, 4]$ neurons and RELU activations. Also in each layer a dropout with rate 0.1 is applied to diminish the overfitting. The network is trained using Adam optimizer and is trained for 150 epochs. The benchmark learner predicts the Bayes error rate better than the DNN, XGBoost and Random Forest classifiers.

Papers	Method	Error rate
(Cirecsan 2010)[24]	Single 6-layer DNN	0.35%
(Ciresan 2011) [25]	Ensemble of 7 CNNs and training data expansion	0.27%
(Cirecsan 2012)[23]	Ensemble of 35 CNNs	0.23%
(Wan 2013) [119]	Ensemble of 5 CNNs and DropConnect regularization	0.21%
Benchmark learner	Ensemble ϵ-ball estimator	0.14%

Table 4.1: Comparison of error probabilities of several the state of the art deep models with the benchmark learner, for the MNIST handwriting image classification dataset

consider a 3-class classification problem with 30-dimensional Rayleigh distributions with parameters $a = 0.7, 1.0, 1.3$. We train a DNN model consisting of 5 layers with $[30, 100, 64, 10, 3]$ neurons and RELU activations. Also in each layer a dropout with rate 0.1 is applied to diminish the overfitting. In Figure. 4.4 we feed in different numbers of samples and compare the error rate of the classifier with the proposed benchmark learner. The network is trained using Adam optimizer for 150 epochs. At around 500 samples, the error rate of the trained DNN is within the confidence interval of the benchmark learner, and one can probably stop increasing the sample number since the error rate of the DNN is close enough to the Bayes error rate. In Figure. 4.5 we feed in 2000 samples to the network and plot the error rate for different training epochs. At around 80 epochs, the error rate of the trained DNN is within the confidence interval of the benchmark learner, and we can stop training the network since the error rate of the DNN is close enough to the Bayes error rate.

4.5 Conclusion

In this chapter, a new framework, benchmark learning, was proposed that learns the Bayes error rate for classification problems. An ensemble of base learners was developed for binary classification and it was shown to converge to the exact Bayes error probability with optimal (parametric) MSE rate. An ensemble estimation technique based on Chebyshev polynomials was proposed that provides closed form expressions

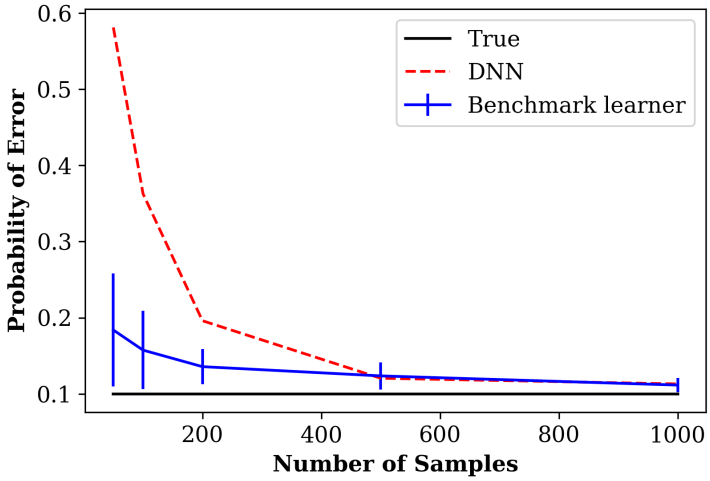


Figure 4.4: Error rate of a DNN classifier compared to the benchmark learner for a 3-class classification problem with 30-dimensional Rayleigh distributions with parameters $a = 0.7, 1.0, 1.3$. We feed in different numbers of samples and compare the error rate of the classifier with the proposed benchmark learner. The network is trained for about 50 epochs. At around 500 samples, the error rate of the trained DNN is within the confidence interval of the benchmark learner, and one can probably stop increasing the sample number since the error rate of the DNN is close enough to the Bayes error rate.

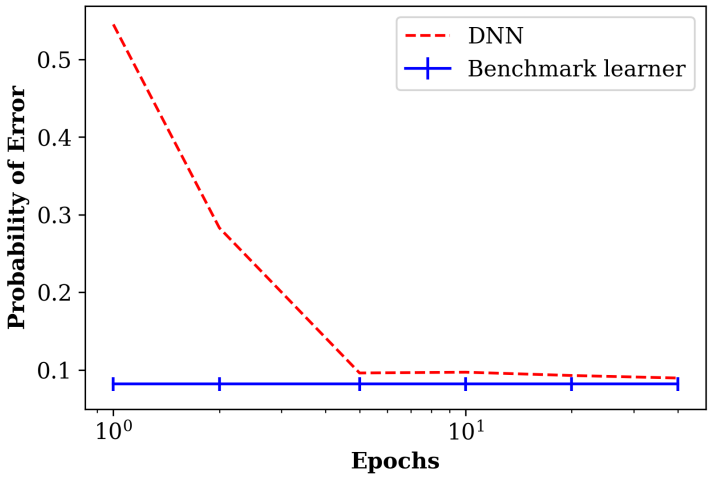


Figure 4.5: Error rate of a DNN classifier compared to the benchmark learner for a 3-class classification problem with 30-dimensional Rayleigh distributions with parameters $a = 0.7, 1.0, 1.3$. We feed in 2000 samples to the network and plot the error rate for different training epochs. At around 40 epochs, the error rate of the trained DNN is within the confidence interval of the benchmark learner, and we can stop training the network since the error rate of the DNN is close enough to the Bayes error rate.

for the optimum weights of the ensemble estimator. Finally, the framework was extended to multi-class classification and the proposed benchmark learner was shown to converge to the Bayes error probability with optimal MSE rates.

CHAPTER V

Bayes Error Based Feature Selection

In order to improve computational complexity and performance of a model in high-dimensional datasets, it is useful to choose a smaller set of features that provides the maxumum distinguishability between different classes. This preprocessing phase is called feature selection. In this chapter we propose a feature selection method based on Bayes error rate. The proposed method resembles the forward selection wrapper methods, yet it is independent of any learning model. The proposed feature selection method uses the Bayes error as a measure of quality of the features. Bayes error rate is defined as misclassification error of the optimum Bayes classifier. Similar to the feature quality measures used in the filter methods, Bayes error is independent of any learning algorithms. However, unlike most of the filter methods, Bayes error is directly related to the error of the classification.

The proposed Bayes error based feature selection (BEFS) method consists of sequential feature selection steps. The method starts with an empty set and at each step we select the feature that decreases the Bayes error feature set the most. Similar to the filter methods, BEFS is computationally efficient. BEFS only involves estimation of the Bayes error rate instead of the computationally expensive training process performed at each step in wrapper methods. We compare the BEFS method

to a few wrapper and filter feature selection methods. The quality of the features are evaluated based on the estimated Bayes error rate using the ϵ-ball estimator defined in equation (4.12). We choose state of the art deep learning and random forest classifiers as the wrapper models in the forward feature selection method.

The structure of this chapter is as follows. In Section 5.1 we introduce the proposed feature selection method. In Section 5.1 we apply BEFS on three real datasets regarding cancer prediction, robot navigation and speech activity detection and we compare the BEFS method to a few wrapper and filter feature selection methods.

5.1 Proposed Feature Selection Method

Assume that the input data, denoted by a random variable X, consists of d features as $X = [X^1, ..., X^d]$ and our task is to select a set of r features ($r < d$) that provides the most distinguishability between the classes. Our proposed forward selection method consists of r steps, where at each step we choose a feature from $\{X^1, ..., X^d\}$ that reduces the Bayes error the most. The Bayes error rates in this method are estimated using the ϵ-ball estimator defined in equation (4.12). The method starts with an empty set and at each step we select the feature that decreases the Bayes error feature set the most. The number of the steps could be specified, or we can continue these steps until the Bayes error rate of the selected features is close enough to the Bayes error of the Bayes error of the full dataset. The pseudocode for the proposed Bayes error based feature selection (BEFS) method is given in Algorithm 7.

Remark: Similar to most of the feature selection methods, BEFS is a greedy algorithm which finds a sub-optimal set of features, however, we show in Section 5.2 that in practice the selected set of features using this method provides a better

Algorithm 7: Bayes Error Based Feature Selection (BEFS)

Input : Input dataset, $\mathbf{X} = \{X_1, ..., X_N\}$
Labels, $\mathbf{Y} = \{Y_1, ..., Y_N\}$
Desired number of output features, r

$\mathcal{F} := \phi$
for *each* $i \in \{1, ..., r\}$ **do**
  $f \leftarrow argmax\{BER\{\mathcal{F}\} - BER\{\mathcal{F} \cup X^j\}$
  Add f into $\mathcal{F}$

Output: $\mathcal{F}$

performance.

Remark: The computational complexity of BEFS method (Algorithm 7) is $O(dmrhN \log N)$, where d, m, r, h and N respectively are dimension, number of classes, number of selected features, the bandwidth of the Bayes error estimator and the number of samples.

5.2 Experimental Results

We apply BEFS method on three real-datasets: breast cancer Wisconsin (diagnostic) dataset [123], wall-following robot navigation dataset [36] and TIMIT acoustic-phonetic continuous speech dataset [40]. For each case we compare the quality of the selected features using BEFS with wrapper and filter methods. The quality of the features are evaluated based on the estimated Bayes error rate. We choose state of the art deep learning and random forest as the wrapper models in the forward feature selection method. Note that choosing a deep neural network as the wrapper model would take an excessive time since we will have to train the network rd times, where d and r are respectively the dimension and number of selected feature. However, this model results in high quality features thanks to the powerful learning model.

5.2.1 Breast Cancer Prediction

We apply the proposed feature selected method on the breast cancer prediction dataset [123]. In this dataset there are 30 features, generated from image analysis of fine needle aspirates (FNA) of breast masses and characterise cell nucleus properties. The task to classify the patients into the classes *malignant* or *benign breast mass*. The set of measured features are mean, standard deviation and the worst value of the measures of: Clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, number of bare nuclei, bland chromatin, number of normal nuclei, and Mitosis. Each of the features is a real value in the range $[0, 1]$.

BEFS Results:

We apply BEFS on this dataset to get the first 5 most important features for the diagnosis. The first 3 feature selection steps of the BEFS method are represented respectively in Figure 5.1 (a)-(c). In the first step all features are sorted according to the Bayes error resulted by a single feature. Then the feature *radius_mean* which provides the minimum Bayes error rate, is selected. In the next steps, the algorithm sorts all of the remaining features according to the Bayes error resulted by adding the feature to the selected set features, and the feature with the minimum Bayes error is added to selected set.

In Figure 5.2 the 5 feature selection steps of the BEFS method are represented. The selected features are respectively *radius_mean*, $frac\_dim\_mean$, *num_concave_mean*, *area_max*, *radius_max* and the corresponding Bayes error rates achieved at each step are 0.087, 0.044,0.040, 0.035, 0.033 (shown by orange bars). The blue bars show the Bayes error rate achieved by all of the 30 features, which is 0.030.

We compare the BEFS results to the wrapper methods with Random Forest and

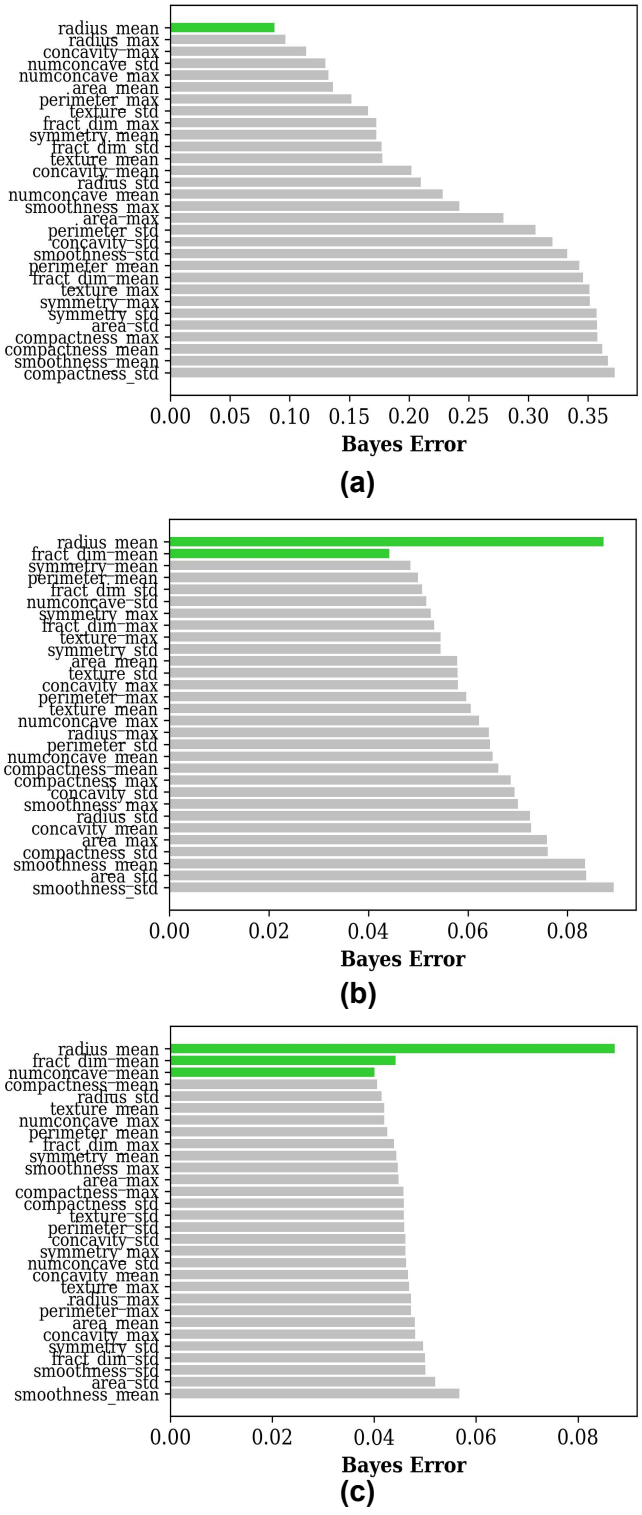


Figure 5.1: The first 3 feature selection steps of the BEFS method are represented in (a)-(c).

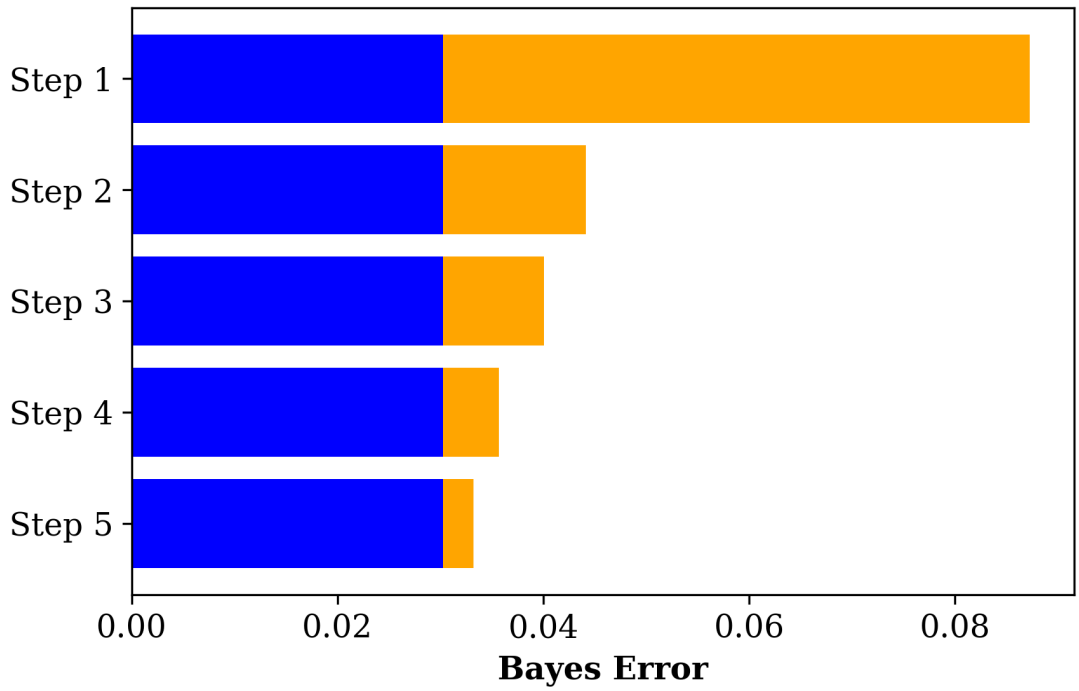


Figure 5.2: The 5 feature selection steps of the BEFS method are represented for the breast cancer dataset. The selected features are respectively *radius_mean*, $frac\_dim\_mean$, *num_concave_mean*, *area_max*, *radius_max* (with the corresponding feature indices $0, 27, 1, 2, 22$) and the Bayes error rates achieved at each step are 0.087, 0.044,0.040, $0.035, 0.033$ (shown by orange bars). The blue bars show the Bayes error rate achieved by all of the 30 features, which is 0.030.

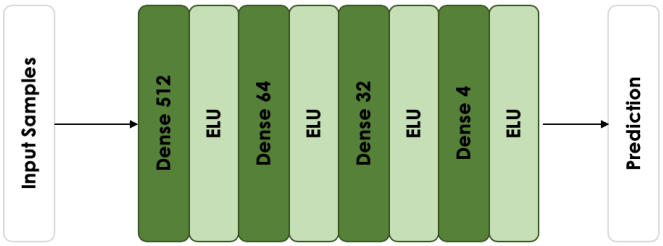


Figure 5.3: The representation of the deep model used as a wrapper forward selection feature selection for the breast cancer prediction.

DNN classifiers, as well as to a filter method with χ-square test score (specified as the Select-Best-k method). In the following we summarize the result on DNN model.

Deep Learning Model: We train a DNN model with 5 fully connected layers. The schematic diagram of the model is represented in figure 5.3. We use this DNN as a wrapper model in the forward selection method. We train the model using Adam optimizer with the learning rate of 0.001 and batch size of 256. For each of the probe subsets we train the model for 500 epochs. The forward selection wrapper method using this network selects the features with the indices $0, 1, 2, 22, 5$ and the classification error rates $0.10, 0.06, 0.07, 0.035, 0.043$ are achieved at each step.

The results of BEFS, DNN, Random Forest and Select-Best-k methos are compared in Figure 5.4. The blue bar shows the Bayes error rate achieved using all of the features, which is 0.030. The orange bars represent the Bayes error rates achieved by the BEFS, Random Forest, DNN and Select-Best-k methods. The selected features using BEFS are $0, 27, 1, 2, 22$, which result in the Bayes error 0.038. On the other hand, the selected features using the DNN method are $0, 1, 2, 22, 5$, which result in the classification error 0.043, which is the same as the Bayes error rate achieved by the selected features. The selected features using the Random Forest wrapper method are $0, 2, 9, 23, 20$, which result in the classification error 0.058 (shown by the red bar), while the Bayes error rate achieved by the selected features is 0.048. Finally, the se-

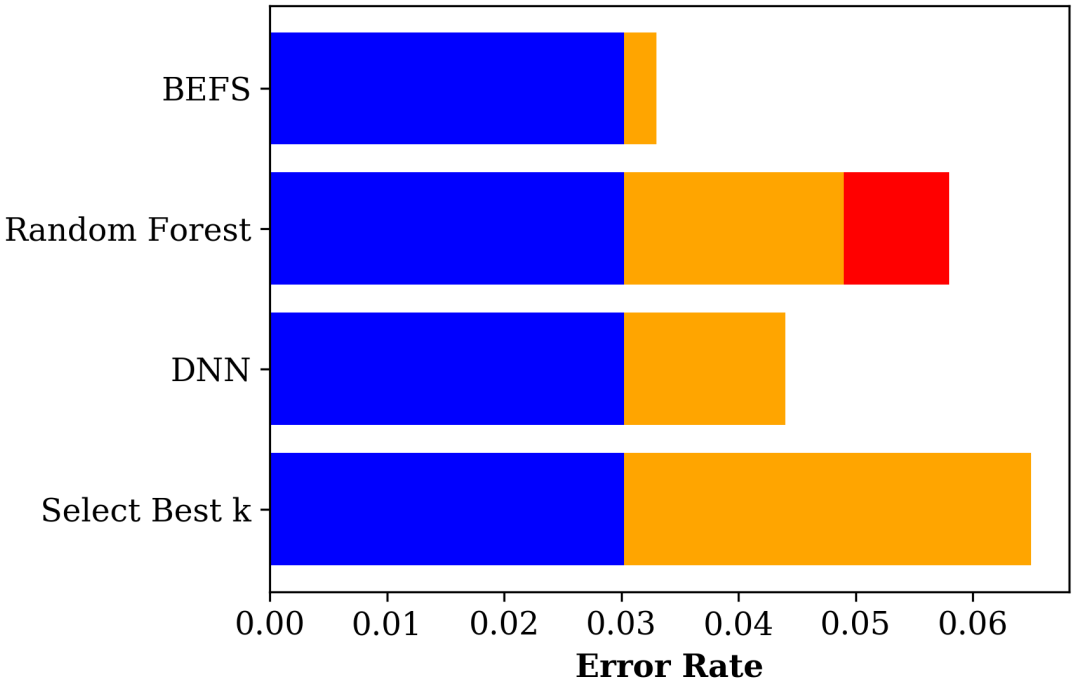


Figure 5.4: The results of BEFS, DNN, Random Forest and Select-Best-k methos are compared. The blue bar shows the Bayes error rate achieved using all of the features, which is 0.030. The orange bars represent the Bayes error rates achieved by the BEFS, Random Forest, DNN and Select-Best-K methods. The features selected using the BEFS method has the least Bayes error rate among others, which shows the effectiveness of the proposed method.

lected features using the Select-Best-k filter method are $1, 2, 3, 21, 22$. The Bayes error rate achieved by the selected features is 0.065. Note that the features $0, 1, 2$ and 22 are commonly selected by most of the investigated methods. The features selected using the BEFS method has the least Bayes error rate among others, which shows the effectiveness of the proposed method.

5.2.2 Wall-Following Robot Navigation Dataset

The second dataset is the wall-following robot navigation data [36]. The data are measurements of the 24 ultrasound sensors arranged circularly around a robot (Figure 5.5). The task for the robot is to navigate in a clockwise direction around

the room, by following the wall. The dataset consists of 5456 recorded timestamps. At each timestamp, the robot gets the measurements from all of the sensors and decides which action should it take. The possible actions include one of the four classes move-forward, sharp-right-turn, slight-right-turn and turn-left.

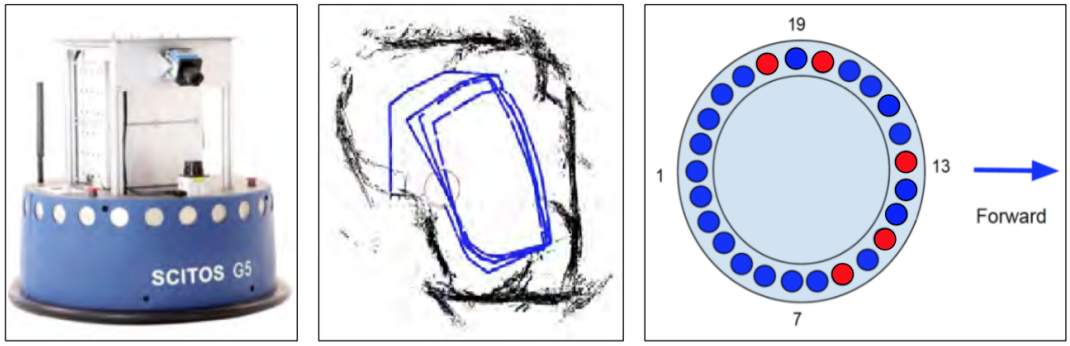


Figure 5.5: The graphs of the SCITOS G5 robot with 24 ultrasound sensors arranged circularly around a robot (left), the followed path by the robot (middle), and the positions and the indexing of the sensors 1 through 24 (right). The task for the robot is to navigate in a clockwise direction around the room, by following the wall. The dataset consists of 5456 recorded timestamps. At each timestamp, the robot gets the measurements from all of the sensors and decides which action should it take. The possible actions include one of the four directions: move-forward, sharp-right-turn, slight-right-turn and turn-left.

BEFS Results: In Figure 5.6 the 5 feature selection steps of the BEFS method are represented. The selected features (sensors) are respectively indexed as $14, 11, 21, 9, 19$ and the corresponding Bayes error rates achieved at each step are 0.22, 0.10,0.049, $0.038, 0.034$ (shown by orange bars). The blue bars show the Bayes error rate achieved by all of the 30 features, which is 0.032.

We compare the BEFS results to the wrapper methods with Random Forest and DNN models, as well as to the Select-Best-k method. In the following we summarize the result on DNN model.

Deep Learning Model: We train a DNN model with 5 fully connected layers with $512, 64, 32, 4$ neurons and ELU activations. We use this DNN as a wrapper model in the forward selection method. We train the model using Adam optimizer

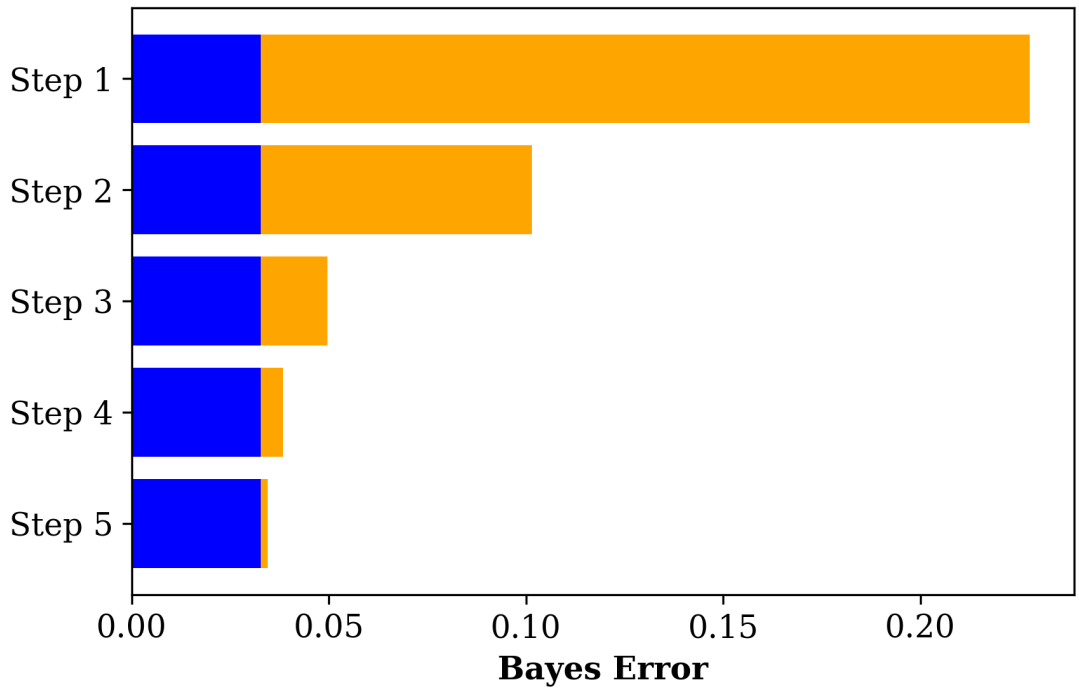


Figure 5.6: The selected features (sensors) are respectively indexed as $14, 11, 21, 9, 19$ and the corresponding Bayes error rates achieved at each step are 0.22, 0.10,0.049, $0.038, 0.034$ (shown by orange bars). The blue bars show the Bayes error rate achieved by all of the 30 features, which is 0.032.

with the learning rate of 0.001 and batch size of 256. For each of the probe subsets we train the model for 500 epochs. The forward selection wrapper method using this network selects the features with the indices $14, 19, 13, 18, 17$ and the classification error rates $0.21, 0.085, 0.071, 0.070, 0.063$ are achieved at each step.

The results of BEFS, DNN, Random Forest and Select-Best-k methos are compared in Figure 5.7. The blue bar shows the Bayes error rate achieved using all of the features, which is 0.032. The orange bars represent the Bayes error rates achieved by the BEFS, Random Forest, DNN and Select-Best-k methods. The selected features using BEFS are $14, 11, 21, 9, 19$ which result in the Bayes error 0.034. On the other hand, the selected features using the DNN method are $14, 19, 13, 18, 17$. Using these features results in the classification error 0.063, which is the same as the Bayes error rate achieved by the selected features. The selected features using the Random Forest wrapper method are $14, 18, 10, 13, 23$, which result in the classification error 0.037 (shown by the red bar), while the Bayes error rate achieved by the selected features is 0.036. Finally, the selected features using the Select-Best-k filter method are $14, 16, 17, 18, 19$. The Bayes error rate achieved by the selected features is 0.093. Note that the features $14, 13$ and 18 are commonly selected by most of the investigated methods. The features selected using the BEFS method has the least Bayes error rate among others, which shows the effectiveness of the proposed method.

5.2.3 Speech Activity Detection

We consider the problem of classifying the speech/non-speech audio. We use the TIMIT acoustic-phonetic continuous speech dataset [40] and random noise (non-speech) samples from [54]. The speech dataset contains samples from 16 speakers from 8 dialect regions (1 male and 1 female from each dialect region). There are totally 160 sentence recordings (10 recordings per speaker). The audio files are

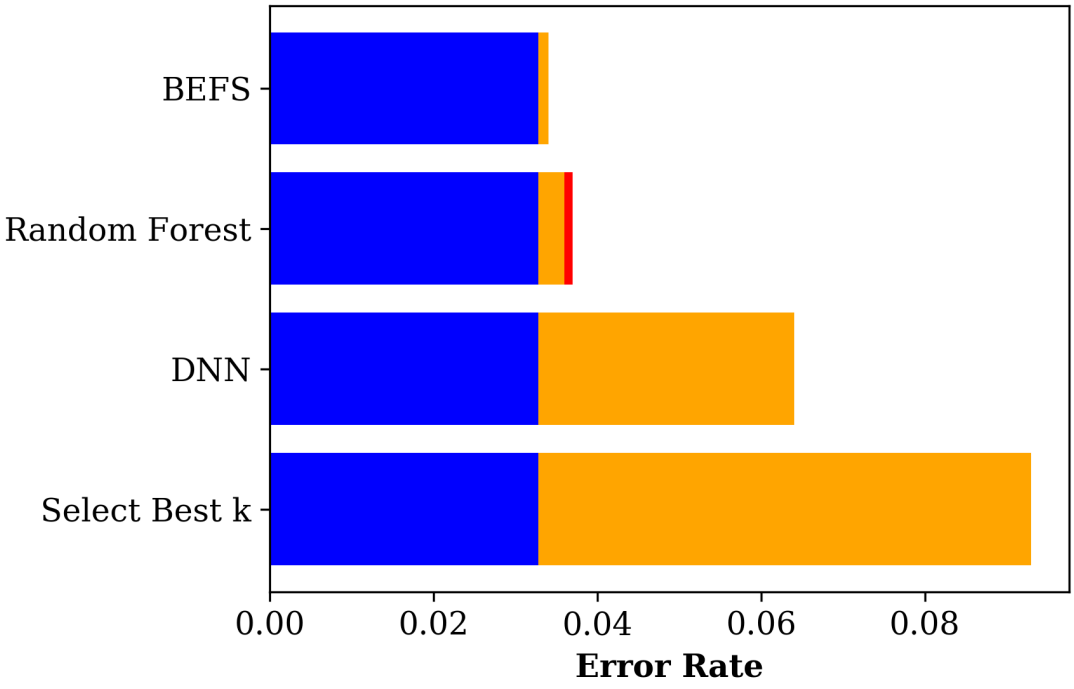


Figure 5.7: The results of BEFS, DNN, Random Forest and Select-Best-k methos are compared. The blue bar shows the Bayes error rate achieved using all of the features, which is 0.030. The orange bars represent the Bayes error rates achieved by the BEFS, Random Forest, DNN and Select-Best-K methods. The features selected using the BEFS method has the least Bayes error rate among others, which shows the effectiveness of the proposed method.

single channel, with 16kHz sampling, 16 bit sample, PCM encoding.

Preprocessing: We consider frames with duration of 10ms and we apply the MFCC feature extraction on each frame which results in 24 features per frame. Each sample contains the features of 21 frames (including 10 frames before and after the central frame). We consider 5000 samples from each of the speech and non-speech classes, where each sample is a 21×24 matrix. Thus, the total number of features is 504. 8000 samples are used for training and 2000 samples are used for testing.

Feature Selection Experiments:

The type of features in this dataset is slightly different from the cancer prediction and robot navigation datasets. In this dataset, each feature contains a single MFCC component in all 21 time frames. We are interested in selecting the MFCC features effective in classification of speech and non-speech audio. In this high-dimensional dataset we compare the BEFS method to the wrapper method with a DNN model. The structure of the DNN model is summarized in the following.

BEFS Results: We apply BEFS with 4 steps on the dataset. The error rates at each step are represented in Figure 5.8. The selected features are respectively $9, 8, 11, 0$ and the corresponding Bayes error rates achieved at each step are 0.10, 0.080,0.065, 0.050. Note that the Bayes error rate achieved by using all of the features is 0.045. Hence, the selected features $9, 8, 11, 0$ achieve the Bayes error rate close the the Bayes error rate achieved by all features.

Deep Learning Model: We train a DNN model with 3 1-D convolutional layers and 4 subsequent fully connected layers. The schematic of the model is represented in figure 5.9. We use this DNN as a wrapper model in the forward selection method. We train the model using Adam optimizer with the learning rate of 0.001 and batch size of 256. After 500 epochs the trained model achieves the test accuracy of %95 using all of

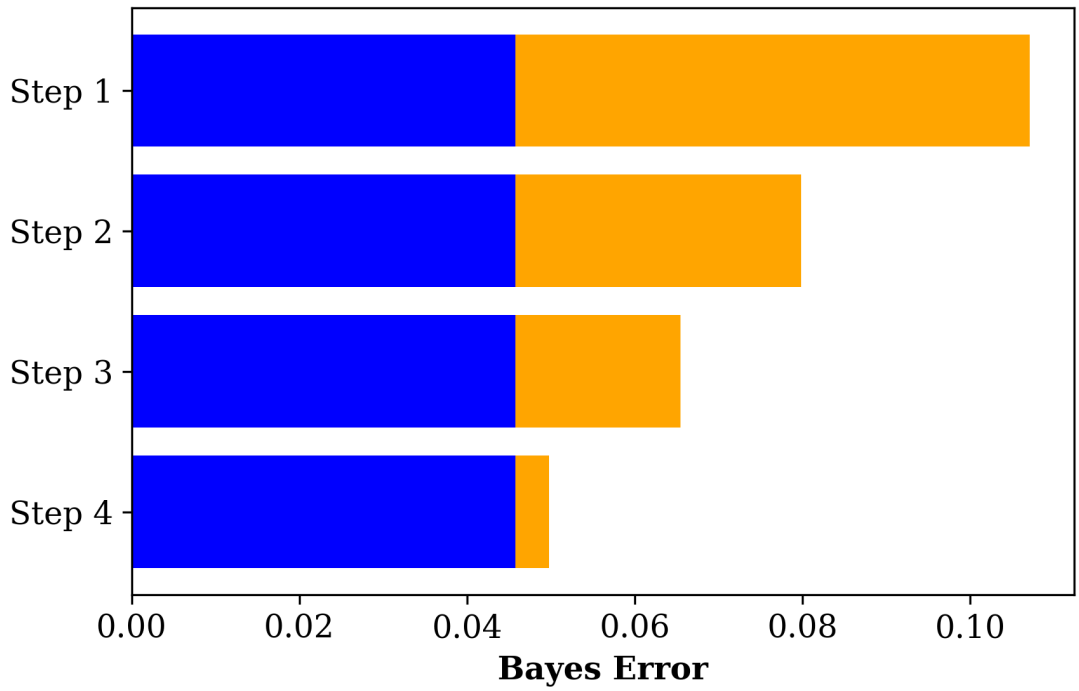


Figure 5.8: The 4 feature selection steps of the BEFS method are represented. The selected features are respectively $9, 8, 11, 0$ and the corresponding Bayes error rates achieved at each step are 0.10, 0.080,0.065, 0.050 (shown by orange bars). The blue bars show the Bayes error rate achieved by all of the 24 features, which is 0.045.

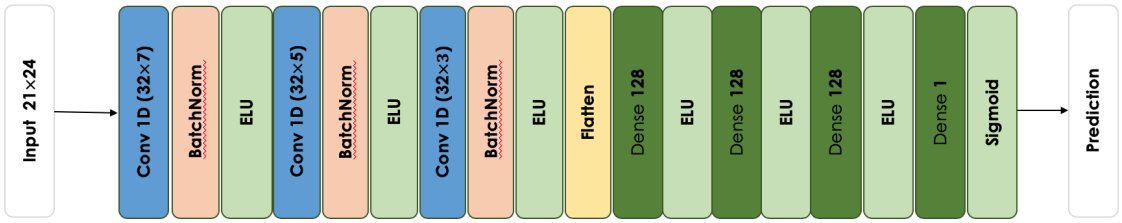


Figure 5.9: The schematic of the deep model used as a wrapper forward selection feature selection.

the 24 features. The forward selection wrapper method using this network selects the features $0, 11, 8, 2$ with corresponding classification error rates $0.16, 0.14, 0.126, 0.124$.

The results of BEFS and DNN methods are compared in Figure 5.10. The blue bar shows the Bayes error rate achieved by all of the features, which is 0.045. The orange bars represent the Bayes error rates achieved by the BEFS and DNN methods. The selected features using BEFS are $9, 8, 11, 0$, which result in the Bayes error 0.050. On the other hand, the selected features using the DNN method are $0, 11, 8, 2$, which result in the classification error 0.12 (shown by the red bar), while the Bayes error rate achieved by the selected features is 0.065. Note that the features $0, 8, 11$ are commonly selected by both of the methods.

5.3 Conclusion

In this chapter we proposed a new feature selection method based on Bayes error rate. The proposed method is independent of any learning model. BEFS method consists of sequential feature selection steps. Similar to the filter methods, BEFS is computationally efficient. BEFS only involves estimation of the Bayes error rate instead of the computationally expensive training process performed at each step in wrapper methods. We applied BEFS on three real datasets regarding cancer prediction, robot navigation and speech activity detection and we compared the

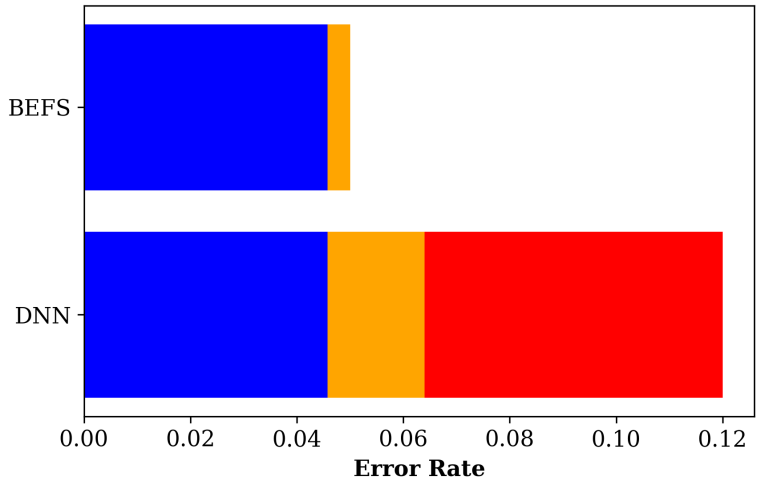


Figure 5.10: The results of BEFS and DNN methdos are compared. The blue bar shows the Bayes error rate achieved by all of the features, which is 0.045. The orange bars represent the Bayes error rates achieved by the BEFS and DNN methods. The selected features using BEFS are $9, 8, 11, 0$, which result in the Bayes error 0.050. On the other hand, the selected features using the DNN method are $0, 11, 8, 2$, which result in the classification error 0.12 (shown by the red bar), while the Bayes error rate achieved by the selected features is 0.065. Note that the features $0, 8, 11$ are commonly selected by both of the methods.

BEFS method to several wrapper and filter feature selection methods.

CHAPTER VI

Hash-based Estimation of Divergence Measure

In this chapter we propose a low complexity divergence estimator that can achieve the optimal MSE rate of $O(1/N)$ for the densities with bounded derivatives of up to d. Our estimator has optimal runtime complexity of $O(N)$, which makes it an appropriate tool for large scale applications. Also in contrast to other competing estimators, our estimator does not require stringent smoothness assumptions on the support set boundary. The structure of the proposed estimator borrows ideas from hash based methods for KNN search and graph constructions problems [128, 75], as well as from the NNR estimator proposed in II.

Hash based methods have previously been used for KNN search and graph constructions problems [128, 75], and they result in fast and low complexity algorithms. The advantage of hash based methods is that they can be used to find the approximate nearest neighbor points with lower complexity as compared to the exact k-NN search methods. This suggests that fast and accurate algorithms for divergence estimation may be derived from hashing approximations of k-NN search. In [94] we considered the k-NN graph of Y in the joint data set (X, Y), and show that the average exponentiated ratio of the number of X points to the number of Y points among all k-NN points is proportional to the Rényi divergence between the X and

Y densities. It turns out that for estimation of the density ratio around each point we really do not need to find the exact k-NN points, but only need sufficient local samples from X and Y around each point. By using a randomized locality sensitive hashing (LSH), we find the closest points in Euclidean space. In this manner, applying ideas from the NNR estimation and hashing techniques to KNN search problem, we obtain a more efficient divergence estimator. Consider two sample sets X and Y with a bounded density support. We use a particular two-level locality sensitive random hashing, and consider the ratio of samples in each bin with a number of Y samples. We prove that the weighted average of these ratios over all of the bins can be made to converge almost surely to f-divergences between the two samples populations. We also argue that using the ensemble estimation technique provided in [79], we can achieve the optimal parametric rate of $O(1/N)$. Furthermore, using a simple algorithm for online estimation method has $O(N)$ complexity and $O(1/N)$ convergence rate, which is the first optimal online estimator of its type.

The rest of this chapter is organized as follows. In Section 6.1, we recall the definition of f-divergence and introduce the Hash-Based (HB) estimator. In sections 6.2 and 6.3, we provide the convergence theorems and propose the Ensemble Hash-Based (EHB) estimator. In Section 6.4, we propose the online version of the proposed HB and EHB estimator. In Section 6.5 we give proofs for the convergence results. Finally, in Section 6.6 we validate our theoretical results using numerical and real data experiments.

We recall the definition of f-divergence measure in the following: Consider two density functions f_1 and f_2 with common bounded support set $\mathcal{X} \subseteq \mathbb{R}^d$. From 1.2,

f-divergence is defined as follows:

$$D_g\left(f_1(x)||f_2(x)\right) := \int g\left(\frac{f_1(x)}{f_2(x)}\right) f_2(x)dx$$
$$= \mathbb{E}_{f_2}\left[g\left(\frac{f_1(x)}{f_2(x)}\right)\right], \tag{6.1}$$

where g is a smooth and convex function such that $g(1) = 0$. KL-divergence, Hellinger distance and total variation distance are particular cases of this family.

6.1 Hash-Based Divergence Estimator

Consider the i.i.d samples $X = \{X_1, ..., X_N\}$ drawn from f_1 and $Y = \{Y_1, ..., Y_M\}$ drawn from f_2. Define the fraction $\eta := M/N$. We define the set $Z := X \cup Y$. We define a positive real valued constant ϵ as a user-selectable parameter of the estimator to be defined in 6.4. We define the hash function $H_1 : \mathbb{R}^d \to \mathbb{Z}^d$ as

$$H_1(x) = \left[h_1(x_1), h_1(x_2), ..., h_1(x_d)\right], \tag{6.2}$$

where x_i is the projection of x on the ith coordinate, and $h_1(x) : \mathbb{R} \to \mathbb{Z}$ is defined as

$$h_1(x) = \left\lfloor \frac{x + b}{\epsilon} \right\rfloor, \tag{6.3}$$

for fixed b. Let $\mathcal{F} := \{1, 2, .., F\}$, where $F := c_H N$ and c_H is a fixed real number. We define a random hash function $H_2 : \mathbb{Z}^d \to \mathcal{F}$ with a uniform density on the output and consider the combined hashing $H(x) := H_2(H_1(x))$, which maps the points in $\mathbb{R}^d$ to $\mathcal{F}$.

Consider the mappings of the sets X and Y using the hash function $H(x)$, and define the vectors $\mathcal{N}$ and $\mathcal{M}$ to respectively contain the number of collisions for each output bucket from the set $\mathcal{F}$. We represent the bins of the vectors $\mathcal{N}$ and $\mathcal{M}$ respectively by N_i and M_i, $1 \leq i \leq F$.

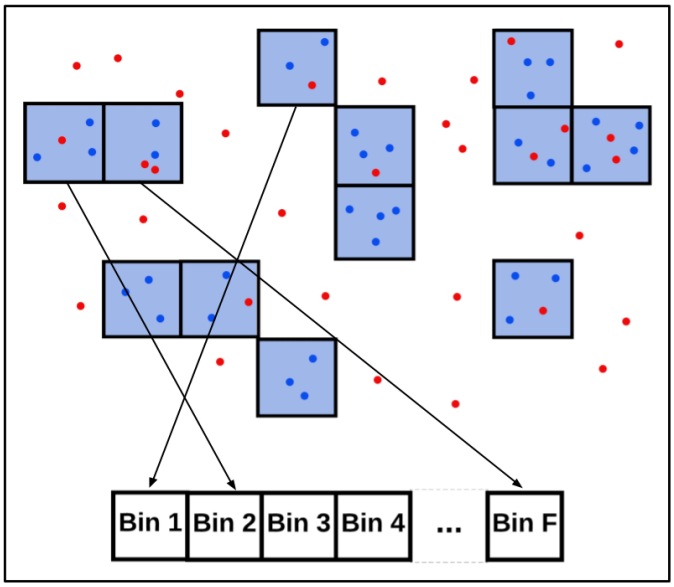


Figure 6.1: Hashing the data points to {1,...,F}.

Figure 6.1 represents an example of the hash function H applied on two different sets of data points.

The hash based f-divergence estimator is defined as

$$\widehat{D}_g(X,Y) := \max\left\{\frac{1}{M}\sum_{\substack{i\leq F\\ M_i>0}} M_i \widetilde{g}\left(\frac{\eta N_i}{M_i}\right), 0\right\}, \tag{6.4}$$

where $\widetilde{g}(x) := \max\{g(x), g\left(C_L/C_U\right)\}$.

Note that if the densities f_1 and f_2 are almost equal, then for each point Y_i, $N_i \approx M_i$, and thus $\widehat{D}_g(X,Y)$ tends to zero, as required. Algorithm 8 shows the HB estimation procedure. We first find the sets of all hashed points in X and Y (lines 1 and 2). Then the number of collisions is counted (lines 3-5), and the divergence estimate is computed (line 6).

Similar to most of LSH structures, computing the hashing output in our estimator is of $O(1)$ complexity, and does not depend on ϵ. Thus, the computational complexity this estimator is $O(M)$.

Remark VI.1. The hash function considered in this chapter is a simple histogram

Algorithm 8: HB Estimator of f-Divergence

Input : Data sets $X = \{X_1, ..., X_N\}$, $Y = \{Y_1, ..., Y_M\}$

`/* Find the sets of all hashed points in X and Y */`

$X' \leftarrow H(X)$.

$Y' \leftarrow H(Y)$.

for *each* $i \in \mathcal{F}$ **do**

  `/* Find the number of collisions at bin i */`

  $N_i \leftarrow |X' = i|$

  $M_i \leftarrow |Y' = i|$

$\widehat{D} \leftarrow \max\left\{\frac{1}{M}\sum_{M_i>0} M_i \widetilde{g}\left(\eta N_i/M_i\right), 0\right\}$,

Output: $\widehat{D}$

binning. In general any other hashing scheme that preserves the locality property might be used for the proposed estimator. For some datasets including images or texts, using the simple histogram binning may not be efficient in practice since the Euclidean metric in the $\mathbb{R}^d$ space may not capture the semantic similarity. For these cases, a hash function which first maps the samples into an appropriate feature space may work more efficiently compared to the simple histogram binning. Further analysis regarding this is left for future work.

6.2 Convergence Theorems

In the following theorems we state upper bounds on the bias and variance rates.

Theorem VI.2. *Assume that f_1 and f_2 are density functions with bounded common support set $\mathcal{X} \in \mathbb{R}^d$ and satisfying γ-Hölder smoothness. The bias of the proposed estimator for f-divergence with function g can be bounded as*

$$\mathbb{B}\left[\widehat{D}_g(X,Y)\right] = O\left(\epsilon^{\gamma}\right) + O\left(\frac{1}{N\epsilon^d}\right).$$

Remark VI.3. In order for the estimator to be asymptotically unbiased, ϵ needs to be a function of N. The optimum bias rate of $O\left(\left(\frac{1}{N}\right)^{\gamma/(\gamma+d)}\right)$ can be achieved for $\epsilon = O\left(\left(\frac{1}{N}\right)^{\gamma/(\gamma+d)}\right)$.

In the following we propose an upper bound on the variance that is independent of ϵ.

Theorem VI.4. *Let $\eta = M/N$ be fixed. The variance of the estimator* (6.4) *can be bounded as*

$$\mathbb{V}\left[\widehat{D}_g(X,Y)\right] \leq O\left(\frac{1}{N}\right). \tag{6.5}$$

Remark VI.5. The same variance bound holds for the random variable $\rho_i := \frac{N_i}{M_i}$. The bias and variance results easily extend to Rényi divergence estimation.

6.3 Ensemble Hash-Based Estimator

We next show that, when f_1 and f_2 belong to the family of differentiable densities, we can improve the bias rate by applying the ensemble estimation approach in [82, 81]. The EHB estimator is defined as follows.

Definition VI.6 (Ensemble Hash-Based Estimator)**.** Assume that the density functions have continuous derivatives up to order $q \geq d$. Let $\mathcal{T} := \{t_1, ..., t_T\}$ be a set of index values with $t_i < c$, where $c > 0$ is a constant. Let $\epsilon(t) := tN^{-1/2d}$. The weighted ensemble estimator is defined as

$$\widehat{D}_w := \sum_{t \in \mathcal{T}} w(t) \widehat{D}_{\epsilon(t)}, \tag{6.6}$$

where $\widehat{D}_{\epsilon(t)}$ is the hash based estimator of f-divergence, with the hashing parameter of $\epsilon(t)$. The following theorem states a sufficient condition for the weight vector w that ensures that the ensemble estimator (D.6) achieves an MSE rate of $O(1/N)$.

Theorem VI.7. *Let $T > d$ and w_0 be the solution to:*

$$\begin{aligned} \min_{w} \quad & \|w\|_2 \\ \textit{subject to} \quad & \sum_{t\in\mathcal{T}} w(t) = 1, \\ & \sum_{t\in\mathcal{T}} w(t)t^i = 0, i \in \mathbb{N}, i \leq d. \end{aligned} \tag{6.7}$$

Then the MSE rate of the ensemble estimator $\widehat{D}_{w_0}$ is $O(1/N)$.

6.4 Online Divergence Estimation

In this section we study the problem of online divergence estimation. In this setting we consider two data steams $X = \{X_1, X_2, ..., X_N\}$ and $Y = \{Y_1, Y_2, ..., Y_N\}$ with i.i.d samples, and we are interested in estimating the divergence between two data sets. The number of samples increase over time and an efficient update of the divergence estimate is desired. The time complexity of a batch update, which uses the entire update batch to compute the estimate at each time point, is $O(N)$, and it may not be so effective in cases which we need quick detection of any change in the divergence function.

Algorithm 9 updates the divergence with amortized runtime complexity of order $O(1)$. Define the sets $X^N := \{X_i\}_{i=1}^N$, $Y^N := \{Y_i\}_{i=1}^N$, the number of X and Y samples in each partition, and the divergence estimate between X^N and Y^N. Consider updating the estimator with new samples X_{N+1} and Y_{N+1}. In the first and second lines of algorithm 9, the new samples are added to the datasets and the values of N_i and M_i of the bins in which the new samples fall. We can find these bins in $O(1)$ using a simple hashing. Note that once N_i and M_i are updated, the divergence measure can be updated, but the number of bins is not increased, by Theorem VII.2, it is clear that the bias will not be reduced. Since increasing the number of bins re-

quires recomputing the bin partitions, a brute force rebinning approach would have order $O(N)$ complexity, and it were updated N times, the total complexity would be $O(N^2)$. Here we use a trick and update the hash function only when $N+1$ is a power of 2. In the following theorem, which is proved in appendix D, we show that the MSE rate of this algorithm is order $O(1/N)$ and the total rebinngn computational complexity is order $O(N)$.

Theorem VI.8. *MSE rate of the online divergence estimator shown in Algorithm 9 is order $O(1/N)$ and the total computational complexity is order $O(N)$.*

Algorithm 9: Online Divergence Estimation

Input : $X^N := \{X_i\}_{i=1}^N, Y^N := \{Y_i\}_{i=1}^N$
$\widehat{D} = \widehat{D}\left(X^N, Y^N\right)$
(N_i, M_i)
(X_{N+1}, Y_{N+1})

Add X_{N+1} and Update N_k s.t $H(X_{N+1}) = k$.
Add Y_{N+1} and Update M_l s.t $H(Y_{N+1}) = l$.
If $N + 1 = 2^i$ for some i, Then
    Update ϵ to the optimum value
    Re-hash X and Y
    Recompute N_i and M_i for $0 \leq i \leq F$
Update $\widehat{D}$

Output: $\widehat{D}$

6.5 Convergence Proofs

In this section we derive the bias bound for the densities in Hölder smoothness class, stated in Theorem VII.2. For the proofs of variance bound in Theorem VII.3, convergence rate of EHB estimator in Theorem VII.5, and online divergence estimator in Theorem VI.8, we refer to Appendix D.

Consider the mapping of the X and Y points by the hash function H_1, and let the vectors $\{V_i\}_{i=1}^L$ represent the distinct mappings of X and Y points under H_1. Here L is the number of distinct outputs of H_1. In the following lemma we prove an

upper bound on L.

Lemma VI.9. *Let $f(x)$ be a density function with bounded support $\mathbb{X} \subseteq \mathbb{R}^d$. Then if L denotes the number of distinct outputs of the hash function H_1 (defined in (7.2)) of i.i.d points with density $f(x)$, we have*

$$L \leq O\left(\frac{1}{\epsilon^d}\right). \tag{6.8}$$

Proof. Let $x = [x_1, x_2, ..., x_d]$ and define $\mathcal{X}_I$ as the region defined as

$$\mathbb{X}_I := \{x | -c_X \leq x_i \leq c_X, 1 \leq i \leq d\}, \tag{6.9}$$

where c_X is a constant such that $\mathbb{X} \subseteq \mathbb{X}_I$.

L is clearly not greater than the total number of bins created by splitting the region $\mathbb{X}$ into partitions of volume ϵ^d. So we have

$$L \leq \frac{(2c_X)^d}{\epsilon^d}. \tag{6.10}$$

□

Proof of Theorem VII.2 Let $\{N_i'\}_{i=1}^L$ and $\{M_j'\}_{j=1}^L$ respectively denote the number of collisions of X and Y points in the bins i and j, using the hash function H_1. E_i stands for the event that there is no collision in bin i for the hash function H_2 with inputs $\{V_i\}_{i=1}^L$. We have

$$\begin{aligned} P(E_i) &= \left(1 - \frac{1}{F}\right)^L + L\left(\frac{1}{F}\right)\left(\frac{F-1}{F}\right)^{L-1} \\ &= 1 - O\left(\frac{L}{F}\right). \end{aligned} \tag{6.11}$$

By definition,

$$\widehat{D}_g(X, Y) := \frac{1}{M} \sum_{\substack{i \leq F \\ M_i > 0}} M_i \widetilde{g}\left(\frac{\eta N_i}{M_i}\right).$$

Therefore,

$$\begin{aligned}\mathbb{E}\left[\widehat{D}_g(X,Y)\right] &= \frac{1}{M}\mathbb{E}\left[\sum_{\substack{i\leq F\\ M_i>0}} M_i\widetilde{g}\left(\frac{\eta N_i}{M_i}\right)\right] \\ &= \frac{1}{M}\sum_{\substack{i\leq F\\ M_i>0}} P(E_i)\mathbb{E}\left[M_i\widetilde{g}\left(\frac{\eta N_i}{M_i}\right)\middle| E_i\right] \\ &\quad + \frac{1}{M}\sum_{\substack{i\leq F\\ M_i>0}} P(\overline{E_i})\mathbb{E}\left[M_i\widetilde{g}\left(\frac{\eta N_i}{M_i}\right)\middle|\overline{E_i}\right]. \end{aligned} \tag{6.12}$$

We represent the second term in (6.12) by $\mathbb{B}_H$. $\mathbb{B}_H$ has the interpretation as the bias error due to collisions in hashing. Remember that $\overline{E_i}$ is defined as the event that there is a collision at bin i for the hash function H_2 with inputs $\{V_i\}_{i=1}^L$. For proving as upper bound on $\mathbb{B}_H$, we first need to compute an upper bound on $\sum_{i=1}^L \mathbb{E}\left[M_i\middle|\overline{E_i}\right]$. This is stated in the following lemma.

Lemma VI.10. *We have*

$$\sum_{\substack{i\leq F\\ M_i>0}} \mathbb{E}\left[M_i\middle|\overline{E_i}\right] \leq O\left(L\right) \tag{6.13}$$

Proof. Define $\mathcal{A}_i := \{j : H_2(V_j) = i\}$. For each i we can rewrite M_i as

$$M_i = \sum_{j=1}^{L} 1_{\mathcal{A}_i}(j) M'_j. \tag{6.14}$$

Thus,

$$\begin{aligned}\sum_{\substack{i\leq F\\ M_i>0}}\mathbb{E}\left[M_i\middle|\overline{E_i}\right] &= \sum_{\substack{i\leq F\\ M_i>0}}\mathbb{E}\left[\sum_{j=1}^{L}1_{\mathcal{A}_i}(j)M'_j\middle|\overline{E_i}\right]\\ &= \sum_{\substack{i\leq F\\ M_i>0}}\sum_{j=1}^{L}M'_j\mathbb{E}\left[1_{\mathcal{A}_i}(j)\middle|\overline{E_i}\right]\\ &= \sum_{\substack{i\leq F\\ M_i>0}}\sum_{j=1}^{L}M'_jP\left(j\in\mathcal{A}_i|\overline{E_i}\right)\\ &= \sum_{\substack{i\leq F\\ M_i>0}}\sum_{j=1}^{L}M'_j\frac{P\left(j\in\mathcal{A}_i,\overline{E_i}\right)}{P(\overline{E_i})}, \end{aligned} \tag{6.15}$$

where $P\left(j\in\mathcal{A}_i,\overline{E_i}\right)$ and $P(\overline{E_i})$ can be derived as

$$P\left(j\in\mathcal{A}_i,\overline{E_i}\right)=\frac{1}{F}\left(1-\left(\frac{F-1}{F}\right)^{L-1}\right)=O\left(\frac{L}{F^2}\right), \tag{6.16}$$

and

$$P(\overline{E_i})=1-P(E_i)=O\left(\frac{L}{F}\right). \tag{6.17}$$

Plugging in (6.16) and (6.17) in (6.15) results in

$$\begin{aligned}\sum_{\substack{i\leq F\\ M_i>0}}\mathbb{E}\left[M_i\middle|\overline{E_i}\right] &= \sum_{\substack{i\leq F\\ M_i>0}}\sum_{j=1}^{L}M'_jO\left(\frac{1}{F}\right)\\ &= \sum_{\substack{i\leq F\\ M_i>0}}O\left(\frac{M}{F}\right)=O\left(L\right), \end{aligned} \tag{6.18}$$

where in the third line we use $\eta = M/N$ and $F = c_H N$. In addition, the number of the terms in the sum is upper bounded by L since L is defined as the number of distinct outputs of hashing the X and Y points. Now in the following lemma we prove a bound on $\mathbb{B}_H$.

□

Lemma VI.11. *Let L denote the number of distinct outputs of the hash function H_1 of the X and Y sample points. The bias of estimator* (6.4) *due to hashing collision can be upper bounded by*

$$\mathbb{B}_H \leq O\left(\frac{L^2}{N^2}\right) \tag{6.19}$$

Proof. From the definition of $\mathbb{B}_H$ we can write

$$\begin{aligned}
\mathbb{B}_H :&= \frac{1}{M}\sum_{\substack{i\leq F\\ M_i>0}} P(\overline{E_i})\mathbb{E}\left[M_i\widetilde{g}\left(\frac{\eta N_i}{M_i}\right)\middle|\overline{E_i}\right] \\
&= \frac{P(\overline{E_1})}{M}\sum_{\substack{i\leq F\\ M_i>0}} \mathbb{E}\left[M_i\widetilde{g}\left(\frac{\eta N_i}{M_i}\right)\middle|\overline{E_i}\right] \\
&\leq \frac{P(\overline{E_1})\widetilde{g}(R_{max})}{M}\sum_{\substack{i\leq F\\ M_i>0}} \mathbb{E}\left[M_i\middle|\overline{E_i}\right] \\
&= \frac{P(\overline{E_1})\widetilde{g}(R_{max})}{M}O(L) \\
&= O\left(\frac{L^2}{N^2}\right), \qquad (6.20)
\end{aligned}$$

where in the second line we used the fact that $P(\overline{E_i}) = P(\overline{E_1})$. In the third line we used the upper bound for $\tilde{g}$, and in the fourth line we used the result in equation (6.18).

□

Now we are ready to continue the proof of the bias bound in (6.12). Let E be defined as the event that there is no collision for the hash function H_2, and all of its outputs are distinct, that is, $E = \cap_{i=1}^{F} E_i$

(6.12) can be written as

$$
\begin{aligned}
&\mathbb{E}\left[\widehat{D}_g(X,Y)\right] \\
&= \frac{1}{M}\sum_{\substack{i\le F\\ M_i>0}} P(E_i)\mathbb{E}\left[M_i\widetilde{g}\left(\frac{\eta N_i}{M_i}\right)\middle| E_i\right] + O\left(\frac{L^2}{N^2}\right) \\
&= \frac{P(E_1)}{M}\sum_{\substack{i\le F\\ M_i>0}} \mathbb{E}\left[M_i\widetilde{g}\left(\frac{\eta N_i}{M_i}\right)\middle| E_i\right] + O\left(\frac{L^2}{N^2}\right) \\
&= \frac{P(E_1)}{M}\sum_{\substack{i\le F\\ M_i>0}} \mathbb{E}\left[M_i\widetilde{g}\left(\frac{\eta N_i}{M_i}\right)\middle| E\right] + O\left(\frac{L^2}{N^2}\right) \qquad (6.21)\\
&= \frac{P(E_1)}{M}\mathbb{E}\left[\sum_{\substack{i\le F\\ M_i>0}} M_i\widetilde{g}\left(\frac{\eta N_i}{M_i}\right)\middle| E\right] + O\left(\frac{L^2}{N^2}\right) \\
&= \frac{P(E_1)}{M}\mathbb{E}\left[\sum_{i=1}^{L} M_i'\widetilde{g}\left(\frac{\eta N_i'}{M_i'}\right)\middle| E\right] + O\left(\frac{L^2}{N^2}\right) \qquad (6.22)\\
&= \frac{1-O(L/F)}{M}\mathbb{E}\left[\sum_{i=1}^{M}\widetilde{g}\left(\frac{\eta N_i'}{M_i'}\right)\right] + O\left(\frac{L^2}{N^2}\right) \qquad (6.23)\\
&= \mathbb{E}_{Y_1\sim f_2(x)}\mathbb{E}\left[\widetilde{g}\left(\frac{\eta N_1'}{M_1'}\right)\middle| Y_1\right] + O\left(\frac{L^2}{N^2}\right), \qquad (6.24)
\end{aligned}
$$

where in (6.21) we have used the fact that conditioned on E_i, N_i and M_i are independent of E_j for $i \neq j$. In (6.22) since there is no collision in H_2, M_i' and N_i' are equal to M_j and N_j for some i and j. Equation (6.23) is because the values M_i' and N_i' are independent of the hash function H_2 and its outputs, and finally in equation (6.24), we used the fact that each set N_i' and M_i' are i.i.d random variables.

At this point, assuming that the variance of $\frac{N_1'}{M_1'}$ is upper bounded by $O(1/N)$ and using (Lemma 3.2 in [94]), we only need to derive $\mathbb{E}\left[\frac{N_1'}{M_1'}\right]$, and then we can simply find the RHS in (6.24). Note that N_i' and M_i' are independent and have binomial distributions with the respective means of NP_i^X and MP_i^Y, where P_i^X and P_i^Y are the probabilities of mapping X and Y points with the respective densities f_0 and f_1

into bin i. Hence,

$$\mathbb{E}\left[\frac{N_1'}{M_1'}\middle|Y_1\right] = \mathbb{E}\left[N_1'|Y_1\right]\mathbb{E}\left[{M_1'}^{-1}\middle|Y_1\right]. \tag{6.25}$$

Let B_i denote the area for which all the points map to the same vector V_i. $\mathbb{E}\left[N_i'\right]$ can be written as:

$$\begin{aligned}
\mathbb{E}\left[N_i'\right] &= N\int_{x\in B_i} f_1(x)dx \\
&= N\int_{x\in B_i} f_1(Y_i) + O(\|x-Y_i\|^\gamma)dx \\
&= N\epsilon^d f_1(Y_i) + N\int_{x\in B_i} O(\|x-Y_i\|^\gamma)dx \\
&= N\epsilon^d f_1(Y_i) + N\int_{x\in B_i+Y_i} O(\|x\|^\gamma)dx, \qquad (6.26)
\end{aligned}$$

where in the second equality we have used the fact that the density functions satisfy Hölder smoothness with parameter γ. Let define $B_i' := \frac{1}{\epsilon}B_i + \frac{1}{\epsilon}Y_i$ and

$$C_\gamma(Y_i) := \int_{x'\in B_i'} \|x'\|^\gamma dx'. \tag{6.27}$$

Note that $C_\gamma(Y_i)$ is a constant independent of ϵ, since the volume of B_i' is independent of ϵ. By defining $x' = x/\epsilon$ we can write

$$\int_{x\in B_i+Y_i} \|x\|^\gamma dx = \int_{x'\in B_i'} \epsilon^\gamma \|x'\|^\gamma(\epsilon^d dx') = C_\gamma(Y_i)\epsilon^{\gamma+d} \tag{6.28}$$

Also note that since the number of X and Y points in each bin are independent we have $\mathbb{E}\left[N_i'|Y_i\right] = \mathbb{E}\left[N_i'\right]$, and therefore

$$\mathbb{E}\left[N_i'|Y_i\right] = N\epsilon^d f_1(Y_i) + O\left(N\epsilon^{\gamma+d}C_\gamma(Y_i)\right). \tag{6.29}$$

Next, note that $\mathbb{E}\left[M_i'|Y_i\right]$ has a non-zero binomial distribution, for which the first order inverse moment can be written as [130]:

$$\mathbb{E}\left[M_i'^{-1}|Y_i\right] = \left[M\epsilon^d f_2(Y_i) + O\left(M\epsilon^{\gamma+d}C(Y_i)\right)\right]^{-1} \times \left(1 + O\left(\frac{1}{M\epsilon^d f_2(Y_i)}\right)\right)$$

$$= \left(M\epsilon^d f_2(Y_i)\right)^{-1}\left[1 + O\left(\epsilon^\gamma\right) + O\left(\frac{1}{M\epsilon^d}\right)\right] \tag{6.30}$$

Thus, (A.33) can be simplified as

$$\mathbb{E}\left[\frac{N_1'}{M_1'}\middle| Y_1\right] = \frac{f_1(Y_1)}{\eta f_2(Y_1)} + O\left(\epsilon^\gamma\right) + O\left(\frac{1}{M\epsilon^d}\right). \tag{6.31}$$

We use (Lemma 3.2 in [94]) and Remark VI.5, and obtain

$$\mathbb{E}\left[\widetilde{g}\left(\frac{\eta N_1'}{M_1'}\right)\middle| Y_1\right] = g\left(\frac{f_1(Y_1)}{f_2(Y_1)}\right) + O\left(\epsilon^\gamma\right) + O\left(\frac{1}{M\epsilon^d}\right) + O(N^{-\frac{1}{2}}). \tag{6.32}$$

Finally from (6.24) we get

$$\mathbb{B}\left[\widehat{D}_g(X,Y)\right] = O\left(\epsilon^\gamma\right) + O\left(\frac{1}{M\epsilon^d}\right) + O(N^{-\frac{1}{2}}) + O\left(\frac{L^2}{N^2}\right)$$

$$= O\left(\epsilon^\gamma\right) + O\left(\frac{1}{N\epsilon^d}\right), \tag{6.33}$$

where in the third line we have used the upper bound on L in Lemma VI.9 and the fact that $M/N = \eta$. Finally note that we can use a similar method with the same steps to prove the convergence of an estimator for Rényi divergence.

6.6 Discussion and Experiments

In this section we compare and contrast the advantages of the proposed estimator with competing estimators, and provide numerical results. These show the efficiency of our estimator in terms of MSE rate and computational complexity.

Table 6.1: Comparison of proposed estimator to Ensemble NNR [94], Ensemble KDE [82] and Mirror KDE [110]

Estimator	HB	NNR	Ensemble KDE	Mirror KDE
MSE Rate	$O(1/N)$	$O(1/N)$	$O(1/N)$	$O(1/N)$
Computational Complexity	$O(N)$	$O(kN \log N)$	$O(N^2)$	$O(N^2)$
Required Smoothness (γ)	d	d	$(d+1)/2$	$d/2$
Extra Smooth Boundaries	No	Yes	Yes	Yes
Online Estimation	Yes	No	No	No
Knowledge about Boundary	No	No	No	Yes

Table 6.1 summarizes the differences between the proposed optimum estimator (EHB) with other competing estimators: Ensemble NNR [94], Ensemble KDE [82] and Mirror KDE [110]. In terms of MSE rate, all of these estimators can achieve the optimal parametric MSE rate of $O(1/N)$. In terms of computational complexity, our estimator has the best runtime compared to others. The smoothness parameter required for the optimum MSE rate is stated in terms of number of required derivatives of the density functions. The proposed estimator is the first divergence estimator that requires no extra smoothness at the boundaries. It is also the first divergence estimator that is directly applicable to online settings, retaining both the accuracy and linear total runtime. Finally, similar to NNR and Ensemble KDE estimators, the proposed estimator does not require any prior knowledge of the support of the densities.

It is also worthwhile to compare the proposed hash-based estimators (HB and EHB) to the histogram plug-in estimator. While the histogram estimator performs poorly when the support set is unknown, the hash based estimator does not rely on the knowledge about the support set. There is a trade-off between bias and variance depending on the bin size parameter in histogram estimators that affects convergence rate. In hash-based estimators the variance is independent of the parameter ϵ., which results in a better performance. In the hash-based estimator, only bins for which $M_i > 0$ are used resulting in reduced memory requirements. Finally, as

discussed before, the computational and space complexity of the hash-based estimator respectively grows linearly with the size of dimension. On the other hand, the histogram estimator suffers from exponential time and space complexity with respect to dimension.

Finally, handling the binning in histogram estimators for the support sets with complex contours makes histogram estimators difficult to implement, especially in high dimension. Implementation of our proposed hash-based estimator does not have this complexity since it does not depend on knowledge of the contours.

We compare the empirical performance of EHB to NNR, and the Ensemble KDE estimators. The experiments are done for two different types of f-divergence; KL-divergence and α-divergence defined in [22]. Assume that X and Y are i.i.d. samples from independent truncated Gaussian densities. Figure 6.2, shows the MSE estimation rate of α-divergence with $\alpha = 0.5$ of two Gaussian densities with the respective expectations of $[0, 0]$ and $[0, 1]$, and equal variances of $\sigma^2 = I_2$ for different numbers of samples. For each sample size we repeat the experiment 50 times, and compute the MSE of each estimator. While all of the estimators have the same asymptotic MSE rate, in practice the proposed estimator performs better. The runtime of this experiment is shown in Figure 6.3. The runtime experiment confirms the advantage of the EHB estimator compared to the previous estimators, in terms of computational complexity. Figure 6.4, shows the comparison of the estimators of KL-divergence between two truncated Gaussian densities with the respective expectations of $[0, 0]$ and $[0, 1]$, and equal covariance matrices of $\sigma_1^2 = \sigma_2^2 = I_2$, in terms of their mean value and %95 confidence band. The confidence band gets narrower for greater values of N, and EHB estimator has the narrowest confidence band. In Figure 6.5 the MSE rates of the three α-divergence estimators are compared in dimension $d = 4$, $\alpha = 2$,

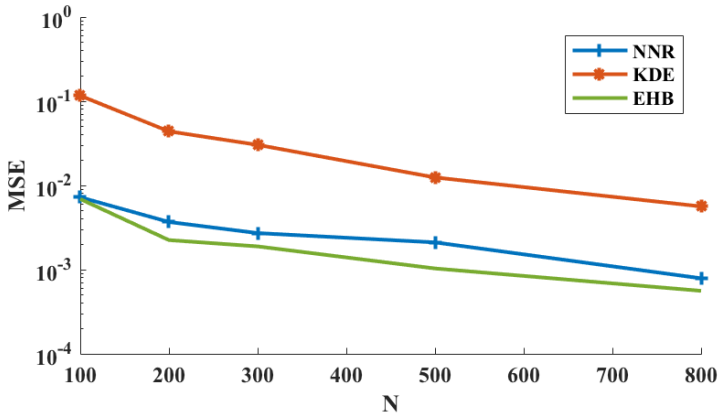


Figure 6.2: MSE comparison of α-divergence estimators with $\alpha = 0.5$ between two independent, mean-shifted truncated 2D Gaussian densities, versus different number of samples.

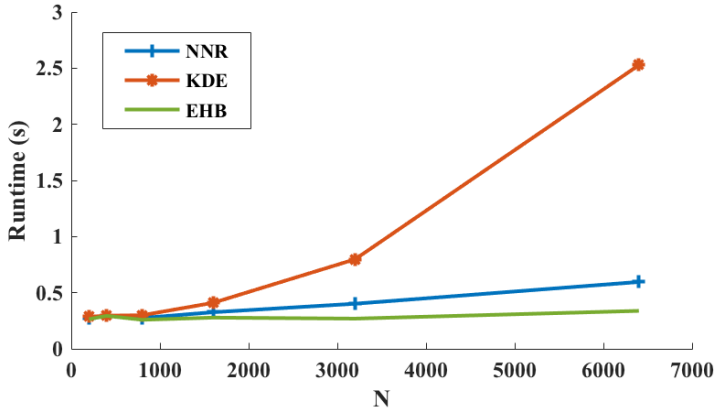


Figure 6.3: Runtime comparison of α-divergence with $\alpha = 0.5$ between two independent, mean-shifted truncated 2D Gaussian densities, versus different number of samples.

for two independent truncated Gaussian densities with the expectations $\mu_1 = \mu_2$ and covariances $\sigma_1^2 = \sigma_2^2 = I_4$, versus different number of samples.

6.7 Conclusion

In this chapter we proposed a fast hash based estimation method for f-divergence. We obtained bias and variance convergence rates of the base estimator. Then, an ensemble estimator was proposed which improved the MSE convergence rate to $O(1/N)$. An algorithm for the online settings was proposed and we analyzed its MSE convergence rate as well as runtime. Further, we validated our results by numerical

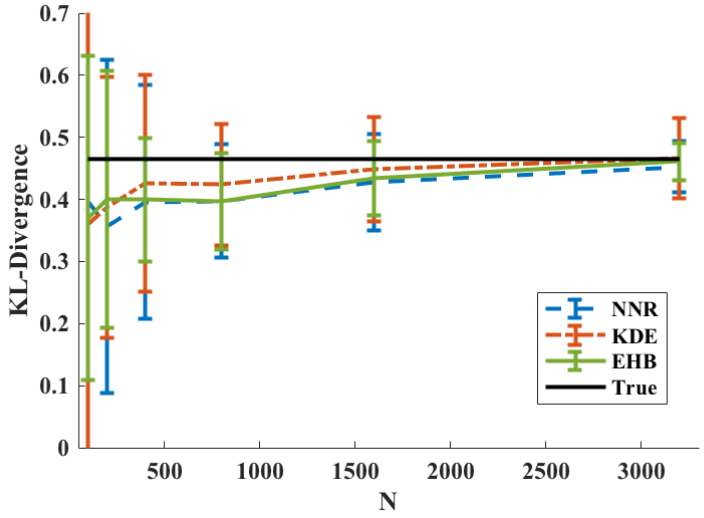


Figure 6.4: Comparison of the estimators of KL-divergence between two mean-shifted truncated 2D Gaussian densities, in terms of their mean value and %95 confidence band.

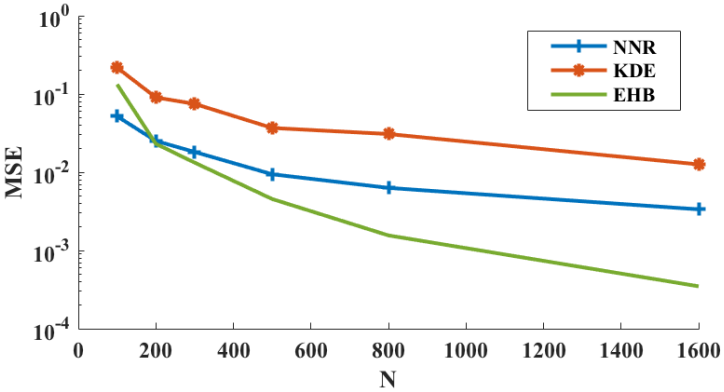


Figure 6.5: MSE estimation rate of α-divergence with $\alpha = 2$ between two identical truncated Gaussian densities with dimension $d = 4$, versus different number of samples.

experiments. Investigating the convergence rate of the hash-based divergence estimation using different hashing schemes is a worthwhile topic for future work.

CHAPTER VII

Hash-based Estimation of Mutual Information

In this chapter we propose a reduced complexity MI estimator called the ensemble dependency graph estimator (EDGE). The estimator combines randomized locality sensitive hashing (LSH), dependency graphs, and ensemble bias-reduction methods. Assume that we have N i.i.d samples of $Z_i = (X_i, Y_i)$. X_i and Y_i are considered a partitioning of the feature vector of Z_i, where in machine learning methods X_i and Y_i are respectively specified as input (explanatory) and output (response) data vectors. A dependence graph is a bipartite directed graph consisting of two sets of nodes V and U. The data points are mapped to the sets V and U using a randomized LSH function H that depends on a hash parameter ϵ. Each node is assigned a weight that is proportional to the number of hash collisions. Likewise, each edge between the vertices v_i and u_j has a weight proportional to the number of (X_k, Y_k) pairs mapped to the node pairs (v_i, u_j). For a given value of the hash parameter ϵ, a base estimator of MI is proposed as a weighted average of non-linearly transformed of the edge weights. The proposed EDGE estimator of MI is obtained by applying the method of weighted ensemble bias reduction [82, 85] to a set of base estimators with different hash parameters. This estimator is a non-trivial extension of the LSH divergence estimator defined in Chapter VI.

In this chapter we represent the mutual informaiton function in a slightly different form. Let $\mathcal{X}$ and $\mathcal{Y}$ be Euclidean spaces and let P_{XY} be a probability measure on the space $\mathcal{X} \times \mathcal{Y}$. For any measurable sets $A \subseteq \mathcal{X}$ and $B \subseteq \mathcal{Y}$, we define the marginal probability measures $P_X(A) := P_{XY}(A \times \mathcal{Y})$ and $P_Y(B) := P_{XY}(\mathcal{X} \times B)$. Similar to [98, 39], the general MI denoted by $I(X, Y)$ is defined as

$$D(P_{XY} \| P_X P_Y) = \mathop{\mathbb{E}}_{P_X P_Y} \left[g\left(\frac{dP_{XY}}{dP_X P_Y} \right) \right], \tag{7.1}$$

where $\frac{dP_{XY}}{dP_X P_Y}$ is the Radon-Nikodym derivative, and $g : (0, \infty) \to \mathbb{R}$ is a convex function with $g(1) = 0$. Shannon mutual information is a particular cases of (7.1) for which $g(x) = x \log x$.

The contributions of this chapter can be summarized as follows:

- To the best of our knowledge the proposed MI estimator is the first estimator to have linear complexity and can achieve the optimal MSE rate of $O(1/N)$.
- The proposed MI estimator provides a simplified and unified treatment of mixed continuous-discrete variables. This is due to the hash function approach that is adopted.
- The proposed dependence graph provides an intuitive way of understanding interdependencies in the data; e.g. sparsity of the graph implies a strong dependency between the covariates, while an equally weighted dense graph implies that the covariates are close to independent.
- EDGE is applied to IB theory of deep learning, and provides evidence that the compression property does indeed occur in ReLu DNNs, contrary to the claims of [103].

The rest of this chapter is organized as follows. In Section 7.2, we introduce the

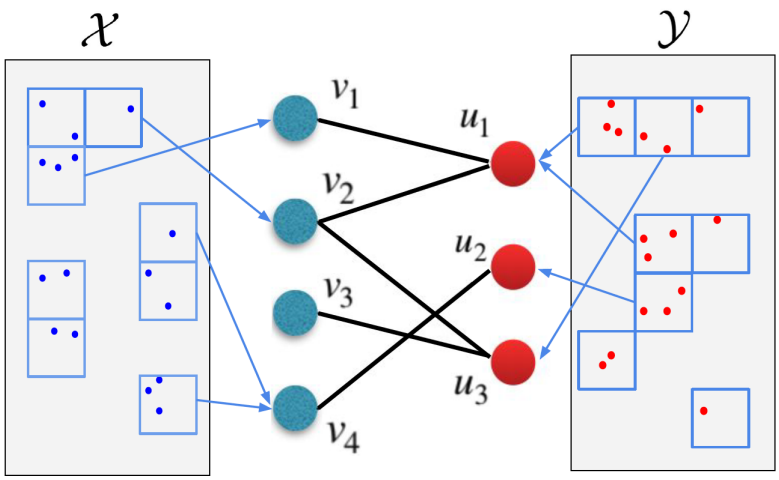


Figure 7.1: Sample dependence graph with 4 and 3 respective distinct hash values of $\mathbf{X}$ and $\mathbf{Y}$ data jointly encoded with LSH, and the corresponding dependency edges.

hash based MI estimator and give theory for the bias and variance. In section 7.4 we introduce the ensemble dependence graph MI estimator (EDGE) and show how the ensemble estimation method can be used to improve the convergence rates. Finally, in Section 3.2.3 we provide numerical results as well as study the IP in DNNs.

7.1 Dependence Graphs

Consider N i.i.d samples (X_i, Y_i), $1 \leq i \leq N$ drawn from the probability measure P_{XY}, defined on the space $\mathcal{X} \times \mathcal{Y}$. Define the sets $\mathbf{X} = \{X_1, X_2, ..., X_N\}$ and $\mathbf{Y} = \{Y_1, Y_2, ..., Y_N\}$. The dependence graph $G(X, Y)$ is a directed bipartite graph, consisting of two sets of nodes V and U with cardinalities denoted as $|V|$ and $|U|$, and the set of edges E_G. Each point in the sets $\mathbf{X}$ and $\mathbf{Y}$ is mapped to the nodes in the sets U and V, respectively, using the hash function H, described as follows.

A vector valued hash function H is defined in a similar way as defined in [92]. First, define the vector valued hash function $H_1 : \mathbb{R}^d \to \mathbb{Z}^d$ as

$$H_1(x) = [h_1(x_1), h_1(x_2), ..., h_1(x_d)], \tag{7.2}$$

where x_i denotes the ith component of the vector x. In (7.2), each scalar hash

function $h_1(x_i) : \mathbb{R} \to \mathbb{Z}$ is given by

$$h_1(x_i) = \left\lfloor \frac{x_i + b}{\epsilon} \right\rfloor, \tag{7.3}$$

for a fixed $\epsilon > 0$, where $\lfloor y \rfloor$ denotes the floor function (the smallest integer value less than or equal to y), and b is a fixed random variable in $[0, \epsilon]$. Let $\mathcal{F} := \{1, 2, .., F\}$, where $F := c_H N$ and c_H is a fixed tunable integer. We define a random hash function $H_2 : \mathbb{Z}^d \to \mathcal{F}$ with a uniform density on the output and consider the combined hashing function

$$H(x) := H_2(H_1(x)), \tag{7.4}$$

which maps the points in $\mathbb{R}^d$ to $\mathcal{F}$.

$H(x)$ reveals the index of the mapped vertex in $G(X, Y)$. The weights ω_i and ω'_j corresponding to the nodes v_i and u_j, and ω_{ij}, the weight of the edge (v_i, u_j), are defined as follows.

$$\omega_i = \frac{N_i}{N}, \qquad \omega'_j = \frac{M_j}{N}, \qquad \omega_{ij} = \frac{N_{ij} N}{N_i M_j}, \tag{7.5}$$

where N_i and M_j respectively are the the number of hash collisions at the vertices v_i and u_j, and N_{ij} is the number of joint collisions of the nodes (X_k, Y_k) at the vertex pairs (v_i, u_j). The number of hash collisions is defined as the number of instances of the input variables map to the same output value. In particular,

$$N_{ij} := \#\{(X_k, Y_k) \text{ s.t } H(X_k) = i \text{ and } H(Y_k) = j\}. \tag{7.6}$$

Fig. 7.1 represents a sample dependence graph. Note that the nodes and edges with zero collisions do not show up in the dependence graph.

7.2 The Base Estimator of Mutual Information

7.2.1 Assumptions

The assumptions we consider here are slightly different from the general assumptions considered in Chapter .I. Thus, we list the following are the assumptions we make on the probability measures and g:

A1. The support sets $\mathcal{X}$ and $\mathcal{Y}$ are bounded.

A2. The following supremum exists and is bounded:

$$\sup_{P_X P_Y} g\left(\frac{dP_{XY}}{dP_X P_Y}\right) \leq U.$$

A3. Let x_D and x_C respectively denote the discrete and continuous components of the vector x. Also let $f_{X_C}(x_C)$ and $p_{X_D}(x_D)$ respectively denote density and pmf functions of these components associated with the probability measure P_X. The density functions $f_{X_C}(x_C)$, $f_{Y_C}(y_C)$, $f_{X_C Y_C}(x_C, y_C)$, and the conditional densities $f_{X_C|X_D}(x_C|x_D)$, $f_{Y_C|Y_D}(y_C|y_D)$, $f_{X_C Y_C|X_D Y_D}(x_C, y_C|x_D, y_D)$ are Hölder continuous.

A4. Assume that the function g in (7.1) is Lipschitz continuous; i.e. g is Hölder continuous with $\gamma = 1$.

7.2.2 Definition of the Base Estimator

For a fixed value of the hash parameter ϵ, we propose the following base estimator of MI (7.1) function based on the dependence graph:

$$\widehat{I}(X,Y) := \sum_{e_{ij} \in E_G} \omega_i \omega_j' \widetilde{g}\left(\omega_{ij}\right), \tag{7.7}$$

where the summation is over all edges $e_{ij} : (v_i \to u_j)$ of $G(X,Y)$ having non-zero weight and $\widetilde{g}(x) := \max\{g(x), U\}$.

When X and Y are strongly dependent, each point X_k hashed into the bucket (vertex) v_i corresponds to a unique hash value for Y_k in U. Therefore, asymptotically

$\omega_{ij} \to 1$ and the mutual information estimation in (7.7) takes its maximum value. On the other hand, when X and Y are independent, each point X_k hashed into the bucket (vertex) v_i may be associated with different values of Y_k, and therefore asymptotically $\omega_{ij} \to \omega_j$ and the Shannon MI estimation tends to 0.

Remark VII.1. Similar to most of LSH structures, computing the hashing output in our estimator is of $O(1)$ complexity, and does not depend on ϵ. Thus, the computational complexity this estimator is $O(N)$. Also note that the computational complexity of computing the summation in (7.7) depends on the number of edges. In general, based on computation of $\mathbb{E}[N_{ij}]$ in Lemma E.4 (Appendix), the average number of the edges of the dependence graph depends on the joint distribution of $P_X Y$ as follow:

$$\mathbb{E}[|e|] = O\left(\sum_{ij} \mathbb{1}_{\{p_{ij}>0\}}\right). \tag{7.8}$$

Note that in the worst case, the number of edges is N, and in the best case, the number of the edges would be $\max\{L_X, L_Y\}$. In the latter case, using (E.1), the computational complexity can be upper bounded by $O(\epsilon^{-d})$.

7.3 Convergence Rate

In the following theorems we state upper bounds on the bias and variance rates of the proposed MI estimator (7.7).

Theorem VII.2. *Let $d = d_X + d_Y$ be the dimension of the joint random variable (X, Y). Under the aforementioned assumptions **A1-A4**, and assuming that the density functions in **A3** have bounded derivatives up to order $q \geq 0$, the following upper*

bound on the bias of the estimator in (7.7) *holds*

$$\mathbb{B}\left[\widehat{I}(X,Y)\right] = \begin{cases} O\left(\epsilon^{\gamma}\right) + O\left(\frac{1}{N\epsilon^{d}}\right), & q = 0 \\ \sum_{i=1}^{q} C_i \epsilon^{i} + O\left(\epsilon^{q}\right) + O\left(\frac{1}{N\epsilon^{d}}\right) & q \geq 1, \end{cases} \tag{7.9}$$

where ϵ *is the hash parameter in* (7.3)*,* γ *is the smoothness parameter in Chapter I, and* C_i *are real constants.*

In (7.9), the hash parameter, ϵ needs to be a function of N to ensure that the bias converges to zero. For the case of $q = 0$, the optimum bias is achieved when $\epsilon = \left(\frac{1}{N}\right)^{\gamma/(\gamma+d)}$. When $q \geq 1$, the optimum bias is achieved for $\epsilon = \left(\frac{1}{N}\right)^{1/(1+d)}$.

Theorem VII.3. *Under the assumptions* ***A1-A4*** *the variance of the proposed estimator can be bounded as* $\mathbb{V}\left[\widehat{I}(X,Y)\right] \leq O\left(\frac{1}{N}\right)$*. Further, the variance of the variable* ω_{ij} *is also upper bounded by* $O(1/N)$*.*

7.4 Ensemble Dependence Graph Estimator (EDGE)

Definition VII.4. Given the expression for the bias in Theorem VII.2, the ensemble estimation technique proposed in [82] can be applied to improve the convergence rate of the MI estimator (7.7). Assume that the densities in **A3** have continuous bounded derivatives up to the order q, where $q \geq d$. Let $\mathcal{T} := \{t_1, ..., t_T\}$ be a set of index values with $t_i < c$, where $c > 0$ is a constant. Let $\epsilon(t) := tN^{-1/2d}$. For a given set of weights $w(t)$ the weighted ensemble estimator is then defined as

$$\widehat{I}_w := \sum_{t \in \mathcal{T}} w(t) \widehat{I}_{\epsilon(t)}, \tag{7.10}$$

where $\widehat{I}_{\epsilon(t)}$ is the mutual information estimator with the parameter $\epsilon(t)$. Using (7.9), for $q > 0$ the bias of the weighted ensemble estimator (7.10) takes the form

$$\mathbb{B}(\hat{I}_w) = \sum_{i=1}^{q} CiN^{-\frac{i}{2d}} \sum_{t \in \mathcal{T}} w(t) t^{i} + O\left(\frac{t^{d}}{N^{1/2}}\right) + O\left(\frac{1}{N\epsilon^{d}}\right) \tag{7.11}$$

Given the form (7.11), as long as $T \geq q$, we can select the weights $w(t)$ to force to zero the slowly decaying terms in (7.11), i.e. $\sum_{t\in\tau} w(t)t^{i/d} = 0$ subject to the constraint that$\sum_{t\in\tau} w(t) = 1$. However, T should be strictly greater than q in order to control the variance, which is upper bounded by the euclidean norm squared of the weights ω.

Theorem VII.5. *For $T > d$ let w_0 be the solution to:*

$$\begin{aligned} \min_{w} \quad & \|w\|_2 \\ \textit{subject to} \quad & \sum_{t\in\mathcal{T}} w(t) = 1, \\ & \sum_{t\in\mathcal{T}} w(t)t^i = 0, i \in \mathbb{N}, i \leq d. \end{aligned} \tag{7.12}$$

Then the MSE rate of the ensemble estimator $\widehat{I}_{w_0}$ is $O(1/N)$.

7.5 Numerical Results

We use a simulated dataset to compare the proposed estimator to the competing MI estimators Ensemble KDE (EKDE) [85], and generalized KSG [39]. Both of these estimators work on mixed continuous-discrete variables.

Fig. 7.2, shows the MSE estimation rate of Shannon MI between the continuous random variables X and Y having the relation $Y = X + aN_U$, where X is a $2D$ Gaussian random variable with the mean $[0, 0]$ and covariance matrix $C = I_2$. Here I_d denote the d-dimensional identity matrix. N_U is a uniform random vector with the support $\mathcal{N}_U = [0, 1] \times [0, 1]$. We compute the MSE of each estimator for different sample sizes. The MSE rates of EDGE, EKDE and KSG are compared for $a = 1/5$. Further, the MSE rate of EDGE is investigated for noise levels of $a = \{1/10, 1/5, 1/2, 1\}$. As the dependency between X and Y increases the MSE rate becomes slower.

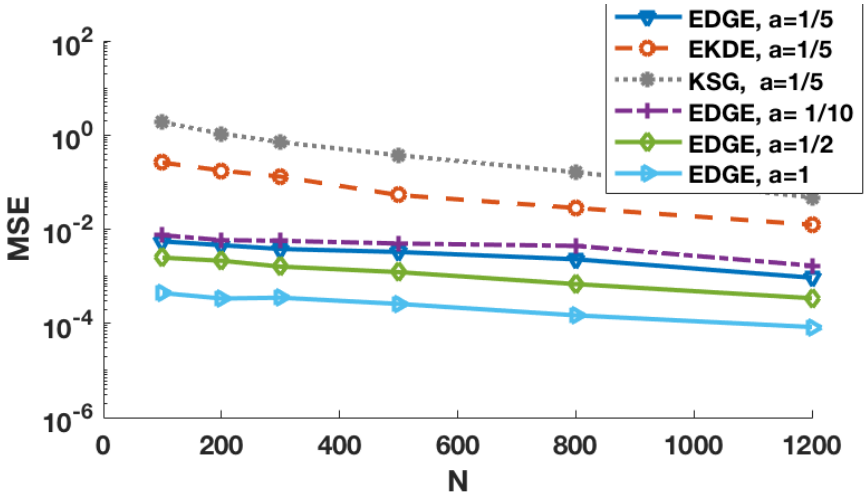


Figure 7.2: MSE comparison of EDGE, EDKE and KSG Shannon MI estimators. X is a $2D$ Gaussian random variable with unit covariance matrix. $Y = X + aN_U$, where N_U is a uniform noise. The MSE rates of EDGE, EKDE and KSG are compared for various values of a.

Fig. 7.3, shows the MSE estimation rate of Shannon MI between a discrete random variables X and a continuous random variable Y. We have $X \in \{1, 2, 3, 4\}$, and each $X = x$ is associated with multivariate Gaussian random vector Y, with $d = 4$, the expectation $[x/2, 0, 0, 0]$ and covariance matrix $C = I_4$. In general in Figures 7.2 and 7.3, EDGE has better convergence rate than EKDE and KSG estimators. Fig. 7.4 represents the runtime comparison for the same experiment as in Fig. 7.3. It can be seen from this graph how fast our proposed estimator performs compared to the other other methods.

7.6 Information Bottlenekc Theory of Deep Learning

Recently, Shwartz-Ziv and Tishby utilized mutual information measure to study the training process in Deep Neural Networks (DNN) [107]. Let X, T and Y respectively denote the input, hidden and output layers. The authors of [107] introduced the information bottleneck (IB) that represents the tradeoff between two mutual information measures: $I(X, T)$ and $I(T, Y)$. They observed that the training process

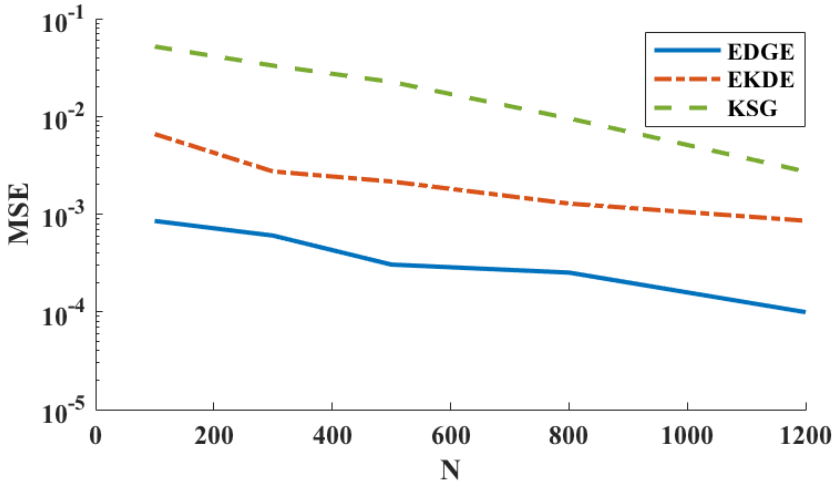


Figure 7.3: MSE comparison of EDGE, EDKE and KSG Shannon MI estimators. $X \in \{1, 2, 3, 4\}$, and each $X = x$ is associated with multivariate Gaussian random vector Y, with $d = 4$, the mean $[x/2, 0, 0, 0]$ and covariance matrix $C = I_4$.

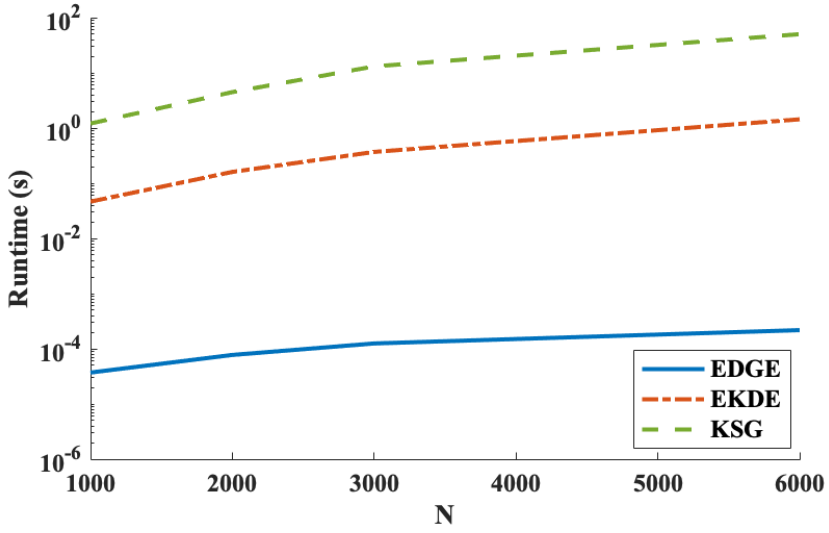


Figure 7.4: Runtime comparison of EDGE, EDKE and KSG Shannon MI estimators. $X \in \{1, 2, 3, 4\}$, and each $X = x$ is associated with multivariate Gaussian random vector Y, with $d = 4$, the mean $[x/2, 0, 0, 0]$ and covariance matrix $C = I_4$.

of a DNN consists of two distinct phases; 1) an initial fitting phase in which $I(T,Y)$ increases, and 2) a subsequent compression phase in which $I(X,T)$ decreases. Saxe *et al* in [103] countered the claim of [107], asserting that this compression property is not universal, rather it depends on the specific activation function. Specifically, they claimed that the compression property does not hold for ReLu activation functions. The authors of [107] challenged these claims, arguing that the authors of [103] had not observed compression due to poor estimates of the MI. We apply our proposed rate-optimal ensemble MI estimator to explore this controversy, observing that our estimator of MI does exhibit the compression phenomenon in the ReLU network studied by [103].

Fig. 7.5 represents the information plane of a DNN with 4 fully connected hidden layers of width $784-1024-20-20-20-10$ with tanh and ReLU activations. The sequence of colored points shows different iterations of the training process. Each gray line connects the points with the same iterations for diferent layers. The left most sequence of points corresponds to the last hidden layer and the right most sequence of points corresponds to the first hidden layer. The network is trained with Adam optimization with a learning rate of 0.003 and cross-entropy loss functions to classify the MNIST handwritten-digits dataset. We repeat the experiment for 20 iterations with different randomized initializations and take the average over all experiments. In both cases of ReLU and tanh activations we observe some degree of compression in all of the hidden layers. However, the amount of compressions is different for ReLU and tanh activations. The average test accuracy in both of these networks are around 0.98. This network is the same as the one studied in [103], for which it is claimed that no compression happens with a ReLU activation. The base estimator used in [103] provides KDE-based lower and upper bounds on the

true MI [64]. According to our experiments (not shown) the upper bound is in some cases twice as large as the lower bound. In contrast, our proposed ensemble method estimates the exact mutual information with significantly higher accuracy.

Fig. 7.6 represents the information plane for another network with 4 fully connected hidden layers of width $784-200-100-60-30-10$ with ReLU activation. The network is trained with Adam optimization with a learning rate of 0.003 and cross-entropy loss functions to classify the MNIST handwritten-digits dataset. Again, we observe compression for this network with ReLU activation.

Finally, we study the information plane curves in a CNN with three convolutioal ReLU layers and a dense ReLU layer. The convolutional layers respectively have depths of $4, 8, 16$ and the dense layer has the dimension 256. Max-pooling functions are used in the second and third layers. Note although for a certain initialization of the weights this model can achieve the test accuracy of 0.99, the average test accuracy (over different weight initializations) is around 0.95. That's why the converged point of the last layer has smaller $I(T, Y)$ compared to the examples in Fig. 7.5, which achieves the average test accuracy of 0.98. Another interesting point about the information plane in CNN is that the convolutional layers have larger $I(T, Y)$ compared to the hidden layers in the fully connected models in 7.5 and 7.6, which implies that the convolutional layers can extract almost all of the useful information about the labels after small number of iterations.

7.7 Conclusion

In this chapter we proposed a fast non-parametric estimation method for MI based on random hashing, dependence graphs, and ensemble estimation. Remarkably, the proposed estimator has linear computational complexity and attains optimal (para-

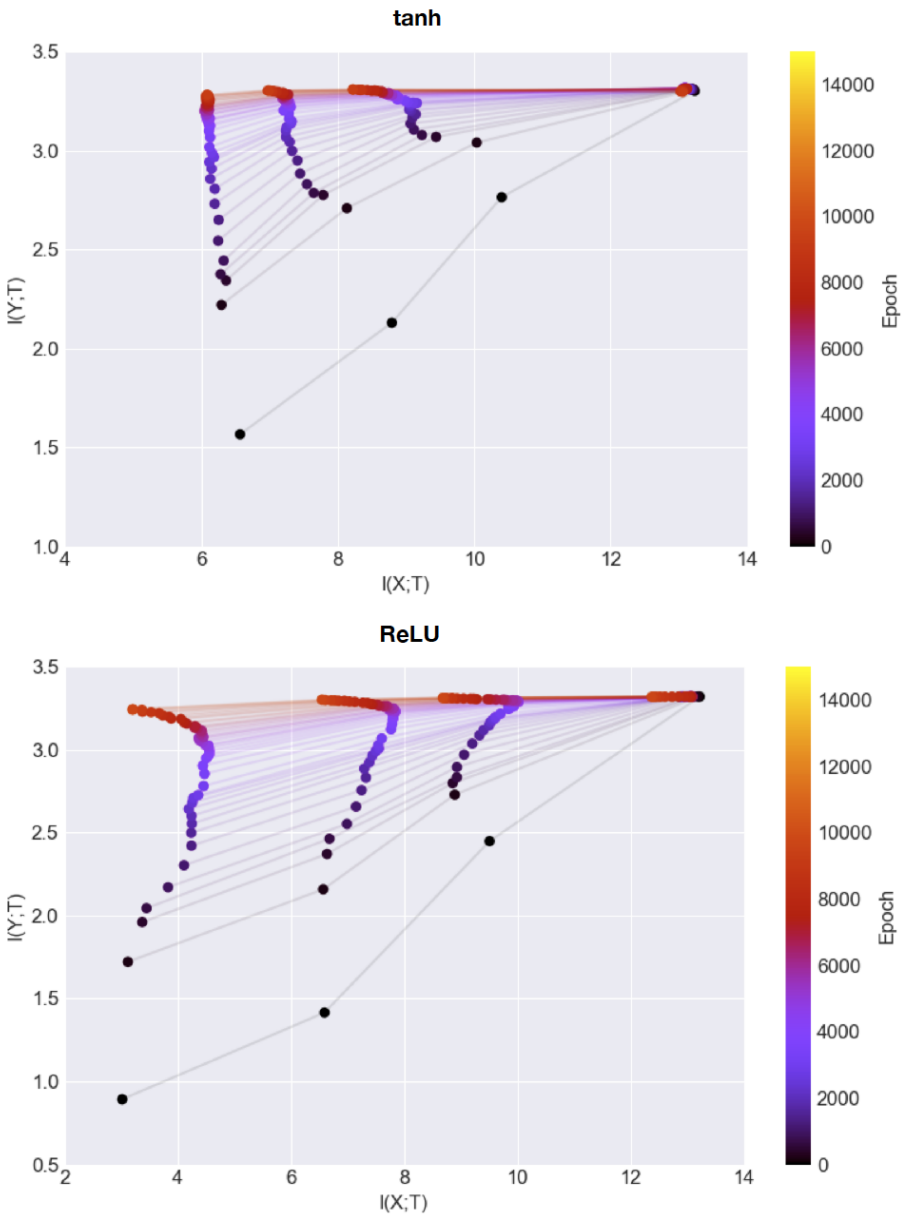


Figure 7.5: Information plane estimated using EDGE for a neural network of size $784 - 1024 - 20 - 20 - 20 - 10$ trained on the MNIST dataset with tanh (top) and ReLU (bottom) activations.

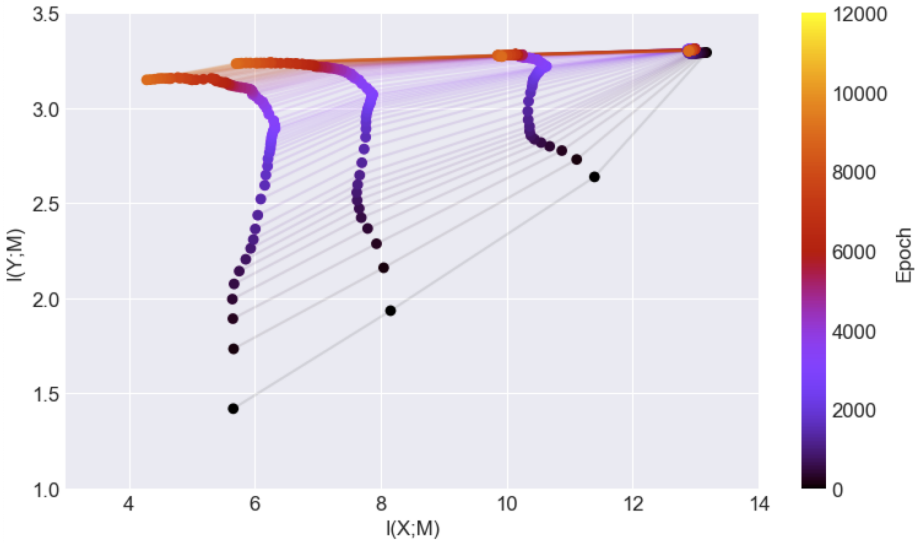


Figure 7.6: Information plane estimated using EDGE for a neural network of size $784 - 200 - 100 - 60 - 30 - 10$ trained on the MNIST dataset with ReLU activation.

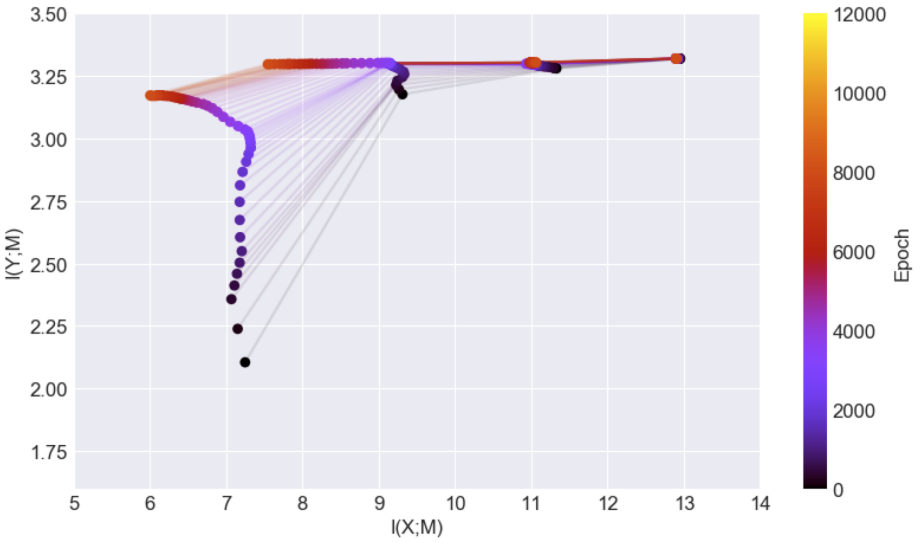


Figure 7.7: Information plane estimated using EDGE for a CNN consisting of three convolutioal ReLU layers with the respective depths of $4, 8, 16$ and a dense ReLU layer with the size of 256.

metric) rates of MSE convergence. We provided bias and variance convergence rate, and validated our results by numerical experiments. We studied the information bottleneck theory of deep learning based on the proposed MI estimator. Finally we proposed a feature quality measure based on the information bottleneck, and applied it on a real-world prediction problem to study the evolution of the features.

CHAPTER VIII

Information Theoretic Structure Learning

In this chapter we consider another application of an optimum estimation method of mutual information on structure learning. We introduce a new method for nonparametric structure discovery that uses weighted ensemble divergence estimators that achieve parametric convergence rates and obey an asymptotic central limit theorem that facilitates hypothesis testing and other types of statistical validation. We focus on two methods of nonparametric structure learning based on ensemble MI estimation. The first method is the Chow-Liu (CL) algorithm which constructs a first order tree from the MI of all pairs of RVs to approximate the joint pdf [21]. Since structure learning approaches can suffer from performance degradation when the model does not match the true distribution, we propose hypothesis testing via MI estimation to determine how well the tree structure imposed by the CL algorithm approximates the joint distribution. The second method learns the structure by performing hypothesis testing on the MI of all pairs of RVs. An edge is assigned between two RVs if the MI is statistically different from zero. In section 8.1 we introduce factor graph learning. In section 8.2 we propose an estimator of MI based on KDE plug-in estimator. Section 8.3 convers the convergence results of the proposed estimator. Finally in section 8.4 we provide the numerical results.

8.1 Factor Graph Learning

We focus on learning a second-order product approximation (i.e. a dependence tree) of the joint probability distribution of the data. Let $\mathbf{X}^{(i)}$ denote the ith component of a d-dimensional random vector $\mathbf{X}$. We use similar notation to [21] where the goal is to approximate the joint probability density $p(\mathbf{X})$ as

$$p'(\mathbf{X}) = \prod_{i=1}^{d} p\left(\mathbf{X}^{(m_i)}|\mathbf{X}^{(m_{j(i)})}\right), \tag{8.1}$$

where $0 \leq j(i) < i$, $(m_1, \ldots, m_d)$ is a (unknown) permutation of 1, 2, $\ldots d$, $p\left(\mathbf{X}^{(i)}|\mathbf{X}^{(0)}\right) = p\left(\mathbf{X}^{(i)}\right)$, and $p\left(\mathbf{X}^{(i)}|\mathbf{X}^{(j)}\right)$ $(j \neq 0)$ is the conditional probability density of $\mathbf{X}^{(i)}$ given $\mathbf{X}^{(j)}$.

The CL algorithm [21] provides an information theoretic method for selecting the second-order terms in (8.1). It chooses the second-order terms that minimize the Kullback-Leibler (KL) divergence between the joint density $p(\mathbf{X})$ and the approximation $p'(\mathbf{X})$. This reduces to constructing the maximal spanning tree where the edge weights correspond to the MI between the RVs at the vertices [21].

In practice, the pairwise MI between each pair of RVs is rarely known and must be estimated from data. Thus accurate MI estimators are required. Furthermore, while the sum of the pairwise MI gives a measure of the quality of the approximation, it does not indicate if the approximation is a sufficiently good fit or whether a different model should be used. This problem can be framed as testing the hypothesis that $p'(\mathbf{X}) = p(\mathbf{X})$ at a prescribed false positive level. This test can be performed using MI estimation.

In addition, we propose that (8.1) can be learned by performing hypothesis testing on the MI of all pairs of RVs and assigning an edge between two RVs if the MI is statistically different from zero. To account for the multiple comparisons bias, we

control the false discovery rate (FDR) [129].

8.2 Mutual Information Estimation Based on KDE

Information theoretic methods for learning nonlinear structures require accurate estimation of MI and estimates of its sample distribution for hypothesis testing. In this section, we employ the ensemble divergence and mutual information estimators and their conditional forms proposed in [86] and use the CLT to justify a large sample Gaussian approximation to the sampling distribution. We consider general MI functionals. Let $g : (0, \infty) \to \mathbb{R}$ be a smooth functional, e.g. $g(u) = \ln u$ for Shannon MI or $g(u) = u^\alpha$, with $\alpha \in [0, 1]$, for Rényi MI. Then we recall the definition of pairwise MI between $\mathbf{X}^{(i)}$ and $\mathbf{X}^{(j)}$ as

$$G_{ij} = \int g\left(\frac{p\left(x^{(i)}\right) p\left(x^{(j)}\right)}{p\left(x^{(i)}, x^{(j)}\right)}\right) p\left(x^{(i)}, x^{(j)}\right) dx^{(i)} dx^{(j)}. \tag{8.2}$$

For hypothesis testing, we are interested in the following

$$G\left(p; p'\right) = \int g\left(\frac{p'(x)}{p(x)}\right) p(x) dx. \tag{8.3}$$

We first define the plug-in KDE estimators. The conditional probability density is defined as the ratio of the joint and marginal densities. Thus the ratio within the g functional in (8.3) can be represented as the ratio of the product of some joint densities with two random variables and the product of marginal densities in addition to p. For example, if $d = 3$ and $p'(\mathbf{X}) = p\left(\mathbf{X}^{(1)}|\mathbf{X}^{(2)}\right) p\left(\mathbf{X}^{(2)}|\mathbf{X}^{(3)}\right) p\left(\mathbf{X}^{(3)}\right)$, then

$$\frac{p'(\mathbf{X})}{p(\mathbf{X})} = \frac{p\left(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}\right) p\left(\mathbf{X}^{(2)}, \mathbf{X}^{(3)}\right)}{p\left(\mathbf{X}^{(2)}\right) p\left(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \mathbf{X}^{(3)}\right)}. \tag{8.4}$$

For the KDEs, assume that we have N i.i.d. samples $\{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$ available from the joint density $p\left(\mathbf{X}\right)$. The KDE of $p(\mathbf{X}_j)$ is

$$\tilde{\mathbf{p}}_{X,h}(\mathbf{X}_j) = \frac{1}{Mh^d} \sum_{\substack{i=1 \\ i \neq j}} K\left(\frac{\mathbf{X}_j - \mathbf{X}_i}{h}\right),$$

where K is a symmetric product kernel function, h is the bandwidth, and $M = N-1$. Define the KDEs $\tilde{\mathbf{p}}_{ik,h}\left(\mathbf{X}_j^{(i)}, \mathbf{X}_j^{(k)}\right)$ and $\tilde{\mathbf{p}}_{i,h}\left(\mathbf{X}_j^{(i)}\right)$ (for $p\left(\mathbf{X}_j^{(i)}, \mathbf{X}_j^{(k)}\right)$ and $p\left(\mathbf{X}_j^{(i)}\right)$, respectively) similarly. Let $\tilde{\mathbf{p}}'_{X,h}(\mathbf{X}_j)$ be defined using the KDEs for the marginal densities and the joint densities with two random variables. For example, in the example given in (8.4), we have

$$\tilde{\mathbf{p}}'_{X,h}(\mathbf{X}_j) = \frac{\tilde{\mathbf{p}}_{12,h}\left(\mathbf{X}_j^{(1)}, \mathbf{X}_j^{(2)}\right)\tilde{\mathbf{p}}_{23,h}\left(\mathbf{X}_j^{(2)}, \mathbf{X}_j^{(3)}\right)}{\tilde{\mathbf{p}}_{2,h}\left(\mathbf{X}_j^{(2)}\right)}.$$

For brevity, we use the same bandwidth and product kernel for each of the KDEs although our method generalizes to differing bandwidths and kernels. The plug-in MI estimator for (8.3) is then

$$\tilde{\mathbf{G}}_h = \frac{1}{N}\sum_{j=1}^{N} g\left(\frac{\tilde{\mathbf{p}}'_{X,h}(\mathbf{X}_j)}{\tilde{\mathbf{p}}_{X,h}(\mathbf{X}_j)}\right).$$

The plug-in estimator $\tilde{\mathbf{G}}_{h,ij}$ for (8.2) is defined similarly.

8.3 Convergence Results

To apply bias-reducing ensemble methods to the plug-in estimators $\tilde{\mathbf{G}}_h$ and $\tilde{\mathbf{G}}_{h,ij}$, similar to the previous chapters, we need to derive their MSE convergence rates. As in [86], we assume that 1) the density $p(\mathbf{X})$ and all other marginal densities and pairwise joint densities are $s \geq 2$ times differentiable and the functional g is infinitely differentiable; 2) $p(\mathbf{X})$ has bounded support set $\mathcal{S}$; 3) all densities are strictly lower bounded on their support sets. Additionally, we make the same assumption on the boundary of the support as in [86]: 4) the support is smooth wrt the kernel $K(u)$ in the sense that the expectation of the area outside of $\mathcal{S}$ wrt any RV u with smooth distribution is a smooth function of the bandwidth h. This assumption is satisfied, for example, when $\mathcal{S}$ is the unit cube and $K(x)$ is the uniform rectangular kernel. For full technical details on the assumptions, see Appendix F.

Theorem VIII.1. *If g is infinitely differentiable, then the bias of $\tilde{\mathbf{G}}_{h,ij}$ and $\tilde{\mathbf{G}}_h$ are*

$$\mathbb{B}\left[\tilde{\mathbf{G}}_{h,ij}\right] = \sum_{m=1}^{\lfloor s \rfloor} c_{5,i,j,m} h^m + O\left(\frac{1}{Nh^2} + h^s\right)$$

$$\mathbb{B}\left[\tilde{\mathbf{G}}_h\right] = \sum_{m=1}^{\lfloor s \rfloor} c_{6,m} h^m + O\left(\frac{1}{Nh^d} + h^s\right). \tag{8.5}$$

If $g\left(t_1/t_2\right)$ has k, l-th order mixed derivatives $\frac{\partial^{k+l} g(t_1/t_2)}{\partial t_1^k \partial t_2^l}$ that depend on t_1, t_2 only through $t_1^\alpha t_2^\beta$ for some α, $\beta \in \mathbb{R}$ for each $1 \leq k, l \leq \lambda$ then the bias of $\tilde{\mathbf{G}}_h$ is

$$\mathbb{B}\left[\tilde{\mathbf{G}}_h\right] = \sum_{m=1}^{\lfloor s \rfloor} c_{6,m} h^m + \sum_{m=0}^{\lfloor s \rfloor} \sum_{q=1}^{\lfloor \lambda/2 \rfloor} \left(\frac{c_{7,1,q,m}}{(Nh^d)^q} + \frac{c_{7,2,q,m}}{(Nh^2)^q}\right) h^m$$

$$+ O\left(\frac{1}{(Nh^d)^{\lambda/2}} + h^s\right). \tag{8.6}$$

The expression in (8.6) allows us to achieve the parametric MSE rate under less restrictive assumptions on the smoothness of the densities ($s > d/2$ for (8.6) compared to $s \geq d$ for (8.5)). The extra condition required on the mixed derivatives of g to obtain the expression in (8.6) is satisfied, for example, for Shannon and Rényi information measures. Note that a similar expression could be derived for the bias of $\tilde{\mathbf{G}}_{h,ij}$. However, since $s \geq 2$ is required and the largest dimension of the densities estimated in $\tilde{\mathbf{G}}_{h,ij}$ is 2, we would not achieve any theoretical improvement in the convergence rate.

Theorem VIII.2. *If the functional $g(t_1/t_2)$ is Lipschitz continuous in both of its arguments with Lipschitz constant C_g, then the variance of both $\tilde{\mathbf{G}}_h$ and $\tilde{\mathbf{G}}_{h,ij}$ is $O(1/N)$.*

The Lipschitz assumption on g is comparable to assumptions required by other nonparametric distributional functional estimators [86, 65, 58, 109] and is ensured for functionals such as Shannon and Rényi informations by our assumption that the densities are bounded away from zero. The proofs of Theorems VIII.1 and VIII.2

share some similarities with the bias and variance proofs for the divergence functional estimators in [86]. The primary differences deal with the product of KDEs. See Appendix F for the full proofs.

From Theorems VIII.1 and VIII.2, letting $h \to 0$ and $Nh^2 \to \infty$ or $Nh^d \to \infty$ is required for the respective MSE of $\tilde{\mathbf{G}}_{h,ij}$ and $\tilde{\mathbf{G}}_h$ to go to zero. Without bias correction, the optimal MSE rate is, respectively, $O\left(N^{-2/3}\right)$ and $O\left(N^{-2/(d+1)}\right)$. Using an optimally weighted ensemble of estimators enables us to perform bias correction and achieve much better (parametric) convergence rates [86, 112].

The ensemble of estimators is created by varying the bandwidth h. Choose $\bar{l} = \{l_1, \ldots, l_L\}$ to be a set of positive real numbers and let $h(l)$ be a function of the parameter $l \in \bar{l}$. Define $w = \{w(l_1), \ldots, w(l_L)\}$ and $\tilde{\mathbf{G}}_w = \sum_{l\in\bar{l}} w(l)\tilde{\mathbf{G}}_{h(l)}$. Theorem 4 in [86] indicates that if enough of the terms in the bias expression of an estimator within an ensemble of estimators are known and the variance is $O(1/N)$, then the weight w_0 can be chosen so that the MSE rate of $\tilde{\mathbf{G}}_{w_0}$ is $O(1/N)$, i.e. the parametric rate. The theorem can be applied as follows. For general g, let $h(l) = lN^{-1/(2d)}$ for $\tilde{\mathbf{G}}_{h(l)}$. Denote $\psi_m(l) = l^m$ with $m \in J = \{1, \ldots, \lfloor s \rfloor\}$. The optimal weight w_0 is obtained by solving

$$
\begin{aligned}
&\min_w && ||w||_2 \\
&\textit{subject to} && \textstyle\sum_{l\in\bar{l}} w(l) = 1, \\
& && \left|\textstyle\sum_{l\in\bar{l}} w(l)\psi_m(l)\right| = 0,\ m \in J,
\end{aligned}
\tag{8.7}
$$

It can then be shown that the MSE of $\tilde{\mathbf{G}}_{w_0}$ is $O(1/N)$ as long as $s \geq d$ [80]. This works by using the last line in (8.7) to cancel the lower-order terms in the bias. Similarly, by using the same optimization problem we can define a weighted ensemble estimator $\tilde{\mathbf{G}}_{w_0,ij}$ of G_{ij} that achieves the parametric rate when $h(l) = lN^{-1/4}$ which results in $\psi_m(l) = l^m$ for $m \in J = \{1, 2\}$. These estimators correspond to the ODin1

estimators defined in [86].

An ODin2 estimator of $G(p;p')$ can be derived using (8.6). Let $\delta > 0$, assume that $s \geq (d+\delta)/2$, and let $h(l) = lN^{-1/(d+\delta)}$. This results in the function $\psi_{1,m,q}(l) = l^{m-dq}$ for $m \in \{0,\ldots,(d+\delta)/2\}$ and $q \in \{0,\ldots,(d+\delta)/\delta\}$ with the restriction that $m+q \neq 0$. Additionally we have $\psi_{2,m,q}(l) = l^{m-2q}$ for $m \in \{0,\ldots,(d+\delta)/2\}$ and $q \in \{1,\ldots,(d+\delta)/(2(d+\delta-2))\}$. These functions correspond to the lower order terms in the bias. Then using (8.7) with these functions results in a weight vector w_0 such that $\tilde{\mathbf{G}}_{w_0}$ achieves the parametric rate as long as $s \geq (d+\delta)/2$. Then since δ is arbitrary, we can achieve the parametric rate for $s > d/2$.

We conclude this section by giving a CLT. This theorem provides justification for performing structural hypothesis testing with the estimators $\tilde{\mathbf{G}}_{w_0}$ and $\tilde{\mathbf{G}}_{w_0,ij}$. The proof uses an application of Slutsky's Theorem preceded by the Efron-Stein inequality that is similar to the proof of the CLT of the divergence ensemble estimators in [86]. The extension of the CLT in [86] to $\tilde{\mathbf{G}}_w$ is analogous to the extension required in the proof of the variance results in Theorem VIII.2.

Theorem VIII.3. *Assume that $h = o(1)$ and $Nh^d \to \infty$. If $\mathbf{S}$ is a standard normal random variable, L is fixed, and g is Lipschitz in both arguments, then*

$$\Pr\left(\left(\tilde{\mathbf{G}}_w - \mathbb{E}\left[\tilde{\mathbf{G}}_w\right]\right)/\sqrt{\mathbb{V}\left[\tilde{\mathbf{G}}_w\right]} \leq t\right) \to \Pr(\mathbf{S} \leq t).$$

8.4 Experiments

We perform multiple experiments to demonstrate the utility of our proposed methods for structure learning of a GM with $d = 3$ nodes whose structure is a nonlinear Markov chain from nodes $i = 1$ to $i = 2$ to $i = 3$. That is, out of a possible 6 edges in a complete graph, only the node pairs $(1,2)$ and $(2,3)$ are connected by edges. In all experiments, $\mathbf{X}^{(1)} \sim \text{Unif}(-0.5, 0.5)$, $\mathbf{N}^{(i)} \sim \mathcal{N}(0, 0.5)$, and $\mathbf{N}^{(1)}$ and $\mathbf{N}^{(2)}$ are

independent. We have $N = 500$ i.i.d. samples from $\mathbf{X}^{(1)}$ and choose an ensemble of bandwidth parameters with $L = 50$ based on the guidelines in [86]. To better control the variance, we calculate the weight w_0 using the relaxed version of (8.7) given in [86]. We compare the results of the ensemble estimators ODin1 and ODin2 ($\delta = 1$ in the latter) to the simple plug-in KDE estimator. All p-values are constructed by applying Theorem VIII.3 where the mean and variance of the estimators are estimated via bootstrapping. In addition, we studentize the data at each node by dividing by the sample standard deviation as is commonly done in entropic machine learning. This introduces some dependency between the nodes that decreases as $O(1/N)$. This studentization has the effect of reducing the dependence of the MI on the marginal distributions and stabilizing the MI estimates. We estimate the Rényi-α integral for Rényi MI with $\alpha = 0.5$; i.e. $g(u) = u^{\alpha}$. Thus if the ratio of densities within (8.2) or (8.3) is 1, the Rényi-α integral is also 1.

In the first type of experiments, we vary the signal-to-noise ratio (SNR) of a Markov chain by varying the parameter a and setting

$$\begin{aligned} \mathbf{X}^{(2)} &= \left(\mathbf{X}^{(1)}\right)^2 + a\mathbf{N}^{(1)}, \\ \mathbf{X}^{(3)} &= \left(\mathbf{X}^{(2)}\right)^2 + a\mathbf{N}^{(2)}. \end{aligned} \tag{8.8}$$

In the second type of experiments, we create a cycle within the graph by setting

$$\begin{aligned} \mathbf{X}^{(2)} &= \left(\mathbf{X}^{(1)}\right)^2 + a\mathbf{N}^{(1)}, \\ \mathbf{X}^{(3)} &= \left(\mathbf{X}^{(2)}\right)^2 + b\mathbf{X}^{(1)} + a\mathbf{N}^{(2)}. \end{aligned} \tag{8.9}$$

We either fix b and vary a or vice versa.

We first use hypothesis testing on the estimated pairwise MI to learn the structure in (8.8). We do this by testing the null hypotheses that each pairwise Rényi-α integral is equal to 1. We do not use the ODin2 estimator in this experiment as there is no

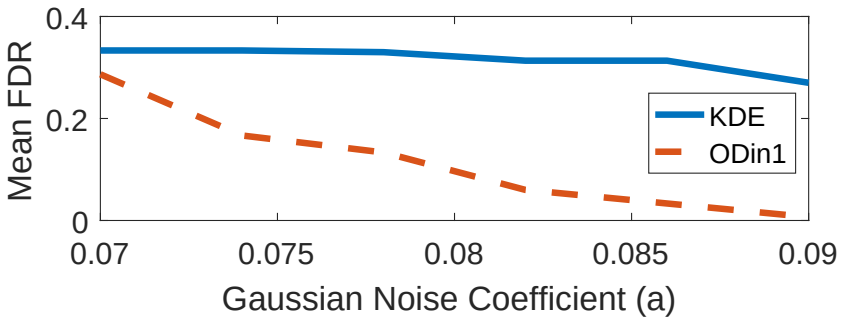


Figure 8.1: The mean FDR from 100 trials as a function of a when estimating the MI between all pairs of RVs for (8.8) with significance level $\gamma = 0.1$. The dependence between $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(3)}$ decreases as the noise increases resulting in lower mean FDR.

theoretical gain in MSE over ODin1 for pairwise MI estimation. Figure 8.1 plots the mean FDR from 100 trials as a function of a under this setting with significance level $\gamma = 0.1$. In ths case, the FDR is either 0 (no false discoveries) or 1/3 (one false discovery). Thus the mean FDR provides an indicator for the number of trials where a false discovery occurs. Figure 8.1 shows that the mean FDR decreases slowly for the KDE estimator and rapidly for the ODin1 estimator as the noise increases. Since $\mathbf{X}^{(3)}$ is a function of $\mathbf{X}^{(2)}$ which is a function of $\mathbf{X}^{(1)}$, then $G_{13} \neq 1$. However, as the noise increases, the relative dependence of $\mathbf{X}^{(3)}$ on $\mathbf{X}^{(1)}$ decreases and thus G_{13} approaches 1. The ODin1 estimator tracks this approach better as the corresponding FDR decreases at a faster rate compared to the KDE estimator.

In the next set of experiments, the CL algorithm estimates the tree structure in (8.8) and we test the hypothesis that $G(p; p') = 1$ to determine if the output of the CL algorithm gives the correct structure. The resulting mean p-value with error bars at the 20th and 80th percentiles from 90 trials is given in Figure 8.2. High p-values indicate that both the CL algorithm performs well and that $G(p; p')$ is not statistically different from 1. The ODin1 estimator generally has higher values than the ODin2 and KDE estimators which indicates better performance.

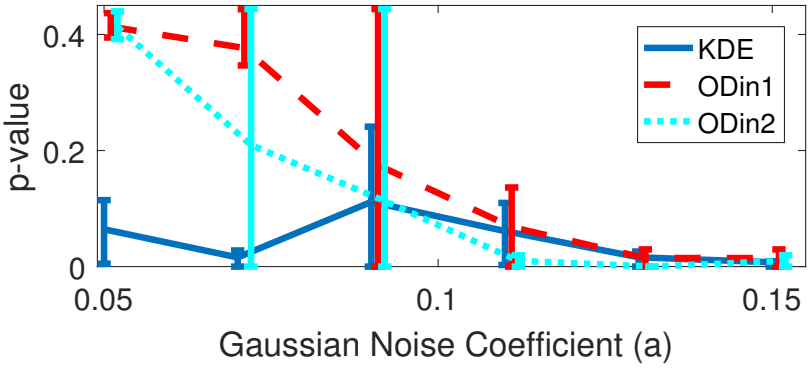


Figure 8.2: The average p-value with error bars at the 20th and 80th percentiles from 90 trials for the hypothesis test that $G(p;p') = 1$ after running the CL algorithm for (8.8). The graphs are offset horizontally for better visualization. Higher noise levels lead to higher error rates in the CL algorithm and thus lower p-values.

The final set of experiments focuses on (8.9). In this case, the CL tree does not include the edge between $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(3)}$ and we report the p-values for the hypothesis that $G\left(p;p'\right) = 1$ when varying either a or b. The mean p-value with error bars at the 20th and 80th percentiles from 100 trials are given in Figure 8.3. In the top figure, we fix $b = 0.5$ and vary the noise parameter a while in the bottom figure we fix $a = 0.05$ and vary b. Thus the true structure does not match the CL tree and low p-values are desired. For low noise in the top figure (fixed dependency coefficient), the ODin estimators perform better than the KDE estimator and have less variability. In the bottom figure (fixed noise), the ODin1 estimator generally outperforms the other estimators.

8.5 Conclusion

We derived the convergence rates for a kernel density plug-in estimator of MI functionals and proposed nonparametric ensemble estimators with a CLT that achieve the parametric rate when the densities are sufficiently smooth. We proposed two approaches for hypothesis testing based on the CLT to learn the structure of the data.

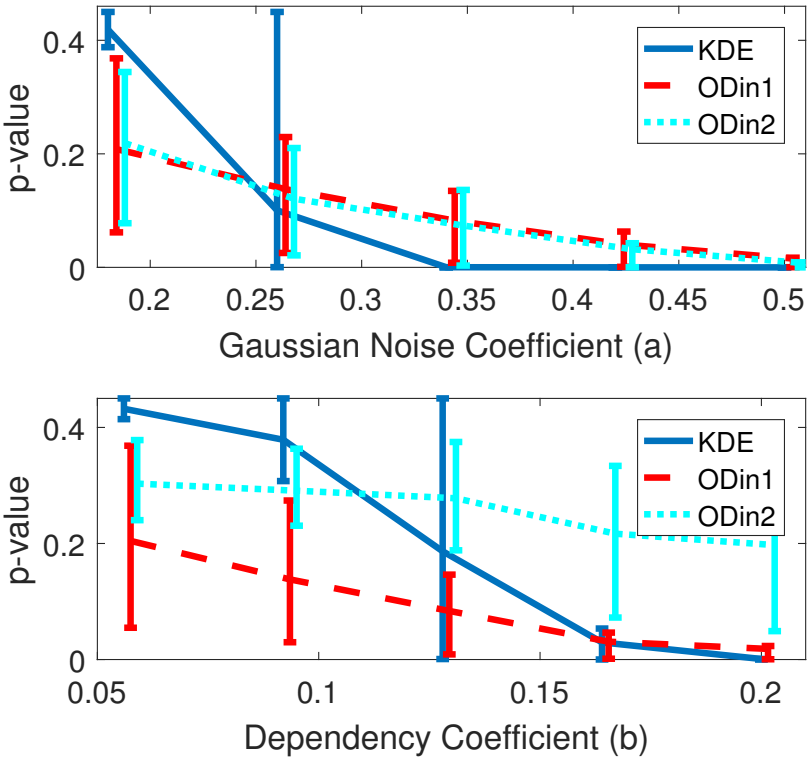


Figure 8.3: The mean p-value with error bars at the 20th and 80th percentiles from 100 trials for the hypothesis test that $G(p;p') = 1$ for (8.9) when the CL tree does not give the correct structure. Top: $b = 0.5$ and a varies. Bottom: $a = 0.05$ and b varies. The graphs are offset horizontally for better visualization. Low p-values indicate better performance. The ODin1 estimator generally matches or outperforms the other estimators, especially in the lower noise cases.

The experiments demonstrated the utility of these approaches in structure learning and demonstrated the improvement of ensemble methods over the plug-in method for a low dimensional example.

CHAPTER IX

Future Work

In this chapter we discuss the future research on some of the problems and challenges related to the topics of the previous chapters. In summary, the suggested future research can be grouped into three categories: Progress in hash-based estimation of divergence and mutual information, progress in estimation of Bayes error rate, and extensive analysis of deep neural networks using information theory. These research lines are discussed in details in the following sections.

9.1 Hash-based Estimation of Information Measures

In chapters VI and VII we proposed hash-based estimators of divergence and mutual inforamation that can achieve the parametric MSE rate of $O(1/N)$ in only linear time complexity. There are several open problems for possible future work. First, the performance of the estimators are sensitive to the hyper parameters such as the bandwidth of the hash-buckets, ϵ. Although we have derived optimal choices of ϵ in terms of the $\Theta()$ notation, choosing the right coefficients in practice is still an open question. In chapter IV we proposed the Chebyshev polynomial approximation approach for ensemble estimation. It would be an interesting idea to apply the Chebyshev weight assigning method for the hash-based estimators and investigate the stability of the results with respect to the hyper-parameters. Another interesting

direction would be to investigate other hashing schemes instead of the simple floor function. A promising hashing approach that fits the proposed estimation methods in this thesis would be a supervised hashing, where the hash function depends on the dataset. Specially for the image type of datasets, supervised hashing schemes, such as autoencoders, could result in better estimates of divergence and mutual information.

9.2 Estimation of Bayes Error Rate and Applications

In chapter IV we proposed an estimator of the Bayes error rate with optimal convergence rate based on ϵ-ball density estimator. Recently training deep neural networks using information theoretic loss functions has found much attraction in the machine learning and deep learning community [1]. Similarly, one could possibly use a loss function based on Bayes error rate. In other words, we can define the loss function as the Bayes error rate of the last layer of the network, and train the network by minimizing the proposed loss function. A problem with defining the loss function based on Bayes error rate is that the proposed Bayes error rate using ϵ-ball is not differentialble. Thus we cannot directly use this estimator for training the network. A possible future work is to find a tight bound on the Bayes error rate such that its computation method is differentiable, and ca be used for training the network.

9.3 Analysis of Deep Neural Networks Using Information Theory

We have applied our hash-based estimator of mutual information (EDGE) to study the information bottleneck theory of deep learning, which was first represented by Shwartz-Ziv and Tishby [107]. Our experiments confirmed that the compression phenomenon happens for a wider range of activation functions such as ReLU and tanh. There are yet many open questions in deep learning that can probably be answered

by information theoretic tools. Some of these questions that we are currently studying are as follows. What information are compressed in each hidden layer? What is the relation between compression and generalization in DNNs? Can we propose an information theoretic cost function (information bottleneck) and implement a DNN based on it, using an efficient estimator of gradient of mutual information?

APPENDICES

APPENDIX A

A.1 Bias Proof of NNR Estimator

In this section we prove Theorem VII.2, which states a bound on the bias of NNR estimator. However, before obtaining the bias bound (VII.2), we need to prove some lemmas. We first begin with some definitions.

Let $\rho_k(x)$ be defined as the k-NN distance on the point x. We define the k-NN ball centered at x as

$$S_k(x) := \{y : d(x, y) \leq \rho_k(x)\}. \tag{A.1}$$

Let $\mathbf{V}_{k,N}(x)$ denote the volume of the k-NN ball with N samples. Set

$$\alpha_k(x) := \frac{\int_{S_k(x)\cap\mathcal{X}} dz}{\int_{S_k(x)} dz}. \tag{A.2}$$

Let $\mathcal{X}_\mathcal{I}$ and $\mathcal{X}_\mathcal{B}$ respectively denote the interior support and boundary of the support. For a point $x \in \mathcal{X}_I$ we have $\alpha_k = 1$, and for $x \in \mathcal{X}_\mathcal{B}$ we have $\alpha_k < 1$. Note that the definition of interior and boundary points depends on k and N.

We need the following lemmas to get a bound on the moments of k-NN distances.

Lemma A.1. *We have the following relation for any $t \in \mathbb{R}$ and for each point $x \in \mathcal{X}_\mathcal{I}$ with density $f(x)$:*

$$\mathbb{E}\left[\rho_k^t(x)\right] = \left(\frac{k}{c_d N f(x)}\right)^{t/d} + O\left(\frac{N^{-t/d}}{k}\right) + u(x)O\left(\left(\frac{k}{N}\right)^{t/d+2}\right) + o\left(\left(\frac{k}{N}\right)^{t/d+2}\right)$$
$$+O\left(\left(\frac{k}{N}\right)^{t/d}\mathcal{C}(k)\right), \tag{A.3}$$

where $u(x) = g'(f(x))h(x)$,and h is some bounded function of the density which is defined in [111].

Proof. We start with a result from [111] (A.25). Let $g : \mathbb{R}^+ \to \mathbb{R}$ be some arbitrary function, then we have the following relation

(A.4)
$$\mathbb{E}\left[g\left(\frac{k}{c_0 n \rho_k^d(x)}\right)\right] = g(f(x))g_1(k,N) + g_2(k,N) + g'(f(x))h(x)(k/N)^2 + o((k/N)^2) + O(\mathcal{C}(k)).$$

where g_1 and g_2 are bias correction functions which depend on g. We also have $\mathcal{C}(k) := exp(-3k^{1-\delta})$ for a fixed $\delta \in (2/3, 1)$. For example, if we set $k = (\log(N))^{1/(1-\delta)}$, then $O(\mathcal{C}(k)) = O(1/N^3)$. Note that this term is negligible compared to other bias terms in our work.

Now according to [111], if we set $g(x) = x^{-\beta}$, then we have $g_1(k,N) = \frac{\Gamma(k)}{\Gamma(k-\beta)(k-1)^\beta}$ and $g_2(k,N) = 0$, which yields

(A.5)
$$\mathbb{E}\left[\rho_k^t(x)\right] = f(x)^{-t/d}\frac{\Gamma(k)}{\Gamma(k-t/d)}c_0' N^{-t/d} + u(x)O((\frac{k}{N})^{t/d+2}) + o((\frac{k}{N})^{t/d+2}) + O((\frac{k}{N})^{t/d}\mathcal{C}(k)).$$

Finally, using the approximation $\frac{\Gamma(k)}{\Gamma(k-\beta)} = k^\beta + O(1/k)$ results in (A.3). □

Now for the case of a bounded support, we derive an upper bound on k-NN distances for the points at the boundary:

Lemma A.2. *For every point $x \in \mathcal{X}_{\mathcal{B}}$ and any $t \in \mathbb{R}$ we have*

$$\mathbb{E}\left[\rho_k^t(x)\right] = O\left((k/N)^{t/d}\right) + O\left(\mathcal{C}(k)\right). \tag{A.6}$$

Proof. Define $V_{k,N}(x) := \frac{k}{N\alpha_k(x)f(x)}$. Let $p(k,N)$ denote any positive function satisfying $p(k,N) = \Theta\left((k/N)^{2/d}\right) + \frac{\sqrt{6}}{k^{\delta/2}}$ for some $\delta > 0$. Further consider the event E_1 as

$$E_1 := \{|\frac{\mathbf{V}_{k,N}(X)}{V_{k,N}(X)} - 1| > p(k,N)\}, \tag{A.7}$$

and E_2 as its complementary event. By using (B.2) in [111] (Appendix B), we have

$$Pr\left(E_1\right) = O\left(\mathcal{C}(k)\right). \tag{A.8}$$

Moreover, we can simplify (A.7) as:

$$|c_d\rho_k^d(x) - \frac{k}{N\alpha_k(x)f(x)}| > \frac{kp(k,N)}{N\alpha_k(x)f(x)}. \tag{A.9}$$

Further we write $\mathbb{E}\left[\rho_k^\gamma(x)\right]$ as the sum of conditional expectations:

$$\begin{aligned}
\mathbb{E}\left[\rho_k^\gamma(x)\right] &= \mathbb{E}\left[\rho_k^\gamma(x)|E_1\right] Pr\left(E_1\right) + \mathbb{E}\left[\rho_k^\gamma(x)|E_2\right] Pr\left(E_2\right) \\
&= O\left(\mathcal{C}(k)\right) + \mathbb{E}\left[\rho_k^\gamma(x)|E_2\right]\left(1 - O\left(\mathcal{C}(k)\right)\right) \\
&= O\left((k/N)^{\gamma/d}\right) + O\left(\mathcal{C}(k)\right), \qquad \text{(A.10)}
\end{aligned}$$

where in the second line we have used (A.8) and also the fact that $\rho_k(x)$ is bounded from above because of the bounded support.

□

Lemma A.3. *Suppose that the density function $f(x)$ belongs to the γ-Hölder smoothness class. Then if $B(x,r)$ denotes the sphere with center x and radius $r = \rho_k(x)$, we have the following smoothness condition:*

$$\mathbb{E}_{\rho_k(x)}\left[\sup_{y\in B(x,\rho_k(x))}|f(y)-f(x)|\right] \leq \epsilon_{\gamma,k}, \tag{A.11}$$

where $\epsilon_{\gamma,k} := O\left((k/N)^{\gamma/d}\right) + O\left(\mathcal{C}(k)\right)$, and we have $\mathcal{C}(k) := exp(-3k^{1-\delta})$ for a fixed $\delta \in (2/3, 1)$.

Proof. From definition of Hölder smoothness, for every $y \in B(x, \rho_k(x))$ we have

$$|f(y)-f(x)| \leq G_f\|y-x\|^\gamma \leq G_f\rho_k^\gamma(x). \tag{A.12}$$

Using Lemmas A.1 and A.2, results in

$$\mathbb{E}_{\rho_k(x)}\left[\sup_{y\in B(x,\rho_k(x))}|f(y)-f(x)|\right] \leq \epsilon_{\gamma,k}, \tag{A.13}$$

where $O\left((k/N)^{\gamma/d}\right)+O\left(\mathcal{C}(k)\right)$. Note that all other terms in (A.3) are of higher order and can be ignored. □

Proof of Theorem VII.2:

Now we are at the stage to prove Theorem VII.2. Note that it is easier to work with $\widehat{J}_\alpha(\mathbf{X},\mathbf{Y})$ defined in (2.2), instead of $\widehat{D}_\alpha(\mathbf{X},\mathbf{Y})$. The following lemma provides the essential tool to make a relation between $\mathbb{B}\left(\widehat{D}\right)$ and $\mathbb{B}\left(\widehat{J}\right)$.

Lemma A.4. *Assume that $g(x) : \mathcal{X} \to \mathbb{R}$ is Lipschitz continuous with constant $H_g > 0$. If $\widehat{Z}$ is a RV estimating a constant value Z with the bias $\mathbb{B}[\widehat{Z}]$ and the variance $\mathbb{V}[\widehat{Z}]$, then the bias of $g(\widehat{Z})$ can be upper bounded by*

$$|\mathbb{E}\left[g(\widehat{Z})-g(Z)\right]| \leq H_g\left(\sqrt{\mathbb{V}\left[\widehat{Z}\right]}+|\mathbb{B}\left[\widehat{Z}\right]|\right). \tag{A.14}$$

Proof.

$$
\begin{aligned}
|\mathbb{E}\left[g\left(\widehat{Z}\right)-g(Z)\right]| &\leq |\mathbb{E}\left[g\left(\widehat{Z}\right)-g\left(\mathbb{E}\left[\widehat{Z}\right]\right)\right]| + |\mathbb{E}\left[g\left(\mathbb{E}\left[\widehat{Z}\right]\right)-g(Z)\right]| \\
&\leq \mathbb{E}\left[|g\left(\widehat{Z}\right)-g\left(\mathbb{E}\left[\widehat{Z}\right]\right)|\right] + H_g|\mathbb{E}\left[\widehat{Z}\right]-Z| \\
&\leq H_g\mathbb{E}\left[|\widehat{Z}-\mathbb{E}\left[\widehat{Z}\right]|\right] + H_g|\mathbb{E}\left[\widehat{Z}\right]-Z| \\
&\leq H_g\left(\sqrt{\mathbb{V}\left[\widehat{Z}\right]}+|\mathbb{B}\left[\widehat{Z}\right]|\right). \qquad \text{(A.15)}
\end{aligned}
$$

In the second line we have used triangle inequality for the first term, and Lipschitz condition for the second term. Again in the third line, we have applied Lipschitz condition for the first term, and finally in the forth line we have used CauchySchwarz inequality.

□

An immediate consequence of this lemma is

$$
|\mathbb{B}\left[\widehat{D}_\alpha(\mathbf{X},\mathbf{Y})\right]| \leq C|\mathbb{B}\left[\widehat{J}_\alpha(\mathbf{X},\mathbf{Y})\right] + \sqrt{\mathbb{V}\left[\widehat{J}_\alpha(\mathbf{X},\mathbf{Y})\right]}|, \qquad \text{(A.16)}
$$

where C is a constant.

From theorem VII.3, $\mathbb{V}\left[\widehat{J}_\alpha(\mathbf{X},\mathbf{Y})\right] = O(1/N)$, so we only need to bound $\mathbb{B}\left[\widehat{J}_\alpha(\mathbf{X},\mathbf{Y})\right]$. If $\eta := M/N$, we have:

$$
\begin{aligned}
\mathbb{E}\left[\widehat{J}_\alpha(\mathbf{X},\mathbf{Y})\right] &= \frac{\eta^\alpha}{M}\mathbb{E}\left[\sum_{i=1}^{M}\left(\frac{N_i}{M_i+1}\right)^\alpha\right] \\
&= \eta^\alpha \mathbb{E}_{Y_1\sim f_Y(x)}\mathbb{E}\left[\left(\frac{N_1}{M_1+1}\right)^\alpha \middle| Y_1\right]. \qquad \text{(A.17)}
\end{aligned}
$$

Now note that N_1 and M_1 are not independent since $N_1 + M_1 = k$. We use the Poissonizing technique [8][55] and assume that $N_1 + M_1 = K$, where K is a Poisson random variable with mean k. We represent the Poissonized variant of $\widehat{J}_\alpha(\mathbf{X},\mathbf{Y})$ by $\overline{J}_\alpha(\mathbf{X},\mathbf{Y})$, and we will show that $\mathbb{E}\left[\widehat{J}_\alpha(\mathbf{X},\mathbf{Y})\right] = \mathbb{E}\left[\overline{J}_\alpha(\mathbf{X},\mathbf{Y})\right] + O(1/k)$. We first compute $Pr\left(Q_i(Y_1)\in\mathbf{X}\right)$ and $Pr\left(Q_i(Y_1)\in\mathbf{Y}\right)$ as follows:

Lemma A.5. *Let $\eta := M/N$ and $k < \min\{N, M\}$. Then for every $i \leq k$, the probability that the point $Q_i(Y_1)$ respectively belongs to the sets $\mathbf{X}$ and $\mathbf{Y}$ is equal to*

$$\begin{aligned} \Pr\left(Q_i(Y_1) \in \mathbf{X}\right) &= \frac{f_X(Y_1)}{f_X(Y_1) + \eta f_Y(Y_1)} + O(\epsilon_{\gamma,k}) \\ \Pr\left(Q_i(Y_1) \in \mathbf{Y}\right) &= \frac{\eta f_Y(Y_1)}{f_X(Y_1) + \eta f_Y(Y_1)} + O(\epsilon_{\gamma,k}), \end{aligned} \tag{A.18}$$

where as defined before, $\epsilon_{\gamma,k} := O\left((k/N)^{\gamma/d}\right) + O\left(\mathcal{C}(k)\right)$.

Proof. Here we prove a more general statement:

Let for any point $y \in \mathcal{X}$ define $\xi_1(y) := f_X(y) - f_X(Y_1)$ and $\xi_2(y) := f_Y(y) - f_Y(Y_1)$. Then $\Pr\left(Q_k(Y_1) \in \mathbf{X}\right)$ can be derived as

$$\Pr\left(Q_k(Y_1) \in X\right) = \frac{f_X(Y_1)}{f_X(Y_1) + \eta f_Y(Y_1)} + \tau_1(Y_1) + \tau_2(Y_1), \tag{A.19}$$

where $\tau_1(Y_1)$ and $\tau_2(Y_1)$ are defined as

$$\begin{aligned} \tau_1(Y_1) &:= \left(f_X(Y_1) + \eta f_Y(Y_1)\right)^{-1} \mathbb{E}_{y \sim f_{Q_k(Y_1)}}\left(\xi_1(y)\right) \\ \tau_2(Y_1) &:= \mathbb{E}_{y \sim f_{Q_k(Y_1)}}\left[\left(\frac{f_X(Y_1)}{f_X(Y_1) + \eta f_Y(Y_1)} + \frac{\xi_1(y)}{f_X(Y_1) + \eta f_Y(Y_1)}\right)\mathcal{U}\left(\frac{\xi_1(y) + \eta\xi_2(y)}{f_X(Y_1) + \eta f_Y(Y_1)}\right)\right], \end{aligned} \tag{A.20}$$

and $\mathcal{U}(x) := 1 + \sum_{i=1}^{\infty}(-1)^i (x)^i$.

To prove this, let $B(Q_k(Y_1), \epsilon)$ be the sphere with the center $Q_k(Y_1)$ (the k-NN point of Y_1) and some small radius $\epsilon > 0$. Also let E_X and E_Z denote the following events:

$$\begin{aligned} E_X &:= \{\exists x \in X \mid x \in B(Q_k(Y_1), \epsilon)\}, \\ E_Z &:= \{\exists x \in Z \mid x \in B(Q_k(Y_1), \epsilon)\}. \end{aligned} \tag{A.21}$$

Let use the notation $\Pr(E_X(y))$ to denote $\Pr(E_X|Q_k(Y_1) = y)$.

Suppose $f_{Q_k(Y_1)}$ be the density function of the RV $Q_k(Y_1)$. Then $Pr(Q_k(Y_1) \in \mathbf{X})$ can be written as:

$$Pr(Q_k(Y_1) \in \mathbf{X}) = \int_{\mathcal{X}} f_{Q_k(Y_1)}(y) Pr(Q_k(Y_1) \in \mathbf{X}|Q_k(Y_1) = y), \tag{A.22}$$

where $Pr(Q_k(Y_1) \in \mathbf{X}|Q_k(Y_1) = y)$ can be formulated using $E_X(y)$ and $E_Y(y)$ as

$$Pr(Q_k(Y_1) \in \mathbf{X}|Q_k(Y_1) = y) = \frac{\Pr(E_X(y))}{\Pr(E_Z(y))}. \tag{A.23}$$

Let $P_f(y, \epsilon)$ denote the probability of the sphere $B(y, \epsilon)$ with density f. Then there exist a function real function $\Delta_1(\epsilon)$ such that for any $\epsilon > 0$ we have

$$P_f(y, \epsilon) = f(y) c_d \epsilon^d + \Delta_1(\epsilon), \tag{A.24}$$

where c_d is volume of the unit ball in dimension d. From definition of the density function we have

$$f(y) = \lim_{\epsilon \to 0} \frac{P_f(y, \epsilon)}{c_d \epsilon^d}. \tag{A.25}$$

So, from (A.24) and (A.25) we get $\lim_{\epsilon \to 0} \Delta_1(\epsilon)/\epsilon^d = 0$.

Now we compute $Pr(E_X(y))$ as

$$\begin{aligned} \Pr(E_X(y)) &= 1 - (1 - P_{f_X}(y, \epsilon))^N \\ &= N P_{f_X}(y, \epsilon) + \Delta_1(\epsilon) + \sum_{i=2}^{N} (-1)^i \binom{N}{i} P_{f_X}(y, \epsilon)^i \\ &= N c_d f_X(y) \epsilon^d + \Delta_2(\epsilon), \end{aligned} \tag{A.26}$$

where $\Delta_2(\epsilon) := \Delta_1(\epsilon) + \sum_{i=2}^{N}(-1)^i \binom{N}{i} P_{f_X}(y,\epsilon)^i$. Note that $\lim_{\epsilon\to 0} \Delta_2(\epsilon)/\epsilon^d = 0$.

Similarly, for $Pr(E_z)$ we can prove that

$$\Pr(E_z) = N c_d f_X(y)\,\epsilon^d + M c_d f_Y(y)\,\epsilon^d + \Delta_2'(\epsilon), \tag{A.27}$$

where $\Delta_2'(\epsilon)$ is a function satisfying $\lim_{\epsilon\to 0} \Delta_2'(\epsilon)/\epsilon^d = 0$.

From (A.23), and considering the fact that (A.26) and (A.27) hold true for any $\epsilon > 0$, we get

$$Pr\left(Q_k(Y_1) \in \mathbf{X} | Q_k(Y_1) = y\right) = \lim_{\epsilon\to 0} \frac{\Pr\left(E_X(y)\right)}{\Pr\left(E_Z(y)\right)} = \frac{f_X(y)}{f_X(y) + \eta f_Y(y)}, \tag{A.28}$$

where $\eta = M/N$. Considering the Taylor expansion of $\frac{A+a}{B+b}$ for any real number A, B, a, b such that $a \ll A$ and $b \ll B$, we have

(A.29)

$$\frac{A+a}{B+b} = \left(\frac{A}{B} + \frac{a}{B}\right)\left(1 + \sum_{i=1}^{\infty}(-1)^i \left(\frac{b}{B}\right)^i\right) = \frac{A}{B} + \frac{a}{B} + \left(\frac{A}{B} + \frac{a}{B}\right)\mathcal{U}\left(\frac{b}{B}\right),$$

where $\mathcal{U}(x) := \sum_{i=1}^{\infty}(-1)^i (x)^i$. Consequently, by using this fact and relation (A.28) we have

$$\begin{aligned} Pr\left(Q_k(Y_1) \in \mathbf{X}\right) &= \int_{\mathcal{X}} f_{Q_k(Y_1)}(y) \frac{f_X(y)}{f_X(y) + \eta f_Y(y)} dy \\ &= \frac{f_X(Y_1)}{f_X(Y_1) + \eta f_Y(Y_1)} + \tau_1(Y_1) + \tau_2(Y_1), \end{aligned} \tag{A.30}$$

and $\tau_1(Y_1)$ and $\tau_2(Y_1)$ are given by

$$\tau_1(Y_1) = \left(f_X(Y_1) + \eta f_Y(Y_1)\right)^{-1} \mathbb{E}_{y\sim f_{Q_k(Y_1)}}\left[\xi_1(y)\right]$$

(A.31)

$$\tau_2(Y_1) = \mathbb{E}_{y\sim f_{Q_k(Y_1)}}\left[\left(\frac{f_X(Y_1)}{f_X(Y_1)+\eta f_Y(Y_1)} + \frac{\xi_1(y)}{f_X(Y_1)+\eta f_Y(Y_1)}\right)\mathcal{U}\left(\frac{\xi_1(y)+\eta\xi_2(y)}{f_X(Y_1)+\eta f_Y(Y_1)}\right)\right].$$

Now from Lemma A.3 we can simply write $\tau_1(Y_1) = O\left(()\,\epsilon_{\gamma,k}\right)$ and $\tau_2(Y_1) = O\left(()\,\epsilon_{\gamma,k}\right)$ which results in:

$$\Pr\left(Q_k(Y_1) \in \mathbf{X}\right) = \frac{f_X(Y_1)}{f_X(Y_1)+\eta f_Y(Y_1)} + O(\epsilon_{\gamma,k}). \tag{A.32}$$

Similarly the second line in (A.18) can be proven in the same way, therefore is omitted. □

Now by partitioning theorem for a Poisson random variable with Bernoulli trials of probabilities $\Pr\left(Q_i(Y_1) \in \mathbf{X}\right)$ and $\Pr\left(Q_i(Y_1) \in \mathbf{Y}\right)$, we argue that N_1 and M_1 are two independent Poisson RVs (conditioned on Y_1). Using the conditional independence of N_1 and M_1 we have

$$\mathbb{E}\left[\frac{N_1}{M_1+1}\middle| Y_1\right] = \mathbb{E}\left[N_1|Y_1\right]\mathbb{E}\left[(M_1+1)^{-1}\middle|Y_1\right]. \tag{A.33}$$

$\mathbb{E}\left[N_1|Y_1\right]$ can be simplified as

$$\begin{aligned}\mathbb{E}\left[N_1|Y_1\right] &= \sum_{i=1}^{k}\Pr\left(Q_i(Y_1)\in X\right)\\ &= k\frac{f_X(Y_1)}{f_X(Y_1)+\eta f_Y(Y_1)} + O(k\epsilon_{\gamma,k}).\end{aligned} \tag{A.34}$$

Also similarly,

$$\mathbb{E}\left[M_1|Y_1\right] = \frac{k\eta f_Y(Y_1)}{f_X(Y_1)+\eta f_Y(Y_1)} + O(k\epsilon_{\gamma,k}).$$

Lemma A.6. *If U is a Poisson random variable with the mean $\lambda > 1$, then*

$$\mathbb{E}\left[(U+1)^{-1}\right] = \frac{1}{\lambda}\left(1 - e^{-\lambda}\right). \tag{A.35}$$

Proof. From definition of Poisson RV, we can write

$$\mathbb{E}\left[(U+1)^{-1}\right] = \sum_{k=0}^{\infty} \frac{1}{k+1}\left(\frac{\lambda^k e^{-\lambda}}{k!}\right) = \frac{1}{\lambda}\sum_{k=0}^{\infty}\frac{\lambda^{k+1}e^{-\lambda}}{(k+1)!} = \frac{1}{\lambda}\left(1 - e^{-\lambda}\right). \tag{A.36}$$

□

Using this lemma for M_1 yields

$$\begin{aligned}&\mathbb{E}\left[(M_1+1)^{-1}\middle|Y_1\right]\\ &\quad= k^{-1}\left[\frac{\eta f_Y(Y_1)}{f_X(Y_1)+\eta f_Y(Y_1)} + O(\epsilon_{\gamma,k})\right]^{-1} + O\left(\frac{e^{-vk}}{k}\right),\end{aligned} \tag{A.37}$$

here v is some positive constant. Therefore, (A.33) becomes

$$\mathbb{E}\left[\frac{N_1}{M_1+1}\middle|Y_1\right] = \frac{f_X(Y_1)}{\eta f_Y(Y_1)} + O(\epsilon_{\gamma,k}) + O\left(e^{-vk}\right). \tag{A.38}$$

Using lemma E.9 and theorem VII.3, we obtain

$$\begin{aligned}\mathbb{E}\left[\left(\frac{N_1}{M_1+1}\right)^{\alpha}\middle|Y_1\right] &= \eta^{-\alpha}\left(\frac{f_X(Y_1)}{f_Y(Y_1)}\right)^{\alpha} +\\ &+ O(\epsilon_{\gamma,k}) + O\left(e^{-vk}\right) + O(N^{-\frac{1}{2}}).\end{aligned} \tag{A.39}$$

By applying an equation similar to (A.17), we get

$$\mathbb{B}\left[\overline{J}_\alpha(\mathbf{X},\mathbf{Y})\right] = O(\epsilon_{\gamma,k}) + O\left(e^{-vk}\right) + O(N^{-\frac{1}{2}}). \tag{A.40}$$

Lemma A.7. *De-Poissonizing $\overline{J}_\alpha(\mathbf{X},\mathbf{Y})$ adds extra error of $O(\frac{1}{k})$:*

$$\mathbb{E}\left[\widehat{J}_\alpha(\mathbf{X},\mathbf{Y})\right] = \mathbb{E}\left[\overline{J}_\alpha(\mathbf{X},\mathbf{Y})\right] + O(1/k). \tag{A.41}$$

Proof. We use the following theorem from [55] to de-possonize the estimator.

Theorem A.8. *Assume a sequence a_n is given, and its poisson transform is $F(Z)$:*

$$F(z) = \sum_{n\geq 0} a_n \frac{z^n}{n!} e^{-z}. \tag{A.42}$$

Consider a linear cone $S_\theta = \{z : |\arg(z)| \leq \theta, \theta < \pi/2\}$. Let the following conditions hold for some constants $R > 0$, $\alpha < 1$ and $\beta \in \mathbb{R}$:

- *For $z \in S_\theta$,*

$$|z| > R \Rightarrow |F(z)| = O\left(z^\beta\right). \tag{A.43}$$

- *For $z \notin S_\theta$,*

$$|z| > R \Rightarrow |F(z)e^z| = O\left(e^{\alpha|z|}\right). \tag{A.44}$$

Then we have the following expansion that holds for every fixed m:

$$a_n = \sum_{i=0}^{m} \sum_{j=0}^{i+m} b_{ij} n^i F^{(j)}(n) + O(n^{\beta-m-1/2}), \tag{A.45}$$

where $\sum_{ij} b_{ij} x^i y^j = \exp\left(x \log(1+y) - xy\right)$.

Let $\widehat{J}_{\alpha,k}(\mathbf{X},\mathbf{Y})$ and $\overline{J}_{\alpha,k}(\mathbf{X},\mathbf{Y})$ respectively represent the RVs $\widehat{J}_\alpha(\mathbf{X},\mathbf{Y})$ and $\overline{J}_\alpha(\mathbf{X},\mathbf{Y})$ with the parameter k.

Using the dePoissonization theorem, we take $a_k := \mathbb{E}\left[\widehat{J}_{\alpha,k}(\mathbf{X},\mathbf{Y})\right]$ and $F(k) := \mathbb{E}\left[\overline{J}_{\alpha,k}(\mathbf{X},\mathbf{Y})\right]$. Since we are only interested in the values of k, for which $\lim_{N\to\infty} \frac{k}{N} = 0$, we can assume $F(z) = O(1)$. So, both the first and second conditions of the Theorem A.8 are satisfied. Then from (A.45)*, for $m = 1$:*

$$\mathbb{E}\left[\widehat{J}_{\alpha,k}(\mathbf{X},\mathbf{Y})\right] = \mathbb{E}\left[\overline{J}_{\alpha,k}(\mathbf{X},\mathbf{Y})\right] + O\left(\frac{1}{k}\right) + \frac{1}{2}O\left(\frac{1}{k^2}\right) + O\left(k^{-3/2}\right), \tag{A.46}$$

where $\beta = 0$.

□

At this point the bias proof of NNR estimator for Rényi divergence is complete, and since $O\left(e^{-vk}\right)$ and $O\left(N^{-\frac{1}{2}}\right)$ are of higher order compared to $O\left(\epsilon_{\gamma,k}\right)$, we obtain the final bias rate in (6.5). The bias proof of NNR estimator for f-divergence is similar, and by using the lemma E.9 with a Lipschitz continuous function g in equation (A.38), we can follow the same steps to prove the bias bound.

APPENDIX B

B.1 Proof of Theorem III.4

In this section, we prove the subadditivity and superadditivity for the mean of FR test statistic. For this, first we need to illustrate the following lemma.

Lemma B.1. *Let $\{Q_i\}_{i=1}^{l^d}$ be a uniform partition of $[0,1]^d$ into l^d subcubes Q_i with edges parallel to the coordinate axes having edge lengths l^{-1} and volumes l^{-d}. Let D_{ij} be the set of edges of MST graph between Q_i and Q_j with cardinality $|D_{ij}|$, then for $|D|$ defined as the sum of $|D_{ij}|$ for all $i, j = 1, \ldots, l^d$, $i \neq j$, we have $\mathbb{E}|D| = O(l^{d-1}\, n^{1/d})$, or more explicitly*

$$\mathbb{E}[|D|] \leq C' l^{d-1} n^{1/d} + O(l^{d-1} n^{(1/d)-s}), \tag{B.1}$$

where $\eta > 0$ is the Hölder smoothness parameter and

$$s = \frac{(1-1/d)\eta}{d\,((1-1/d)\eta + 1)}.$$

Here and in what follows, denote $\Xi_{MST}(\mathfrak{X}_n)$ the length of the shortest spanning tree on $\mathfrak{X}_n = \{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$, namely

$$\Xi_{MST}(\mathfrak{X}_n) := \min_T \sum_{e \in T} |e|,$$

where the minimum is over all spanning trees T of the vertex set $\mathfrak{X}_n$. Using the subadditivity relation for Ξ_{MST} in [125], with the uniform partition of $[0,1]^d$ into l^d subcubes Q_i with edges parallel to the coordinate axes having edge lengths l^{-1} and volumes l^{-d}, we have

$$\Xi_{MST}(\mathfrak{X}_n) \leq \sum_{i=1}^{l^d} \Xi_{MST}(\mathfrak{X}_n \cap Q_i) + C\, l^{d-1}, \tag{B.2}$$

where C is constant. Denote D the set of all edges of $MST\Big(\bigcup_{i=1}^{M} Q_i\Big)$ which intersect two different subcubes Q_i and Q_j with cardinality $|D|$. Let $|e_i|$ be the length of i-th edge in set D. We can write

$$\sum_{i\in|D|} |e_i| \leq Cl^{d-1} \text{ and } \mathbb{E}\sum_{i\in|D|} |e_i| \leq Cl^{d-1},$$

also we know that

$$\mathbb{E}\sum_{i\in|D|} |e_i| = \mathbb{E}_D \sum_{i\in|D|} \mathbb{E}\big[|e_i|\big|D\big]. \tag{B.3}$$

Note that using the result from ([51], Proposition 3), for some constants C_{i1} and C_{i2}, we have

$$\mathbb{E}|e_i| \leq C_{i1} n^{-1/d} + C_{i2} n^{-(1/d)-s}, \quad i \in |D|. \tag{B.4}$$

Now let $C_1 = \max_i\{C_{i1}\}$ and $C_2 = \max_i\{C_{i2}\}$, hence we can bound the expectation (B.3) as

$$\mathbb{E}|D|\ (C_1 n^{-1/d} + C_2(n^{-(1/d)-s})) \leq Cl^{d-1},$$

which implies

$$\mathbb{E}|D| \leq (C_1 n^{-1/d} + O(n^{-(1/d)-s}))$$

$$\leq C' l^{d-1} n^{1/d} + O(l^{d-1} n^{(1/d)-s}).$$

To aim toward the goal (3.9), we partition $[0,1]^d$ into $M := l^d$ subcubes Q_i of side $1/l$. Recalling Lemma 2.1 in [115] we therefore have the set inclusion:

$$MST\Big(\bigcup_{i=1}^{M} Q_i\Big) \subset \bigcup_{i=1}^{M} MST(Q_i) \cup D, \tag{B.5}$$

where D is defined as in Lemma B.1. Let m_i and n_i be the number of sample $\{\mathbf{X}_1, \ldots, \mathbf{X}_m\}$ and $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$ respectively falling into the partition Q_i, such that $\sum_i m_i = m$ and $\sum_i n_i = n$. Introduce sets A and B as

$$A := MST\Big(\bigcup_{i=1}^{M} Q_i\Big), \quad B := \bigcup_{i=1}^{M} MST(Q_i).$$

Since set B has fewer edges than set A, thus (B.5) implies that the difference set of B and A contains at most $2|D|$ edges, where $|D|$ is the number of edges in D. On the other word

$$|A \Delta B| \leq |A - B| + |B - A| = |D| + |B - A|$$

$$= |D| + (|B| - |B \cap A| \leq |D| + (|A| - |B \cap A|) = 2|D|.$$

The number of edge linked nodes from different samples in set A is bounded by the number of edge linked nodes from different samples in set B plus $2|D|$:

$$\mathfrak{R}_{m,n}(\mathfrak{X}_m, \mathfrak{Y}_n) \leq \sum_{i=1}^{M} \mathfrak{R}_{m_i,n_i}\big((\mathfrak{X}_m, \mathfrak{Y}_n) \cap Q_i\big) + 2|D|. \tag{B.6}$$

Here $\mathfrak{R}_{m_i,n_i}$ stands with the number edge linked nodes from different samples in partition Q_i, M. Next, we address the reader to Lemma B.1, where it has been shown that there is a constant c such that $\mathbb{E}|D| \leq c\, l^{d-1}\, (m+n)^{1/d}$. This concludes the claimed assertion (3.9). Now to accomplish the proof, the lower bound term in (3.10) is obtained with similar methodology and the set inclusion:

$$\bigcup_{i=1}^{M} MST(Q_i) \subset MST\Big(\bigcup_{i=1}^{M} Q_i\Big) \cup D. \tag{B.7}$$

This completes the proof.

B.2 Proof of Theorem III.2

As many of continuous subadditive functionals on $[0,1]^d$, in the case of FR statistic there exist a dual superadditive functional $\mathfrak{R}^*_{m,n}$ based on dual MST, MST*, proposed in Definition III.5. Note that in MST* graph, the degrees of the corner points are bounded by c_d where only depends on dimension d, and is the bound for degree of every node in MST graph. The following properties hold true for dual FR test statistic, $\mathfrak{R}^*_{m,n}$:

Lemma B.2. *Given samples* $\mathfrak{X}_m = \{\mathbf{X}_1, \ldots, \mathbf{X}_m\}$ *and* $\mathfrak{Y}_n = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$*, the following inequalities hold true:*

(i) For constant c_d *which depends on* d*:*

$$\mathfrak{R}^*_{m,n}(\mathfrak{X}_m, \mathfrak{Y}_n) \leq \mathfrak{R}_{m,n}(\mathfrak{X}_m, \mathfrak{Y}_n) + c_d\, 2^d,$$

(B.8)

$$\mathfrak{R}_{m,n}(\mathfrak{X}_m, \mathfrak{Y}_n) \leq \mathfrak{R}^*_{m,n}(\mathfrak{X}_m, \mathfrak{Y}_n).$$

*(ii) (*Subadditivity on $\mathbb{E}[\mathfrak{R}^*_{m,n}]$ and Superadditivity*) Partition* $[0,1]^d$ *into* l^d *subcubes* Q_i *such that* m_i*,* n_i *be the number of sample* $\mathfrak{X}_m = \{\mathbf{X}_1, \ldots, \mathbf{X}_m\}$ *and* $\mathfrak{Y}_n = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$ *respectively falling into the partition* Q_i *with dual* $\mathfrak{R}^*_{m_i,n_i}$*. Then we have*

(B.9)

$$\mathbb{E}\Big[\mathfrak{R}^*_{m,n}(\mathfrak{X}_m, \mathfrak{Y}_n)\Big] \leq \sum_{i=1}^{l^d} \mathbb{E}\Big[\mathfrak{R}^*_{m_i,n_i}((\mathfrak{X}_m, \mathfrak{Y}_n) \cap Q_i)\Big] + c\, l^{d-1}\, (m+n)^{1/d},$$

$$\mathfrak{R}^*_{m,n}(\mathfrak{X}_m, \mathfrak{Y}_n) \geq \sum_{i=1}^{l^d} \mathfrak{R}^*_{m_i,n_i}((\mathfrak{X}_m, \mathfrak{Y}_n) \cap Q_i) - 2^d c_d l^d.$$

where c *is a constant.*

(i) Consider the nodes connected to the corner points. Since $\mathrm{MST}(\mathfrak{X}_m, \mathfrak{Y}_n)$ *and* $\mathrm{MST}^*(\mathfrak{X}_m, \mathfrak{Y}_n)$ *can only be different in the edges connected to these nodes, and in*

$\mathfrak{R}^*(\mathfrak{X}_m, \mathfrak{Y}_n)$ *we take all of the edges between these nodes and corner nodes into account, so we obviously have the second relation in* (B.8). *Also for the first inequality in* (B.8) *it is enough to say that the total number of edges connected to the corner nodes is upper bounded by* $2^d\, c_d$.

(ii) Let $|D^*|$ *be the set of edges of the* MST* *graph which intersect two different partitions. Since MST and* MST* *are only different in edges of points connected to the corners and edges crossing different partitions. Therefore* $|D^*| \leq |D|$. *By eliminating one edge in set* D *in worse scenario we would face with two possibilities: either the corresponding node is connected to the corner which is counted anyways or any other point in MST graph which wouldn't change the FR test statistic. This implies the following subadditivity relation:*

$$\mathfrak{R}^*_{m,n}(\mathfrak{X}_m, \mathfrak{Y}_n) - |D| \leq \sum_{i=1}^{l^d} \mathfrak{R}^*_{m_i,n_i}\big((\mathfrak{X}_m, \mathfrak{Y}_n) \cap Q_i\big).$$

Further from Lemma B.1, we know that there is a constant c *such that* $\mathbb{E}|D| \leq c\, l^{d-1}\, (m+n)^{1/d}$. *Hence the first inequality in (B.9) is obtained. Next consider* $|D^*_c|$ *which represents the total number of edges from both samples only connected to the all corners points in* MST* *graph. Therefore one can easily claim:*

$$\mathfrak{R}^*_{m,n}(\mathfrak{X}_m, \mathfrak{Y}_n) \geq \sum_{i=1}^{l^d} \mathfrak{R}^*_{m_i,n_i}\big((\mathfrak{X}_m, \mathfrak{Y}_n) \cap Q_i\big) - |D^*_c|.$$

Also we know that $|D^*_c| \leq 2^d l^d c_d$ *where* c_d *stands with the largest possible degree of any vertex. One can write*

$$\mathfrak{R}^*_{m,n}(\mathfrak{X}_m, \mathfrak{Y}_n) \geq \sum_{i=1}^{l^d} \mathfrak{R}^*_{m_i,n_i}\big((\mathfrak{X}_m, \mathfrak{Y}_n) \cap Q_i\big) - 2^d c_d l^d.$$

The following list of Lemmas B.3, B.4 and B.6 are inspired from [45] and are required to prove Theorem B.7. See the Supplementary Materials for their proofs.

Lemma B.3. *Let $g(\mathbf{x})$ be a density function with support $[0,1]^d$ and belong to the Hölder class $\Sigma(\eta, L)$, $0 < \eta \leq 1$, stated in Definition* **??**. *Also, assume that $P(\mathbf{x})$ is a η-Hölder smooth function, such that its absolute value is bounded from above by a constant. Define the quantized density function with parameter l and constants ϕ_i as*

$$\widehat{g}(\mathbf{x}) = \sum_{i=1}^{M} \phi_i \mathbf{1}\{\mathbf{x} \in Q_i\}, \quad \textit{where } \phi_i = l^d \int_{Q_i} g(\mathbf{x})\, \mathrm{d}\mathbf{x}. \tag{B.10}$$

Let $M = l^d$ and $Q_i = \{\mathbf{x}, \mathbf{x}_i : \|\mathbf{x} - \mathbf{x}_i\| < l^{-d}\}$. Then

$$\int \Big\| \big(g(\mathbf{x}) - \widehat{g}(\mathbf{x})\big) P(\mathbf{x}) \Big\| \, \mathrm{d}\mathbf{x} \leq O(l^{-d\eta}). \tag{B.11}$$

Lemma B.4. *Denote $\Delta(\mathbf{x}, \mathcal{S})$ the degree of vertex $\mathbf{x} \in \mathcal{S}$ in the MST over set $\mathcal{S}$ with the n number of vertices. For given function $P(\mathbf{x}, \mathbf{x})$, one obtains*

$$\int P(\mathbf{x}, \mathbf{x}) g(\mathbf{x}) \mathbb{E}[\Delta(\mathbf{x}, \mathcal{S})] \, \mathrm{d}\mathbf{x} = 2 \int P(\mathbf{x}, \mathbf{x}) g(\mathbf{x}) \, \mathrm{d}\mathbf{x} + \varsigma_\eta(l, n), \tag{B.12}$$

where for constant $\eta > 0$,

$$\varsigma_\eta(l, n) = \Big(O\big(l/n\big) - 2\, l^d/n \Big) \int g(\mathbf{x}) P(\mathbf{x}, \mathbf{x}) \, \mathrm{d}\mathbf{x} + O(l^{-d\eta}). \tag{B.13}$$

Lemma B.5. *Assume that for given k, $g_k(\mathbf{x})$ is a bounded function belong to $\Sigma(\eta, L)$. Let $P : \mathbb{R}^d \times \mathbb{R}^d \mapsto [0,1]$ be a symmetric, smooth, jointly measurable function, such that, given k, for almost every $\mathbf{x} \in \mathbb{R}^d$, $P(\mathbf{x}, .)$ is measurable with $\mathbf{x}$ a Lebesgue point of the function $g_k(.)P(\mathbf{x}, .)$. Assume that the first derivative P is bounded. For each k, let $\mathbf{Z}_1^k, \mathbf{Z}_2^k, \ldots, \mathbf{Z}_k^k$ be independent d-dimensional variable with common density function g_k. Set $\mathfrak{Z}_k = \{\mathbf{Z}_1^k, \mathbf{Z}_2^k \ldots, \mathbf{Z}_k^k\}$ and $\mathfrak{Z}_k^{\mathbf{x}} = \{\mathbf{x}, \mathbf{Z}_2^k, \mathbf{Z}_3^k \ldots, \mathbf{Z}_k^k\}$. Then*

$$\mathbb{E}\Big[\sum_{j=2}^{k} P(\mathbf{x}, \mathbf{Z}_j^k) \mathbf{1}\big\{(\mathbf{x}, \mathbf{Z}_j^k) \in MST(\mathfrak{Z}_k^{\mathbf{x}})\big\} \Big] = P(\mathbf{x}, \mathbf{x})\, \mathbb{E}\big[\Delta(\mathbf{x}, \mathfrak{Z}_k^{\mathbf{x}})\big] + \Big\{ O\big(k^{-\eta/d}\big) + O\big(k^{-1/d}\big) \Big\}. \tag{B.14}$$

Lemma B.6. *Consider the notations and assumptions in Lemma B.5. Then*

$$\left|k^{-1}\sum\sum_{1\le i<j\le k} P(\mathbf{Z}_i^k,\mathbf{Z}_j^k)\mathbf{1}\{(\mathbf{Z}_i^k,\mathbf{Z}_j^k)\in MST(\mathfrak{Z}_k)\}-\int_{\mathbb{R}^d} P(\mathbf{x},\mathbf{x})g_k(\mathbf{x})\,\mathrm{d}\mathbf{x}\right| \tag{B.15}$$
$$\le \varsigma_\eta(l,k)+O(k^{-\eta/d})+O(k^{-1/d}).$$

Here $MST(\mathcal{S})$ denotes the MST graph over nice and finite set $\mathcal{S}\subset\mathbb{R}^d$ and η is the smoothness Hölder parameter. Note that $\varsigma_\eta(l,k)$ is given as before in Lemma B.4 (B.13).

Theorem B.7. *Assume $\mathfrak{R}_{m,n}:=\mathfrak{R}(\mathfrak{X}_m,\mathfrak{Y}_n)$ denotes the FR test statistic and densities f_0 and f_1 belong to the Hölder class $\Sigma(\eta,L)$, $0<\eta\le 1$. Then the rate for the bias of the $\mathfrak{R}_{m,n}$ estimator for $d\ge 2$ is of the form:*

$$\left|\frac{\mathbb{E}\big[\mathfrak{R}_{m,n}\big]}{m+n}-2pq\int\frac{f_0(\mathbf{x})f_1(\mathbf{x})}{pf_0(\mathbf{x})+qf_1(\mathbf{x})}\,\mathrm{d}\mathbf{x}\right|\le O\big(l^d(m+n)^{-\eta/d}\big)+O(l^{-d\eta}). \tag{B.16}$$

The proof and a more explicit form for the bound on the RHS are given in Supplementary Materials.

Now, we are at the position to prove the assertion in (**??**). Without lose of generality assume that $(m+n)l^{-d}>1$. In the range $d\ge 2$ and $0<\eta\le 1$, we select l as a function of $m+n$ to be the sequence increasing in $m+n$ which minimizes the maximum of these rates:

$$l(m+n)=arg\ \min_l \max\Big\{l^d(m+n)^{-\eta/d},\ l^{-\eta d}\Big\}.$$

The solution $l=l(m+n)$ occurs when $l^d(m+n)^{-\eta/d}=l^{-\eta d}$, or equivalently $l=\lfloor(m+n)^{\eta/(d^2(\eta+1))}\rfloor$. Substitute this into l in the bound (B.16), the RHS expression in (**??**) for $d\ge 2$ is established.

B.3 Proof of Theorems III.3

To bound the variance we will apply one of the first concentration inequalities which was proved by Efron and Stein [33] and further was improved by Steele [114].

Lemma B.8. (The Efron-Stein Inequality) *Let* $\mathfrak{X}_m = \{\mathbf{X}_1, \ldots, \mathbf{X}_m\}$ *be a random vector on the space* $\mathcal{S}$. *Let* $\mathfrak{X}' = \{\mathbf{X}'_1, \ldots, \mathbf{X}'_m\}$ *be the copy of random vector* $\mathfrak{X}_m$. *Then if* $f : \mathcal{S} \times \cdots \times \mathcal{S} \to \mathbb{R}$, *we have*

$$\mathbb{V}\big[f(\mathfrak{X}_m)\big] \leq \frac{1}{2}\sum_{i=1}^{m}\mathbb{E}\Big[\big(f(\mathbf{X}_1, \ldots, \mathbf{X}_m) - f(\mathbf{X}_1, \ldots, \mathbf{X}'_i, \ldots, \mathbf{X}_m)\big)^2\Big]. \tag{B.17}$$

Consider two set of nodes $\mathbf{X}_i$, $1 \leq i \leq m$ and $\mathbf{Y}_j$ for $1 \leq j \leq n$. Without loss of generality, assume that $m < n$. Then consider the $n - m$ virtual random points $\mathbf{X}_{m+1}, ..., \mathbf{X}_n$ with the same distribution as $\mathbf{X}_i$, and define $\mathbf{Z}_i := (\mathbf{X}_i, \mathbf{Y}_i)$. Now for using the Efron-Stein inequality on set $\mathfrak{Z}_n = \{\mathbf{Z}_1, ..., \mathbf{Z}_n\}$, we involve another independent copy of $\mathfrak{Z}_n$ as $\mathfrak{Z}'_n = \{\mathbf{Z}'_1, ..., \mathbf{Z}'_n\}$, and define $\mathfrak{Z}_n^{(i)} := (\mathbf{Z}_1, ..., \mathbf{Z}_{i-1}, \mathbf{Z}'_i, \mathbf{Z}_{i+1}, ..., \mathbf{Z}_n)$, then $\mathfrak{Z}_n^{(1)}$ becomes $(\mathbf{Z}'_1, \mathbf{Z}_2, ..., \mathbf{Z}_n) = \big\{(\mathbf{X}'_1, \mathbf{Y}'_1), (\mathbf{X}_2, \mathbf{Y}_2), \ldots, (\mathbf{X}_m, \mathbf{Y}_n)\big\} =: (\mathfrak{X}_m^{(1)}, \mathfrak{Y}_n^{(1)})$ where $(\mathbf{X}'_1, \mathbf{Y}'_1)$ is independent copy of $(\mathbf{X}_1, \mathbf{Y}_1)$. Next define the function $r_{m,n}(\mathfrak{Z}_n) := \mathfrak{R}_{m,n}/(m+n)$, which means that we discard the random samples $\mathbf{X}_{m+1}, ..., \mathbf{X}_n$, and find the previously defined $\mathfrak{R}_{m,n}$ function on the nodes $\mathbf{X}_i$, $1 \leq i \leq m$ and $\mathbf{Y}_j$ for $1 \leq j \leq n$, and multiply by some coefficient to normalize it. Then, according to the Efron-Stein inequality we have

$$Var(r_{m,n}(\mathfrak{Z}_n)) \leq \frac{1}{2}\sum_{i=1}^{n}\mathbb{E}\left[(r_{m,n}(\mathfrak{Z}_n) - r_{m,n}(\mathfrak{Z}_n^{(i)}))^2\right].$$

Now we can divide the RHS as

$$\begin{aligned}\frac{1}{2}\sum_{i=1}^{n}\mathbb{E}\left[(r_{m,n}(\mathfrak{Z}_n) - r_{m,n}(\mathfrak{Z}_n^{(i)}))^2\right] &= \frac{1}{2}\sum_{i=1}^{m}\mathbb{E}\left[(r_{m,n}(\mathfrak{Z}_n) - r_{m,n}(\mathfrak{Z}_n^{(i)}))^2\right] \\ &+\frac{1}{2}\sum_{i=m+1}^{n}\mathbb{E}\left[(r_{m,n}(\mathfrak{Z}_n) - r_{m,n}(\mathfrak{Z}_n^{(i)}))^2\right].\end{aligned} \tag{B.18}$$

The first summand becomes

$$= \frac{1}{2}\sum_{i=1}^{m}\mathbb{E}\left[(r_{m,n}(\mathfrak{Z}_n) - r_{m,n}(\mathfrak{Z}_n^{(i)}))^2\right] = \frac{m}{2\,(m+n)^2}\mathbb{E}\left[(\mathfrak{R}_{m,n}(\mathfrak{X}_m, \mathfrak{Y}_n) - \mathfrak{R}_{m,n}(\mathfrak{X}_m^{(1)}, \mathfrak{Y}_n^{(1)}))^2\right],$$

which can also be upper bounded as follows:

$$\left|\mathfrak{R}_{m,n}(\mathfrak{X}_m,\mathfrak{Y}_n)-\mathfrak{R}_{m,n}(\mathfrak{X}_m^{(1)},\mathfrak{Y}_n^{(1)})\right| \leq \left|\mathfrak{R}_{m,n}(\mathfrak{X}_m,\mathfrak{Y}_n)-\mathfrak{R}_{m,n}(\mathfrak{X}_m^{(1)},\mathfrak{Y}_n)\right| \tag{B.19}$$
$$+\left|\mathfrak{R}(\mathfrak{X}_m^{(1)},\mathfrak{Y}_n)-\mathfrak{R}_{m,n}(\mathfrak{X}_m^{(1)},\mathfrak{Y}_n^{(1)})\right|.$$

For deriving an upper bound on the second line in (B.19) we should observe how much changing a point's position modifies the amount of $\mathfrak{R}_{m,n}(\mathfrak{X}_m,\mathfrak{Y}_n)$. We consider two steps of changing $\mathbf{X}_1$'s position: we first remove it from the graph, and then add it to the new position. Removing it would change $\mathfrak{R}_{m,n}(\mathfrak{X}_m,\mathfrak{Y}_n)$ at most by 2 c_d, because X_1 has a degree of at most c_d, and c_d edges will be removed from the MST graph, and c_d edges will be added to it. Similarly, adding $\mathbf{X}_1$ to the new position will affect $\mathfrak{R}_{m,n}(\mathfrak{X}_{m,n},\mathfrak{Y}_{m,n})$ at most by $2c_d$. So, we have

$$\left|\mathfrak{R}_{m,n}(\mathfrak{X}_m,\mathfrak{Y}_n)-\mathfrak{R}_{m,n}(\mathfrak{X}_m^{(1)},\mathfrak{Y}_n)\right| \leq 4\ c_d,$$

and we can also similarly reason that

$$\left|\mathfrak{R}_{m,n}(\mathfrak{X}_m^{(1)},\mathfrak{Y}_n)-\mathfrak{R}_{m,n}(\mathfrak{X}_m^{(1)},\mathfrak{Y}_n^{(1)})\right| \leq 4\ c_d.$$

Therefore totally we would have

$$\left|\mathfrak{R}_{m,n}(\mathfrak{X}_m,\mathfrak{Y}_n)-\mathfrak{R}_{m,n}(\mathfrak{X}_m^{(1)},\mathfrak{Y}_n^{(1)})\right| \leq 8\ c_d.$$

Furthermore, the second summand in (B.18) becomes

$$=\frac{1}{2}\sum_{i=m+1}^{n}\mathbb{E}\left[(r_{m,n}(\mathfrak{Z}_n)-r_{m,n}(\mathfrak{Z}_n^{(i)}))^2\right]=K_{m,n}\mathbb{E}\left[(\mathfrak{R}_{m,n}(\mathfrak{X}_m,\mathfrak{Y}_n)-\mathfrak{R}_{m,n}(\mathfrak{X}_m^{(m+1)},\mathfrak{Y}_n^{(m+1)}))^2\right],$$

where $K_{m,n}=\frac{n-m}{2\,(m+n)^2}$. Since in $(\mathfrak{X}_m^{(m+1)},\mathfrak{Y}_n^{(m+1)})$, the point $\mathbf{X}'_{m+1}$ is a copy of virtual random point $\mathbf{X}_{m+1}$, therefore this point doesn't change the FR test statistic $\mathfrak{R}_{m,n}$. Also following the above arguments we have

$$\left|\mathfrak{R}_{m,n}(\mathfrak{X}_m,\mathfrak{Y}_n)-\mathfrak{R}_{m,n}(\mathfrak{X}_m,\mathfrak{Y}_n^{(m+1)})\right| \leq 4\ c_d.$$

Hence we can bound the variance as below:

$$Var(r_{m,n}(\mathfrak{Z}_n)) \leq \frac{8c_d^2(n-m)}{(m+n)^2} + \frac{32\ c_d^2\ m}{(m+n)^2}. \tag{B.20}$$

Combining all results with the fact that $\frac{n}{m+n} \to q$ concludes the proof.

APPENDIX C

C.1 Proof of Theorem IV.4

Theorem IV.4 consists of two parts: bias and variance bounds. For the bias proof, from equation (4.5) we can write

$$\begin{aligned}\mathbb{E}\left[\widehat{\mathcal{E}}_{\varepsilon}(\mathbf{X}_1,\mathbf{X}_2)\right] &= \mathbb{E}\left[\min(\hat{p}_1,\hat{p}_2) - \frac{1}{N_2}\sum_{i=1}^{N_2}\tilde{t}\left(\widehat{U}_i\right)\right] \\ &= \min(\hat{p}_1,\hat{p}_2) - \frac{1}{N_2}\sum_{i=1}^{N_2}\mathbb{E}\left[\tilde{t}\left(\widehat{U}_i\right)\right] \\ &= \min(\hat{p}_1,\hat{p}_2) - \mathbb{E}_{X_{2,1}\sim f_2}\mathbb{E}\left[\tilde{t}\left(\widehat{U}_1\right)|X_{2,1}\right] \qquad \text{(C.1)}\end{aligned}$$

Now according to equation (33) of noshad2018hash, for any region for which its geometry is independent of the samples and the largest diameter within the region is equal to $c\varepsilon$, where c is a constant, then we have

$$\mathbb{E}\left[\tilde{t}\left(\widehat{U}_1\right)|X_{2,1}=x\right] = \tilde{t}\left(\frac{f_1(x)}{f_2(x)}\right) + O\left(\varepsilon^{\gamma}\right) + O\left(\frac{1}{N\varepsilon^{d}}\right). \qquad \text{(C.2)}$$

Thus, plugging (C.2) in (C.1) results in

$$\mathbb{E}\left[\widehat{\mathcal{E}}_{\varepsilon}(\mathbf{X}_1,\mathbf{X}_2)\right] = \min(\hat{p}_1,\hat{p}_2) - \mathbb{E}_{f_2}\left[\tilde{t}\left(\frac{f_1(X)}{f_2(X)}\right)\right] + O\left(\varepsilon^{\gamma}\right) + O\left(\frac{1}{N\varepsilon^{d}}\right), \qquad \text{(C.3)}$$

which completes the bias proof.

Remark C.1. It can easily be shown that if we use the NNR density ratio estimator (defined in noshad2017direct) with parameter k, the Bayes error estimator defined in (4.5) achieves the bias rate of $O\left(\left(\frac{k}{N}\right)^{\gamma/d}\right) + O\left(\frac{1}{k}\right)$.

The approach for the proof of the variance bound is similar to the Hash-based estimator noshad2018hash. Consider the two sets of nodes $X_{1,i}$, $1 \leq i \leq N_1$ and $X_{2,j}$, $1 \leq j \leq N_2$. For simplicity we assume that $N_1 = N_2$, however, similar to the variance proofs in noshad2017direct,noshad2018hash, by considering a number of virtual points one can easily extend the proof to general N_1 and N_2. Let $Z_i := (X_{1,i}, X_{2,i})$. For using the EfronStein inequality on $\mathbf{Z} := (Z_1, ..., Z_{N_1})$, we consider another independent copy of Z as $\mathbf{Z}' := (Z'_1, ..., Z'_{N_1})$ and define $\mathbf{Z}^{(i)} := (Z_1, ..., Z_{i-1}, Z'_i, Z_{i+1}, ..., Z_{N_1})$. In the following we use the EfronStein inequality. Note that we use the shorthand $\mathcal{E}(\mathbf{Z}) := \widehat{\mathcal{E}}_\varepsilon(\mathbf{X}_1, \mathbf{X}_2)$.

$$
\begin{aligned}
\mathbb{V}\left([]\,\mathcal{E}(\mathbf{Z})\right] &\leq \frac{1}{2}\sum_{i=1}^{N_1}\mathbb{E}\left[\left(\mathcal{E}(\mathbf{Z}) - \mathcal{E}(\mathbf{Z}^{(i)})\right)^2\right] \\
&= \frac{N_1}{2}\mathbb{E}\left[\left(\mathcal{E}(\mathbf{Z}) - \mathcal{E}(\mathbf{Z}^{(1)})\right)^2\right] \\
&\leq \frac{N_1}{2}\mathbb{E}\left(\frac{1}{N_1}\sum_{i=1}^{N_1}\widetilde{t}\left(\frac{\eta N_{i,1}}{N_{i,2}}\right) - \frac{1}{N_1}\sum_{i=1}^{N_1}\widetilde{t}\left(\frac{\eta N^{(1)}_{1,i}}{N^{(1)}_{2,i}}\right)\right)^2 \\
&= \frac{1}{2N_1}\mathbb{E}\left(\widetilde{t}\left(\frac{\eta N_{1,1}}{N_{1,2}}\right) - \widetilde{t}\left(\frac{\eta N^{(1)}_{1,1}}{N^{(1)}_{2,1}}\right)\right)^2 \\
&= \frac{1}{2N}O\left(1\right) = O(\frac{1}{N}). \qquad \text{(C.4)}
\end{aligned}
$$

Thus, the variance proof is complete.

C.2 Proof of Theorem IV.5

In this section we provide the proof of theorem IV.5. For simplicity we assume that $N_1 = N_2$ and we use the notation $N := N_1$. Also note that for simplicity we use the notation $\widehat{U}_i := \widehat{U}_i^{(\varepsilon)}$ Using the definition of $\widehat{\mathcal{E}}_\varepsilon(\mathbf{X}_1, \mathbf{X}_2)$ we have

$$\begin{aligned}
\sqrt{N}\left(\widehat{\mathcal{E}}_\varepsilon(\mathbf{X}_1,\mathbf{X}_2) - \mathbb{E}\left[\widehat{\mathcal{E}}_\varepsilon(\mathbf{X}_1,\mathbf{X}_2)\right]\right) &= \sqrt{N}\left(\frac{1}{2} - \frac{1}{N}\sum_{i=1}^{N}\tilde{t}\left(\widehat{U}_i\right) - \mathbb{E}\left[\frac{1}{2} - \frac{1}{N}\sum_{i=1}^{N}\tilde{t}\left(\widehat{U}_i\right)\right]\right) \\
&= \frac{1}{\sqrt{N}}\sum_{i=1}^{N}\left(\tilde{t}\left(\widehat{U}_i\right) - \mathbb{E}\left[\tilde{t}\left(\widehat{U}_i\right)\right]\right) \\
&= \frac{1}{\sqrt{N}}\sum_{i=1}^{N}\left(\tilde{t}\left(\widehat{U}_i\right) - \mathbb{E}_{\bar{i}}\left[\tilde{t}\left(\widehat{U}_i\right)\right]\right) \\
&\quad + \frac{1}{\sqrt{N}}\sum_{i=1}^{N}\left(\mathbb{E}_{\bar{i}}\left[\tilde{t}\left(\widehat{U}_i\right)\right] - \mathbb{E}\left[\tilde{t}\left(\widehat{U}_i\right)\right]\right), \qquad \text{(C.5)}
\end{aligned}$$

where $\mathbb{E}_{\bar{i}}$ denotes the expectation over all samples $\mathbf{X}_1, \mathbf{X}_2$ except $X_{2,i}$. In the above equation, we denote the first and second terms respectively by $S_1(\mathbf{X})$ and $S_2(\mathbf{X})$, where $\mathbf{X} := (\mathbf{X}_1, \mathbf{X}_2)$. In the following we prove that $S_2(\mathbf{X})$ converges to a normal random variable, and $S_1(\mathbf{X})$ converges to zero in probability. Therefore, using the Slutsky's theorem, the left hand side of (C.5) converges to a normal random variable.

Lemma C.2. *Let $N \to \infty$. Then, $S_2(\mathbf{X})$ converges to a normal random variable.*

Proof. Let $A_i(\mathbf{X}) := \mathbb{E}_{\bar{i}}\left[\tilde{t}\left(\widehat{U}_i\right)\right] - \mathbb{E}\left[\tilde{t}\left(\widehat{U}_i\right)\right]$. Since for all $i \in \{1, ..., N\}$, $A_i(\mathbf{X})$ are i.i.d. random variables, using the standard central limit theorem durrett2019probability, $S_2(\mathbf{X})$ converges to a normal random variable. □

Lemma C.3. *Let $\varepsilon \to 0$ and $\frac{1}{\varepsilon^d N} \to 0$. Then, $S_1(\mathbf{X})$ converges to 0 in mean square.*

Proof. In order to prove that MSE converges to zero, we need to compute the bias

and variance terms separately. The bias term is obviously equal to zero since

$$\begin{aligned}\mathbb{E}[S_1(\mathbf{X})] &= \mathbb{E}\left[\frac{1}{\sqrt{N}}\sum_{i=1}^{N}\left(\tilde{t}\left(\widehat{U}_i\right) - \mathbb{E}_{\bar{i}}\left[\tilde{t}\left(\widehat{U}_i\right)\right]\right)\right] \\ &= \frac{1}{\sqrt{N}}\sum_{i=1}^{N}\left(\mathbb{E}\left[\tilde{t}\left(\widehat{U}_i\right)\right] - \mathbb{E}\left[\tilde{t}\left(\widehat{U}_i\right)\right]\right) \\ &= 0. \qquad \text{(C.6)}\end{aligned}$$

Next, we find an upper bound on the variance of $S_1(\mathbf{X})$ using the Efron-Stein inequality. Let $\mathbf{X}' := (\mathbf{X}_1', \mathbf{X}_2')$ denote another copy of $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2)$ with the same distribution. We define the resampled dataset as

(C.7)

$$\mathbf{X}^{(j)} := \begin{cases} (X_{1,1}, ..., X_{1,j-1}, X'_{1,j}, X_{1,j+1}, ..., X_{1,N}, X_{2,1}, ..., X_{2,N}) & \text{if} \quad N+1 \leq j \leq 2N \\ (X_{1,1}, ..., X_{1,N}, X_{2,1}, ..., X_{2,j-1}, X'_{2,j}, X_{2,j+1}, ..., X_{2,N}) & \text{if} \quad 1 \leq j \leq N \end{cases}$$

Let $\Delta_i =: \tilde{t}\left(\widehat{U}_i\right) - \mathbb{E}_{\bar{i}}\left[\tilde{t}\left(\widehat{U}_i\right)\right] - \tilde{t}\left(\widehat{U}_i^{(1)}\right) + \mathbb{E}_{\bar{i}}\left[\tilde{t}\left(\widehat{U}_i^{(1)}\right)\right]$. Using the Efron-Stein inequality we can write

$$\begin{aligned}\mathbb{V}\left[S_1(\mathbf{X}_1, \mathbf{X}_2)\right] &\leq \frac{1}{2}\sum_{j=1}^{2N}\mathbb{E}\left[\left(S_1(\mathbf{X}) - S_1(\mathbf{X}^{(j)})\right)^2\right] \\ &= N\mathbb{E}\left[\left(S_1(\mathbf{X}) - S_1(\mathbf{X}^{(1)})\right)^2\right] \\ &= \mathbb{E}\left[\left(\sum_{i=1}^{N}\Delta_i\right)^2\right], \\ &= \sum_{i=1}^{N}\mathbb{E}\left[\Delta_i^2\right] + \sum_{i\neq j}\mathbb{E}\left[\Delta_i\Delta_j\right]. \qquad \text{(C.8)}\end{aligned}$$

We obtain bounds on the first and second terms in equation (C.8). First, we obtain

separate bounds on $\mathbb{E}\left[\Delta_i^2\right]$ for $i = 1$ and $i \neq 1$. We have

$$\begin{aligned}
\mathbb{E}\left[\Delta_1^2\right] &= \mathbb{E}\left[\left(\tilde{t}\left(\widehat{U}_i\right) - \mathbb{E}_{\bar{i}}\left[\tilde{t}\left(\widehat{U}_i\right)\right] - \tilde{t}\left(\widehat{U}_i^{(1)}\right) + \mathbb{E}_{\bar{i}}\left[\tilde{t}\left(\widehat{U}_i^{(1)}\right)\right]\right)^2\right] \\
&= \mathbb{E}\left[\left(\tilde{t}\left(\widehat{U}_i\right) - \mathbb{E}_{\bar{i}}\left[\tilde{t}\left(\widehat{U}_i\right)\right]\right)^2\right] + \mathbb{E}\left[\left(\tilde{t}\left(\widehat{U}_i^{(1)}\right) - \mathbb{E}_{\bar{i}}\left[\tilde{t}\left(\widehat{U}_i^{(1)}\right)\right]\right)^2\right] \\
&\quad - 2\mathbb{E}\left[\left(\tilde{t}\left(\widehat{U}_i\right) - \mathbb{E}_{\bar{i}}\left[\tilde{t}\left(\widehat{U}_i\right)\right]\right)\left(\tilde{t}\left(\widehat{U}_i^{(1)}\right) - \mathbb{E}_{\bar{i}}\left[\tilde{t}\left(\widehat{U}_i^{(1)}\right)\right]\right)\right] \\
&\leq 4\mathbb{E}\left[\left(\tilde{t}\left(\widehat{U}_i\right) - \mathbb{E}_{\bar{i}}\left[\tilde{t}\left(\widehat{U}_i\right)\right]\right)^2\right] \qquad \text{(C.9)} \\
&\leq 4\mathbb{E}_{X_1}\left[\mathbb{E}_{X_{\bar{1}}}\left[\left(\tilde{t}\left(\widehat{U}_i\right) - \mathbb{E}_{\bar{i}}\left[\tilde{t}\left(\widehat{U}_i\right)\right]\right)^2 \middle| X_1 = x\right]\right] \\
&\leq 4\mathbb{E}_{X_1}\left[\mathbb{V}\left[\tilde{t}\left(\widehat{U}_i\right)\right]\right] \qquad \text{(C.10)} \\
&\leq O(\frac{1}{N}). \qquad \text{(C.11)}
\end{aligned}$$

Now for the case of $i \neq 1$ note that $\mathbb{E}_{\bar{i}}\left[\tilde{t}\left(\widehat{U}_i\right)\right] = \mathbb{E}_{\bar{i}}\left[\tilde{t}\left(\widehat{U}_i^{(1)}\right)\right]$. Thus, we can bound $\mathbb{E}\left[\Delta_i^2\right]$ as

$$\begin{aligned}
\mathbb{E}\left[\Delta_i^2\right] &= \mathbb{E}\left[\left(\tilde{t}\left(\widehat{U}_i\right) - \tilde{t}\left(\widehat{U}_i^{(1)}\right)\right)^2\right] \\
&\leq O\left(\varepsilon^d\right)\left(1 - O\left(\varepsilon^d\right)\right) O\left(\left(\frac{1}{\varepsilon^d N}\right)^2\right) = \frac{1}{N} O\left(\frac{1}{\varepsilon^d N}\right). \qquad \text{(C.12)}
\end{aligned}$$

Hence, using (C.11) and (C.12) we get

$$\sum_{i=1}^{N} \mathbb{E}\left[\Delta_i^2\right] \leq O\left(\frac{1}{\varepsilon^d N}\right). \qquad \text{(C.13)}$$

Note that we can similarly prove that the bound $\sum_{i \neq j} \mathbb{E}\left[\Delta_i \Delta_j\right] \leq O\left(\frac{1}{\varepsilon^d N}\right)$. Thus, from equation (C.8) we have $\mathbb{V}\left[S_1(\mathbf{X}_1, \mathbf{X}_2)\right] \leq O\left(\frac{1}{\varepsilon^d N}\right)$, which convergence to zero if the assumption $\frac{1}{\varepsilon^d N} \to 0$ holds.

□

C.3 Proof of Theorem IV.8

First note that since $N_{1,1}$ and $N_{2,1}$ are independent we can write

$$\mathbb{E}\left[\frac{N_{1,i}}{N_{2,i}} \middle| X_{2,i}\right] = \mathbb{E}\left[N_{1,i} | X_{2,i}\right] \mathbb{E}\left[N_{2,i}^{-1} \middle| X_{2,i}\right]. \qquad \text{(C.14)}$$

From (37) and (38) of noshad2018hash we have

(C.15)

$$\mathbb{E}\left[N_{1,i}\right] = N_1\epsilon^d\left([]\right) f_1(X_{2,i}) + \sum_{l=1}^{q} C_l(X_{2,i})\epsilon^l + O\left(C_q(X_{2,i})\epsilon^q\right)],$$

(C.16)

$$\mathbb{E}\left[(N_{2,i})^{-1}\right] = N_2^{-1}\epsilon^{-d}\left([]\right) f_2(X_{2,i}) + \sum_{l=1}^{q} C_l(X_{2,i})\epsilon^l + O\left(C_q(X_{2,i})\epsilon^q\right)]^{-1}\left(1 + O\left(\frac{1}{N_2\epsilon^d f_2(X_{2,i})}\right)\right),$$

where $C_i(x)$ for $1 \leq i \leq q$ are functions of x. Plugging equations (C.15) and (C.16) into (C.14) results in

$$\mathbb{E}\left[\frac{\eta N_{1,i}}{N_{2,i}}\middle| X_{2,i}\right] = \frac{f_1(X_{2,i})}{f_2(X_{2,i})} + \sum_{i=1}^{q} C_i''\epsilon^i + O\left(\frac{1}{N\epsilon^d}\right), \tag{C.17}$$

where $C_1'', ..., C_q''$ are constants.

Now apply the ensemble theorem (moon2018ensemble, Theorem 4). Let $\mathcal{T} := \{t_1, ..., t_T\}$ be a set of index values with $t_i < c$, where $c > 0$ is a constant. Define $\epsilon(t) := \left\lfloor tN^{-1/2d}\right\rfloor$. According to the ensemble theorem in (moon2018ensemble, Theorem 4) if we choose the parameters $\psi_i(t) = t^{i/d}$ and $\phi'_{i,d}(N) = \phi_{i,\kappa}(N)/N^{i/d}$, the following weighted ensemble converges to the true value with the MSE rate of $O(1/N)$:

$$\widehat{U}_i^{\mathbf{w}} := \sum_{l=1}^{L} w_l \widehat{U}_i, \tag{C.18}$$

where the weights w_l are the solutions of the optimization problem in equation (4.14). Thus, the bias of the ensemble estimator can be written as

$$\mathbb{E}_{\bar{X}_i}\left[\widehat{U}_i^{\mathbf{w}}\middle| X_{2,i}\right] = \frac{f_1(X_{2,i})}{f_2(X_{2,i})} + O(1/\sqrt{N_1}). \tag{C.19}$$

By Lemma 4.4 in noshad2017direct and the fact that function $t(x) := |p_1 x - p_2| - p_1 x$ is Lipschitz continuous with constant $2p_1$,

(C.20)
$$\left|\mathbb{E}_{\bar{X}_i}[t(\widehat{U}_i^{\mathbf{w}})|X_{2,i}] - t\left(\frac{f_1(X_{2,i})}{f_2(X_{2,i})}\right)\right| \leq 2p_1\left(\sqrt{\mathbb{V}_{\bar{X}_i}[\widehat{U}_i^{\mathbf{w}}|X_{2,i}]} + \left|\mathbb{B}_{\bar{X}_i}[\widehat{U}_i^{\mathbf{w}}|X_{2,i}]\right|\right).$$

Here $\mathbb{B}$ and $\mathbb{V}$ represent bias and variance, respectively. By (C.19), we have $\mathbb{B}_{\bar{X}_i}[\widehat{U}_i^{\mathbf{w}}|X_{2,i}] = O(1/\sqrt{N_1})$; and by Theorem 2.2 in noshad2017direct, $\mathbb{V}_{\bar{X}_i}[\widehat{U}_i^{\mathbf{w}}|X_{2,i}] = O(1/N_1)$. Thus,

(C.21)
$$\mathbb{E}_{\bar{X}_i}[t(\widehat{U}_i^{\mathbf{w}})|X_{2,i}] - t\left(\frac{f_1(X_{2,i})}{f_2(X_{2,i})}\right) = O(1/\sqrt{N_1}).$$

So the bias of the estimator $\mathcal{F}(\mathbf{X}_1, \mathbf{X}_2)$ is given by

$$\mathbb{B}(\mathcal{F}(\mathbf{X}_1, \mathbf{X}_2)) = \left|\mathbb{E}_{\mathbf{X}_1,\mathbf{X}_2}\left[\frac{1}{2N_2}\sum_{i=1}^{N_2} t(\widehat{U}_i^{\mathbf{w}})\right] - \frac{1}{2}\mathbb{E}_{X_{2,i}}\left[t\left(\frac{f_1(X_{2,i})}{f_2(X_{2,i})}\right)\right]\right|$$

(C.22)
$$= \frac{1}{2N_2}\sum_{i=1}^{N_2}\left|\mathbb{E}_{X_{2,i}}\left[\mathbb{E}_{\bar{X}_i}[t(\widehat{U}_i^{\mathbf{w}})|X_{2,i}] - t\left(\frac{f_1(X_{2,i})}{f_2(X_{2,i})}\right)\right]\right| = O(1/\sqrt{N_1}).$$

Finally, since the variance of $\widehat{U}_i^{\mathbf{w}}$ can easily be upper bounded by $O(1/N)$ using the Efron-Stein inequality using the same steps in Appendix. C.1.

C.4 Proof of Theorem IV.9

In order to prove the theorem we first prove that the solutions of the constraint in (4.14) for $t_i = s_i$ can be written as a function of the shifted Chebyshev polynomials. Then we find the optimal solutions of w_i which minimize $\|w\|_2^2$.

Lemma C.4. *All solutions of the constraint*

(C.23)
$$\begin{aligned}\sum_{k=0}^{L-1}\omega_k s_k^j &= 0, \quad \forall j \in \{1, ..., d\}\\ \sum_{k=0}^{L-1}\omega_k &= 1,\end{aligned}$$

have the following form

$$w_i = \sum_{k=0}^{d} \frac{2T_k^\alpha(0)}{L} T_k^\alpha(s_i) + \sum_{k=d+1}^{L-1} c_k T_k^\alpha(s_i) - \frac{1}{L} \qquad \forall i \in \{0, ..., L-1\}, \tag{C.24}$$

for some $c_k \in \mathbb{R}$, $k \in \{d+1, ..., L-1\}$, *and for any* $c_k \in \mathbb{R}$, $k \in \{d+1, ..., L-1\}$, w_i *given by* (C.24) *satisfy the equations in* (C.23).

Proof. We can rewrite (C.23) as

$$\sum_{j=0}^{d} \sum_{k=0}^{L-1} \omega_k x_j s_k^j = x_0 \quad \forall x_j \in \mathbb{R}. \tag{C.25}$$

Note that setting $\forall i \in \{1, ..., d\}, x_i = 0$ in (C.25) yields the second constraint in (4.14), and $\forall i \neq j, x_i = 0$ results in the first set of d constraints in (4.14). Using the fact that $\sum_j \sum_k \omega_k x_j s_k^j = \sum_k \omega_k \sum_j x_j s_k^j$ we can equivalently write the constraint as

$$\sum_{k=0}^{L-1} \omega_k f(s_k) = f(o) \quad \forall f \in P_d, \tag{C.26}$$

where P_d is the family of the polynomials of degree d. One can expand the polynomial $f(x) \in P_d$ defined in $[0, \alpha]$ in the Chebyshev polynomial basis:

$$f(x) = \sum_{i=0}^{d} r_i T_i^\alpha(x).$$

Thus, we can write the constraint in (C.26) as

$$\sum_{k=0}^{L-1} \omega_k \sum_{j=0}^{d} r_j T_j^\alpha(s_k) = \sum_{j=0}^{d} r_j T_j^\alpha(0) \quad \forall r_j \in \mathbb{R}, \tag{C.27}$$

which can be further formulated as

$$\sum_{j=0}^{d} r_j \sum_{k=0}^{L-1} \omega_k T_j^\alpha(s_k) = \sum_{j=0}^{d} r_j T_j^\alpha(0) \quad \forall r_j \in \mathbb{R}, \tag{C.28}$$

which is equivalent to the following constraint in the Chebyshev polynomials basis:

$$\sum_{k=0}^{L-1} \omega_k T_j^{\alpha}(s_k) = T_j^{\alpha}(0) \quad \forall j \in \{0, ..., d\}. \tag{C.29}$$

Now we use the Chebyshev polynomial approximation method in order to simplify the optimization problem in equation (4.14). Define a function $f : [0, \alpha] \to \mathbb{R}$ such that $f(s_i) = w_i, i \in \{0, ..., L-1\}$.

We can write $f(x)$ in terms of Chebyshev interpolation polynomials with the L points $0 < s_0, ..., s_{L-1} < 1$ as

$$f(x) = \sum_{k=0}^{L-1} c_k T_k^{\alpha}(x) - \frac{c_0}{2} + R(x), \tag{C.30}$$

where $R(x)$ is the error of approximation and is given by

$$R(x) = \frac{f^{(L)}(\xi)}{L!} \prod_{j=0}^{L-1} (x - s_j), \tag{C.31}$$

for some $\xi \in [0, \alpha]$. Thus we have

$$w_i = f(s_i) = \sum_{k=0}^{L-1} c_k T_k^{\alpha}(s_i) - \frac{c_0}{2} \qquad \forall i \in \{0, ..., L-1\}. \tag{C.32}$$

The interpolation coefficients in (C.30) can be computed as follows

$$c_k = \frac{2}{L} \sum_{j=0}^{L-1} f(s_j) T_k^{\alpha}(s_j) \qquad \forall k \in \{0, ..., L-1\}. \tag{C.33}$$

Comparing the equation (C.33) with the constraint in (C.29) we get

$$c_k = \frac{2T_k^\alpha(0)}{L} \quad \forall k \in \{0, ..., d\}. \tag{C.34}$$

Thus, we can write equation (C.32) as

$$w_i = f(s_i) = \sum_{k=0}^{d} \frac{2T_k^\alpha(0)}{L} T_k^\alpha(s_i) + \sum_{k=d+1}^{L-1} c_k T_k^\alpha(s_i) - \frac{1}{L} \qquad \forall i \in \{0, ..., L-1\}. \tag{C.35}$$

Next, for any $c_k \in \mathbb{R}$, $k \in \{d+1, ..., L-1\}$, w_i given by (C.24) satisfy equation (C.29), which is an equivalent form of the original constraints in equation (C.23). Using (C.35) we can write:

$$\begin{aligned}\sum_{i=0}^{L-1} \omega_i T_j^\alpha(s_i) &= \sum_{i=0}^{L-1} T_j^\alpha(s_i) \left[\sum_{k=0}^{d} \frac{2T_k^\alpha(0)}{L} T_k^\alpha(s_i) + \sum_{k=d+1}^{L-1} c_k T_k^\alpha(s_i) - \frac{c_0}{2} \right] \\ &= \sum_{k=0}^{d} \frac{2T_k^\alpha(0)}{L} \sum_{i=0}^{L-1} T_j^\alpha(s_i) T_k^\alpha(s_i) + \sum_{k=d+1}^{L-1} c_k \sum_{i=0}^{L-1} T_j^\alpha(s_i) T_k^\alpha(s_i) - \sum_{i=0}^{L-1} T_j^\alpha(s_i) \frac{T_0^\alpha(s_i)}{L},\end{aligned} \tag{C.36}$$

where for the last term we have used the fact that $c_0 = \frac{2T_0^\alpha(0)}{L} = \frac{2T_0^\alpha(s_i)}{L} = \frac{2}{L}$ from equation (C.34). Now in order to simplify equation (C.36), we use the orthogonality property of the Chebyshev (and shifted Chebyshev) polynomials. That is, if s_i are the zeros of T_L^*, then

$$\sum_{i=0}^{L-1} T_j^\alpha(s_i) T_k^\alpha(s_i) = K_j \delta_{kj}, \tag{C.37}$$

where $K_j = L$ for $j = 0$ and $K_j = L/2$ for $L - 1 \geq j > 0$. Hence, (C.36) simplifies to

$$\sum_{i=0}^{L-1} \omega_i T_j^\alpha(s_i) = \sum_{k=0}^{d} \frac{2T_k^\alpha(0)}{L} K_j \delta_{kj} + \sum_{k=d+1}^{L-1} c_k K_j \delta_{kj} - K_0 \delta_{0j} \frac{1}{L}. \tag{C.38}$$

Thus, for $j = 0$ we get

$$\sum_{i=0}^{L-1} \omega_i T_j^{\alpha}(s_i) = 2T_0^{\alpha}(0) - 1 = T_0^{\alpha}(0), \tag{C.39}$$

and for $d \geq j > 0$ we get

$$\sum_{i=0}^{L-1} \omega_i T_j^{\alpha}(s_i) = T_j^{\alpha}(0), \tag{C.40}$$

which shows that w_i satisfy the constraint in equation (C.29), which is an equivalent form of the original constraints in equation (C.23). The proof of the lemma is complete. □

Proof of Theorem IV.9: In (C.24), c_k, $k \in \{d+1, ..., L-1\}$ will be determined such that the term $\|w\|_2^2$ in the original optimization problem is minimized. Using (C.24), the objective function of the optimization problem in (4.14) can be simplified as

$$\begin{aligned} \|w\|_2^2 &= \sum_{i=0}^{L-1} w_i^2 \\ &= \sum_{i=0}^{L-1} f(s_i)^2 \\ &= \sum_{i=0}^{L-1} A_i^2 + \sum_{i=0}^{L-1} 2A_i \sum_{k=d+1}^{L-1} c_k T_k^{\alpha}(s_i) + \sum_{i=0}^{L-1} \left(\sum_{k=d+1}^{L-1} c_k T_k^{\alpha}(s_i) \right)^2 \end{aligned} \tag{C.41}$$

where $A_i := \sum_{k=0}^{d} \frac{2T_k^*(0)}{L} T_k^{\alpha}(s_i) - \frac{1}{L}$. Note that since the first term in (C.41) is constant, the minimization of $\|w\|_2^2$ is equivalent to minimization of the following quadratic expression in terms of the variables $\{c_{d+1}, ..., c_{L-1}\}$:

$$G(c_{d+1}, ..., c_{L-1}) := \sum_{i=0}^{L-1} 2A_i \sum_{k=d+1}^{L-1} c_k T_k^{\alpha}(s_i) + \sum_{i=0}^{L-1} \left(\sum_{k=d+1}^{L-1} c_k T_k^{\alpha}(s_i) \right)^2. \tag{C.42}$$

We first show that the first term in (C.42) is equal to zero.

$$\begin{aligned}\sum_{i=0}^{L-1} 2A_i \sum_{k=d+1}^{L-1} c_k T_k^{\alpha}(s_i) &= \sum_{i=0}^{L-1} 2\left(\sum_{k=0}^{d} \frac{2T_k^*(0)}{L} T_k^{\alpha}(s_i) - \frac{1}{L}\right) \sum_{k=d+1}^{L-1} c_k T_k^{\alpha}(s_i) \\ &= \frac{2}{L} \sum_{i=0}^{L-1} \sum_{k=0}^{d} \sum_{j=d+1}^{L-1} 2T_k^*(0) T_k^{\alpha}(s_i) c_j T_j^{\alpha}(s_i) - \sum_{i=0}^{L-1} \sum_{j=d+1}^{L-1} c_j T_j^{\alpha}(s_i) \\ &= \frac{2}{L} \sum_{k=0}^{d} \sum_{j=d+1}^{L-1} 2T_k^*(0) c_j \sum_{i=0}^{L-1} T_k^{\alpha}(s_i) T_j^{\alpha}(s_i) - \sum_{j=d+1}^{L-1} c_j \sum_{i=0}^{L-1} T_j^{\alpha}(s_i) T_0^{\alpha}(s_i) \\ &= 0. \end{aligned} \tag{C.43}$$

Note that in the third line, we have used the identity $T_0^*(s_i) = 1$. In the fourth line we have used the orthogonality identity (C.37). Finally, setting $c_{d+1} = ... = c_{L-1} = 0$ minimizes the second term and as a result $G(c_{d+1}, ..., c_{L-1})$. Thus, the optimal solutions of w_i are given as

$$w_i = \frac{2}{L} \sum_{k=0}^{d} T_k^{\alpha}(0) T_k^{\alpha}(s_i) - \frac{1}{L} \qquad \forall i \in \{0, ..., L-1\}, \tag{C.44}$$

which completes the proof.

C.5 Proof of Theorem IV.10

Bias proof: In the following we state a multivariate generalization of Lemma 3.2 in noshad2017direct.

Lemma C.5. *Assume that $g(x_1, x_2, \ldots, x_k) : \mathcal{X} \times \cdots \times \mathcal{X} \to \mathbb{R}$ is Lipschitz continuous with constant $H_g > 0$, with respect to $x_1, \ldots, x_k$. If $\widehat{T}_i$ where $0 \leq i \leq k$ be random variables, each one with a variance $\mathbb{V}[\widehat{T}_i]$ and a bias with respect to given constant values T_i, defined as $\mathbb{B}[\widehat{T}_i] := T_i - \mathbb{E}[\widehat{T}_i]$, then the bias of $g(\widehat{T}_1, \ldots, \widehat{T}_k)$ can be upper bounded by*

$$\left|\mathbb{E}\left[g(\widehat{T}_1, \ldots, \widehat{T}_k) - g(T_1, \ldots, T_k)\right]\right| \leq H_g \sum_{i=1}^{k} \left(\sqrt{\mathbb{V}[\widehat{T}_i]} + \left|\mathbb{B}[\widehat{T}_i]\right|\right). \tag{C.45}$$

Proof:

$$\left|\mathbb{E}\left[g(\widehat{T}_1,\ldots,\widehat{T}_\lambda)-g(T_1,\ldots,T_\lambda)\right]\right| \leq \sum_{i=1}^{\lambda}\left|\mathbb{E}\left[g(\widehat{T}_1,\ldots,\widehat{T}_i,T_{i+1},\ldots,T_\lambda)-g(T_1,\ldots,T_\lambda)\right]\right|$$

$$\leq \sum_{i=1}^{\lambda} H_g\left(\sqrt{\mathbb{V}[\widehat{T}_i]}+\left|\mathbb{B}[\widehat{T}_i]\right|\right), \tag{C.46}$$

where in the last inequality we have used Lemma 3.2 in noshad2017direct, by assuming that g is only a function of $\widehat{T}_i$.

Now, we plug $\widehat{U}_i^{\mathbf{w}}$ defined in (C.18) into $\widehat{T}_i$ in (C.45). Using equation (C.19) and the fact that $\mathbb{V}_{\bar{X}_i}[\widehat{U}_i^{\mathbf{w}}|X_{2,i}] = O(1/N_1)$ (as mentioned in Appendix C.3), concludes the bias proof.

Variance proof: Without loss of generality, we assume that $N_\lambda = \max(N_1, N_2, \ldots, N_\lambda)$. We consider $(N_\lambda - N_l)$ virtual random nodes $X_{l,N_l+1},\ldots,X_{l,N_\lambda}$ for $1 \leq l \leq \lambda-1$ which follow the same distribution as $X_{l,1},\ldots,X_{l,N_l}$. Let $Z_i := (X_{1,i}, X_{2,i},\ldots,X_{\lambda,i})$. Now we consider $\mathbf{Z} := (Z_1,\ldots,Z_{N_\lambda})$ and another independent copy of $\mathbf{Z}$ as $\mathbf{Z}' := (Z'_1,\ldots,Z'_{N_\lambda})$, where $Z_i := (X'_{1,i}, X'_{2,i},\ldots,X'_{\lambda,i})$. Let $\mathbf{Z}^{(i)} := (Z_1,\ldots,Z_{i-1},Z'_i,Z_{i+1},\ldots,Z_{N_\lambda})$ and $\mathcal{E}_k(\mathbf{Z}) := \mathcal{E}_k(\mathbf{X}_1,\mathbf{X}_2,\ldots,\mathbf{X}_\lambda)$. Let

$$B_{\alpha,i} := \tilde{t}\left(\widehat{U}^{\mathbf{w}}_{(1/\lambda)}(X_{\lambda,i}),\widehat{U}^{\mathbf{w}}_{(2/\lambda)}(X_{\lambda,i}),\ldots,\widehat{U}^{\mathbf{w}}_{((\lambda-1)/\lambda))}(X_{\lambda,i})\right)$$

$$-\tilde{t}\left(\widehat{U}^{\mathbf{w}}_{(1/\lambda)}(X'_{\lambda,i}),\widehat{U}^{\mathbf{w}}_{(2/\lambda)}(X'_{\lambda,i}),\ldots,\widehat{U}^{\mathbf{w}}_{((\lambda-1)/\lambda)}(X'_{\lambda,i})\right). \tag{C.47}$$

We have

$$\frac{1}{2}\sum_{i=1}^{N_\lambda}\mathbb{E}\left[\left(\mathcal{E}_k(\mathbf{Z})-\mathcal{E}_k(\mathbf{Z}^{(i)})\right)^2\right] = \frac{1}{2N_\lambda}\mathbb{E}\left[\sum_{i=1}^{N_\lambda}B_{\alpha,i}\right]^2$$

$$= \frac{1}{2N_\lambda}\sum_{i=1}^{N_\lambda}\mathbb{E}[B^2_{\alpha,i}] + \frac{1}{2N_\lambda}\sum_{i\neq j}\mathbb{E}[B_{\alpha,i}B_{\alpha,j}] = \frac{1}{2}\mathbb{E}[B^2_{\alpha,2}] + \frac{N_\lambda}{2}\mathbb{E}[B_{\alpha,2}]^2. \tag{C.48}$$

The last equality follows from $\mathbb{E}[B_{\alpha,i}B_{\alpha,j}] = \mathbb{E}[B_{\alpha,i}]\mathbb{E}[B_{\alpha,j}] = \mathbb{E}[B_{\alpha,i}]^2$ for $i \neq j$. With a parallel argument in the proof of Lemma 4.10 in noshad2017direct, we have

$$\mathbb{E}[B_{\alpha,2}] = O\left(\frac{\lambda}{N_\lambda}\right) \text{ and } \mathbb{E}[B_{\alpha,2}^2] = O\left(\frac{\lambda^2}{N_\lambda}\right). \tag{C.49}$$

Then applying Efron-Stein inequality, we obtain

$$\mathbb{V}[\mathcal{E}_k(\mathbf{Z})] \leq \frac{1}{2}\sum_{i=1}^{M}\mathbb{E}\left[\left(\mathcal{E}_k(\mathbf{Z}) - \mathcal{E}_k(\mathbf{Z}^{(i)})\right)^2\right] = O\left(\frac{\lambda^2}{N_\lambda}\right). \tag{C.50}$$

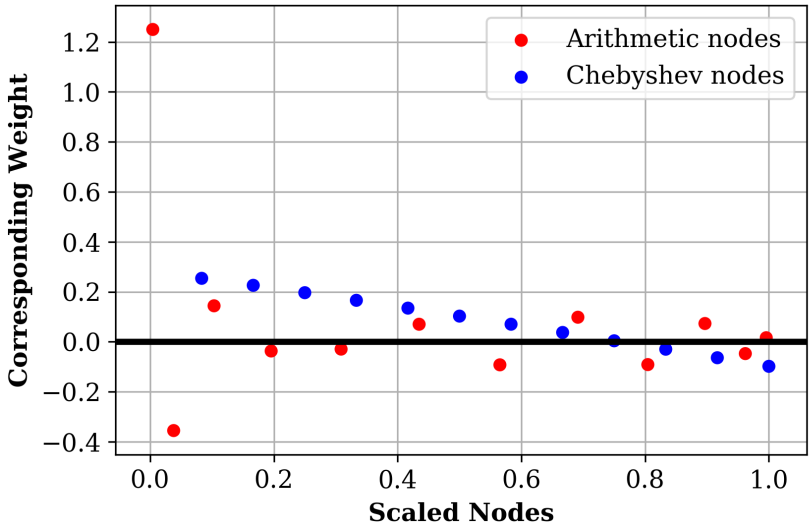
Since the ensemble estimator is a convex combination of some single estimators, the proof is complete.

C.6 Supplementary Numerical Results

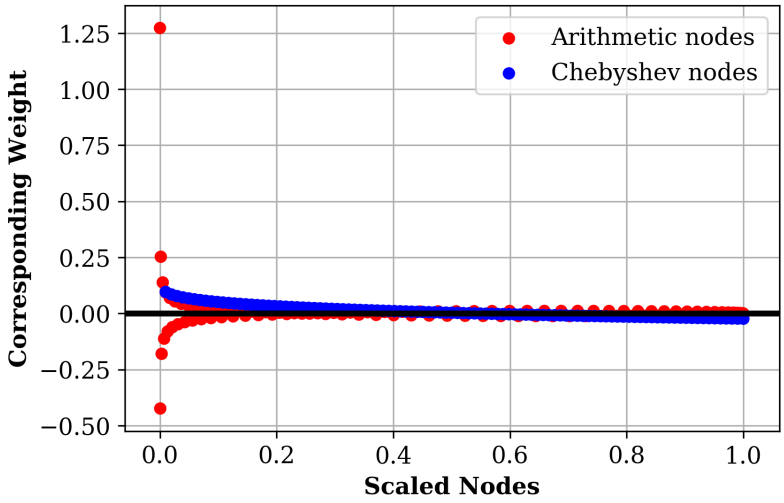
In this section we perform extended experiments on the proposed benchmark learner. We perform experiments on different simulated datasets with Gaussian, beta, Rayleigh and concentric distributions of various dimensions of up to $d = 100$.

Figure C.1 represents the scaled coefficients of the base estimators and their corresponding weights in the ensemble estimator using the arithmetic and Chebyshev nodes for (a) $d = 10$ ($L = 11$) and (b) $d = 100$ ($L = 101$). The optimal weights for the arithmetic nodes decreases monotonically. However, the optimal weights for the Chebyshev nodes has an oscillating pattern.

In Figures C.2 and C.3 we consider binary classification problems respectively with 4-dimensional and 100-dimensional isotropic normal distributions with covariance matrix $\sigma\mathbf{I}$, where the means are separated by 2 units in the first dimension. We plot the Bayes error estimates for different methods of Chebyshev, arithmetic and uniform weight assigning methods for different sample sizes, in terms of (a) MSE rate and (b) mean estimates with %95 confidence intervals. Although both the Chebyshev and arithmetic weight assigning methods are asymptotically optimal, in our experiments



(a) $d = 10$



(b) $d = 100$

Figure C.1: The scaled coefficients of the base estimators and their corresponding optimal weights in the ensemble estimator using the arithmetic and Chebyshev nodes for (a) $d = 10$ and (b) $d = 100$. The optimal weights for the arithmetic nodes decreases monotonically. However, the optimal weights for the Chebyshev nodes has an oscillating pattern.

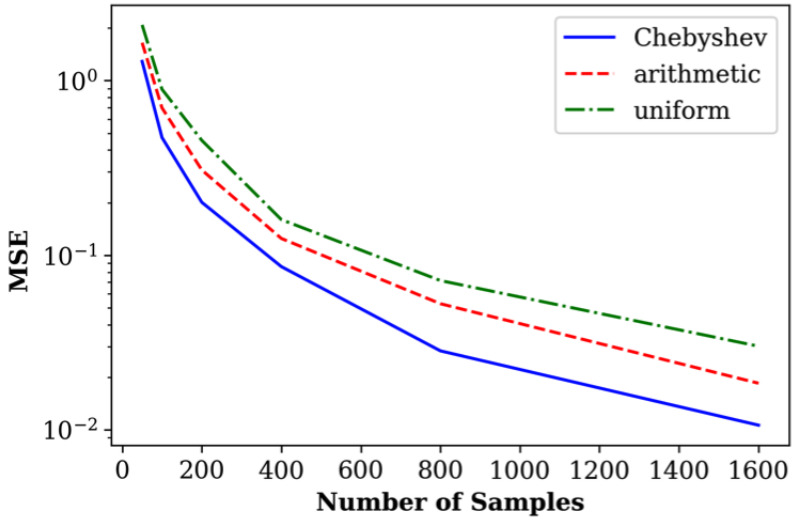
the benchmark learner with Chebyshev nodes has a better convergence rate for finite number of samples. For example in Figures C.2 and C.3, for 1600 samples, MSE of the Chebyshev method is respectively %10 and %92 less than MSE of the arithmetic method.

In Figures C.4 (a) and (b) we compare the Bayes error estimates for ensemble estimator with Chebyshev nodes with different scaling coefficients $\alpha = 0.1, 0.3, 0.5, 1.0$ for binary classification problems respectively with 10-dimensional and 50-dimensional isotropic normal distributions with covariance matrix $2\mathbf{I}$, where the means are separated by 5 units in the first dimension.

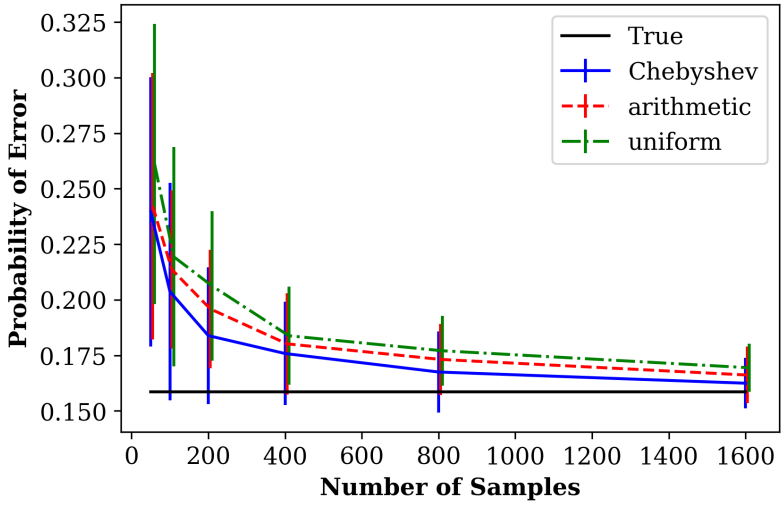
Figure C.5 compares of the Bayes error estimates for ensemble estimator with Chebyshev nodes with different scaling coefficients $\alpha = 0.1, 0.3, 0.5, 1.0$ for a 3-class classification problems, where the distributions of each class are 50-dimensional beta distributions with parameters $(3, 1)$, $(3, 1.5)$ and $(3, 2)$. All of the experiments in Figures C.4 and C.5 show that the performance of the estimator does not significantly vary for the scaling factor in the range $\alpha \in [0.3, 0.5]$ and a good performance can be achieved for the scaling factor $\alpha \in [0.3, 0.5]$.

Figure C.6 compares the optimal benchmark learner with the Bayes error lower and upper bounds using HP-divergence, for a 3-class classification problem with 10-dimensional Rayleigh distributions with parameters $a = 2, 4, 6$. While the HP-divergence bounds have a large bias, the proposed benchmark learner converges to the true value by increasing sample size.

In Figure C.7 we compare the optimal benchmark learner (Chebyshev method) with XGBoost and Random Forest classifiers, for a 4-class classification problem 100-dimensional isotropic mean-shifted Gaussian distributions with identity covariance matrix, where the means are shifted by 5 units in the first dimension. The

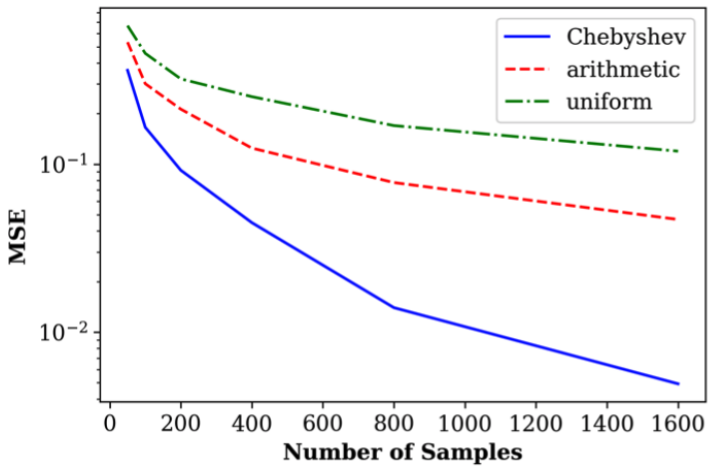


(a) Mean square error

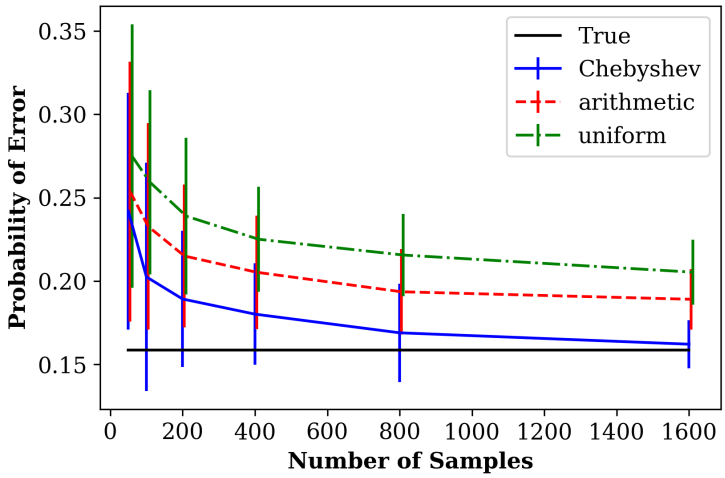


(b) Mean estimates with %95 confidence intervals

Figure C.2: Comparison of the Bayes error estimates for different methods of Chebyshev, arithmetic and uniform weight assigning methods for a binary classification problem with 4-dimensional isotropic normal distributions. The Chebyshev method provides a better convergence rate.

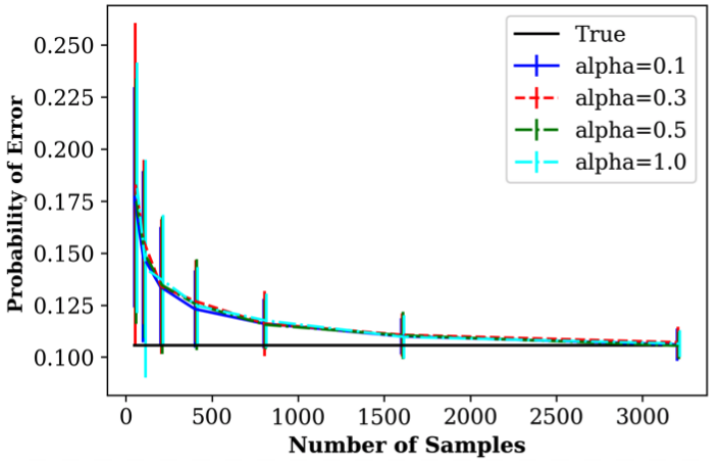


(a) Mean square error

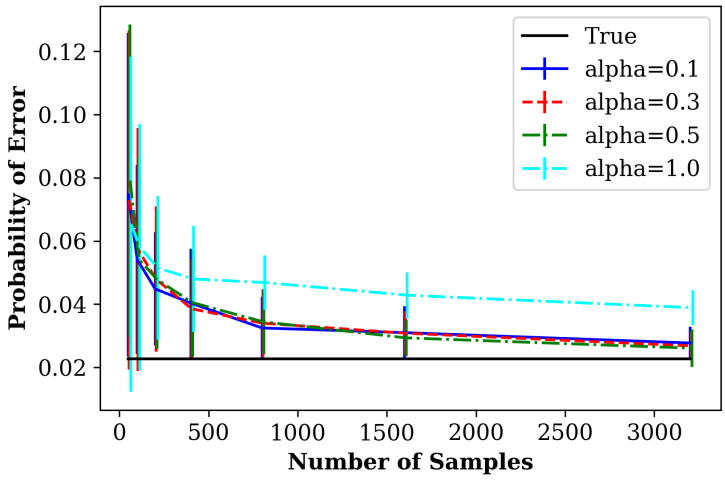


(b) Mean estimates with %95 confidence intervals

Figure C.3: Comparison of the Bayes error estimates for different methods of Chebyshev, arithmetic and uniform weight assigning methods for a binary classification problem with 100-dimensional isotropic normal distributions. The Chebyshev method provides a better convergence rate compared to the arithmetic and uniform methods.



(a) Mean square error



(b) Mean estimates with %95 confidence intervals

Figure C.4: Comparison of the Bayes error estimates for ensemble estimator with Chebyshev nodes with different scaling coefficients $\alpha = 0.1, 0.3, 0.5, 1.0$ for binary classification problems with (a) 10-dimensional and (b) 100-dimensional isotropic normal distributions with covariance matrix $2\mathbf{I}$, where the means are shifted by 5 units in the first dimension.

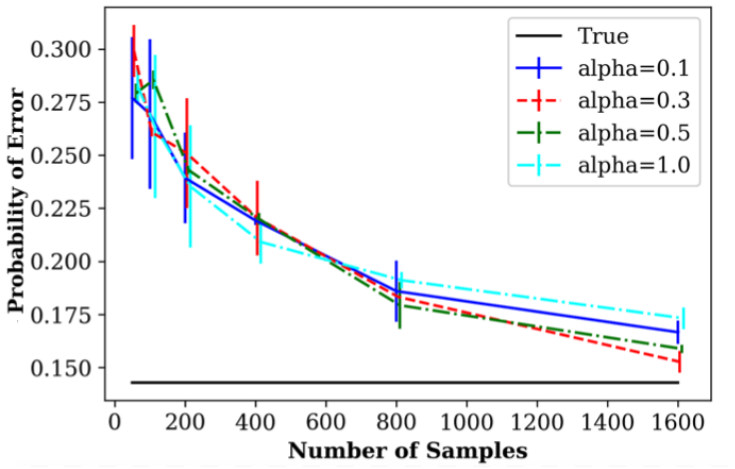


Figure C.5: Comparison of the Bayes error estimates for ensemble estimator with Chebyshev nodes with different scaling coefficients $\alpha = 0.1, 0.3, 0.5, 1.0$ for a 3-class classification problems, where the distributions of each class are 50-dimensional beta distributions with parameters $(3, 1)$, $(3, 1.5)$ and $(3, 2)$.

benchmark learner predicts the error rate bound better than XGBoost and Random Forest classifiers.

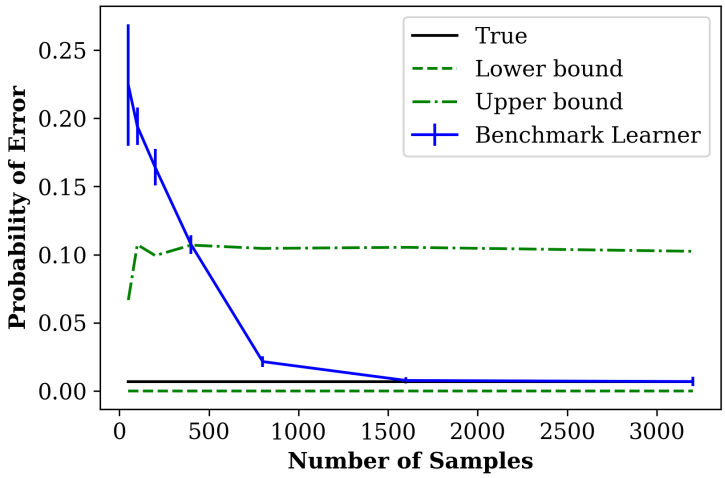


Figure C.6: Comparison of the optimal benchmark learner (Chebyshev method) with the Bayes error lower and upper bounds using HP-divergence, for a 3-class classification problem with 10-dimensional Rayleigh distributions with parameters $a = 2, 4, 6$. While the HP-divergence bounds have a large bias, the proposed benchmark learner converges to the true value by increasing sample size.

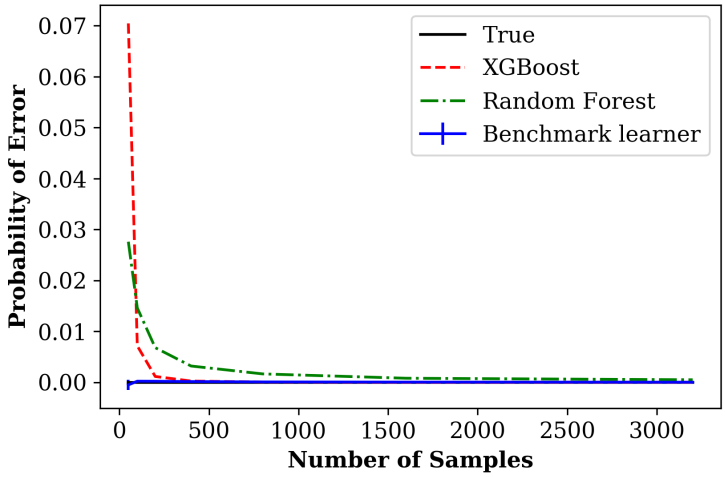


Figure C.7: Comparison of the optimal benchmark learner (Chebyshev method) with XGBoost and Random Forest classifiers, for a 4-class classification problem 100-dimensional isotropic mean-shifted Gaussian distributions with identity covariance matrix, where the means are shifted by 5 units in the first dimension. The benchmark learner predicts the Bayes error rate better than XGBoost and Random Forest classifiers.

APPENDIX D

D.1 Variance Proof

Proof of Theorem VII.3: The proof is based on Efron-Stein inequality. We follow similar steps used to prove the variance of NNR estimator in [94]. Note that the proof for variance of $\rho_i = N_i/(M_i)$ is contained in the the variance proof for $\widehat{D}_g(X,Y)$. Assume that we have two sets of nodes X_i, $1 \leq i \leq N$ and Y_j for $1 \leq j \leq M$. Here for simplicity we assume that $N = M$, however, the extension of the proof to the case when M and N are not equal, is straightforward, by considering a number of virtual points, as considered in [94]. Define $Z_i := (X_i, Y_i)$. For using the EfronStein inequality on $Z := (Z_1, ..., Z_N)$, we consider another independent copy of Z as $Z' := (Z'_1, ..., Z'_N)$ and define $Z^{(i)} := (Z_1, ..., Z_{i-1}, Z'_i, Z_{i+1}, ..., Z_N)$. Define $\widehat{D}_g(Z) := \widehat{D}_g(X,Y)$. By applying EfronStein inequality we have

$$
\begin{aligned}
\mathbb{V}\left([]\,\widehat{D}_g(Z)\right] &\leq \frac{1}{2}\sum_{i=1}^{N}\mathbb{E}\left[\left(\widehat{D}_g(Z)-\widehat{D}_g(Z^{(i)})\right)^2\right] \\
&= \frac{N}{2}\mathbb{E}\left[\left(\widehat{D}_g(Z)-\widehat{D}_g(Z^{(1)})\right)^2\right] \\
&\leq \frac{N}{2}\mathbb{E}\left[\left(\frac{1}{N}\sum_{\substack{i\leq F\\ M_i>0}} M_i\widetilde{g}\left(\frac{\eta N_i}{M_i}\right)-\frac{1}{N}\sum_{\substack{i\leq F\\ M_i>0}} M_i^{(1)}\widetilde{g}\left(\frac{\eta N_i^{(1)}}{M_i^{(1)}}\right)\right)^2\right] \\
&= \frac{1}{2N}\mathbb{E}\left[\left(\sum_{\substack{i\leq F\\ M_i>0}}\left(M_i\widetilde{g}\left(\frac{\eta N_i}{M_i}\right)-M_i^{(1)}\widetilde{g}\left(\frac{\eta N_i^{(1)}}{M_i^{(1)}}\right)\right)\right)^2\right] \\
&= \frac{1}{2N}O\left(1\right) = O(\frac{1}{N}). \qquad \text{(D.1)}
\end{aligned}
$$

where in the last line we used the fact that M_i and M_i' can be different just for two of $i \leq F$, and that difference is just $O(1)$. So, the proof is complete. □

D.2 Proof of Theorem VII.5

Assume that the densities have bounded derivatives up to the order q. Then the Taylor expansion of $f(y)$ around $f(x)$ is as follows

$$
f(y) = f(x) + \sum_{|i|\leq q}\frac{D^i f(x)}{i!}\|y-x\|^i + O\left(\|y-x\|^q\right). \qquad \text{(D.2)}
$$

Therefore, similar to (6.26) and using (6.28) we can write

$$
\begin{aligned}
\mathbb{E}\left[N_i'\right] &= N\int_{x\in B_i} f_1(x)dx \\
&= N\int_{x\in B_i} f(Y_i) + \sum_{|j|\leq q}\frac{D^j f(Y_i)}{j!}\|x-Y_i\|^j + O\left(\|x-Y_i\|^q\right)dx \\
&= N\epsilon^d f_1(Y_i) + N\sum_{|j|\leq q}\frac{D^j f(Y_i)}{j!}C_j(Y_i)\epsilon^{|j|+d} + O\left(NC_q(Y_i)\epsilon^{q+d}\right) \\
&= N\epsilon^d\left([]\,f_1(Y_i) + \sum_{l=1}^{q} C_l'(Y_i)\epsilon^l + O\left(C_q(Y_i)\epsilon^q\right)\right], \qquad \text{(D.3)}
\end{aligned}
$$

where

$$C'_{|j|}(Y_i) := \sum_{|j|\leq q} \frac{D^j f(Y_i)}{j!} C_j(Y_i).$$

Similarly we obtain

(D.4)

$$\mathbb{E}\left[(M'_i)^{-1}\right] = M^{-1}\epsilon^{-d}\left([]\, f_2(Y_i) + \sum_{l=1}^{q} C'_l(Y_i)\epsilon^l + O\left(C_q(Y_i)\epsilon^q\right)\right]^{-1}\left(1 + O\left(\frac{1}{M\epsilon^d f_2(Y_i)}\right)\right).$$

The rest of the proof follows by using the same steps as used in equations (6.31)-(6.33), and we get

$$\mathbb{B}\left([]\, \widehat{D}_g(X,Y)\right] = \sum_{i=1}^{q} C''_i \epsilon^i + O\left(\frac{1}{N\epsilon^d}\right), \tag{D.5}$$

where $C''_1, ..., C''_2$ are constants. Now are ready to apply the ensemble theorem ([82], Theorem 4). Let $\mathcal{T} := \{t_1, ..., t_T\}$ be a set of index values with $t_i < c$, where $c > 0$ is a constant. Let $\epsilon(t) := \left\lfloor tN^{-1/2d} \right\rfloor$. The proof completes by using the ensemble theorem in ([82], Theorem 4) with the parameters $\psi_i(t) = t^{i/d}$ and $\phi'_{i,d}(N) = \phi_{i,\kappa}(N)/N^{i/d}$. So the following weighted ensemble has the MSE convergence rate of $O(1/N)$:

$$\widehat{D}_w := \sum_{t\in\mathcal{T}} w(t)\widehat{D}_{\epsilon(t)}. \tag{D.6}$$

C. Proof of Theorem VI.8

We first argue that amortized runtime complexity of the online estimation algorithm is order $O(1)$ for each update after adding new samples. Note that when we add a new pair of samples X_{N+1} and Y_{N+1}, if $N+1 \neq 2^k$ for some integer k, we only find $H(X_{N+1})$ and $H(Y_{N+1})$ and update the corresponding M_i and N_i, which take a

constant time. But, only when $N+1=2^k$ for some integer k, we need $O(N)$ time complexity to update ϵ and therefore the hash function. Thus, if we have $N=2^k$ nodes added to the estimation algorithm, the total complexity due to rehashing, T_H, is as follows:

$$T_H = 1+2+2^2+...+2^k = 2^{k+1}-1 = 2N-1. \tag{D.7}$$

So, the amortized runtime complexity per each time step is $O(\frac{2N-1}{N}) = O(1)$. So overall, the amortized computational complexity is order $O(1)$. Finally, note that since we update ϵ when N doubles, it is at most by a factor of 2 away from the optimum ϵ. Since constant factor doesn't affect the asymptotic order of the bias error, the bias bound always holds for online estimation algorithm.

APPENDIX E

E.1 Bias Proof

We first prove a theorem that establishes an upper bound on the number of vertices in V and U.

Lemma E.1. *Cardinality of the sets U and V are upper bounded as $|V| \leq O\left(\epsilon^{-d}\right)$ and $|U| \leq O\left(\epsilon^{-d}\right)$, respectively.*

Proof. Let $\{\widetilde{X}_i\}_{i=1}^{L_X}$ and $\{\widetilde{Y}_i\}_{i=1}^{L_Y}$ respectively denote distinct outputs of H_1 with the N i.i.d points X_k and Y_k as input. Then according to [92] (Lemma 4.1), we have

$$L_X \leq O\left(\epsilon^{-d}\right), \quad L_Y \leq O\left(\epsilon^{-d}\right). \tag{E.1}$$

Simply, because of the deterministic feature of H_2, the number of its distinct inputs is greater than or equal to the number of its outputs. So, $|V| \leq L_X$ and $|U| \leq L_Y$. Using the bounds in (E.1) completes the proof. □

The bias proof is based on analyzing the hash function defined in (7.4). The proof consists of two main steps: 1) Finding the expectation of hash collisions of H_1; and 2) Analyzing the collision error of H_2. An important point about H_1 and H_2 is that collision of H_1 plays a crucial role in our estimator, while the collision of H_2 adds

extra bias to the estimator. We introduce the following events to formally define these two biases:

E_{ij} :The event that there is an edge between the vertices v_i and u_j.

$E_{\mathcal{E}}$:The event that $\mathcal{E}$ is the set of all edges in G, i.e. $\mathcal{E} = E_G$.

$E_{v_i}^{>0}$:The event that there is at least one vector from $\{\widetilde{X}_i\}_{i=1}^{L_X}$ that maps to v_i using H_2 .

$E_{v_i}^{=1}$:The event that there is exactly one vector from $\{\widetilde{X}_i\}_{i=1}^{L_X}$ that maps to v_i using H_2 .

(E.2)

$E_{v_i}^{>1}$:The event that there are at least two vectors from $\{\widetilde{X}_i\}_{i=1}^{L_X}$ that map to v_i using H_2 .

$E_{u_i}^{>0}$, $E_{u_i}^{=1}$ and $E_{u_i}^{>1}$ are defined similarly. Further, let for any event E, $\overline{E}$ denote the complementary event. Let $E_{ij}^{=1} := E_{v_i}^{=1} \cap E_{u_i}^{=1}$. Finally, we define $E^{=1} := \left(\cap_{i=1}^{L_X} E_{v_i}^{=1}\right) \cap \left(\cap_{j=1}^{L_Y} E_{u_j}^{=1}\right)$, which represent the event of no collision.

Consider the notation $\widetilde{I}(X,Y) := \sum_{e_{ij}\in E_G} \omega_i \omega_j' \widetilde{g}\left(\omega_{ij}\right)$ (Notice the difference from the definition in (7.7)). We can derive its expectation as

$$
\begin{aligned}
\mathbb{E}\left[\widetilde{I}(X,Y)\right] &= \mathbb{E}\left[\sum_{e_{ij}\in E_G} \omega_i \omega_j' \widetilde{g}\left(\omega_{ij}\right) \Big| E_G\right] \\
&= \sum_{e_{ij}\in E_G} \mathbb{E}\left[\omega_i \omega_j' \widetilde{g}\left(\omega_{ij}\right) | E_{ij}\right] \\
&= \sum_{e_{ij}\in E_G} P(E_{ij}^{=1}|E_{ij}) \mathbb{E}\left[\omega_i \omega_j' \widetilde{g}\left(\omega_{ij}\right) \Big| E_{ij}^{=1}, E_{ij}\right] \\
&\quad + \sum_{e_{ij}\in E_G} P\left(\overline{E_{ij}^{=1}}|E_{ij}\right) \mathbb{E}\left[\omega_i \omega_j' \widetilde{g}\left(\omega_{ij}\right) \Big| \overline{E_{ij}^{=1}}, E_{ij}\right].
\end{aligned} \tag{E.3}
$$

Note that the second term in (E.3) is the bias due to collision of H_2 and we denote this term by $\mathbb{B}_H$.

E.1.1 Bias Due to Collision

The following lemma states an upper bound on the bias error caused by H_2.

Lemma E.2. *The bias error due to collision of H_2 is upper bounded as*

$$\mathbb{B}_H \leq O\left(\frac{1}{\epsilon^d N}\right). \tag{E.4}$$

Before proving this lemma, we provide the following lemma.

Lemma E.3. *$P(E_{ij}^{=1}|E_{ij})$ is given by*

$$P(E_{ij}^{=1}|E_{ij}) = 1 - O\left(\frac{1}{\epsilon^d N}\right). \tag{E.5}$$

Proof. Let $\widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{Y}} = \widetilde{\mathbf{y}}$ respectively abbreviate the equations $\widetilde{X}_1 = \widetilde{x}_1, ..., \widetilde{X}_{L_X} = \widetilde{x}_{L_X}$ and $\widetilde{Y}_1 = \widetilde{y}_1, ..., \widetilde{Y}_{L_Y} = \widetilde{y}_{L_Y}$. Let $\widetilde{\mathbf{x}} := \{\widetilde{x}_1, \widetilde{x}_2, ..., \widetilde{x}_{L_X}\}$ and $\widetilde{\mathbf{y}} := \{\widetilde{y}_1, \widetilde{y}_2, ..., \widetilde{y}_{L_Y}\}$. Define $\widetilde{\mathbf{z}} := \widetilde{\mathbf{x}} \cup \widetilde{\mathbf{y}}$ and $L_Z := |\widetilde{\mathbf{z}}|$.

$$P(E_{ij}^{=1}|E_{ij}) = \sum_{\widetilde{\mathbf{x}},\widetilde{\mathbf{y}}} P\left(\widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}, \widetilde{\mathbf{Y}} = \widetilde{\mathbf{y}}|E_{ij}\right) P(E_{ij}^{=1}|E_{ij}, \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}, \widetilde{\mathbf{Y}} = \widetilde{\mathbf{y}}). \tag{E.6}$$

Define $a = 2$ for the case $i \neq j$ and $a = 1$ for the case $i = j$. Then we have

$$\begin{aligned}
P(E_{ij}^{=1}|E_{ij}) &= \sum_{\widetilde{\mathbf{x}},\widetilde{\mathbf{y}}} P\left(\widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}, \widetilde{\mathbf{Y}} = \widetilde{\mathbf{y}}|E_{ij}\right) O\left(\left(\frac{F-a}{F}\right)^{L_Z - a}\right) \\
&= \sum_{\widetilde{\mathbf{x}},\widetilde{\mathbf{y}}} P\left(\widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}, \widetilde{\mathbf{Y}} = \widetilde{\mathbf{y}}|E_{ij}\right) \left(1 - O\left(\frac{L_Z}{F}\right)\right) \\
&\leq \sum_{\widetilde{\mathbf{x}},\widetilde{\mathbf{y}}} P\left(\widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}, \widetilde{\mathbf{Y}} = \widetilde{\mathbf{y}}|E_{ij}\right) \left(1 - O\left(\frac{L_X + L_Y}{F}\right)\right) \\
&= \sum_{\widetilde{\mathbf{x}},\widetilde{\mathbf{y}}} P\left(\widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}, \widetilde{\mathbf{Y}} = \widetilde{\mathbf{y}}|E_{ij}\right) \left(1 - O\left(\frac{1}{\epsilon^d N}\right)\right) \\
&= \left(1 - O\left(\frac{1}{\epsilon^d N}\right)\right) \sum_{\widetilde{\mathbf{x}},\widetilde{\mathbf{y}}} P\left(\widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}, \widetilde{\mathbf{Y}} = \widetilde{\mathbf{y}}|E_{ij}\right) \\
&= \left(1 - O\left(\frac{1}{\epsilon^d N}\right)\right), \qquad \text{(E.7)}
\end{aligned}$$

where in the fourth line we have used (E.1). □

Proof of E.2. N'_i and M'_j respectively are defined as the number of the input points $\mathbf{X}$ and $\mathbf{Y}$ mapped to the buckets $\widetilde{X}_i$ and $\widetilde{Y}_j$ using H_1. Define $\mathcal{A}_i := \{j : H_2(\widetilde{X}_j) = i\}$ and $\mathcal{B}_i := \{j : H_2(\widetilde{Y}_j) = i\}$. For each i we can rewrite N_i and M_i as

$$N_i = \sum_{j=1}^{L_X} \mathbb{1}_{\mathcal{A}_i}(j) N'_j, \;\; M_i = \sum_{j=1}^{L_Y} \mathbb{1}_{\mathcal{B}_i}(j) M'_j. \tag{E.8}$$

Thus,

$$\mathbb{B}_H \leq \sum_{i,j\in\mathcal{F}} P\left(E_{ij}^{>1}\right)\mathbb{E}\left[\mathbb{1}_{E_{ij}}\omega_i\omega_j'\widetilde{g}\left(\omega_{ij}\right)\middle|E_{ij}^{>1}\right]$$

$$= \sum_{i,j\in\mathcal{F}} P\left(E_{ij}^{>1}\right)\left(P\left(E_{ij}|E_{ij}^{>1}\right)\mathbb{E}\left[\omega_i\omega_j'\widetilde{g}\left(\omega_{ij}\right)\middle|E_{ij}^{>1},E_{ij}\right] + P\left(\overline{E_{ij}}|E_{ij}^{>1}\right)\mathbb{E}\left[\omega_i\omega_j'\widetilde{g}\left(\omega_{ij}\right)\middle|E_{ij}^{>1},\overline{E_{ij}}\right]\right)$$

(E.9)

$$= \sum_{i,j\in\mathcal{F}} P\left(E_{ij}\right)P\left(E_{ij}^{>1}|E_{ij}\right)\mathbb{E}\left[\omega_i\omega_j'\widetilde{g}\left(\omega_{ij}\right)\middle|E_{ij}^{>1},E_{ij}\right]$$

(E.10)

$$\leq O\left(\frac{U}{\epsilon^d N}\right)\sum_{i,j\in\mathcal{F}} P\left(E_{ij}\right)\mathbb{E}\left[\omega_i\omega_j'\middle|E_{ij}^{>1},E_{ij}\right]$$

$$= O\left(\frac{U}{\epsilon^d N^3}\right)\sum_{i,j\in\mathcal{F}} P\left(E_{ij}\right)\mathbb{E}\left[N_iM_j\middle|E_{ij}^{>1},E_{ij}\right]$$

$$= O\left(\frac{U}{\epsilon^d N^3}\right)\sum_{\widetilde{\mathbf{x}},\widetilde{\mathbf{y}}} p_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}}}\left(\widetilde{\mathbf{x}},\widetilde{\mathbf{y}}\right)\sum_{i,j\in\mathcal{F}} P\left(E_{ij}\right)\mathbb{E}\left[N_iM_j\middle|E_{ij}^{>1},E_{ij},\widetilde{\mathbf{X}}=\widetilde{\mathbf{x}},\widetilde{\mathbf{Y}}=\widetilde{\mathbf{y}}\right]$$

$$= O\left(\frac{U}{\epsilon^d N^3}\right)\sum_{\widetilde{\mathbf{x}},\widetilde{\mathbf{y}}} p_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}}}\sum_{i,j\in\mathcal{F}}$$

(E.11)

$$P\left(E_{ij}\right)\mathbb{E}\left[\left(\sum_{r=1}^{L_X}\mathbb{1}_{\mathcal{A}_i}(r)N_r'\right)\left(\sum_{s=1}^{L_Y}\mathbb{1}_{\mathcal{B}_j}(s)M_s'\right)\middle|E_{ij}^{>1},E_{ij},\widetilde{\mathbf{X}}=\widetilde{\mathbf{x}},\widetilde{\mathbf{Y}}=\widetilde{\mathbf{y}}\right]$$

$$= O\left(\frac{U}{\epsilon^d N^3}\right)\sum_{\widetilde{\mathbf{x}},\widetilde{\mathbf{y}}} p_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}}}\sum_{i,j\in\mathcal{F}}$$

$$P\left(E_{ij}\right)\sum_{r=1}^{L_X}\sum_{s=1}^{L_Y}\mathbb{E}\left[\left(\mathbb{1}_{\mathcal{A}_i}(r)\right)\left(\mathbb{1}_{\mathcal{B}_j}(s)\right)\middle|E_{ij}^{>1},E_{ij},\widetilde{\mathbf{X}}=\widetilde{\mathbf{x}},\widetilde{\mathbf{Y}}=\widetilde{\mathbf{y}}\right]\mathbb{E}\left[N_r'M_s'|E_{ij},\widetilde{\mathbf{X}}=\widetilde{\mathbf{x}},\widetilde{\mathbf{Y}}=\widetilde{\mathbf{y}}\right]$$

$$= O\left(\frac{U}{\epsilon^d N^3}\right)\sum_{\widetilde{\mathbf{x}},\widetilde{\mathbf{y}}} p_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}}}\sum_{i,j\in\mathcal{F}} P\left(E_{ij}\right)$$

(E.12)

$$\sum_{r=1}^{L_X}\sum_{s=1}^{L_Y} P\left(r\in\mathcal{A}_i, s\in\mathcal{B}_j\middle|E_{ij}^{>1},E_{ij},\widetilde{\mathbf{X}}=\widetilde{\mathbf{x}},\widetilde{\mathbf{Y}}=\widetilde{\mathbf{y}}\right)\mathbb{E}\left[N_r'M_s'|E_{ij},\widetilde{\mathbf{X}}=\widetilde{\mathbf{x}},\widetilde{\mathbf{Y}}=\widetilde{\mathbf{y}}\right],$$

where in (E.9) we have used the Bayes rule, and the fact that $\widetilde{g}\left(\omega_{ij}\right)=0$ conditioned on the event $\overline{E_{ij}}$. In (E.10) we have used the bound in Lemma E.3, and

the upper bound on $\widetilde{g}(\omega_{ij})$. Equation (E.11) is due to (E.8). Now we simplify $P\left(r \in \mathcal{A}_i, s \in \mathcal{B}_j \middle| E_{ij}^{>1}, E_{ij}, \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}, \widetilde{\mathbf{Y}} = \widetilde{\mathbf{y}}\right)$ in (E.12) as follows. First assume that $\tilde{X}_r \neq \tilde{Y}_s$.

$$\begin{aligned} P\left(r \in \mathcal{A}_i, s \in \mathcal{B}_j \middle| E_{ij}^{>1}, E_{ij}, \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}, \widetilde{\mathbf{Y}} = \widetilde{\mathbf{y}}\right) &\leq P\left(r \in \mathcal{A}_i, s \in \mathcal{B}_j \middle| E_{v_i}^{>1}, E_{u_j}^{>1}, \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}, \widetilde{\mathbf{Y}} = \widetilde{\mathbf{y}}\right) \\ &= P\left(r \in \mathcal{A}_i \middle| E_{v_i}^{>1}, \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}\right) P\left(s \in \mathcal{B}_j \middle| E_{u_j}^{>1}, \widetilde{\mathbf{Y}} = \widetilde{\mathbf{y}}\right), \end{aligned} \tag{E.13}$$

where the second line is because the hash function H_2 is random and independent for different inputs. $P\left(r \in \mathcal{A}_i \middle| E_{v_i}^{>1}, \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}\right)$ in (E.13) can be written as

$$P\left(r \in \mathcal{A}_i \middle| E_{v_i}^{>1}, \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}\right) = \frac{P\left(r \in \mathcal{A}_i, E_{v_i}^{>1} \middle| \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}\right)}{P\left(E_{v_i}^{>1} \middle| \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}\right)}. \tag{E.14}$$

We first find $P\left(E_{v_i}^{>1} \middle| \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}\right)$:

$$\begin{aligned} P\left(E_{v_i}^{>1} \middle| \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}\right) &= 1 - P\left(E_{v_i}^{=0} \middle| \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}\right) - P\left(E_{v_i}^{=1} \middle| \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}\right) \\ &= 1 - \left(\frac{F-1}{F}\right)^{L_X} - \left(\frac{L_X}{F}\left(\frac{F-1}{F}\right)^{L_x - 1}\right) \\ &= \frac{L_X^2}{2F^2} + o\left(\frac{L_X^2}{2F^2}\right). \end{aligned} \tag{E.15}$$

Next, we find $P\left(r \in \mathcal{A}_i, E_{v_i}^{>1} \middle| \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}\right)$ in (E.14) as follows.

$$\begin{aligned} P\left(r \in \mathcal{A}_i, E_{v_i}^{>1} \middle| \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}\right) &= P\left(E_{v_i}^{>1} \middle| r \in \mathcal{A}_i, \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}\right) P\left(r \in \mathcal{A}_i \middle| \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}\right) \\ &= \left(1 - \left(\frac{F-1}{F}\right)^{L_X - 1}\right)\left(\frac{1}{F}\right) = O\left(\frac{L_X}{F^2}\right) \end{aligned} \tag{E.16}$$

Thus, using (E.15) and (E.16) yields

$$P\left(r \in \mathcal{A}_i \middle| E_{v_i}^{>1}, \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}\right) = O\left(\frac{1}{L_X}\right). \tag{E.17}$$

Similarly, we have

$$P\left(s \in \mathcal{B}_j \middle| E_{u_j}^{>1}, \widetilde{\mathbf{Y}} = \widetilde{\mathbf{y}}\right) = O\left(\frac{1}{L_Y}\right). \tag{E.18}$$

Now assume the case $\tilde{X}_r = \tilde{Y}_s$. Then since $H_2(\tilde{X}_r) = H_2(\tilde{Y}_s)$, we can simplify $P\left(r \in \mathcal{A}_i, s \in \mathcal{B}_j \middle| E_{ij}^{>1}, E_{ij}, \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}, \widetilde{\mathbf{Y}} = \widetilde{\mathbf{y}}\right)$ in (E.12) as

(E.19)

$$P\left(r \in \mathcal{A}_i, s \in \mathcal{B}_j \middle| E_{ij}^{>1}, E_{ij}, \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}, \widetilde{\mathbf{Y}} = \widetilde{\mathbf{y}}\right) = \delta_{ij} P\left(r \in \mathcal{A}_i \middle| E_{v_i}^{>1}, \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}, \widetilde{\mathbf{Y}} = \widetilde{\mathbf{y}}\right).$$

Recalling the definition $\widetilde{\mathbf{z}} := \widetilde{\mathbf{x}} \cup \widetilde{\mathbf{y}}$ and $L_Z := |\widetilde{\mathbf{z}}|$, similar to

$$P\left(r \in \mathcal{A}_i \middle| E_{v_i}^{>1}, \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}, \widetilde{\mathbf{Y}} = \widetilde{\mathbf{y}}\right) = O\left(\frac{1}{L_Z}\right). \tag{E.20}$$

By using equations (E.13), (E.17), (E.18) and (E.20) in (E.12), we can write the following upper bound for the bias estimator due to collision.

$$
\begin{aligned}
\mathbb{B}_H &\leq O\left(\frac{U}{\epsilon^d N^3}\right) \sum_{\widetilde{\mathbf{x}},\widetilde{\mathbf{y}}} p_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}}} \sum_{i,j\in\mathcal{F}} P\left(E_{ij}\right) \sum_{r=1}^{L_X}\sum_{s=1}^{L_Y} \mathbb{E}\left[N'_r M'_s | E_{ij}, \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}, \widetilde{\mathbf{Y}} = \widetilde{\mathbf{y}}\right] \\
&\quad \left(O\left(\frac{1}{L_X L_Y}\right) + \delta_{ij} O\left(\frac{1}{L_Z}\right)\right) \\
&= O\left(\frac{U}{\epsilon^d N^3}\right) \sum_{\widetilde{\mathbf{x}},\widetilde{\mathbf{y}}} p_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}}} \sum_{i,j\in\mathcal{F}} P\left(E_{ij}\right) \mathbb{E}\left[\sum_{r=1}^{L_X} N'_r \sum_{s=1}^{L_Y} M'_s | E_{ij}, \widetilde{\mathbf{X}} = \widetilde{\mathbf{x}}, \widetilde{\mathbf{Y}} = \widetilde{\mathbf{y}}\right] \\
&\quad \left(O\left(\frac{1}{L_X L_Y}\right) + \delta_{ij} O\left(\frac{1}{L_Z}\right)\right) \\
&= O\left(\frac{U}{\epsilon^d N^3}\right) \sum_{\widetilde{\mathbf{x}},\widetilde{\mathbf{y}}} p_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}}} \sum_{i,j\in\mathcal{F}} P\left(E_{ij}\right) N^2 \left(O\left(\frac{1}{L_X L_Y}\right) + \delta_{ij} O\left(\frac{1}{L_Z}\right)\right) \\
&= O\left(\frac{U}{\epsilon^d N^3}\right) \sum_{\widetilde{\mathbf{x}},\widetilde{\mathbf{y}}} p_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}}} \left(O\left(\frac{N^2}{L_X L_Y}\right) + O\left(\frac{N}{L_Z}\right)\right) \sum_{i,j\in\mathcal{F}} P\left(E_{ij}\right) \\
&= O\left(\frac{U}{\epsilon^d N^3}\right) \sum_{\widetilde{\mathbf{x}},\widetilde{\mathbf{y}}} p_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}}} \left(O\left(\frac{N^2}{L_X L_Y}\right) + O\left(\frac{N}{L_Z}\right)\right) \mathbb{E}\left[\sum_{i,j\in\mathcal{F}} \mathbb{1}_{E_{ij}}\right] \\
&\leq O\left(\frac{U}{\epsilon^d N^3}\right) \sum_{\widetilde{\mathbf{x}},\widetilde{\mathbf{y}}} p_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}}} \left(O\left(\frac{N^2}{L_X L_Y}\right) + O\left(\frac{N}{L_Z}\right)\right) (L_X L_Y) \\
&\leq O\left(\frac{1}{\epsilon^d N}\right). \qquad \text{(E.21)}
\end{aligned}
$$

□

E.1.2 Bias without Collision

A key idea in proving Theorem VII.2 is to show that the expectation of the edge weights ω_{ij} are proportional to the Radon-Nikodym derivative $dP_{XY}/dP_X P_Y$ at the points that correspond to the vertices v_i and u_j. This fact is stated in the following lemma:

Lemma E.4. *Under the assumptions **A1-A4**, and assuming that the density functions in **A3** have bounded derivatives up to order $q \geq 0$ we have:*

$$
\mathbb{E}\left[\omega_{ij}\right] = \frac{dP_{XY}}{dP_X P_Y} + \mathbb{B}(N, \epsilon, q, \gamma), \qquad \text{(E.22)}
$$

where

$$\text{(E.23)} \qquad \mathbb{B}(N,\epsilon,q,\gamma) := \begin{cases} O\left(\epsilon^{\gamma}\right) + O\left(\frac{1}{N\epsilon^{d}}\right), & q = 0 \\ \sum_{i=1}^{q} C_i \epsilon^{i} + O\left(\epsilon^{q}\right) + O\left(\frac{1}{N\epsilon^{d}}\right), & q \geq 1, \end{cases}$$

and C_i are real constants.

Note that since $\omega_{ij} = N_{ij}N/N_iM_j$, and N_{ij}, N_i and N_j are not independent variables, deriving the expectation is not trivial. In the following we give a lemma that provides conditions under which the expectation of a function of random variables is close to the function of expectations of the random variables. We will use the following lemma to simplify $\mathbb{E}\left[\omega_{ij}\right]$.

Lemma E.5. *Assume that $g(Z_1, Z_2, ..., Z_k) : \mathcal{Z}_1 \times ... \times \mathcal{Z}_k \rightarrow R$ is a Lipschitz continuous function with constant $H_g > 0$ with respect to each of variables $Z_i, 1 \leq i \leq k$. Let $\mathbb{V}\left[Z_i\right]$ and $\mathbb{V}\left[Z_i|X\right]$ respectively denote the variance and the conditional variance of each variable Z_i for a given variable X. Then we have*

(E.24)

a) $\left|\mathbb{E}\left[g\left(Z_1, Z_2, ..., Z_k\right)\right] - g\left(\mathbb{E}\left[Z_1\right], \mathbb{E}\left[Z_2\right], ..., \mathbb{E}\left[Z_k\right]\right)\right| \leq H_g \sum_{i=1}^{k} \sqrt{\mathbb{V}\left[Z_i\right]},$

(E.25)

b) $\left|\mathbb{E}\left[g\left(Z_1, Z_2, ..., Z_k\right)|X\right] - g\left(\mathbb{E}\left[Z_1|X\right], \mathbb{E}\left[Z_2|X\right], ..., \mathbb{E}\left[Z_k|X\right]\right)\right| \leq H_g \sum_{i=1}^{k} \sqrt{\mathbb{V}\left[Z_i|X\right]}.$

Proof.

$$
\begin{aligned}
&|\mathbb{E}\left[g\left(Z_1, Z_2, ..., Z_k\right)\right] - g\left(\mathbb{E}\left[Z_1\right], \mathbb{E}\left[Z_2\right], ..., \mathbb{E}\left[Z_k\right]\right)| \\
&= |\mathbb{E}\left[g\left(Z_1, Z_2, ..., Z_k\right) - g\left(\mathbb{E}\left[Z_1\right], \mathbb{E}\left[Z_2\right], ..., \mathbb{E}\left[Z_k\right]\right)\right]| \\
&\leq \mathbb{E}\left[|g\left(Z_1, Z_2, ..., Z_k\right) - g\left(\mathbb{E}\left[Z_1\right], \mathbb{E}\left[Z_2\right], ..., \mathbb{E}\left[Z_k\right]\right)|\right] \qquad \text{(E.26)} \\
&\leq \mathbb{E}[|g\left(Z_1, Z_2, ..., Z_k\right) - g\left(\mathbb{E}\left[Z_1\right], Z_2, ..., Z_k\right) + \\
&\quad + g\left(\mathbb{E}\left[Z_1\right], Z_2, ..., Z_k\right) - g\left(\mathbb{E}\left[Z_1\right], \mathbb{E}\left[Z_2\right], ..., Z_k\right) \\
&\quad + ... \\
&\quad + g\left(\mathbb{E}\left[Z_1\right], \mathbb{E}\left[Z_2\right], ..., \mathbb{E}\left[Z_{k-1}\right], Z_k\right) - g\left(\mathbb{E}\left[Z_1\right], \mathbb{E}\left[Z_2\right], ..., \mathbb{E}\left[Z_k\right]\right)|] \\
&\leq \mathbb{E}\left[\left|g\left(Z_1, Z_2, ..., Z_k\right) - g\left(\mathbb{E}\left[Z_1\right], Z_2, ..., Z_k\right)\right|\right] \\
&\quad + \mathbb{E}\left[\left|g\left(\mathbb{E}\left[Z_1\right], Z_2, ..., Z_k\right) - g\left(\mathbb{E}\left[Z_1\right], \mathbb{E}\left[Z_2\right], ..., Z_k\right)\right|\right] \\
&\quad + ... \\
&\quad + \mathbb{E}\left[\left|g\left(\mathbb{E}\left[Z_1\right], ..., \mathbb{E}\left[Z_{k-1}\right], Z_k\right) - g\left(\mathbb{E}\left[Z_1\right], ..., \mathbb{E}\left[Z_k\right]\right)\right|\right] \qquad \text{(E.27)} \\
&\leq H_g \mathbb{E}\left[|Z_1 - \mathbb{E}\left[Z_1\right]|\right] + H_g \mathbb{E}\left[|Z_2 - \mathbb{E}\left[Z_2\right]|\right] + ... + H_g \mathbb{E}\left[|Z_k - \mathbb{E}\left[Z_k\right]|\right] \qquad \text{(E.28)} \\
&\leq H_g \sum_{i=1}^{k} \sqrt{\mathbb{V}\left[Z_i\right]}. \qquad \text{(E.29)}
\end{aligned}
$$

In (E.26) and (E.27) we have used triangle inequalities. In (E.28) we have applied Lipschitz condition, and finally in (E.29) we have used CauchySchwarz inequality. Since the proofs of parts (a) and (b) are similar, we omit the proof of part (b). □

Lemma E.6. *Define $\nu_{ij} = N_{ij}/N$, and recall the definitions $\omega_{ij} = N_{ij}N/N_iN_j$, $\omega_i = N_i/N$, and $\omega'_j = N_j/N$. Then we can write*

$$
\mathbb{E}\left[\omega_{ij}\right] = \frac{\mathbb{E}\left[\nu_{ij}\right]}{\mathbb{E}\left[\omega_i\right]\mathbb{E}\left[\omega'_j\right]} + O\left(\sqrt{\frac{1}{N}}\right) \qquad \text{(E.30)}
$$

Proof. The proof follows by Lemma E.5 and the fact that $\mathbb{V}\left[\omega_{ij}\right] \leq O\left(1/N\right)$ (proved in Lemma E.10). □

Let x_D and x_C respectively denote the discrete and continuous components of the vector x, with dimensions d_D and d_C. Also let $f_{X_C}(x_C)$ and $p_{X_D}(x_D)$ respectively denote density and pmf functions of these components associated with the probability measure P_X. Let $S(x,r)$ be the set of all points that are within the distance $r/2$ of x in each dimension i, i.e.

$$S(x,r) : \{x | \forall i \leq d, |X_i - x_i| < r/2\}. \tag{E.31}$$

Denote $P_r(x) := P(x \in S(x,r))$. Then we have the following lemma.

Lemma E.7. *Let $r < s_{\mathcal{X}}$, where $s_{\mathcal{X}}$ is the smallest possible distance in the discrete components of the support set, $\mathcal{X}$. Under the assumption **A3**, and assuming that the density functions in **A3** have bounded derivatives up to the order $q \geq 0$, we have*

$$P_r(x) = P(X_D = x_D) r^{d_C} \left(f(x_C|x_D) + \mu(r, \gamma, q, \mathbf{C}_X) \right), \tag{E.32}$$

where

$$\mu(r, \gamma, q, \mathbf{C}_X) := \begin{cases} O\left(r^{\gamma}\right), & q = 0 \\ \sum_{i=1}^{q} C_i r^i + O\left(r^q\right), & q \geq 1. \end{cases} \tag{E.33}$$

In the above equation, $\mathbf{C}_X := (C_1, C_2, ..., C_q)$, and C_i are real constants depending on the probability measure P_X.

Proof. The proof is straightforward by using (**??**) for the case $q = 0$ (similar to (27)-(29) in [92]), and using the Taylor expansion of $f(x_C|x_D)$ for the case $q \geq 1$ (similar to (36)-(37) in [92]). □

Lemma E.8. *Let $H(x) = i, H(y) = j$. Under the assumptions **A1-A3**, and assuming that the density functions in **A3** have bounded derivatives up to the order $q \geq 0$, we have*

$$\mathbb{E}\left[\omega_{ij}|E_{\overline{ij}}^{\leq 1}\right] = \frac{dP_{XY}}{dP_X P_Y}(x,y) + \mu(\epsilon,\gamma,q,\mathbf{C}'_{XY}) + O\left(\frac{1}{\sqrt{N}}\right), \tag{E.34}$$

where $\mu(\epsilon,\gamma,q,\mathbf{C}'_{XY})$ *is defined in* (E.33).

Proof. Define $\nu_{ij} = N_{ij}/N$, and recall the definitions $\omega_{ij} = N_{ij}N/N_iN_j$, $\omega_i = N_i/N$, and $\omega'_j = N_j/N$. Using Lemma E.5 we have

$$\mathbb{E}\left[\omega_{ij}|E_{\overline{ij}}^{\leq 1}\right] = \frac{\mathbb{E}\left[\nu_{ij}|E_{\overline{ij}}^{\leq 1}\right]}{\mathbb{E}\left[\omega_i|E_{\overline{ij}}^{\leq 1}\right]\mathbb{E}\left[\omega'_j|E_{\overline{ij}}^{\leq 1}\right]} + O\left(\frac{1}{\sqrt{N}}\right) \tag{E.35}$$

Assume that $H(x) = i$. Let $\mathcal{X}$ have d_C and d_D continuous and discrete components, respectively. Also let $\mathcal{Y}$ have d'_C and d'_D continuous and discrete components, respectively. Then we can write

$$\begin{aligned}
\mathbb{E}\left[\omega_i|E_{\overline{ij}}^{\leq 1}\right] &= \frac{1}{N}\mathbb{E}\left[N_i|E_{\overline{ij}}^{\leq 1}\right] \\
&= P(X \in S(x,\epsilon)) \\
&= P(X_D = x_D)\epsilon^{d_C}\left(f(x_C|x_D) + \mu(\epsilon,\gamma,q,\mathbf{C}_X)\right), \qquad \text{(E.36)}
\end{aligned}$$

where in the third line we have used Lemma E.7. Similarly we can write

$$\mathbb{E}\left[\omega'_j|E_{\overline{ij}}^{\leq 1}\right] = P(Y_D = y_D)\epsilon^{d'_C}\left(f(y_C|y_D) + \mu(\epsilon,\gamma,q,\mathbf{C}_X)\right),$$

(E.37)

$$\mathbb{E}\left[\nu_{ij}|E_{\overline{ij}}^{\leq 1}\right] = P(X_D = x_D, Y_D = y_D)\epsilon^{(d_C+d'_C)}\left(f(x_C,y_C|x_D,y_D) + \mu(\epsilon,\gamma,q,\mathbf{C}_{XY})\right).$$

Using (E.36) and (E.37) in (E.35) results in

(E.38)

$$\mathbb{E}\left[\omega_{ij}|E_{\overline{ij}}^{\leq 1}\right] = \frac{P(X_D = x_D)P(Y_D = y_D)f(x_C|x_D)f(y_C|y_D)}{P(X_D = x_D, Y_D = y_D)f(x_C,y_C|x_D,y_D)} + \mu(\epsilon,\gamma,q,\mathbf{C'}_{XY}) + O\left(\frac{1}{\sqrt{N}}\right),$$

where $\mathbf{C}'_{XY}$ depends only on P_{XY}. Now note that using Lemma E.7, $\frac{dP_{XY}}{dP_X P_Y}(x,y)$ can be simplified as

(E.39)
$$\frac{dP_{XY}}{dP_X P_Y}(x,y) = \frac{\frac{dP_{XY,r}}{dr}(x,y)}{\frac{dP_{X,r}P_{Y,r}}{dr}(x,y)} = \frac{P(X_D = x_D)P(Y_D = y_D)f(x_C|x_D)f(y_C|y_D)}{P(X_D = x_D, Y_D = y_D)f(x_C, y_C|x_D, y_D)} + \mu(\epsilon, \gamma, q, \mathbf{C}''_{XY}).$$

Finally, using (E.39) in (E.38) gives

$$\mathbb{E}\left[\omega_{ij}|E_{ij}^{\leq 1}\right] = \frac{dP_{XY}}{dP_X P_Y}(x,y) + \mu(\epsilon, \gamma, q, \widetilde{\mathbf{C}}_{XY}) + O\left(\frac{1}{\sqrt{N}}\right), \tag{E.40}$$

where $H(x) = i, H(y) = j$. □

Proof of Lemma E.4. Lemma E.4 is a simple consequence of Lemma E.8. We have

$$\mathbb{E}\left[\omega_{ij}\right] = P\left(E_{ij}^{\leq 1}\right)\mathbb{E}\left[\omega_{ij}|E_{ij}^{\leq 1}\right] + P\left(E_{ij}^{>1}\right)\mathbb{E}\left[\omega_{ij}|E_{ij}^{>1}\right]. \tag{E.41}$$

Recall the definitions $\widetilde{\mathbf{X}} := \left(\widetilde{X}_1, \widetilde{X}_2, ..., \widetilde{X}_{L_X}\right)$ and $\widetilde{\mathbf{Y}} := \left(\widetilde{Y}_1, \widetilde{Y}_2, ..., \widetilde{Y}_{L_Y}\right)$ as the mapped $\mathbf{X}$ and $\mathbf{Y}$ points through H_1. Let $\widetilde{\mathbf{Z}} := \widetilde{\mathbf{X}} \cup \widetilde{\mathbf{Y}}$ and $L_Z := |\widetilde{\mathbf{Z}}|$. We first find $P\left(E_{ij}^{\leq 1}\right)$ as follows. For a fixed set $\widetilde{\mathbf{Z}}$ we have

$$\begin{aligned}
P\left(E_{ij}^{\leq 1}\right) &= P\left(E_{v_i}^{=0} \cap E_{u_j}^{=0}\right) + P\left(E_{v_i}^{=0} \cap E_{u_j}^{=1}\right) + P\left(E_{v_i}^{=1} \cap E_{u_j}^{=0}\right) + P\left(E_{v_i}^{=1} \cap E_{u_j}^{=1}\right) \\
&= \frac{(F-2)^{L_Z}}{F^{L_Z}} + \frac{L_Y(F-2)^{L_Z-1}}{F^{L_Z}} + \frac{L_X(F-2)^{L_Z-1}}{F^{L_Z}} + \frac{L_Y L_X(F-2)^{L_Z-2}}{F^{L_Z}} \\
&= 1 - O\left(\frac{L_Z}{F}\right) \\
&\leq 1 - O\left(\frac{L_X + L_Y}{F}\right) \\
&= 1 - O\left(\frac{1}{\epsilon^d N}\right). \qquad \text{(E.42)}
\end{aligned}$$

Now note that the second term in (E.41) is the bias due to collision of H_2, and similar to (E.21) it is upper bounded by $O\left(\frac{1}{\epsilon^d N}\right)$. Thus, (E.42) and (E.41) give rise to

$$\mathbb{E}\left[\omega_{ij}\right] = \frac{dP_{XY}}{dP_X P_Y}(x,y) + \mu(\epsilon,\gamma,q,\widetilde{\mathbf{C}}_{XY}) + O\left(\frac{1}{\sqrt{N}}\right) + O\left(\frac{1}{\epsilon^d N}\right). \tag{E.43}$$

which completes the proof.

□

In the following lemma we make a relation between the bias of an estimator and the bias of a function of that estimator.

Lemma E.9. *Assume that $g(x) : \mathcal{X} \to \mathbb{R}$ is infinitely differentiable. If $\widehat{Z}$ is a random variable estimating a constant Z with the bias $\mathbb{B}[\widehat{Z}]$ and the variance $\mathbb{V}[\widehat{Z}]$, then the bias of $g(\widehat{Z})$ can be written as*

$$\mathbb{E}\left[g(\widehat{Z}) - g(Z)\right] = \sum_{i=1}^{\infty} \xi_i \left(\mathbb{B}\left[\widehat{Z}\right]\right)^i + O\left(\sqrt{\mathbb{V}\left[\widehat{Z}\right]}\right), \tag{E.44}$$

where ξ_i are real constants.

Proof.

$$\begin{aligned}
\mathbb{E}\left[g\left(\widehat{Z}\right) - g(Z)\right] &= g\left(\mathbb{E}\left[\widehat{Z}\right]\right) - g(Z) + \mathbb{E}\left[g\left(\widehat{Z}\right) - g\left(\mathbb{E}\left[\widehat{Z}\right]\right)\right] \\
&= \sum_{i=1}^{\infty} \left(\mathbb{E}\left[\widehat{Z}\right] - Z\right)^i \frac{g^{(i)}(Z)}{i!} + O\left(\mathbb{E}\left[|g\left(\widehat{Z}\right) - g\left(\mathbb{E}\left[\widehat{Z}\right]\right)|\right]\right) \\
&= \sum_{i=1}^{\infty} \xi_i \left(\mathbb{B}\left[\widehat{Z}\right]\right)^i + O\left(\sqrt{\mathbb{V}\left[\widehat{Z}\right]}\right). \qquad \text{(E.45)}
\end{aligned}$$

In the second line we have used Taylor expansion for the first term, and triangle inequality for the second term. In the third line we have used the definition $\xi_i := g^{(i)}(Z)/i!$, and the CauchySchwarz inequality for the second term.

□

In the following we compute the expectation of the first term in (E.3) and prove Theorem VII.2.

Proof of Theorem VII.2. Recall that N'_i and M'_j respectively are defined as the number of the input points $\mathbf{X}$ and $\mathbf{Y}$ mapped to the buckets $\widetilde{X}_i$ and $\widetilde{Y}_j$ using H_1. Similarly, N'_{ij} is defined as the number of input pairs $(\mathbf{X}, \mathbf{Y})$ mapped to the bucket pair $\left(\widetilde{X}_i, \widetilde{Y}_j\right)$ using H_1. Define the notations $r(i) := H_2^{-1}(i)$ for $i \in \mathcal{F}$ and $s(x) := H_1(x)$ for $x \in \mathcal{X} \cup \mathcal{Y}$. Then from (E.38) since there is no collision of mapping with H_2 into v_i and u_j we have

$$\mathbb{E}\left[\frac{N'_{s(x)s(y)}N}{N'_{s(x)}N'_{s(y)}}\right] = \frac{dP_{XY}}{dP_X P_Y}(x,y) + \mu(\epsilon, \gamma, q, \widetilde{\mathbf{C}}_{XY}) + O\left(\frac{1}{\sqrt{N}}\right), \tag{E.46}$$

By using (E.42) and defining $\tilde{h}(x) = \tilde{g}(x)/x$ we can simplify the first term of (E.3) as

$$
\begin{aligned}
&\sum_{i,j\in\mathcal{F}} P\left(E_{ij}^{\leq 1}\right)\mathbb{E}\left[\mathbb{1}_{E_{ij}}\omega_i\omega_j'\widetilde{g}\left(\omega_{ij}\right)\middle|E_{ij}^{\leq 1}\right]\\
&=\left(1-O\left(\frac{1}{\epsilon^d N}\right)\right)\sum_{i,j\in\mathcal{F}}\mathbb{E}\left[\mathbb{1}_{E_{ij}}\omega_i\omega_j'\widetilde{g}\left(\omega_{ij}\right)\middle|E_{ij}^{\leq 1}\right]\\
&=\sum_{i,j\in\mathcal{F}}\mathbb{E}\left[\mathbb{1}_{E_{ij}}\frac{N_iM_j}{N^2}\widetilde{g}\left(\frac{N_{ij}N}{N_iM_j}\right)\middle|E_{ij}^{\leq 1}\right]+O\left(\frac{1}{\epsilon^d N}\right)\\
&=\sum_{i,j\in\mathcal{F}}\mathbb{E}\left[\mathbb{1}_{E_{ij}}\frac{N'_{r(i)}M'_{r(j)}}{N^2}\widetilde{g}\left(\frac{N'_{r(i)r(j)}N}{N'_{r(i)}M'_{r(j)}}\right)\right]+O\left(\frac{1}{\epsilon^d N}\right)\\
&=\sum_{i,j\in\mathcal{F}}\mathbb{E}\left[\mathbb{1}_{E_{ij}}\frac{N'_{r(i)r(j)}}{N}\widetilde{h}\left(\frac{N'_{r(i)r(j)}N}{N'_{r(i)}M'_{r(j)}}\right)\right]+O\left(\frac{1}{\epsilon^d N}\right)\\
&=\frac{1}{N}\sum_{i,j\in\mathcal{F}}\mathbb{E}\left[N'_{r(i)r(j)}\widetilde{h}\left(\frac{N'_{r(i)r(j)}N}{N'_{r(i)}M'_{r(j)}}\right)\right]+O\left(\frac{1}{\epsilon^d N}\right) \qquad \text{(E.47)}\\
&=\frac{1}{N}\mathbb{E}\left[\sum_{i,j\in\mathcal{F}}N'_{r(i)r(j)}\widetilde{h}\left(\frac{N'_{r(i)r(j)}N}{N'_{r(i)}M'_{r(j)}}\right)\right]+O\left(\frac{1}{\epsilon^d N}\right)\\
&=\frac{1}{N}\mathbb{E}\left[\sum_{i=1}^{N}\widetilde{h}\left(\frac{N'_{s(X)s(Y)}N}{N'_{s(X)}M'_{s(Y)}}\right)\right]+O\left(\frac{1}{\epsilon^d N}\right)\\
&=\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[\widetilde{h}\left(\frac{N'_{s(X)s(Y)}N}{N'_{s(X)}M'_{s(Y)}}\right)\right]+O\left(\frac{1}{\epsilon^d N}\right)\\
&=\mathbb{E}_{(X,Y)\sim P_{XY}}\left[\mathbb{E}\left[\widetilde{h}\left(\frac{N'_{s(X)s(Y)}N}{N'_{s(X)}M'_{s(Y)}}\right)\middle|X=x,Y=y\right]\right]+O\left(\frac{1}{\epsilon^d N}\right)\\
&=\mathbb{E}_{(X,Y)\sim P_{XY}}\left[\frac{dP_{XY}}{dP_XP_Y}\right]+\mu(\epsilon,\gamma,q,\overline{\mathbf{C}}_{XY})+O\left(\frac{1}{\sqrt{N}}\right)+O\left(\frac{1}{\epsilon^d N}\right). \qquad \text{(E.48)}
\end{aligned}
$$

(E.49)

(E.47) is due to the fact that $N'_{r(i)r(j)}=0$ if there is no edge between v_i and u_j. Also, (E.48) is due to (E.46).

From (E.48) and (E.3) we obtain

$$\mathbb{E}\left[\widetilde{I}(X,Y)\right] = \mathbb{E}\left[\sum_{e_{ij}\in E_G} \omega_i \omega_j' \widetilde{g}\left(\omega_{ij}\right)\right] = \mathbb{E}_{(X,Y)\sim P_{XY}}\left[\frac{dP_{XY}}{dP_X P_Y}\right] + \mu(\epsilon,\gamma,q,\overline{\mathbf{C}}_{XY}) +$$

$$O\left(\frac{1}{\sqrt{N}}\right) + O\left(\frac{1}{\epsilon^d N}\right). \tag{E.50}$$

Finally using Lemma E.9 results in (7.9).

□

E.2 Variance Proof

In this section we first prove bounds on the variances of the edge and vertex weights and then we provide the proof of Theorem VII.3.

Lemma E.10. *Under the assumptions **A1-A4**, the following variance bounds hold true.*

(E.51)

$$\mathbb{V}\left[\omega_i\right] \leq O\left(\frac{1}{N}\right), \qquad \mathbb{V}\left[\omega_j'\right] \leq O\left(\frac{1}{N}\right), \qquad \mathbb{V}\left[\omega_{ij}\right] \leq O\left(\frac{1}{N}\right), \qquad \mathbb{V}\left[\nu_{ij}\right] \leq O\left(\frac{1}{N}\right).$$

Proof. Here we only provide the variance proof of ω_i. The variance bounds of ω_j', ω_{ij} and ν_{ij} can be proved in the same way. The proof is based on Efron-Stein inequality. Define $Z_i := (X_i, Y_i)$. For using the EfronStein inequality on $\mathbf{Z} := (Z_1, ..., Z_N)$, we consider another independent copy of $\mathbf{Z}$ as $\mathbf{Z}' := (Z_1', ..., Z_N')$ and define $\mathbf{Z}^{(i)} := (Z_1, ..., Z_{i-1}, Z_i', Z_{i+1}, ..., Z_N)$. Define $\omega_i(\mathbf{Z})$ as the weight of vertex v_i in the dependence graph constructed by the set $\mathbf{Z}$. By applying EfronStein inequality [94] we have

$$\begin{aligned}
\mathbb{V}\left[\omega_i\right] &\leq \frac{1}{2}\sum_{i=1}^{N}\mathbb{E}\left[\left(\omega_i\left(\mathbf{Z}\right)-\omega_i\left(\mathbf{Z}^{(j)}\right)\right)^2\right] \\
&= \frac{1}{2N^2}\sum_{i=1}^{N}\mathbb{E}\left[\left(N_i\left(\mathbf{Z}\right)-N_i\left(\mathbf{Z}^{(j)}\right)\right)^2\right] \\
&\leq \frac{1}{2N^2}O\left(N\right) \\
&\leq O\left(\frac{1}{N}\right). \qquad \text{(E.52)}
\end{aligned}$$

In the third line we have used the fact that the absolute value of $N_i\left(\mathbf{Z}\right)-N_i\left(\mathbf{Z}^{(j)}\right)$ is at most 1.

□

Proof of Theorem VII.3 . We follow similar steps as the proof of Lemma E.10. Define $\widehat{I}_g(\mathbf{Z})$ as the mutual information estimation using the set $\mathbf{Z}$. By applying EfronStein inequality we have

$$\begin{aligned}
\mathbb{V}\left[\widehat{I}(X,Y)\right] &\leq \frac{1}{2}\sum_{k=1}^{N}\mathbb{E}\left[\left(\widehat{I}(\mathbf{Z})-\widehat{I}(\mathbf{Z}^{(k)})\right)^2\right] \\
&\leq \frac{N}{2}\mathbb{E}\left[\left(\sum_{e_{ij}\in E_G}\omega_i\left(\mathbf{Z}\right)\omega_j'\left(\mathbf{Z}\right)\widetilde{g}\left(\omega_{ij}\left(\mathbf{Z}\right)\right)-\sum_{e_{ij}\in E_G}\omega_i\left(\mathbf{Z}^{(k)}\right)\omega_j'\left(\mathbf{Z}^{(k)}\right)\widetilde{g}\left(\omega_{ij}\right)\left(\mathbf{Z}^{(k)}\right)\right)^2\right] \\
&= \frac{N}{2N^4}\mathbb{E}\left[\left(\sum_{e_{ij}\in E_G}N_i\left(\mathbf{Z}\right)M_j\left(\mathbf{Z}\right)\widetilde{g}\left(\frac{N_{ij}\left(\mathbf{Z}\right)N}{N_i\left(\mathbf{Z}\right)M_j\left(\mathbf{Z}\right)}\right)-\right. \\
&\qquad \text{(E.53)} \\
&\left.\sum_{e_{ij}\in E_G}N_i\left(\mathbf{Z}^{(k)}\right)M_j\left(\mathbf{Z}^{(k)}\right)\widetilde{g}\left(\frac{N_{ij}\left(\mathbf{Z}^{(k)}\right)N}{N_i\left(\mathbf{Z}^{(k)}\right)M_j\left(\mathbf{Z}^{(k)}\right)}\right)^2\right] \qquad \text{(E.54)} \\
&\leq \frac{1}{2N^3}\mathbb{E}\left[\left(\Sigma_{n_1}+\Sigma_{n_2}+\Sigma_{m_1}+\Sigma_{m_1}+D_{n_1m_1}+D_{n_2m_2}\right)^2\right].
\end{aligned}$$

Note that in equation (E.54), when (X_k, Y_k) is resampled, at most two of N_i for $i \in \mathcal{F}$ are changed exactly by one (one decrease and the other increase). The same statement holds true for M_j. Let these vertices be v_{n_1}, v_{n_2}, v_{m_1} and v_{m_2}. Also the pair collision counts N_{ij} are fixed except possibly $N_{n_1m_1}$ and $N_{n_2m_2}$ that may change by one. So, in the fourth line Σ_{n_1} and Σ_{n_2} account for the changes in MI estimation due to the changes in N_{n_1} and N_{n_2}, and Σ_{m_1} and Σ_{m_2} account for the changes in M_{m_1} and v_{m_2}, respectively. Finally $D_{n_1m_1}$ and $D_{n_2m_2}$ account for the changes in MI estimation due to the changes in $N_{n_1m_1}$ and $N_{n_2m_2}$. For example, Σ_{n_1} is precisely defined as follows:

$$\Sigma_{n_1} := \sum_{j:e_{mj}\in E_G} N_m M_j \widetilde{g}\left(\frac{N_{mj}N}{N_m N_j}\right) - (N_m+1)M_j\widetilde{g}\left(\frac{N_{mj}N}{(N_m+1)M_j}\right) \tag{E.55}$$

where we have used the notations N_i and $N_i^{(k)}$ instead of $N_i(\mathbf{Z})$ and $N_i(\mathbf{Z}^{(k)})$ for simplicity. Now note that by assumption **A4** we have

$$\begin{aligned} |\widetilde{g}\left(\frac{N_{mj}N}{N_m M_j}\right) - \widetilde{g}\left(\frac{N_{mj}N}{(N_m+1)M_j}\right)| &\leq G_g \left|\frac{N_{mj}N}{N_m M_j} - \frac{N_{mj}N}{(N_m+1)M_j}\right| \\ &\leq O\left(\frac{N_{mj}N}{N_m^2 M_j}\right). \end{aligned} \tag{E.56}$$

Thus, using (E.56), Σ_{n_1} can be upper bounded as follows

$$\Sigma_{n_1} \leq \sum_{j:e_{mj}\in E_G} O\left(\frac{N_{mj}N}{N_m^2}\right) = O\left(\frac{N}{N_m}\right) \leq O(N). \tag{E.57}$$

It can similarly be shown that N_{n_2}, Σ_{m_1}, Σ_{m_2}, $D_{n_1m_1}$ and $D_{n_2m_2}$ are upper bounded by $O(N)$. Thus, (E.54) simplifies as follows

$$\mathbb{V}\left[\widehat{I}(X,Y)\right] \leq \frac{36O(N^2)}{2N^3} = O(\frac{1}{N}). \tag{E.58}$$

□

E.3 Optimum MSE Rates of EDGE

In this short section we prove Theorem VII.5.

Proof of Theorem VII.5. The proof simply follows by using the ensemble theorem in ([82], Theorem 4) with the parameters $\psi_i(t) = t^i$ and $\phi_{i,d}(N) = N^{-i/2d}$ for the bias result in Theorem VII.2. Thus, the following weighted ensemble estimator (EDGE) can achieve the optimum parametric MSE convergence rate of $O(1/N)$ for $q \geq d$.

$$\widehat{I}_w := \sum_{t \in \mathcal{T}} w(t) \widehat{I}_{\epsilon(t)}, \tag{E.59}$$

□

APPENDIX F

F.1 Assumptions

In the proofs, in addition to the Hölder smoothness $\Sigma(s, H)$ on the densities,

we make the following assumptions on the densities and the functional g, which we adapt from [84], are

- ($\mathcal{A}$.0): Assume that the kernel K is a symmetric product kernel with bounded support in each dimension.
- ($\mathcal{A}$.1): Assume there exist constants $\epsilon_0, \epsilon_\infty$ such that $0 < \epsilon_0 \leq p(x) \leq \epsilon_\infty < \infty$, $\forall x \in \mathcal{S}$ and similarly that the marginal densities and joint pairwise densities are bounded above and below.
- ($\mathcal{A}$.2): Assume that each of the densities belong to $\Sigma(s, H)$ in the interior of their support sets with $s \geq 2$.
- ($\mathcal{A}$.3): Assume that $g(t_1/t_2)$ has an infinite number of mixed derivatives wrt t_1 and t_2.
- ($\mathcal{A}$.4): Assume that $\left|\frac{\partial^{k+l} g(t_1,t_2)}{\partial t_1^k \partial t_2^l}\right| / (k!l!)$, $k, l = 0, 1, \ldots$ are strictly upper bounded for $\epsilon_0 \leq t_1, t_2 \leq \epsilon_\infty$.

- ($\mathcal{A}.5$): Assume the following boundary smoothness condition: Let $p_x(u) : \mathbb{R}^d \to \mathbb{R}$ be a polynomial in u of order $q \leq r = \lfloor s \rfloor$ whose coefficients are a function of x and are $r - q$ times differentiable. Then assume that

$$\int_{x \in \mathcal{S}} \left(\int_{u:K(u)>0,\, x+uh \notin \mathcal{S}} K(u) p_x(u) du \right)^t dx = v_t(h),$$

where $v_t(h)$ admits the expansion

$$v_t(h) = \sum_{i=1}^{r-q} e_{i,q,t} h^i + o\left(h^{r-q}\right),$$

for some constants $e_{i,q,t}$.

It has been shown that assumption $\mathcal{A}.5$ is satisfied when $\mathcal{S}$ is rectangular (e.g. $\mathcal{S} = [-1, 1]^d$) and K is the uniform rectangular kernel [84]. Thus it can be applied to any densities in $\Sigma(s, H)$ on this support.

F.2 Proof of Bias Results

We prove Theorem B.7 in this appendix. The proof shares some similarities with the bias proof of the divergence functional estimators in [84]. The primary differences lie in handling the possible dependencies between random variables. We focus on the more difficult case of $\tilde{\mathbf{G}}_h$ as the bias derivation for $\tilde{\mathbf{G}}_{h,ij}$ is similar.

Recall that $\tilde{\mathbf{p}}'_{X,h}$ is a ratio of two products of KDEs. The numerator is a product of 2-dimensional KDEs while the denominator is a product of marginal (1-dimensional) KDEs, all with the same bandwidth. Let $\gamma \subset \{(i, j) : i, j \in \{1, \ldots, d\}\}$ denote the set of index pairs that denote the components of $\mathbf{X}$ that have joint KDEs that are in the product in the numerator of $\tilde{\mathbf{p}}'_{X,h}$. Let β denote the set of indices that denote the components of $\mathbf{X}$ that have marginal KDEs that are in the product in the denominator of $\tilde{\mathbf{p}}'_{X,h}$. Note that $|\gamma| = d - 1$ and $|\beta| = d - 2$. As an example, in

the example given in (8.4), we have $\gamma = \{(1,2),(2,3)\}$ and $\beta = \{2\}$. The bias of $\tilde{\mathbf{G}}_h$ can then be expressed as

$$\begin{aligned}
\mathbb{B}\left[\tilde{\mathbf{G}}_h\right] &= \mathbb{E}\left[g\left(\frac{\tilde{\mathbf{p}}'_{X,h}(\mathbf{X})}{\tilde{\mathbf{p}}_{X,h}(\mathbf{X})}\right) - g\left(\frac{p'(\mathbf{X})}{p(\mathbf{X})}\right)\right] \\
&= \mathbb{E}\left[g\left(\frac{\tilde{\mathbf{p}}'_{X,h}(\mathbf{X})}{\tilde{\mathbf{p}}_{X,h}(\mathbf{X})}\right) - g\left(\frac{\prod_{(i,j)\in\gamma}\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{ij,h}\left(\mathbf{X}^{(i)},\mathbf{X}^{(j)}\right)\right]}{\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{X,h}(\mathbf{X})\right]\prod_{k\in\beta}\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{k,h}\left(\mathbf{X}^{(k)}\right)\right]}\right)\right] \\
&\quad + \mathbb{E}\left[g\left(\frac{\prod_{(i,j)\in\gamma}\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{ij,h}\left(\mathbf{X}^{(i)},\mathbf{X}^{(j)}\right)\right]}{\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{X,h}(\mathbf{X})\right]\prod_{k\in\beta}\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{k,h}\left(\mathbf{X}^{(k)}\right)\right]}\right) - g\left(\frac{p'(\mathbf{X})}{p(\mathbf{X})}\right)\right], \qquad \text{(F.1)}
\end{aligned}$$

where $\mathbf{X}$ is drawn from p and $\mathbb{E}_{\mathbf{X}}$ denotes the conditional expectation given $\mathbf{X}$. We can view these terms as a variance-like component (the first term) and a bias-like component, where the respective Taylor series expansions depend on variance-like or bias-like terms of the KDEs.

We first consider the bias-like term, i.e. the second term in (F.1). The Taylor series expansion of $g\left(\frac{\prod_{(i,j)\in\gamma}\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{ij,h}\left(\mathbf{X}^{(i)},\mathbf{X}^{(j)}\right)\right]}{\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{X,h}(\mathbf{X})\right]\prod_{k\in\beta}\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{k,h}\left(\mathbf{X}^{(k)}\right)\right]}\right)$ around $\prod_{(i,j)\in\gamma} p\left(\mathbf{X}^{(i)},\mathbf{X}^{(j)}\right)$ and $p(\mathbf{X})\prod_{k\in\gamma}p(\mathbf{X}^{(k)})$ gives an expansion with terms of the form of

$$\begin{aligned}
\mathbb{B}^m_{\mathbf{X}}\left[\prod_{(i,j)\in\gamma}\tilde{\mathbf{p}}_{ij,h}\left(\mathbf{X}^{(i)},\mathbf{X}^{(j)}\right)\right] &= \left(\prod_{(i,j)\in\gamma}\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{ij,h}\left(\mathbf{X}^{(i)},\mathbf{X}^{(j)}\right)\right] - \prod_{(i,j)\in\gamma}p\left(\mathbf{X}^{(i)},\mathbf{X}^{(j)}\right)\right)^m, \\
\mathbb{B}^m_{\mathbf{X}}\left[p(\mathbf{X})\prod_{k\in\gamma}p(\mathbf{X}^{(k)})\right] &= \left(\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{X,h}(\mathbf{X})\right]\prod_{k\in\beta}\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{k,h}\left(\mathbf{X}^{(k)}\right)\right] - p(\mathbf{X})\prod_{k\in\gamma}p(\mathbf{X}^{(k)})\right)^m.
\end{aligned}$$

Since we are not doing boundary correction, we need to consider separately the cases when $\mathbf{X}$ is in the interior of the support $\mathcal{S}$ and when $\mathbf{X}$ is close to the boundary of the support. For precise definitions, a point $X \in \mathcal{S}$ is in the interior of $\mathcal{S}$ if for all $X' \notin \mathcal{S}$, $K\left(\frac{X-X'}{h}\right) = 0$, and a point $X \in \mathcal{S}$ is near the boundary of the support if it is not in the interior. Since K is a product kernel, $X \in \mathcal{S}$ is in the interior if and only if all of the components of X are in their respective interiors.

Assume that $\mathbf{X}$ is drawn from the interior of $\mathcal{S}$. By a Taylor series expansion of

the probability density p, we have that

$$\begin{aligned}\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{X,h}(\mathbf{X})\right] &= \frac{1}{h^d}\int K\left(\frac{\mathbf{X}-x}{h}\right)p\left(x\right)dx \\ &= \int K(u)p(\mathbf{X}-uh)du \\ &= p(\mathbf{X}) + \sum_{j=1}^{\lfloor s/2\rfloor} c_{X,j}(\mathbf{X})h^{2j} + O\left(h^s\right). \qquad \text{(F.2)}\end{aligned}$$

Similar expressions can be obtained for $\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{ij,h}\left(\mathbf{X}^{(i)},\mathbf{X}^{(j)}\right)\right]$ and $\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{k,h}\left(\mathbf{X}^{(k)}\right)\right]$.

Now assume that $\mathbf{X}$ lies near the boundary of the support $\mathcal{S}$. In this case, we extend the expectation beyond the support of the density:

$$\begin{aligned}\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{X,h}(\mathbf{X})\right] - p(\mathbf{X}) &= \frac{1}{h^d}\int_{x:x\in\mathcal{S}} K\left(\frac{\mathbf{X}-x}{h}\right)p(x)dx - p(\mathbf{X}) \\ &= \left[\frac{1}{h^d}\int_{x:K\left(\frac{\mathbf{X}-x}{h}\right)>0} K\left(\frac{\mathbf{X}-x}{h}\right)p(x)dx - p(\mathbf{X})\right] \\ &\quad - \left[\frac{1}{h^d}\int_{x:x\notin\mathcal{S}} K\left(\frac{\mathbf{X}-x}{h}\right)p(x)dx\right] \\ &= T_{1,X}(\mathbf{X}) - T_{2,X}(\mathbf{X}). \qquad \text{(F.3)}\end{aligned}$$

We only evaluate the density p and its derivatives at points within the support when we take its Taylor series expansion. Thus the exact manner in which we define the extension of p does not matter as long as the Taylor series remains the same and as long as the extension is smooth. Thus the expected value of $T_{1,X}(\mathbf{X})$ gives an expression of the form of (F.2). For the $T_{2,X}(\mathbf{X})$ term, we perform a similar Taylor series expansion and then apply the condition in assumption $\mathcal{A}.5$ to obtain

$$\mathbb{E}\left[T_{2,X}(\mathbf{X})\right] = \sum_{i=1}^{r} e_i h^i + o\left(h^r\right).$$

Similar expressions can be found for $\tilde{\mathbf{p}}_{ij,h}\left(\mathbf{X}^{(i)},\mathbf{X}^{(j)}\right)$, $\tilde{\mathbf{p}}_{k,h}\left(\mathbf{X}^{(k)}\right)$, and when (F.3) is raised to a power t. Applying this result gives for the second term in (F.1),

$$\sum_{j=1}^{r} c_{g,p,j}h^j + O\left(h^s\right), \qquad \text{(F.4)}$$

where the constants $c_{g,p,j}$ depend on the densities, their derivatives, and the functional g and its derivatives.

For the first term in (F.1), a Taylor series expansion of $g\left(\frac{\tilde{\mathbf{p}}'_{X,h}(\mathbf{X})}{\tilde{\mathbf{p}}_{X,h}(\mathbf{X})}\right)$ around $\prod_{(i,j)\in\gamma}\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{ij,h}\left(\mathbf{X}^{(i)},\mathbf{X}^{(j)}\right)\right]$ and $\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{X,h}(\mathbf{X})\right]\prod_{k\in\beta}\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{k,h}\left(\mathbf{X}^{(k)}\right)\right]$ gives an expansion with terms of the form of

$$\begin{aligned}
\tilde{\mathbf{e}}^q_{1,h}(\mathbf{X}) &= \left(\prod_{(i,j)\in\gamma}\tilde{\mathbf{p}}_{ij,h}\left(\mathbf{X}^{(i)},\mathbf{X}^{(j)}\right) - \prod_{(i,j)\in\gamma}\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{ij,h}\left(\mathbf{X}^{(i)},\mathbf{X}^{(j)}\right)\right]\right)^q, \\
\tilde{\mathbf{e}}^q_{2,h}(\mathbf{X}) &= \left(\tilde{\mathbf{p}}_{X,h}(\mathbf{X})\prod_{k\in\beta}\tilde{\mathbf{p}}_{k,h}\left(\mathbf{X}^{(k)}\right) - \mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{X,h}(\mathbf{X})\right]\prod_{k\in\beta}\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{k,h}\left(\mathbf{X}^{(k)}\right)\right]\right)^q.
\end{aligned}$$

To control these terms, we need expressions for $\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{e}}^q_{1,h}(\mathbf{X})\right]$ and $\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{e}}^q_{2,h}(\mathbf{X})\right]$. We'll derive the expression only for $\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{e}}^q_{1,h}(\mathbf{X})\right]$ as the expression for $\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{e}}^q_{2,h}(\mathbf{X})\right]$ is obtained in a similar manner.

For simplicity of exposition, we assume that $d = 3$ and that $\gamma = \{(1,2),(2,3)\}$. Note that our method extends easily to the general case where notation can be cumbersome. Define

$$\begin{aligned}
\mathbf{V}_{i,j}(\mathbf{X}) &= K_1\left(\frac{\mathbf{X}^{(1)}_i - \mathbf{X}^{(1)}}{h}\right)K_2\left(\frac{\mathbf{X}^{(2)}_i - \mathbf{X}^{(2)}}{h}\right)K_2\left(\frac{\mathbf{X}^{(2)}_j - \mathbf{X}^{(2)}}{h}\right)K_3\left(\frac{\mathbf{X}^{(3)}_j - \mathbf{X}^{(3)}}{h}\right) \\
&\quad - \mathbb{E}_{\mathbf{X}}\left[K_1\left(\frac{\mathbf{X}^{(1)}_i - \mathbf{X}^{(1)}}{h}\right)K_2\left(\frac{\mathbf{X}^{(2)}_i - \mathbf{X}^{(2)}}{h}\right)\right]\mathbb{E}_{\mathbf{X}}\left[K_2\left(\frac{\mathbf{X}^{(2)}_j - \mathbf{X}^{(2)}}{h}\right)K_3\left(\frac{\mathbf{X}^{(3)}_j - \mathbf{X}^{(3)}}{h}\right)\right] \\
&= \eta_{ij}(\mathbf{X}) - \mathbb{E}_{\mathbf{X}}\left[\eta_i(\mathbf{X})\right]\mathbb{E}_{\mathbf{X}}\left[\eta'_j(\mathbf{X})\right].
\end{aligned}$$

Therefore,

$$\tilde{\mathbf{e}}_{1,h}(\mathbf{X}) = \frac{1}{(Nh^2)^{|\gamma|}}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{V}_{i,j}(\mathbf{X}).$$

By the binomial theorem,

$$\mathbb{E}_{\mathbf{X}}\left[\mathbf{V}^k_{i,j}(\mathbf{X})\right] = \sum_{l=0}^{k}\binom{l}{k}\mathbb{E}_{\mathbf{X}}\left[\eta^l_{ij}(\mathbf{X})\right]\left[\mathbb{E}_{\mathbf{X}}\left[\eta_i(\mathbf{X})\right]\mathbb{E}_{\mathbf{X}}\left[\eta'_j(\mathbf{X})\right]\right]^{k-l}.$$

It can then be seen using a similar Taylor Series analysis as in the derivation of (F.2) that for $\mathbf{X}$ in the interior of $\mathcal{S}$ and $i \neq j$, we have that

$$\mathbb{E}_{\mathbf{X}}\left[\eta_{ij}^{l}(\mathbf{X})\right] = h^{2|\gamma|}\sum_{m=0}^{\lfloor s/2 \rfloor} c_{2,1,m,l}(\mathbf{X})h^{2m}.$$

Combining these results gives for $i \neq j$

$$\mathbb{E}_{\mathbf{X}}\left[\mathbf{V}_{i,j}^{k}(\mathbf{X})\right] = h^{2|\gamma|}\sum_{m=0}^{\lfloor s/2 \rfloor} c_{2,2,m,k}(\mathbf{X})h^{2m} + O\left(h^{4|\gamma|}\right).$$

If $i = j$, we obtain

$$\mathbb{E}_{\mathbf{X}}\left[\eta_{ii}^{l}(\mathbf{X})\right] = h^{d}\sum_{m=0}^{\lfloor s/2 \rfloor} c_{2,2,m}(\mathbf{X})h^{2m}.$$

This then gives

$$\mathbb{E}_{\mathbf{X}}\left[\mathbf{V}_{i,i}^{k}(\mathbf{X})\right] = h^{d}\sum_{m=0}^{\lfloor s/2 \rfloor} c_{2,m,k}(\mathbf{X})h^{2m} + O\left(h^{4|\gamma|}\right)$$

Here the constants $c_{2,i,m,k}(\mathbf{X})$ depend on the density p, its derivatives, and the moments of the kernels.

Let $n(q)$ be the set of integer divisors of q including 1 but excluding q. Then due to the independence of the different samples, it can then be shown that

$$\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{e}}_{1,h}^{q}(\mathbf{X})\right] = \sum_{i\in n(q)}\sum_{m=0}^{\lfloor s/2 \rfloor}\left(\frac{c_{3,1,m,q}(\mathbf{X})}{(Nh^2)^{(q-i)}} + \frac{c_{3,2,m,q}(\mathbf{X})}{(Nh)^{(q-i)}}\right)h^{2m} + o\left(\frac{1}{N}\right). \tag{F.5}$$

By a similar procedure, we can show that

$$\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{e}}_{2,h}^{q}(\mathbf{X})\right] = \sum_{i\in n(q)}\sum_{m=0}^{\lfloor s/2 \rfloor}\left(\sum_{j=0}^{|\beta|}\frac{c_{4,1,j,m,q}(\mathbf{X})}{\left(Nh^{d}\left(Nh\right)^{j}\right)^{q-i}} + \frac{c_{4,2,m,q}(\mathbf{X})}{(Nh)^{q-i}}\right)h^{2m} + o\left(\frac{1}{N}\right). \tag{F.6}$$

When $\mathbf{X}$ is near the boundary of the support, we can obtain similar expressions as in (F.5) and (F.6) by following a similar procedure as in the derivation of (F.4). The primary difference is that we will then have powers of h^m instead of h^{2m}.

For general g, we can only guarantee that

$$c\left(\frac{\prod_{(i,j)\in\gamma}\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{ij,h}\left(\mathbf{X}^{(i)},\mathbf{X}^{(j)}\right)\right]}{\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{X,h}(\mathbf{X})\right]\prod_{k\in\beta}\mathbb{E}_{\mathbf{X}}\left[\tilde{\mathbf{p}}_{k,h}\left(\mathbf{X}^{(k)}\right)\right]}\right)=c\left(\frac{p'(\mathbf{X})}{p(\mathbf{X})}\right)+o(1),$$

where $c(t_1,t_2)$ is a functional of the derivatives of g. Applying this gives the final result in this case. However, we can obtain higher order terms by making stronger assumptions on the functional g and its derivatives. Specifically, if $c(t_1,t_2)$ includes functionals of the form of $t_1^{\alpha}t_2^{\beta}$ with $\alpha,\beta<0$, then we can apply the generalized binomial theorem to use the higher order expressions in (F.5) and (F.6). This completes the proof.

F.3 Proof of Variance Results

To bound the variance of the plug-in estmiator, we use the Efron-Stein inequality [34]: (Efron-Stein Inequality) Let $\mathbf{X}_1,\ldots,\mathbf{X}_n,\mathbf{X}_1',\ldots,\mathbf{X}_n'$ be independent random variables on the space $\mathcal{S}$. Then if $f:\mathcal{S}\times\cdots\times\mathcal{S}\to\mathbb{R}$, we have that

$$\mathbb{V}\left[f(\mathbf{X}_1,\ldots,\mathbf{X}_n)\right]\leq\frac{1}{2}\sum_{i=1}^{n}\mathbb{E}\left[\Big(f(\mathbf{X}_1,\ldots,\mathbf{X}_n)-f(\mathbf{X}_1,\ldots,\mathbf{X}_i',\ldots,\mathbf{X}_n)\Big)^2\right].$$

The Efron-Stein inequality bounds the variance by the sum of the expected squared difference between the plug-in estimator with the original samples and the plug-in estimator where one of the samples is replaced with another iid sample.

In our case, consider the sample sets $\{\mathbf{X}_1,\ldots,\mathbf{X}_n\}$ and $\{\mathbf{X}_1',\mathbf{X}_2\ldots,\mathbf{X}_n\}$ and denote the respective plug-in estimators as $\tilde{\mathbf{G}}_h$ and $\tilde{\mathbf{G}}_h'$. Using the triangle inequality, we have

(F.7)

$$\left|\tilde{\mathbf{G}}_h-\tilde{\mathbf{G}}_h'\right|\leq\frac{1}{N}\left|g\left(\frac{\tilde{\mathbf{p}}_{X,h}'(\mathbf{X}_1)}{\tilde{\mathbf{p}}_{X,h}(\mathbf{X}_1)}\right)-g\left(\frac{\tilde{\mathbf{p}}_{X,h}'(\mathbf{X}_1')}{\tilde{\mathbf{p}}_{X,h}(\mathbf{X}_1')}\right)\right|+\frac{1}{N}\sum_{j=2}^{N}\left|g\left(\frac{\tilde{\mathbf{p}}_{X,h}'(\mathbf{X}_j)}{\tilde{\mathbf{p}}_{X,h}(\mathbf{X}_j)}\right)-g\left(\frac{\left(\tilde{\mathbf{p}}_{X,h}'(\mathbf{X}_j)\right)'}{\left(\tilde{\mathbf{p}}_{X,h}(\mathbf{X}_j)\right)'}\right)\right|,$$

where $\left(\tilde{\mathbf{p}}'_{X,h}(\mathbf{X}_j)\right)'$ and $\left(\tilde{\mathbf{p}}_{X,h}(\mathbf{X}_j)\right)'$ are the respective KDEs with $\mathbf{X}_1$ replaced with $\mathbf{X}'_1$. Then since g is Lipschitz continuous with constant C_g, we can write

$$\left| g\left(\frac{\tilde{\mathbf{p}}'_{X,h}(\mathbf{X}_1)}{\tilde{\mathbf{p}}_{X,h}(\mathbf{X}_1)}\right) - g\left(\frac{\tilde{\mathbf{p}}'_{X,h}(\mathbf{X}'_1)}{\tilde{\mathbf{p}}_{X,h}(\mathbf{X}'_1)}\right)\right| \leq C_g \left| \prod_{(i,j)\in\gamma} \tilde{\mathbf{p}}_{ij,h}\left(\mathbf{X}_1^{(i)}, \mathbf{X}_1^{(j)}\right) - \prod_{(i,j)\in\gamma} \tilde{\mathbf{p}}_{ij,h}\left(\mathbf{X}_1^{\prime(i)}, \mathbf{X}_1^{\prime(j)}\right)\right|$$
$$+ C_g \left| \tilde{\mathbf{p}}_{X,h}(\mathbf{X}_1) \prod_{k\in\beta} \tilde{\mathbf{p}}_{k,h}\left(\mathbf{X}_1^{(k)}\right) - \tilde{\mathbf{p}}_{X,h}(\mathbf{X}'_1) \prod_{k\in\beta} \tilde{\mathbf{p}}_{k,h}\left(\mathbf{X}_1^{\prime(k)}\right)\right|.$$

To bound the expected squared value of these terms, we split the product of KDEs into separate cases. For example, if we consider the case where the KDEs are all evaluated at the same point which occurs M times, we obtain

$$\frac{M}{(Mh^2)^{2|\gamma|}} \sum_{m=2}^{N} \mathbb{E}\left[\left(\prod_{(i,j)\in\gamma} K_i\left(\frac{\mathbf{X}_1^{(i)} - \mathbf{X}_m^{(i)}}{h}\right) K_j\left(\frac{\mathbf{X}_1^{(j)} - \mathbf{X}_m^{(j)}}{h}\right) - \right.$$
$$\left. \prod_{(i,j)\in\gamma} K_i\left(\frac{\mathbf{X}_1^{\prime(i)} - \mathbf{X}_m^{(i)}}{h}\right) K_j\left(\frac{\mathbf{X}_1^{\prime(j)} - \mathbf{X}_m^{(j)}}{h}\right)^2 \right. \qquad \textit{program@epst}$$
$$\leq \frac{1}{M^2} \prod_{(i,j)\in\gamma} ||K_i K_j||_\infty^2. \qquad \text{(F.8)}$$

By considering the other $|\gamma| - 1$ cases where the KDEs are evaluated at different points (e.g. 2 KDEs evaluated at the same point while all others are evaluated at different points, etc.), applying Jensen's inequality gives

$$\mathbb{E}\left[\left| \prod_{(i,j)\in\gamma} \tilde{\mathbf{p}}_{ij,h}\left(\mathbf{X}_1^{(i)}, \mathbf{X}_1^{(j)}\right) - \prod_{(i,j)\in\gamma} \tilde{\mathbf{p}}_{ij,h}\left(\mathbf{X}_1^{\prime(i)}, \mathbf{X}_1^{\prime(j)}\right)\right|^2\right] \leq C_1 \prod_{(i,j)\in\gamma} ||K_i K_j||_\infty^2,$$

where $C_1 < \infty$ is some constant that is $O(1)$. Similarly, we obtain

$$\mathbb{E}\left[\left| \tilde{\mathbf{p}}_{X,h}(\mathbf{X}_1) \prod_{k\in\beta} \tilde{\mathbf{p}}_{k,h}\left(\mathbf{X}_1^{(k)}\right) - \tilde{\mathbf{p}}_{X,h}(\mathbf{X}'_1) \prod_{k\in\beta} \tilde{\mathbf{p}}_{k,h}\left(\mathbf{X}_1^{\prime(k)}\right)\right|^2\right] \leq C_2 ||K||_\infty^2 \prod_{k\in\beta} ||K_k||_\infty^2.$$

Combining these results gives

$$\mathbb{E}\left[\left| g\left(\frac{\tilde{\mathbf{p}}'_{X,h}(\mathbf{X}_1)}{\tilde{\mathbf{p}}_{X,h}(\mathbf{X}_1)}\right) - g\left(\frac{\tilde{\mathbf{p}}'_{X,h}(\mathbf{X}'_1)}{\tilde{\mathbf{p}}_{X,h}(\mathbf{X}'_1)}\right)\right|^2\right] \leq C_3, \qquad \text{(F.9)}$$

where $C_3 = O(1)$.

As before, the Lipschitz condition can be applied to the second term in (F.7) to obtain

$$\left| g\left(\frac{\tilde{\mathbf{p}}'_{X,h}(\mathbf{X}_m)}{\tilde{\mathbf{p}}_{X,h}(\mathbf{X}_m)}\right) - g\left(\frac{\left(\tilde{\mathbf{p}}'_{X,h}(\mathbf{X}_m)\right)'}{\left(\tilde{\mathbf{p}}_{X,h}(\mathbf{X}_m)\right)'}\right)\right| \leq C_g \left| \prod_{(i,j)\in\gamma} \tilde{\mathbf{p}}_{ij,h}\left(\mathbf{X}_m^{(i)}, \mathbf{X}_m^{(j)}\right) - \prod_{(i,j)\in\gamma} \tilde{\mathbf{p}}'_{ij,h}\left(\mathbf{X}_m^{(i)}, \mathbf{X}_m^{(j)}\right)\right|$$
$$+ C_g \left| \tilde{\mathbf{p}}_{X,h}(\mathbf{X}_m) \prod_{k\in\beta} \tilde{\mathbf{p}}_{k,h}\left(\mathbf{X}_m^{(k)}\right) - \tilde{\mathbf{p}}'_{X,h}(\mathbf{X}_m) \prod_{k\in\beta} \tilde{\mathbf{p}}'_{k,h}\left(\mathbf{X}_m^{(k)}\right)\right|.$$

For the first term, we again consider the $|\gamma|$ cases where the KDEs are evaluated at different points. As a concrete example, consider the example given in (8.4). Then we can write by the triangle inequality

$$\left| \prod_{(i,j)\in\gamma} \tilde{\mathbf{p}}_{ij,h}\left(\mathbf{X}_m^{(i)}, \mathbf{X}_m^{(j)}\right) - \prod_{(i,j)\in\gamma} \tilde{\mathbf{p}}'_{ij,h}\left(\mathbf{X}_m^{(i)}, \mathbf{X}_m^{(j)}\right)\right|$$

$$\leq \frac{1}{M^2h^4}\left| K_1\left(\frac{\mathbf{X}_m^{(1)} - \mathbf{X}_1^{(1)}}{h}\right) K_2^2\left(\frac{\mathbf{X}_m^{(2)} - \mathbf{X}_1^{(2)}}{h}\right) K_3\left(\frac{\mathbf{X}_m^{(3)} - \mathbf{X}_1^{(3)}}{h}\right) - \right.$$
$$\left. K_1\left(\frac{\mathbf{X}_m^{(1)} - \mathbf{X}_1'^{(1)}}{h}\right) K_2^2\left(\frac{\mathbf{X}_m^{(2)} - \mathbf{X}_1'^{(2)}}{h}\right) K_3\left(\frac{\mathbf{X}_m^{(3)} - \mathbf{X}_1'^{(3)}}{h}\right) \right. \qquad \textit{prod}$$
$$+ \frac{1}{M^2h^4}\left(\sum_{\substack{n=2 \\ n\neq m}}^{N} K_1\left(\frac{\mathbf{X}_m^{(1)} - \mathbf{X}_n^{(1)}}{h}\right) K_2\left(\frac{\mathbf{X}_m^{(2)} - \mathbf{X}_n^{(2)}}{h}\right) \left| K_2\left(\frac{\mathbf{X}_m^{(2)} - \mathbf{X}_1^{(2)}}{h}\right) K_3\left(\frac{\mathbf{X}_m^{(3)} - \mathbf{X}_1^{(3)}}{h}\right) \right.\right.$$
$$\left.\left. - K_2\left(\frac{\mathbf{X}_m^{(2)} - \mathbf{X}_1'^{(2)}}{h}\right) K_3\left(\frac{\mathbf{X}_m^{(3)} - \mathbf{X}_1'^{(3)}}{h}\right)\right| + \left| K_1\left(\frac{\mathbf{X}_m^{(1)} - \mathbf{X}_1^{(1)}}{h}\right) K_2\left(\frac{\mathbf{X}_m^{(2)} - \mathbf{X}_1^{(2)}}{h}\right) - \right.$$
$$\left. K_1\left(\frac{\mathbf{X}_m^{(1)} - \mathbf{X}_1'^{(1)}}{h}\right) K_2\left(\frac{\mathbf{X}_m^{(2)} - \mathbf{X}_1'^{(2)}}{h}\right) \right. \qquad \textit{prod}$$
$$\left. \times \sum_{\substack{n=2 \\ n\neq m}}^{N} K_2\left(\frac{\mathbf{X}_m^{(2)} - \mathbf{X}_n^{(2)}}{h}\right) K_3\left(\frac{\mathbf{X}_m^{(3)} - \mathbf{X}_n^{(3)}}{h}\right)\right).$$

(F.10)

$$\implies \mathbb{E}\left[\left| \prod_{(i,j)\in\gamma} \tilde{\mathbf{p}}_{ij,h}\left(\mathbf{X}_m^{(i)}, \mathbf{X}_m^{(j)}\right) - \prod_{(i,j)\in\gamma} \tilde{\mathbf{p}}'_{ij,h}\left(\mathbf{X}_m^{(i)}, \mathbf{X}_m^{(j)}\right)\right|^2\right] \leq \frac{4 + 6(M-2)^2}{M^4} ||K_1K_2^2K_3||_\infty^2.$$

For more general γ, it can be shown that the LHS of (F.10) is $O\left(\frac{1}{M^2}\right)$. Similarly, we can check that

$$\mathbb{E}\left[\left|\tilde{\mathbf{p}}_{X,h}(\mathbf{X}_m)\prod_{k\in\beta}\tilde{\mathbf{p}}_{k,h}\left(\mathbf{X}_m^{(k)}\right)-\tilde{\mathbf{p}}'_{X,h}(\mathbf{X}_m)\prod_{k\in\beta}\tilde{\mathbf{p}}'_{k,h}\left(\mathbf{X}_m^{(k)}\right)\right|^2\right]=O\left(\frac{1}{M^2}\right).$$

Applying the Cauchy-Schwarz inequality with these results then gives

$$\mathbb{E}\left[\left(\sum_{j=2}^{N}\left|g\left(\frac{\tilde{\mathbf{p}}'_{X,h}(\mathbf{X}_j)}{\tilde{\mathbf{p}}_{X,h}(\mathbf{X}_j)}\right)-g\left(\frac{\left(\tilde{\mathbf{p}}'_{X,h}(\mathbf{X}_j)\right)'}{\left(\tilde{\mathbf{p}}_{X,h}(\mathbf{X}_j)\right)'}\right)\right|\right)^2\right]=O(1). \tag{F.11}$$

Combining (F.9) and (F.11) with (F.7) gives

$$\mathbb{E}\left[\left|\tilde{\mathbf{G}}_h-\tilde{\mathbf{G}}'_h\right|^2\right]=O\left(\frac{1}{N^2}\right).$$

Applying the Efron-Stein inequality then gives

$$\mathbb{V}\left[\tilde{\mathbf{G}}_h\right]=O\left(\frac{1}{N}\right).$$

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[2] S. Ali and S. D. Silvey. A general class of coefficients of divergence of one distribution from another. *J. Royal Statist. Soc. Ser. B (Methodology.)*, pages 131–142, 1996.

[3] Syed Mumtaz Ali and Samuel D Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 131–142, 1966.

[4] Syed Mumtaz Ali and Samuel D Silvey. A general class of coefficients of divergence of one distribution from another. *J. Royal Stat. Soc. Ser. B (Methodol.)*, pages 131–142, 1966.

[5] Animashree Anandkumar and Ragupathyraj Valluvan. Learning loopy graphical models with latent variables: Efficient methods and guarantees. *The Annals of Statistics*, 41(2):401–435, 2013.

[6] Hiromasa Arai, Crystal Maung, Ke Xu, and Haim Schweitzer. Unsupervised feature selection by heuristic search with provable bounds on suboptimality. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[7] D. Banks, M. Lavine, and H.J. Newton. The minimal spanning tree for nonparametric regression and structure discovery. In *computing Science and Statistics, Proceedings of the 24th Symposium on the Interface, H. Joseph Newton, Ed.*, pages 370–374, 1992.

[8] Andrew D Barbour, Lars Holst, and Svante Janson. *Poisson approximation*. Clarendon Press Oxford, 1992.

[9] Martyna Bator. UCI machine learning repository, 2013.

[10] Jillian Beardwood, John H Halton, and John Michael Hammersley. The shortest path through many points. In *Math Proc Cambridge*, volume 55, pages 299–327. Cambridge Univ Press, 1959.

[11] Rodrigo Benenson. https://rodrigob.github.io.

[12] V. Berisha and A.O. Hero. Empirical non-parametric estimation of the fisher information. *IEEE Signal Process. Lett.*, 22(7):988–992, 2015.

[13] V. Berisha, A. Wisler, A.O. Hero, and A. Spanias. Empirically estimable classification bounds based on a nonparametric divergence measure. *IEEE Trans. on Signal Process.*, 64(3):580–591, 2016.

[14] Visar Berisha, Alan Wisler, Alfred O Hero, and Andreas Spanias. Empirically estimable classification bounds based on a nonparametric divergence measure. *IEEE Transactions on Signal Processing*, 64(3):580–591, 2016.

[15] Visar Berisha, Alan Wisler, Alfred O Hero, and Andreas Spanias. Empirically estimable classification bounds based on a nonparametric divergence measure. *IEEE Trans. Signal Process.*, 64(3):580–591, Feb. 2016.

[16] Rajen B. Bhatt, Gaurav Sharma, Abhinav Dhall, and Santanu Chaudhury. Efficient skin region segmentation using low complexity fuzzy decision tree model. In *IEEE-INDICON, Dec 16-18, Ahmedabad, India*, pages 1–4, 2009.

[17] Dumitru Brinza, Matthew Schultz, Glenn Tesler, and Vineet Bafna. Rapid detection of gene–gene interactions in genome-wide association studies. *Bioinformatics*, 26(22):2856–2862, 2010.

[18] S.H. Cha. Comprehensive survey on distance/similarity measures between probability density functions. *Int. J. Math. Models Methods Appl. Sci.*, 1(4):300–307, 2007.

[19] Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Ann. Math. Stat.*, pages 493–507, 1952.

[20] Myung Jin Choi, Vincent YF Tan, Animashree Anandkumar, and Alan S Willsky. Learning latent tree graphical models. *Journal of Machine Learning Research*, 12(May):1771–1812, 2011.

[21] C Chow and C Liu. Approximating discrete probability distributions with dependence trees. *IEEE transactions on Information Theory*, 14(3):462–467, 1968.

[22] Andrzej Cichocki, Hyekyoung Lee, Yong-Deok Kim, and Seungjin Choi. Non-negative matrix factorization with α-divergence. *Pattern Recognition Letters*, 29(9):1433–1440, 2008.

[23] Dan Cireşan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. *arXiv preprint arXiv:1202.2745*, 2012.

[24] Dan Claudiu Cireşan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. Deep, big, simple neural nets for handwritten digit recognition. *Neural computation*, 22(12):3207–3220, 2010.

[25] Dan Claudiu Ciresan, Ueli Meier, Jonathan Masci, Luca Maria Gambardella, and Jürgen Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[26] I Ciszar. Information-type measures of difference of probability distributions and indirect observations. *Studia Sci. Math. Hungar.*, 2:299–318, 1967.

[27] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[28] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[29] Imre Csiszár and Paul C Shields. Information theory and statistics: A tutorial. *Foundations and Trends in Communications and Information Theory*, 1(4):417–528, Dec. 2004.

[30] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

[31] John C Duchi, Khashayar Khosravi, and Feng Ruan. Multiclass classification, information, divergence, and surrogate risk. *arXiv preprint arXiv:1603.00126*, 2016.

[32] David Edwards. *Introduction to graphical modelling*. Springer Science & Business Media, 2012.

[33] B. Efron and C. stein. The jackknife estimate of variance. *Annals of Statistics*, pages 586–596, 1981.

[34] Bradley Efron and Charles Stein. The jackknife estimate of variance. *The Annals of Statistics*, pages 586–596, 1981.

[35] Ananda Freire et al. UCI machine learning repository, 2010.

[36] Ananda L Freire, Guilherme A Barreto, Marcus Veloso, and Antonio T Varela. Short-term memory mechanisms in neural network learning of robot navigation tasks: A case study. In *2009 6th Latin American Robotics Symposium (LARS 2009)*, pages 1–6. IEEE, 2009.

[37] J. H. Friedman and L. C. Rafsky. Multivariate generalizations of the wald-wolfowitz and smirnov two-sample tests. *Ann. Statist.*, pages 697–717, 1979.

[38] Jerome H Friedman and Lawrence C Rafsky. Multivariate generalizations of the wald-wolfowitz and smirnov two-sample tests. *The Annals of Statistics*, pages 697–717, 1979.

[39] Weihao Gao, Sreeram Kannan, Sewoong Oh, and Pramod Viswanath. Estimating mutual information for discrete-continuous mixtures. In *Advances in Neural Information Processing Systems*, pages 5988–5999, 2017.

[40] et al Garofolo, John S. Timit acoustic-phonetic continuous speech corpus ldc93s1, 1993.

[41] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Efficient learning with sets of features. *Journal of Machine Learning Research*, 8(Apr):725–760, 2007.

[42] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[43] Isabelle Guyon, Steve Gunn, Masoud Nikravesh, and Lofti A Zadeh. *Feature extraction: foundations and applications*, volume 207. Springer, 2008.

[44] Mark Heimann, Tara Safavi, and Danai Koutra. Distribution of node embeddings as multiresolution features for graphs. *ICDM*, 2019.

[45] N. Henze and M.D. Penrose. On the multivarite runs test. *Ann. Statist.*, 27(1):290–298, 1999.

[46] Norbert Henze. A multivariate two-sample test based on the number of nearest neighbor type coincidences. *The Annals of Statistics*, pages 772–783, 1988.

[47] Norbert Henze and Mathew D Penrose. On the multivariate runs test. *Ann. Stat.*, pages 290–298, Feb. 1999.

[48] Norbert Henze and Mathew D Penrose. On the multivariate runs test. *Annals of statistics*, pages 290–298, 1999.

[49] Alfred O Hero, J Costa, and Bing Ma. Asymptotic relations between minimal graphs and alpha-entropy. *Comm. and Sig. Proc. Lab.(CSPL), Dept. EECS, University of Michigan, Ann Arbor, Tech. Rep*, 334, 2003.

[50] Alfred O Hero, Bing Ma, Olivier JJ Michel, and John Gorman. Applications of entropic spanning graphs. *IEEE signal processing magazine*, 19(5):85–95, 2002.

[51] A.O. Hero, J.A. Costa, and B. Ma. Asymptotic relations between minimal graphs and alpha-entropy. *Comm. and Sig. Proc. Lab.(CSPL), Dept. EECS, University of Michigan, Ann Arbor, Tech. Rep.*, 2003.

[52] A.O. Hero, J.A. Costa, and B. Ma. Convergence rates of minimal graphs with random vertices. *http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.8.4480 rep=rep1 type=pdf*, 2003.

[53] R. Hoffman and A.K. Jain. A test of randomness based on the minimal spanning tree. *Pattern Recognition Letters*, 1:175–180, 1983.

[54] https://freesound.org.

[55] Philippe Jacquet and Wojciech Szpankowski. Analytical depoissonization and its applications. *Theor Comput Sci*, 201(1):1–62, 1998.

[56] Thomas Kailath. The divergence and Bhattacharyya distance measures in signal selection. *IEEE Trans. Commun. Tech.*, 15(1):52–60, Feb. 1967.

[57] Kirthevasan Kandasamy, Akshay Krishnamurthy, Barnabas Poczos, Larry Wasserman, et al. Nonparametric Von Mises estimators for entropies, divergences and mutual informations. In *NIPS*, pages 397–405, 2015.

[58] Kirthevasan Kandasamy, Akshay Krishnamurthy, Barnabas Poczos, Larry Wasserman, and James Robins. Nonparametric von mises estimators for entropies, divergences and mutual informations. In *Advances in Neural Information Processing Systems*, pages 397–405, 2015.

[59] AD Kennedy. Approximation theory for matrices. *Nuclear Physics B-Proceedings Supplements*, 128:107–116, 2004.

[60] R. Kindermann and J.L. Snell. *Markov Random Fields and Their Applications*. American Mathematical Society, 1980.

[61] Kenji Kira and Larry A Rendell. A practical approach to feature selection. In *Machine Learning Proceedings 1992*, pages 249–256. Elsevier, 1992.

[62] Hisashi Koga, Tetsuo Ishibashi, and Toshinori Watanabe. Fast agglomerative hierarchical clustering algorithm using locality-sensitive hashing. *Knowledge and Information Systems*, 12(1):25–53, 2007.

[63] Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.

[64] Artemy Kolchinsky and Brendan D Tracey. Estimating mixture entropy with pairwise distances. *Entropy*, 19(7):361, 2017.

[65] A. Krishnamurthy, K. Kandasamy, B. Poczos, and L. Wasserman. Nonparametric estimation of renyi divergence and friends. In *Proceedings of The 31st International Conference on Machine Learning*, pages 919–927, 2014.

[66] Akshay Krishnamurthy, Kirthevasan Kandasamy, Barnabas Poczos, and Larry Wasserman. Nonparametric estimation of renyi divergence and friends. *arXiv preprint arXiv:1402.2966*, 2014.

[67] S. Kullback and R.A Leibler. On information and sufficiency. *The annals of Mathematical Statistics*, 22(1):79–86, 1951.

[68] Solomon Kullback and Richard A Leibler. On information and sufficiency. *Ann. Math. Stat.*, 22(1):79–86, 1951.

[69] Thomas Navin Lal, Olivier Chapelle, Jason Weston, and André Elisseeff. Embedded methods. In *Feature extraction*, pages 137–165. Springer, 2006.

[70] S.L. Lauritzen. *Graphical Models*. Clarendon Press, 1996.

[71] Yuh-Jye Lee, Chien-Chung Chang, and Chia-Huang Chao. Incremental forward feature selection with application to microarray gene expression data. *Journal of biopharmaceutical statistics*, 18(5):827–840, 2008.

[72] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):94, 2018.

[73] M. Lichman. UCI machine learning repository, 2013.

[74] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Trans. Inform. Theory*, 37(1):145–151, Jan. 1991.

[75] Qin Lv, William Josephson, Zhe Wang, Moses Charikar, and Kai Li. Multi-probe LSH: efficient indexing for high-dimensional similarity search. In *Proceedings of the 33rd International Conference on Very Large Data Bases*, pages 950–961. VLDB Endowment, 2007.

[76] Karthik Mohan, Mike Chung, Seungyeop Han, Daniela Witten, Su-In Lee, and Maryam Fazel. Structured learning of gaussian graphical models. In *Advances in neural information processing systems*, pages 620–628, 2012.

[77] K. R. Moon, K. Sricharan, Greenewald K., and Alfred O Hero. Ensemble estimation of information divergence. In *Entropy (Special Issue Inform. Theory Machine Learning)*, volume 20, page 560, Jul. 2018.

[78] Kevin Moon and Alfred Hero. Multivariate f-divergence estimation with confidence. In *Advances in Neural Information Processing Systems*, pages 2420–2428, 2014.

[79] Kevin R Moon and Alfred O Hero. Ensemble estimation of multivariate f-divergence. In *Information Theory (ISIT), 2014 IEEE International Symposium on*, pages 356–360. IEEE, 2014.

[80] Kevin R Moon, Morteza Noshad, Salimeh Yasaei Sekeh, and Alfred O Hero. Information theoretic structure learning with confidence. In *Proc. of IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, pages 6095–6099, 2017.

[81] Kevin R Moon, Morteza Noshad, Salimeh Yasaei Sekeh, and Alfred O Hero III. Information theoretic structure learning with confidence. In *Proc IEEE Int Conf Acoust Speech Signal Process*, 2017.

[82] Kevin R Moon, Kumar Sricharan, Kristjan Greenewald, and Alfred O Hero. Improving convergence of divergence functional ensemble estimators. In *IEEE International Symposium Inf Theory*, pages 1133–1137. IEEE, 2016.

[83] Kevin R Moon, Kumar Sricharan, Kristjan Greenewald, and Alfred O Hero. Improving convergence of divergence functional ensemble estimators. In *2016 IEEE Int. Symp. Inform. Theory*, pages 1133–1137, 2016.

[84] Kevin R Moon, Kumar Sricharan, Kristjan Greenewald, and Alfred O Hero III. Nonparametric ensemble estimation of distributional functionals. *arXiv preprint arXiv:1601.06884v2*, 2016.

[85] Kevin R Moon, Kumar Sricharan, and Alfred O Hero III. Ensemble estimation of mutual information. *Proceedings of the IEEE Intl Symp. on Information Theory (ISIT), Aachen, June 2017*.

[86] K.R Moon, K. Sricharan, K. Greenewald, and A.O. Hero. Improving convergence of divergence functional ensemble estimators. In *IEEE International Symposium on Information Theory (ISIT)*, pages 1133–1137, 2016.

[87] K.R Moon, K. Sricharan, K. Greenewald, and A.O. Hero. Nonparametric ensemble estimation of distributional functionals. *arXiv preprint arXiv: 1601.06884v2*, 2016.

[88] Elchanan Mossel. Distorted metrics on trees and phylogenetic forests. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 4(1):108–116, 2007.

[89] XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by penalized convex risk minimization. In *NIPS*, pages 1089–1096, 2007.

[90] Morteza Noshad and Alfred Hero. Scalable hash-based estimation of divergence measures. In *International Conference on Artificial Intelligence and Statistics*, pages 1877–1885, 2018.

[91] Morteza Noshad and Alfred O Hero. Rate-optimal meta learning of classification error. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2481–2485, 2018.

[92] Morteza Noshad and Alfred O Hero III. Scalable hash-based estimation of divergence measures. *Proceedings of the 22nd Conference on Artificial Intelligence and Statistics, Canary Islands, March 2018, arXiv:1801.00398*.

[93] Morteza Noshad, Kevin R Moon, Salimeh Yasaei Sekeh, and Alfred O Hero. Direct estimation of information divergence using nearest neighbor ratios. In *2017 IEEE Int. Symp. Inform. Theory*, pages 903–907, Jun. 2017.

[94] Morteza Noshad, Kevin R Moon, Salimeh Yasaei Sekeh, and Alfred O Hero III. Direct estimation of information divergence using nearest neighbor ratios. *Proc of the IEEE Intl Symp. on Information Theory (ISIT), Aachen, June 2017*.

[95] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 2014.

[96] Barnabás Póczos and Jeff G Schneider. On the estimation of alpha-divergences. In *AISTATS*, pages 609–617, 2011.

[97] Barnabás Póczos and Jeff G Schneider. Nonparametric estimation of conditional information and divergences. In *AISTATS*, pages 914–923, 2012.

[98] Yury Polyanskiy and Yihong Wu. Strong data-processing inequalities for channels and bayesian networks. In *Convexity and Concentration*, pages 211–249. Springer, 2017.

[99] A. Rényi. On measures of entropy and information. In *Fourth Berkeley Sympos. on Mathematical Stat. and Prob.*, pages 547–561, 1961.

[100] Alfréd Rényi. On measures of entropy and information. Technical report, Hungarian Academy of Sciences, 1961.

[101] A. Rukhin. Optimal estimator for the mixture parameter by the method of moments and information affinity. In *Proc. Trans. 12th Prague Conf. Inf. Theory*, pages 214–219, 1994.

[102] Alfrd Rnyi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1*, pages 547–561. University of California Press, 1961.

[103] Andrew Michael Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan Daniel Tracey, and David Daniel Cox. On the information bottleneck theory of deep learning. *ICLR*.

[104] Stephen Senn and William Richardson. The first t-test. *Statistics in medicine*, 13(8):785–803, 1994.

[105] Claude Elwood Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

[106] Alexander Shishkin, Anastasia Bezzubtseva, Alexey Drutsa, Ilia Shishkov, Ekaterina Gladkikh, Gleb Gusev, and Pavel Serdyukov. Efficient high-order interaction-aware feature selection based on conditional mutual information. In *Advances in neural information processing systems*, pages 4637–4645, 2016.

[107] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[108] Shashank Singh and Barnabás Póczos. Exponential concentration of a density functional estimator. In *Advances in Neural Information Processing Systems*, pages 3032–3040, 2014.

[109] Shashank Singh and Barnabás Póczos. Exponential concentration of a density functional estimator. In *Adv. Neural Inform. Process. Syst.*, pages 3032–3040, 2014.

[110] Shashank Singh and Barnabás Póczos. Generalized exponential concentration inequality for renyi divergence estimation. In *ICML*, pages 333–341, 2014.

[111] Kumar Sricharan, Raviv Raich, and Alfred O Hero III. Estimation of nonlinear functionals of densities with confidence. *Information Theory, IEEE Transactions on*, 58(7):4135–4159, 2012.

[112] Kumar Sricharan, Dennis Wei, and Alfred O Hero. Ensemble estimators for multivariate entropy estimation. *IEEE transactions on information theory*, 59(7):4374–4388, Jul. 2013.

[113] J Michael Steele. *Probability theory and combinatorial optimization*, volume 69. Siam, 1997.

[114] J.M Steele. An efron-stein inequality for nonsymmetric statistics. *Annals of Statistics*, 14:753–758, 1986.

[115] J.M. Steele, L.A. Shepp, and W.F. Eddy. On the number of leaves of a euclidean minimal spanning tree. *J. Appl. Prob.*, 24:809–826, 1987.

[116] G.T. Toussaint. The relative neighborhood graph of a finite planar set. *Pattern Recognition*, 12:261–268, 1980.

[117] Wallace Ugulino, Débora Cardador, Katia Vega, Eduardo Velloso, Ruy Milidiú, and Hugo Fuks. Wearable computing: Accelerometers data classification of body postures and movements. In *Brazilian Symposium on Artificial Intelligence*, pages 52–61. Springer, 2012.

[118] Pravin M Vaidya. An o (n logn) algorithm for the all-nearest-neighbors problem. *Discrete & Computational Geometry*, 4(1):101–115, 1989.

[119] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pages 1058–1066, 2013.

[120] Qing Wang, Sanjeev R Kulkarni, and Sergio Verdú. Divergence estimation of continuous distributions based on data-dependent partitions. *IEEE Trans. Inform. Theory*, 51(9):3064–3074, Sept. 2005.

[121] Qing Wang, Sanjeev R Kulkarni, and Sergio Verdú. Divergence estimation for multidimensional densities via-nearest-neighbor distances. *IEEE Transactions on Information Theory*, 55(5):2392–2405, 2009.

[122] Alan Wisler, Visar Berisha, Andreas Spanias, and Alfred O Hero. A data-driven basis for direct estimation of functionals of distributions. *arXiv preprint arXiv:1702.06516*, 2017.

[123] William H Wolberg, W Nick Street, and Olvi L Mangasarian. Image analysis and machine learning applied to breast cancer diagnosis and prognosis. *Analytical and Quantitative cytology and histology*, 17(2):77–87, 1995.

[124] Ming Yuan and Yi Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

[125] J.E. Yukich. *Probability theory of classical Euclidean optimization*. Vol. 1675 of lecture notes in Mathematics, Springer-Verlag, Berlin, 1998.

[126] Joseph E Yukich. *Probability theory of classical Euclidean optimization problems*. 1998.

[127] C.T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. on Computers*, C-20:68–86, 1971.

[128] Yanming Zhang, Kaizhu Huang, Guanggang Geng, and Chenglin Liu. Fast kNN graph construction with locality sensitive hashing. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 660–674, 2013.

[129] Dongxiao Zhu, Alfred O Hero, Zhaohui S Qin, and Anand Swaroop. High throughput screening of co-expressed gene pairs with controlled false discovery rate (fdr) and minimum acceptable strength (mas). *Journal of Computational Biology*, 12(7):1029–1045, 2005.

[130] Marko Znidaric. Asymptotic expansion for inverse moments of binomial and Poisson distributions. *arXiv preprint math/0511226*, 2005.