Data Reduction in Non-parametric Statistical Analysis and Optimal Transport Methods

by

Cheng Meng

(Under the Direction of Wenxuan Zhong and Ping Ma)

Abstract

With advances in science and technologies in the past decade, the amount of data generated and recorded has grown enormously in virtually all fields of industry and science. This extraordinary amount of data provides unprecedented opportunities for data-driven decision-making and knowledge discovery. However, the task of analyzing such large-scale dataset poses significant challenges and calls for innovative statistical methods specifically designed for faster speed and higher efficiency. In this thesis, I will cover some state-of-the-art data reduction methods for large-scale data analysis, with a focus on the design-based subsampling methods and some applications of sufficient dimension reduction in optimal transport methods.

Index words: Subsampling, dimension reduction, space-filling design, smoothing splines, optimal transport

Data Reduction in Non-parametric Statistical Analysis and Optimal Transport Methods

by

Cheng Meng

B.S., Tsinghua University, China, 2015

A Dissertation Submitted to the Graduate Faculty of the University of Georgia in Partial Fulfillment of the Requirements for the Degree

Doctor of Philosophy

Athens, Georgia

2020

Data Reduction in Non-parametric Statistical Analysis and Optimal Transport Methods

by

Cheng Meng

Major Professor:	Wenxuan Zhong, Ping Ma
Committee:	Tharuvai N Sriram
	Changying Li
	Tianming Liu

Electronic Version Approved:

Ron Walcott
Interim Dean of the Graduate School
The University of Georgia
May 2020

Table of Contents

List of Figures

List of Tables

Chapter 1

LowCon: A Design-based Subsampling Approach in a Misspecified Linear Model

We consider a measurement constrained supervised learning problem, that is, (1) full sample of the predictors are given;(2) the response observations are unavailable and expensive to measure. Thus, it is ideal to select a subsample of predictor observations, measure the corresponding responses, and then fit the supervised learning model on the subsample of the predictors and responses. However, model fitting is a dynamic process, the postulated model for the data could be misspecified. Our empirical studies demonstrate that most of the existing subsampling methods have unsatisfactory performance when the models are misspecified. In this chapter, we develop a novel subsampling method, called "LowCon", in a misspecified linear model, where there is an unknown misspecified term in addition to linear terms of predictors. Our method utilizes the Latin hypercube design in experimental design to achieve a robust estimation. We show the estimator using the proposed subsample approximately minimizes the so-called "worst-case" bias, with respect to many possible misspecification terms. Both the simulation and real-data analysis demonstrate the proposed estimator is more robust than several subsample least squares estimators obtained by state-of-the-art subsampling methods.

1.1 Introduction

Measurement constrained supervised learning is an emerging problem in machine learning (Derezinski et al., 2018; Settles, 2012; Y. Wang et al., 2017). In this problem, the predictor observations (also called unlabeled data points in machine learning literature) are collected, but the response observations are unavailable and difficult or expensive to obtain. Take speech recognition as an example, one may easily get plenty of unlabeled audio data, but the accurate labeling of speech utterances is extremely time-consuming and requires trained linguists. For an unlabeled speech of one minute, it can take up to ten minutes for the word-level annotation and nearly seven hours for the phoneme-level annotation (Zhu et al., 2005). A more concrete example is the task of predicting the soil functional property, where the soil function property refers to the property related to a soil's capacity to support essential ecosystem service (Hengl et al., 2015). Suppose one wants to model the relationship between the soil functional property and some predictors that can be easily derived from Earth remote sensing data. To get the response, the accurate measurement of the soil property, a sample of soil from the target area is needed. The response thus can be extremely time-consuming or even impractical to obtain, especially when the target area is off the beaten path. Thus, it is ideal to select a subsample of predictor observations, measure the corresponding responses, and then fit the supervised learning model on the subsample of the predictors and responses.

In this chapter, we study the subsampling method and postulate a general linear model for linking the response and predictors. One of the natural subsampling methods is the uniform subsampling method (also called the simple random subsampling method), i.e., selecting a subsample with the uniform sampling probability. For many problems, it is straightforward to construct "worst-case" input for which uniform subsampling method will perform poorly (Cochran, 2007; Thompson, 2012). Motivated by this, there has been a great deal of work on developing random subsampling methods that select a subsample with a data-dependent non-uniform sampling probability (Mahoney et al., 2011). One popular choice of the sampling probability is the normalized statistical leverage scores, leading to the *algorithmic leveraging* approach (Ma & Sun, 2015; Meng et al., 2017; X. Zhang et al., 2018).

Such an approach has already yielded impressive algorithmic and theoretical benefits in the linear regression model (Drineas et al., 2012; Ma, Mahoney, et al., 2015; Mahoney et al., 2011). Besides linear models, the idea of *algorithmic leveraging* is also widely applied in logistic regression, and Nyström method (Alaoui & Mahoney, 2015).

Different from random subsampling methods, there also exist some deterministic subsampling methods which select the subsample based on certain rules, especially the optimality criteria. Optimality criteria are often used in the context of the design of experiments (Pukelsheim, 2006), which includes but not limited to A-, D- and E-optimality.

Y. Wang et al., 2017 proposed a computationally tractable subsampling approach based on the A-optimality criterion. D-optimality criterion was considered in H. Wang et al., 2018, in which the author introduced the information-based optimal subdata selection method for selecting the most informative subdata.

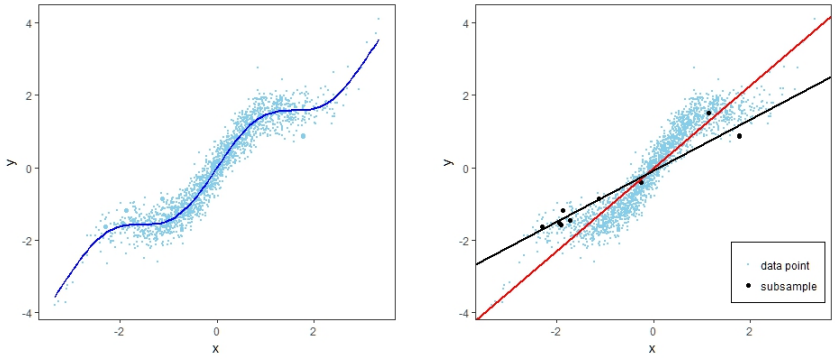

Figure 1.1: Example of subsampling in misspecified linear model

While the existing subsampling methods have already shown impressive performance on coefficient estimation and model prediction, the performance highly relies on the model assumption. However, model fitting is a dynamic process, and a postulated model could be misspecified. When the model is misspecified, most of these methods may lead to unacceptable results. Take the random subsampling method as an example; note that different model assumptions yield different sampling probabilities. Hence the selected subsample may be misleading when the model is misspecified, resulting in poor estimation or prediction. For deterministic subsampling methods, the key to the success of these methods is the optimality criteria they used. The optimality criteria, however, differs from model to model. An optimality criterion derived from a postulated model does not necessarily lead to a decent subsampling method for the true model. We now demonstrate the issue of model misspecification using a toy example. In this example, data are generated from the model $y_i = x_i + \sin(x_i^2)/2 + \epsilon_i$, $i = 1, 2, \ldots, n$, where $\{\epsilon_i\}_{i=1}^n$ are the i.i.d. standard normal errors. In Figure 1.1, the data points (blue points) and the true function (blue curve) are shown in the left panel. The red line in the right panel shows the full-sample

linear regression line based on only x_i, with the nonlinear term removed. We postulate a linear model without the nonlinear term and randomly select a subsample of size ten (black dots) using the leverage subsampling method (Ma, Mahoney, et al., 2015). The subsample linear regression line is shown as the black line, which deviates severely from the red line. Such an observation suggests the performance of a subsample least squares estimator may deteriorate significantly when the model is misspecified.

In practice, the explicit form of the underlying true model is almost always unknown to the practitioner. The subsample hence is highly desirable to be robust to possible model misspecification. To achieve the goal, Tsao and Ling, 2012 proposed to construct a robust estimator using bootstrap. One limitation of this method is that it can not be applied under the measurement-constrained setting since the value of the whole response vector is needed in this method to obtain the estimator.

In this chapter, we bridge the gap by providing the first statistical analysis of the subsampling method under the scenario where the underlying linear model contains unknown misspecification. We do so in the context of coefficient estimation by performing the least squares estimation on the selected subsample. Our major theoretical contribution is to provide an analytic framework for evaluating the mean squared error (MSE) of the subsample least squares (SLS) estimator in a misspecified linear model. Within this framework, we show it is very easy to construct a "worst-case" sample and a misspecification term for which an SLS estimator will have an arbitrary large mean squared error. We also show that an SLS estimator is robust if and only if the information matrix of the selected subsample has a relatively-low condition number, a traditional concept from numerical analysis (Trefethen & Bau III, 1997).

Based on these theoretical results, we propose and analyze a novel subsampling algorithm, called "LowCon". LowCon is designed to select the subsample, which takes the balance of variance and bias to the estimation of the coefficient. This algorithm involves selecting the subsample, which approximates a set of orthogonal Latin hypercube design points, a technique in experimental design (Ye, 1998). We show the proposed SLS estimator has a finite upper bound of the mean squared error, and it approximately minimizes the "worst-case" bias, with respect to all the possible misspecification terms. Our main empirical contribution is to provide a detailed evaluation of the robustness of the SLS estimators on both synthetic and real datasets. The empirical results indicate the proposed estimator is the only one among all that is robust to various types of misspecification terms. We thus recommend the use of LowCon in the measurement-constrained linear regression problem in the future.

1.2 Model Setup

In this section, we first introduce the linear model that contains unknown misspecification. We then consider the subsample least squares estimator, and we derive the mean squared error of these estimators under this model. We show that an SLS estimator is robust if and only if the information matrix of the selected subsample has a relatively-low condition number.

Throughout this chapter, $||\cdot||$ represents the Euclidean norm. Let $\lambda_{min}(\cdot)$ and $\lambda_{max}(\cdot)$ be the smallest and the largest eigenvalue of a matrix, and $\boldsymbol{\mu}_{min}(\cdot)$ and $\boldsymbol{\mu}_{max}(\cdot)$ be the corresponding eigenvectors, respectively. We use $s_1(\cdot)$ and $s_p(\cdot)$ to denote the largest and the smallest non-zero singular value of a matrix with p columns, respectively.

1.2.1 Misspecified Linear Model

Suppose the underlying true model has the form

$$y_i = \boldsymbol{x}_i^T \mathcal{B}_0 + u_i, \quad i = 1, 2, \ldots, n, \tag{1.1}$$

where y_is are the input responses, $\boldsymbol{x}_i$s are the input predictors, and $\mathcal{B}_0 \in \mathbb{R}^p$ ($p \ll n$) is the coefficient. The random errors $\{u_i\}_{i=1}^n$ are independently distributed, and u_i follows non-centered normal distribution $N(h(\boldsymbol{x}_i), \sigma^2)$, $i = 1, \ldots, n$. In this chapter, we consider the unknown multivariate function h that satisfies

$$\max_{\boldsymbol{x}} \frac{|h(\boldsymbol{x})|}{||\boldsymbol{x}||} = \alpha, \tag{1.2}$$

where $\alpha > 0$ is a constant. Let $\boldsymbol{y} = (y_1, \ldots, y_n)^T$ be the response vector, $\mathcal{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T$ as the predictor matrix, and $\boldsymbol{h}_X = (h(\boldsymbol{x}_1), \ldots, h(\boldsymbol{x}_n))^T$ as the misspecification term. For model-identifiability, we assume the matrix $[\mathcal{X}; \boldsymbol{h}_X]$ has a full column rank. When $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})^T$ has bounded value, some examples of h include $h(\boldsymbol{x}_i) = \sin(x_{i1})$ and $h(\boldsymbol{x}_i) = x_{i1}x_{i2}$.

We consider the scenario that the practitioner has no prior information about the true model (1.1) and the practitioner postulates a classical linear model

$$y_i = \boldsymbol{x}_i^T \mathcal{B}_0 + \epsilon_i, \quad i = 1, 2, \ldots, n, \tag{1.3}$$

where the random errors $\{\epsilon_i\}_{i=1}^n$ follow i.i.d. normal distribution $N(0, \sigma^2)$. Model (1.3) is thus a misspecified linear model, and the existence of the misspecification term in the true model (1.1) may result in the degenerated performance

of the coefficient estimation and model prediction. For example, the full-sample ordinary least squares (OLS) estimator, known as the best linear unbiased estimator, will lead to a biased estimation for the true coefficient when the model is misspecified (Box & Draper, 1959). We refer to Kiefer, 1975 and Sacks and Ylvisaker, 1978 for more discussion about the misspecified linear model.

In our measurement-constrained setting, the practitioner is initially given the full sample of predictors $\{\boldsymbol{x}_i\}_{i=1}^n$. Although required as the input in model (1.1), the responses $\{y_i\}_{i=1}^n$, however, are hidden unless explicitly requested. The practitioner is then allowed to reveal a subset of $\{y_i\}_{i=1}^n$, denoted by $\boldsymbol{y}^* = (y_1^*, \ldots, y_r^*)^T$, where $p < r \ll n$. The goal is to estimate the true coefficient $\mathcal{B}_0$ by using $\boldsymbol{y}^*$ and the corresponding subsample $\{\boldsymbol{x}_i^*\}_{i=1}^r$. The subsample predictor matrix is denoted by $\mathbf{R} = (\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_r^*)^T$. A natural estimator for the coefficient $\mathcal{B}_0$ is the subsample least squares estimator (Y. Wang et al., 2017),

$$\widetilde{\boldsymbol{\beta}}_{\mathbf{R}} = (\mathbf{R}^T\mathbf{R})^{-1}\mathbf{R}^T\boldsymbol{y}^*.$$

We derive the mean squared error and the worst-case MSE of this estimator, in the next subsection.

1.2.2 Worst-case MSE

Let $\mathbf{Q} = (\mathbf{R}^T\mathbf{R})^{-1}\mathbf{R}^T$ and $\boldsymbol{h} = (h(\boldsymbol{x}_1^*), \ldots, h(\boldsymbol{x}_r^*))^T \in \mathbb{R}^r$. The mean squared error of the estimator $\widetilde{\boldsymbol{\beta}}_{\mathbf{R}}$ (conditional on $\mathcal{X}$) thus can be decomposed as

$$\begin{aligned}
\text{MSE}(\widetilde{\boldsymbol{\beta}}_{\mathbf{R}}) &= \text{tr}(\text{Var}(\widetilde{\boldsymbol{\beta}}_{\mathbf{R}})) + [\text{bias}(\widetilde{\boldsymbol{\beta}}_{\mathbf{R}})]^T[\text{bias}(\widetilde{\boldsymbol{\beta}}_{\mathbf{R}})] \\
&= \sigma^2\text{tr}[(\mathbf{R}^T\mathbf{R})^{-1}] + [(\mathbf{R}^T\mathbf{R})^{-1}\mathbf{R}^T\boldsymbol{h}]^T[(\mathbf{R}^T\mathbf{R})^{-1}\mathbf{R}^T\boldsymbol{h}] \\
&= \sigma^2\text{tr}[(\mathbf{R}^T\mathbf{R})^{-1}] + \boldsymbol{h}^T\mathbf{Q}^T\mathbf{Q}\boldsymbol{h}, \qquad (1.4)
\end{aligned}$$

where the bias term $\boldsymbol{h}^T\mathbf{Q}^T\mathbf{Q}\boldsymbol{h}$ is associated with the model misspecification. Note that when $\boldsymbol{h}_X = \mathbf{0}$, i.e., the model is correctly specified, the bias term will vanish, and thus minimizing MSE is equivalent to minimizing the variance term. Further discussion following this line of thinking can be found in Y. Wang et al., 2017 and H. Wang et al., 2018, in which the authors focused on selecting the subsample that minimizes the variance term. In our setting, where the model is misspecified, however, minimizing the variance term does not necessarily lead to a small MSE.

Recall that our goal is to select a subsample such that the corresponding SLS estimator is robust to various model misspecification. Since the misspecification term $\boldsymbol{h}_X$ is unknown to the practitioner, a natural and intuitive approach is

to find the subsample that minimizes the so-called "worst-case" MSE, i.e., the maximum value of MSE($\widetilde{\boldsymbol{\beta}}_{\mathbf{R}}$) with respect to all the possible choices of the misspecification term $\boldsymbol{h}_X$. The following lemma gives an explicit form of the worst-case MSE, see the appendix for the proof.

Lemma 1.2.1 (Worst-case MSE)**.** *Under the regularity condition (1.2), the following inequality holds:*

$$MSE(\widetilde{\boldsymbol{\beta}}_{\mathbf{R}}) \leq \sigma^2 tr[(\mathbf{R}^T\mathbf{R})^{-1}] + \alpha^2 \frac{tr(\mathbf{R}^T\mathbf{R})}{\lambda_{min}(\mathbf{R}^T\mathbf{R})}. \tag{1.5}$$

The right-hand side of (1.5) is called the worst-case MSE, and it can be achieved when

$$\boldsymbol{h} = \sqrt{\alpha^2 tr(\mathbf{R}^T\mathbf{R})} \cdot \boldsymbol{\mu}_{max}(\mathbf{Q}^T\mathbf{Q}).$$

Two conclusions can be made from Lemma 1.2.1. First, the worst-case MSE of an SLS estimator can be inflated to arbitrarily large values by a very small value of $\lambda_{min}(\mathbf{R}^T\mathbf{R})$. It is thus very easy to construct a "worst-case" sample and a misspecification term for which an SLS estimator will have unacceptable performance. Second, $\widetilde{\boldsymbol{\beta}}_{\mathbf{R}}$ is the most robust SLS estimator if and only if the selected subsample minimizes the worst-case MSE. Such a subsample, however, is impossible to obtain for real examples, since both the values of σ^2 and α^2 are unknown to the practitioners.

In many real-life datasets, with a large sample size particularly, some outliers in the response vector often exist (Kriegel, Zimek, et al., 2008; Zimek et al., 2012). The existence of these extreme outliers indicates the value of α^2 can be considerable in practice. Motivated by this, we thus are more interested in the setting where the value of α^2 is large enough such that, on the right-hand side of the inequality (1.5), the second term dominates the first term. Under this setting, the desired subsample $\mathbf{R}$ should yield a relatively small value of $\text{tr}(\mathbf{R}^T\mathbf{R})/\lambda_{min}(\mathbf{R}^T\mathbf{R})$. Notice that

$$\text{tr}(\mathbf{R}^T\mathbf{R})/\lambda_{min}(\mathbf{R}^T\mathbf{R}) \geq p, \tag{1.6}$$

where the equality holds when the condition number of the subsample information matrix, i.e., $\kappa(\mathbf{R}^T\mathbf{R}) \stackrel{\text{def}}{=} \lambda_{max}(\mathbf{R}^T\mathbf{R})/\lambda_{min}(\mathbf{R}^T\mathbf{R})$, takes the minimum value 1. The inequality (1.6) thus indicates the desired subsample $\mathbf{R}$ is the one such that $\kappa(\mathbf{R}^T\mathbf{R})$ has a relatively small value.

We now give another intuition about how is $\kappa(\mathbf{R}^T\mathbf{R})$ related to the robustness of the SLS estimator. In the literature, the condition number of the

full-sample information matrix, $\kappa(\mathcal{X}^T\mathcal{X})$ is known to be related to the robustness of the full-sample OLS estimator $\widehat{\boldsymbol{\beta}}_{ols}$. Casella, 1985 considered the question, how much would $\widehat{\boldsymbol{\beta}}_{ols}$ change if there were perturbation in $\mathcal{X}^T\boldsymbol{y}$, and the author showed that

$$\frac{||\delta\widehat{\boldsymbol{\beta}}_{ols}||}{||\widehat{\boldsymbol{\beta}}_{ols}||} = \frac{||\delta(\mathcal{X}^T\mathcal{X})^{-1}\mathcal{X}^T\boldsymbol{y}||}{||(\mathcal{X}^T\mathcal{X})^{-1}\mathcal{X}^T\boldsymbol{y}||} \leq \kappa(\mathcal{X}^T\mathcal{X})\frac{||\delta\mathcal{X}^T\boldsymbol{y}||}{||\mathcal{X}^T\boldsymbol{y}||},$$

where $\delta\widehat{\boldsymbol{\beta}}_{ols}$ and $\delta\mathcal{X}^T\boldsymbol{y}$ represent the perturbation. Thus the length of the estimator is more stable if $\kappa(\mathcal{X}^T\mathcal{X})$ has a small value. Analogously, one can also show that

$$\frac{||\delta\widetilde{\boldsymbol{\beta}}_{\mathbf{R}}||}{||\widetilde{\boldsymbol{\beta}}_{\mathbf{R}}||} \leq \kappa(\mathbf{R}^T\mathbf{R})\frac{||\delta\mathbf{R}^T\boldsymbol{y}^*||}{||\mathbf{R}^T\boldsymbol{y}^*||}. \tag{1.7}$$

Inequality (1.7) thus indicates that smaller the value of $\kappa(\mathbf{R}^T\mathbf{R})$ is, more robust the estimator $\widetilde{\boldsymbol{\beta}}_{\mathbf{R}}$ will be.

It worth noting that the subsample matrix $\mathbf{R}$ which minimize the worst-case MSE does not necessarily minimize $\kappa(\mathbf{R}^T\mathbf{R})$ simultaneously since both the value of σ^2 and α^2 are not available in practice. A robust subsample $\mathbf{R}$ should at least yield a relatively small value of $\kappa(\mathbf{R}^T\mathbf{R})$ and take the balance of the variance and the bias in the equation (1.4). Following this line of thinking, we propose a novel subsampling algorithm, and the details are presented in the next section.

1.3 LowCon Algorithm

In this section, we present our main algorithm, called "Low condition number pursuit" or "LowCon." We first introduce the notion of orthogonal Latin hypercube designs (OLHD), and how these can be used to generate a design matrix $\mathbf{L}$ such that $\kappa(\mathbf{L}^T\mathbf{L})$ has a relatively small value. Next, we present the detail of the proposed algorithm which incorporates the idea of OLHD. We conclude the section by presenting the theoretical property of the proposed SLS estimator, which is obtained by LowCon algorithm. We show the proposed estimator has a relatively small upper bound of the MSE.

1.3.1 Orthogonal Latin Hypercube Design

Space-filling design techniques have been used as standard practice for computer experiments (Fang et al., 2005; Kleijnen, 2008). These techniques focus

on the problem that how to draw the design points that cover a continuous design space as uniformly as possible. Note that one fundamental difference between such a problem and the subsampling problem, however, is that in the latter, the selected points cannot be freely designed in a continuous space and they must come from the given finite sample pool $\{\boldsymbol{x}_i\}_{i=1}^n$. To bridge the gap between these two problems, we propose to round the design point to its nearest neighbor in the sample. Details are provided in Section 3.2.

We now introduce the Latin hypercube design (LHD), a specific space-fillling design technique that is of our interest. Latin hypercube design is known for the best one-dimensional marginal projection property (McKay et al., 2000; Stein, 1987), and its definition is shown in the following.

Definition 1.3.1 (Latin hypercube design)**.** Given the space $\mathcal{X} = [-1, 1]^p$, $\mathbf{L} \in \mathbb{R}^{r\times p}$ is called a Latin hypercube design matrix if each column of $\mathbf{L}$ is a random permutation of $\{\frac{1-r}{r}, \frac{3-r}{r}, \ldots, \frac{r-1}{r}\}$ (Steinberg & Lin, 2006).

Intuitively, if one divides the design space $[-1, 1]^p$ into r equal-size slices according to the jth ($j = 1, \ldots, p$) dimension, a Latin hypercube design ensures that there is exactly one design point in each slice. The left panel of Figure 1.2 shows an example of a set of Latin hypercube design points (black dots). Although uniformly distributed on the marginal, the Latin hypercube design points do not necessarily spread out within the whole design space. That is to say, a set of LHD points may not be "space-filling" enough. To improve the "space-filling" property of LHD, various methods have been developed (Fang et al., 2002; Joseph & Hung, 2008; J.-S. Park, 1994; Tang, 1993). Of particular interest in this chapter is the orthogonal Latin hypercube design (OLHD) which achieves the goal by reducing the pairwise correlations of LHD (Ye, 1998), see the right panel of Figure 1.2 for an example.

Consider the information matrix $\mathbf{L}^T\mathbf{L}$ where $\mathbf{L}$ is an OLHD matrix. Intuitively, the matrix $\mathbf{L}^T\mathbf{L}$ has a relatively small condition number, since all the diagonal elements of $\mathbf{L}^T\mathbf{L}$ are the same and all the off-diagonal elements of $\mathbf{L}^T\mathbf{L}$ have relatively small absolute value. Although there is a lack of theoretical guarantee, empirically, it is known that $\kappa(\mathbf{L}^T\mathbf{L})$ is in general no greater than 1.13 (Cioppa & Lucas, 2007). Such a fact motivates us to select the subsample that approximates a set of orthogonal Latin hypercube design points.

1.3.2 LowCon Subsampling Algorithm

We now provide the details of LowCon subsampling algorithm. Intuitively, given an OLHD with r design points, the proposed algorithm searches and

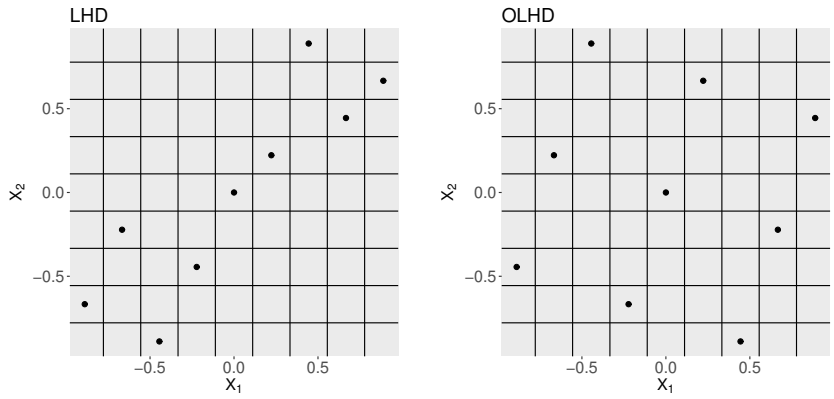

Figure 1.2: Example of LHD (left panel) and OLHD (right panel) with nine design points in $[-1, 1]^2$.

selects the nearest neighbor from the data points in the full-sample for every design point. The algorithm is summarized in the following.

Algorithm 1 " Low Condition Number Pursuit (LowCon)" subsampling algorithm

---

Data normalization: The data points $\{\boldsymbol{x}_i\}_{i=1}^n$ are first scaled to $[-1, 1]^p$.
Generate OLHD points: Given a design space $\mathcal{X} \subset [-1, 1]^p$, generate a set of orthogonal Latin hypercube design points $\{\boldsymbol{s}_i\}_{i=1}^r$.
Nearest neighbor search: Select the nearest neighbor for each design point $\boldsymbol{s}_i$ from $\{\boldsymbol{x}_i\}_{i=1}^n$, denoted by $\boldsymbol{s}_i^*$. The selected subsample is thus given by $\{\boldsymbol{s}_i^*\}_{i=1}^r$.

Figure 1.3 illustrates LowCon algorithm. The synthetic data points in the left panel were generated from a bivariate normal distribution, and are scaled to $[-1, 1]^2$. A set of orthogonal Latin hypercube design points are then generated, labeled as black triangles in the middle panel. For each design point, the nearest data point is selected, marked as black dots in the right panel. Observe that the selected points can well-approximate the design points.

Comment 1. The set of design points generated by orthogonal Latin hypercube design technique is not unique and different sets of design points may result in different subsamples. Algorithm 1 thus is a random subsampling method instead of a deterministic subsampling method. In practice, the set of design points $\{\boldsymbol{s}_i\}_{i=1}^r$ in Algorithm 1 can be randomly generated.

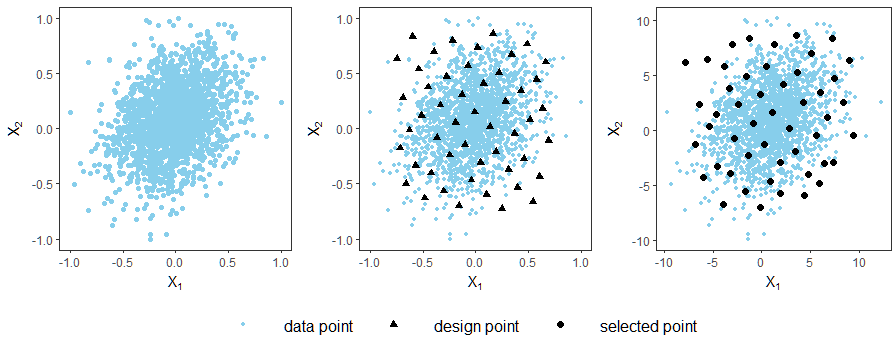

Figure 1.3: Illustration for Algorithm 1.

Comment 2. The key to the success of Algorithm 1 is that the selected subsample $\{\boldsymbol{s}_i^*\}_{i=1}^r$ can well-represent the set of design points $\{\boldsymbol{s}_i\}_{i=1}^r$. This is to say, the design point $\boldsymbol{s}_i$ is close-enough to its nearest neighbor $\boldsymbol{s}_i^*$, $i = 1, \ldots, r$. We provide more discussion in Section 3.3 about when such a requirement is met in practice. Empirically, we find $[-1, 1]^p$ may not be a good choice for the design space $\mathcal{X}$. This is because in such a scenario, the design points, which close to the boundary of $[-1, 1]^p$, may be too far away from its nearest neighbor, especially when the population density function has a heavy tail. As a result, a design space that is slightly smaller than $[-1, 1]^p$ would be a safer choice. We opt to set the design space as $\mathcal{X} = [\theta_{j1}, \theta_{j2}]^p$, where θ_{j1} and θ_{j2} are the θ-percentile and $(100 - \theta)$-percentile of the jth column of the scaled data points, respectively. Through all the experiments in this chapter, θ is set as 1.

1.3.3 Theoretical Results

We now present the theoretical property of the subsample least squares estimator, obtained by the LowCon algorithm. Recall that $\mathbf{L}$ represents an orthogonal Latin hypercube design matrix. Let $\mathbf{R}_L$ be the subsample matrix obtained by the proposed algorithm. One thus can decompose $\mathbf{R}_L$ into a sum of the design matrix $\mathbf{L}$ and a matrix $\mathbf{D} = (\boldsymbol{d}_1, \cdots, \boldsymbol{d}_r)^T$, i.e., $\mathbf{R}_L = \mathbf{L} + \mathbf{D}$.

Following the notations in Algorithm 1, one can write $\mathbf{L} = (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_r)^T$ and $\mathbf{R}_L = (\boldsymbol{s}_1^*, \ldots, \boldsymbol{s}_r^*)^T$, where $\boldsymbol{s}_i$ and $\boldsymbol{s}_i^*$ represent the ith design point and its corresponding nearest neighbor from the sample, respectively. One thus has $\boldsymbol{d}_i = \boldsymbol{s}_i^* - \boldsymbol{s}_i$, for $i = 1, \ldots, r$. Intuitively, $\mathbf{D}$ is a random perturbation matrix, and the selected data points can well-approximate the design points if $\mathbf{D}$ is "negligible". In such a case, $MSE(\widetilde{\boldsymbol{\beta}}_{\mathbf{R}_L})$, which is a function of $\mathbf{R}_L$,

can be expanded around $MSE(\widetilde{\boldsymbol{\beta}}_{\mathbf{L}})$ through Taylor expansion. From this, we can establish our main theorem in the following. The proof is relegated to the appendix.

Theorem 1.3.1. *Suppose the data follow the model (1.1) and the regularity condition (1.2) is satisfied. Assume $s_p(\mathbf{L}) > s_1(\mathbf{D})$, where $s_1(\cdot)$ and $s_p(\cdot)$ represent the largest and the smallest singular value of a matrix of p columns, respectively. A Taylor expansion of $MSE(\widetilde{\boldsymbol{\beta}}_{\mathbf{R}_L})$ around the point $\mathbf{R}_L = \mathbf{L}$ yields the following upper bound,*

$$MSE(\widetilde{\boldsymbol{\beta}}_{\mathbf{R}_L}) \leq \sigma^2 p^2 \frac{\kappa(\mathbf{L}^T\mathbf{L})}{tr(\mathbf{L}^T\mathbf{L})} + \alpha^2 p\kappa(\mathbf{L}^T\mathbf{L}) + W. \tag{1.8}$$

Here, $W = O(s_1(\mathbf{D}))$ is the Taylor expansion remainder.

When the Taylor expansion in Theorem 1.3.1 is valid, three significant conclusions can be made. First, the theorem indicates that the MSE of the proposed estimator is finite. Recall that Lemma 1.2.1 shows the worst-case MSE of an SLS estimator can be inflated to arbitrarily large value by a very small value of $\lambda_{min}(\mathbf{R}^T\mathbf{R})$. The fact that the proposed estimator has a finite MSE thus indicates the proposed estimator is robust, i.e., the value of which will not be inflated to arbitrary large.

Second, the upper bound of the squared bias of the proposed estimator, which equals $\alpha^2 p\kappa(\mathbf{L}^T\mathbf{L})$, is very close to the minimum value of the worst-case squared bias. To see this, combining the inequality (1.6) and the inequality (1.5) yields the worst-case squared bias has the minimum value of $\alpha^2 p$. In Section 3.1, we discussed the value of $\kappa(\mathbf{L}^T\mathbf{L})$ is known to be close to 1. Combining these two facts together yields the second conclusion. Consider the common situation when the value of α^2 is large enough such that, in inequality (1.5), the bias term dominates the variance term. Under such a situation, the second conclusion thus indicates, the proposed estimator is very close to the "most robust" estimator which minimizes the worst-case squared bias.

Third, the proposed estimator has finite variance. This is because the value of $tr(\mathbf{L}^T\mathbf{L})$ is finite when the design space of $\mathbf{L}$ equals $[-1, 1]^p$, a direct conclusion from the definition of the Latin hypercube design. Recall that in Algorithm 1, sometimes we may choose a design space $\mathcal{X} \subset [-1, 1]^p$. The value of $tr(\mathbf{L}^T\mathbf{L})$ will decrease in such cases, compared to the case when the design space equals $[-1, 1]^p$. The variance of the proposed estimator thus will increase in such cases. Nevertheless, the variance term will not be inflated to arbitrarily large, as long as the design space is not too small.

There are two essential assumptions in Theorem 1.3.1. One is that $s_p(\mathbf{L}) > s_1(\mathbf{D})$ and the other is that the Taylor expansion is valid, i.e., when $s_1(\mathbf{D})$ is

"small". Although we will evaluate the quality of the proposed estimator empirically in the next section, a precise theoretical characterization of when these two assumptions are valid is currently not available. Here, we simply give an example such that $s_1(\mathbf{D})$ converges to zero as n goes to infinity, in which case the desired Taylor expansion is valid apparently. The assumption $s_p(\mathbf{L}) > s_1(\mathbf{D})$ is also satisfied in such a case, as n goes to infinity, since the value of $s_p(\mathbf{L})$ is not relevant to n. Consider the case when the non-zero support of the population distribution is $[-1, 1]^p$. This is to say, the sample and the design points have the same domain. In such a case, the distance between each design point and its nearest neighbor converges to zero, as n goes to infinity. As a result, each entry of the matrix $\mathbf{D}$ converges to zero, and thus $s_1(\mathbf{D})$ converges to zero as well, as n goes to infinity. Consequently, the desired Taylor expansion is valid in such a case.

1.4 Simulation Results

To show the effectiveness of the proposed method in misspecified linear models, we compare it with existing subsampling methods in terms of estimation error. The subsampling methods considered here are uniform subsampling (UNIF), basic leverage subsampling (BLEV), shrinkage leverage subsampling (SLEV), unweighted-leverage subsampling (LEVUNW) (Ma, Mahoney, et al., 2015; Ma & Sun, 2015), and information-based optimal subset selection (IBOSS) (H. Wang et al., 2018). The parameter for SLEV is set as 0.9, as suggested in Ma, Mahoney, et al., 2015. The parameter θ for the proposed method is set as 1.

We simulate the data from the model (1.1) with $n = 10^4, p = \{10, 20\}$ and $r = \{2p, 4p, \ldots, 10p\}$. Three different population distributions for the data points are considered: **(D1)** $N(\mathbf{1}, \mathbf{\Sigma})$; **(D2)** $0.5N(\mathbf{0}, 2\mathbf{\Sigma}) + 0.5N(\mathbf{1}, \mathbf{\Sigma})$; **(D3)** $t_{10}(\mathbf{1}, \mathbf{\Sigma})$, where $\mathbf{\Sigma}_{ij} = 10 \times 0.6^{|i-j|}$ for $i, j = 1, \ldots, p$. For the coefficient $\boldsymbol{\beta}_0$, the first 20% and the last 20% entries were set to be 1 and the rest of them were set to be 0.1. To show the robustness of the proposed estimator under various of misspecification terms, we consider five different scenarios: **(H1)** $h(\boldsymbol{x}_i) = 0$; **(H2)** $h(\boldsymbol{x}_i) = 10\sin(x_{i3})$; **(H3)** $h(\boldsymbol{x}_i) = c_1 \cdot x_{i3}x_{i8}$; **(H4)** $h(\boldsymbol{x}_i) = c_2 \cdot x_{i3}\sin(x_{i8})$ and **(H5)** $h(\boldsymbol{x}_i) = c_3 \cdot x_{i3}^2$. To ensure that the response will not be dominated by the misspecification term, we select the constants c_1, c_2 and c_3 such that $\max_{\boldsymbol{x}}(\{|h(\boldsymbol{x})|\}_{i=1}^n) = 10$, respectively. In Figure 1.4 we depict the heat map of the last four misspecified terms, where the data are generated from distribution $N(\mathbf{1}, \mathbf{\Sigma})$. Only the third and eighth predictors are shown for illustration.

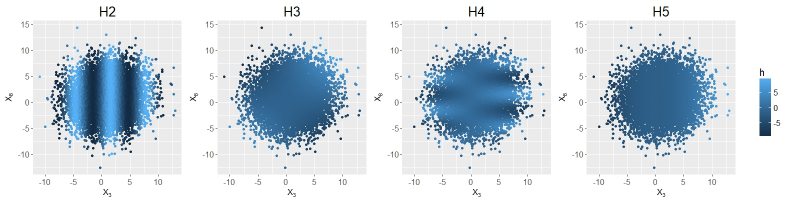

Figure 1.4: The scatterplots of ten thousand data points generated from distribution **D1** with ten predictors.

Figure 1.5 compares the subsamples selected by different subsampling methods. The LEVUNW method is omitted here since the subsample identified by LEVUNW is the same as the subsample identified by BLEV. The data points (blue dots) are generated from distribution **D3** with $n = 10^4$ and $p = 10$, only the third and the eighth predictors are shown. In each panel, a subsample of size 40 is selected (black dots). Figure 1.5 reveals some interesting facts. We first observe the subsamples selected by BLEV and SLEV are more dispersed than the subsample selected by UNIF. Such an observation can be attributed to the fact that BLEV and SLEV give more weight to the high-leverage-score data points. For the IBOSS method, the selected subsample includes all the "extreme" data points from all predictors. Such a subsample is most informative when the linear model assumption is valid (H. Wang et al., 2018). Finally, we observe that the subsample chosen by the proposed LowCon algorithm is most "uniformly-distributed" among all. Intuitively, such a pattern indicates the selected subsample yields an information matrix that has a relatively small condition number.

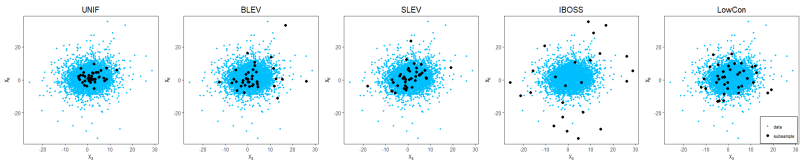

Figure 1.5: An illustration of five subsamples identified by different subsampling methods.

To compare the performance for different SLS estimators, we calculate the mean squared error for each of the SLS estimator based on 100 replicates, MSE $= \sum_{i=1}^{100} ||\widehat{\boldsymbol{\beta}}^{(i)} - \boldsymbol{\beta}_0||^2/100$, where $\widehat{\boldsymbol{\beta}}^{(i)}$ represents the SLS estimator in

the ith replication. Figure 1.6 and Figure 1.7 show the log(MSE) versus different subsample size under various settings, when $p = 10$ and 20, respectively. In both figures, each row represents a particular data distribution (**D1**–**D3**) and each column represents a particular misspecification term (**H1**–**H5**).

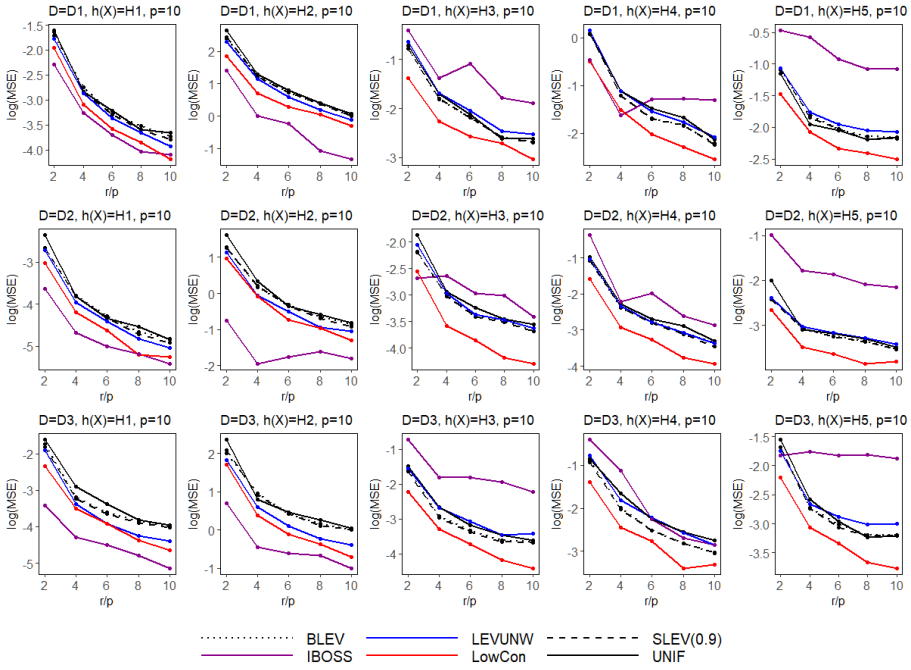

Figure 1.6: Comparison of different estimators when $p = 10$.

From Figure 1.6 and Figure 1.7, we first observe that UNIF (black solid line), as expected, is not performing well. As two of the random subsampling methods, BLEV (black dashed line) and SLEV (black dotted line) perform similarly, and both have superior performance than UNIF in most of the cases. Such a phenomenon is attributed to the fact that both methods tend to select the data points with high leverage-scores, and these points are more informative for estimating the coefficient, compared to randomly selected points.

Next, we find both LEVUNW (blue line) and IBOSS (purple line) have decent performance when the misspecification term equals zero (the most left column). Their performance, however, is inconsistent when the non-zero misspecification term exists, i.e., they perform well in some cases and perform poorly on others. Note that these two methods, at times, are even inferior to the UNIF method. Such an observation indicates that these two methods are effective when the linear model assumption is correct, but are not robust when the model is misspecified. We attribute this observation to the fact that the most informative data points derived under the postulated model do not necessarily lead to a decent estimator when the postulated model is incorrect. On the contrary, the

selected subsample can even be misleading and may dramatically pull back the performance of the subsample estimator.

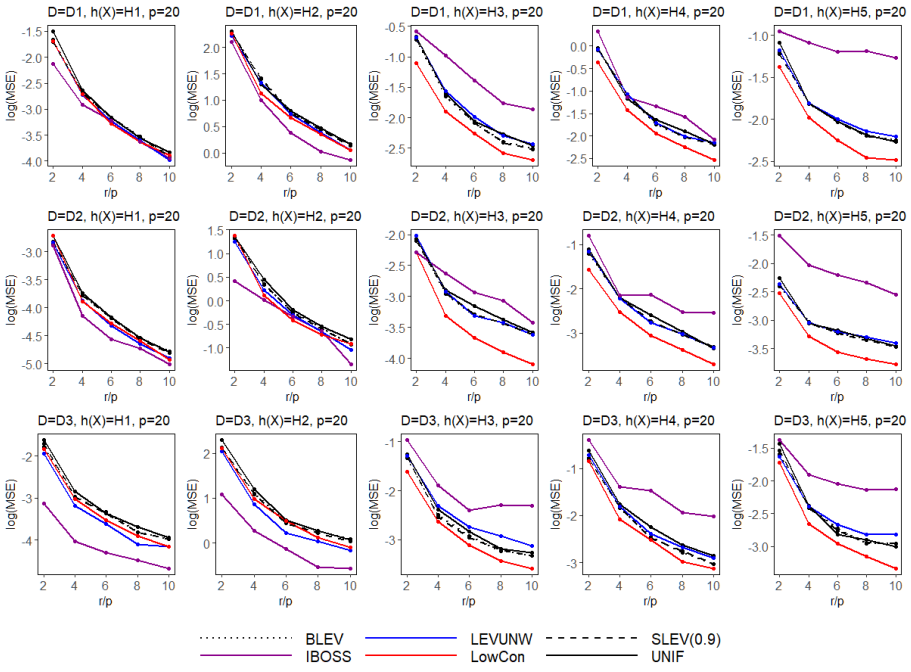

Figure 1.7: Comparison of different estimators when $p = 20$.

Finally, we observe that the proposed LowCon (red line) method is consistently better than the UNIF method. Furthermore, LowCon has a decent performance in most of the cases, especially when the model is misspecified. This observation indicates LowCon is able to give a robust estimator under various misspecified linear models. Such success can be attributed to the fact that the proposed estimator has a relatively small upper bound for the worst-case MSE.

1.5 Real Data Analysis

In this section, we evaluate the performance of different SLS estimators on two real-world datasets. One problem in real data analysis is that one does not know the true coefficient. It is thus impossible to calculate the mean squared error of a coefficient estimate. To overcome this problem, we consider the full-sample OLS estimator $\widehat{\boldsymbol{\beta}}_{OLS}$ and the following two estimators as the surrogates for the true coefficient $\boldsymbol{\beta}_0$. One of them is the M-estimator $\widehat{\boldsymbol{\beta}}_M$, which is a well-known estimator in robust linear regression (Meer et al., 1991). M-estimator can be calculated by using iterated re-weighted least squares, and it is known that such

an estimator is more robust to the potential outliers in the data, compared to the OLS estimator (Andersen, 2008). We obtained the M-estimator using the R package `MASS` with default parameters. The other estimator we considered is the cubic smoothing spline estimator for the "null space" (C. Gu, 2013; Wahba, 1990), denoted by $\widehat{\boldsymbol{\beta}}_{SS}$. We now briefly introduce the cubic smoothing spline estimator in the following.

Suppose the response y_i and the vector of predictors $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})^T$ are related through the unknown functions η such that $y_i = \eta(\boldsymbol{x}_i) + \epsilon_i$, where $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$. A widely used approach for estimating η is using minimizing the penalized likelihood function,

$$\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \eta(\boldsymbol{x}_i)\right)^2 + \lambda J(\eta), \tag{1.9}$$

where λ is the tuning parameter and $J(\eta)$ is a penalty term. We refer to C. Gu, 2013 for how to select the tuning parameter and how to construct the penalty term. The standard formulation of cubic smoothing splines performs the minimization of (1.9) in a reproducing kernel Hilbert space $\mathcal{H}$. In this case, the well-known representer theorem (Wahba, 1990) states that there exist vectors $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ and $\boldsymbol{c} = (c_1, \ldots, c_n)^T$, such that the minimizer of (1.9) is given by $\eta(\boldsymbol{x}) = \sum_{j=1}^{p}\beta_j x_{ij} + \sum_{i=1}^{n} c_i H(\boldsymbol{x}_i, \boldsymbol{x})$. Here, the bivariate function $H(\cdot, \cdot)$ is related to the reproducing kernel of $\mathcal{H}$, and we refer to C. Gu, 2013 for technical details. Let $\mathbf{H}$ be an $n \times n$ matrix with (i, j)-th element equals $H(\boldsymbol{x}_i, \boldsymbol{x}_j)$. By construction of $\mathcal{H}$, one has $J(\eta) = \boldsymbol{c}^T\mathbf{H}\boldsymbol{c}$ (C. Gu, 2013). Solving the minimization problem in (1.9) thus is equivalent to solving

$$(\widehat{\boldsymbol{\beta}}_{SS}, \widehat{\boldsymbol{c}}) = \underset{\boldsymbol{\beta}, \boldsymbol{c}}{\operatorname{argmin}} \frac{1}{n}(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{H}\boldsymbol{c})^T(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{H}\boldsymbol{c}) + \lambda \boldsymbol{c}^T\mathbf{H}\boldsymbol{c}. \tag{1.10}$$

We could thus view the estimated $\widehat{\boldsymbol{\beta}}_{SS}$ in (1.10) as the "corrected" estimate of the true coefficient $\boldsymbol{\beta}_0$ that takes into consideration of the misspecified terms quantified by $\mathbf{H}\widehat{\boldsymbol{c}}$. We calculate such an estimate using the R package `gss` with the default parameters.

To compare the performance of different SLS estimators, we calculate the empirical MSE (EMSE) through a hundred replicates. In the ith replicate, each subsampling method selects a subsample, leading to an SLS estimator $\widehat{\boldsymbol{\beta}}^{(i)}$. For each of the three full-sample estimators ($\widehat{\boldsymbol{\beta}}_{OLS}$, $\widehat{\boldsymbol{\beta}}_{M}$ and $\widehat{\boldsymbol{\beta}}_{SS}$), the correspond-

ing EMSE is then calculated as,

$$\text{EMSE}_{OLS} = \sum_{i=1}^{100} ||\widehat{\boldsymbol{\beta}}^{(i)} - \widehat{\boldsymbol{\beta}}_{OLS}||^2/100,$$
$$\text{EMSE}_{M} = \sum_{i=1}^{100} ||\widehat{\boldsymbol{\beta}}^{(i)} - \widehat{\boldsymbol{\beta}}_{M}||^2/100,$$
$$\text{EMSE}_{SS} = \sum_{i=1}^{100} ||\widehat{\boldsymbol{\beta}}^{(i)} - \widehat{\boldsymbol{\beta}}_{SS}||^2/100.$$

We emphasize that none of the three full-sample estimators can be regarded as the gold standard. However, a robust SLS estimator should at least be relatively "close" to all of these three estimators. That is to say, intuitively, a robust SLS estimator yields relatively small values of EMSE_{OLS}, EMSE_{M} and EMSE_{SS}.

Throughout this section, we set the parameter θ for the proposed LowCon method as 1. We opt to choose the subsample size r as $5p$, $10p$ and $20p$. The results in this section show the proposed SLS estimator yields the smallest empirical mean squared error.

1.5.1 Africa Soil Property Prediction

Soil functional properties refer to the properties related to a soil's capacity to support essential ecosystem services, which includes primary productivity, nutrient and water retention, and resistance to soil erosion (Hengl et al., 2015). The soil functional properties are thus important for planning sustainable agricultural intensification and natural resources management. To measure the soil functional properties in a target area, a natural paradigm is to first collect a sample of soil in this area, then analyze the sample using the technique of diffuse reflectance infrared spectroscopy (Shepherd & Walsh, 2002). Such a paradigm might be time-consuming or even impractical if the desired sample of soil from the target area is difficult to obtain. Predicting the soil functional properties is thus a measurement-constrained problem.

With the help of greater availability of Earth remote sensing data, the practitioners are provided new opportunities to predict soil functional properties at unsampled locations. One of the Earth remote sensing databases is provided by the Shuttle Radar Topography Mission (SRTM), which aims to generate the most complete high-resolution digital topographic database of Earth (Farr et al., 2007). In this section, we consider the *Africa Soil Property Prediction* dataset, which contains the soil samples from 1157 different areas ($n = 1157$). We aim

to analyze the relationship between the sand content, one of the soil functional properties, and the five features ($p = 5$) derived from the SRTM data. The features include: compound topographic index calculated from SRTM elevation data (CTI); SRTM elevation data (ELEV); topographic Relief calculated from SRTM elevation data (RELI); mean annual precipitation of average long-term Tropical Rainfall Monitoring Mission data (TMAP); and modified Fournier index of average long-term Tropical Rainfall Monitoring Mission data (TMFI). We assume the data follow the model

$$\begin{aligned} y_i &= \beta_0 + \beta_1 CTI_i + \beta_2 ELEV_i + \beta_3 RELI_i + \beta_4 TMAP_i \\ &\quad + \beta_5 TMFI_i + u_i, \quad i = 1, 2, \ldots, n, \end{aligned} \tag{1.11}$$

where u_i follow i.i.d. normal distribution $N(h(\boldsymbol{x}_i), \sigma^2)$. Here, one has $\boldsymbol{x}_i = (1, CTI_i, ELEV_i, RELI_i, TMAP_i, TMFI_i)^T$ and $h(\cdot)$ represents a multivariate function that is unknown to the practitioner. The postulated model is thus a misspecified linear model. In our measurement-constrained setting, we assume the response vector is hidden unless explicitly requested. We then estimate the true coefficient of the model (1.11), i.e., $(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5)^T$, using subsampling methods.

The methods considered here are uniform subsampling (UNIF), basic leverage subsampling (BLEV), shrinkage leverage subsampling (SLEV) with parameter $\alpha = 0.9$, unweighted-leverage subsampling (LEVUNW) (Ma, Mahoney, et al., 2015; Ma & Sun, 2015), information-based optimal subset selection (IBOSS) (H. Wang et al., 2018) and the proposed LowCon method. Table 1 summarizes the EMSEs for all six SLS estimators, and the best result in each row is in bold letter. We observe that the proposed LowCon method yields the best result in every row.

Table 1.1: EMSEs for the *Africa Soil Property Prediction* dataset

r	EMSE	UNIF	BLEV	SLEV	LEVUNW	IBOSS	LowCon
$5p$	OLS	6.09	3.23	2.82	2.56	34.87	**1.17**
	M	6.12	3.24	2.82	2.56	34.07	**1.14**
	SS	9.89	6.56	6.58	5.84	28.74	**2.81**
$10p$	OLS	1.92	1.26	1.23	0.91	18.62	**0.57**
	M	1.92	1.27	1.24	0.93	17.97	**0.56**
	SS	5.56	4.76	4.73	4.38	18.01	**2.69**
$20p$	OLS	0.78	0.55	0.53	0.44	2.84	**0.33**
	M	0.79	0.57	0.54	0.47	2.64	**0.33**
	SS	4.31	3.86	3.96	3.74	5.48	**2.48**

1.5.2 Diamond Price Prediction

The second real-data example we consider is the *Diamond Price Prediction* dataset, which contains the prices and the features of around 54,000 diamonds. Of interest is to analyze the relationship between the price of the diamond, and three continuous features (p=3): weight of the diamond (*caret*), total depth percentage (*depth*); and width of top of diamond relative to widest point (*table*).

As the same setting used in Section 5.1, we assume the data follow a misspecified linear model,

$$y_i = \beta_0 + \beta_1 caret_i + \beta_2 depth_i + \beta_3 table_i + u_i, \quad i = 1, 2, \ldots, n.$$

Here, u_i follow i.i.d. non-centered normal distribution $N(h(\boldsymbol{x}_i), \sigma^2)$, where $\boldsymbol{x}_i = (1, caret_i, depth_i, table_i)^T$, and $h(\cdot)$ is a multivariate function that is unknown to the practitioner. Note that the price of a diamond might be time-consuming or even impossible to obtain if the diamond has not been on the market yet. We thus assume the value of the response vector is hidden unless explicitly requested, and we estimate the true coefficient using subsampling methods.

Table 2 summarizes the EMSEs for all the subsample estimators, and the best result in each row is in bold letter. From Table 2, we observe that the proposed LowCon algorithm yields the best result in most of the cases.

Table 1.2: EMSEs for the *Diamond Price Prediction* data

r	EMSE	UNIF	BLEV	SLEV	LEVUNW	IBOSS	LowCon
$5p$	OLS	7.80	4.27	4.77	4.82	8.96	**3.40**
	M	8.29	4.39	5.17	4.94	6.07	**4.09**
	SS	12.42	8.43	9.33	9.05	9.36	**7.98**
$10p$	OLS	2.84	2.20	2.08	2.59	8.68	**1.50**
	M	3.10	2.71	2.37	2.83	5.82	**2.12**
	SS	7.14	6.90	6.44	7.02	9.29	**5.94**
$20p$	OLS	1.30	0.95	1.09	1.13	8.16	**0.78**
	M	1.70	**1.23**	1.36	1.24	5.38	1.32
	SS	5.82	5.29	5.43	5.29	8.79	**5.14**

1.6 Concluding Remarks

We considered the problem of estimating the coefficients in a misspecified linear model, under the measurement-constrained setting. When the model is correctly specified, various subsampling methods have been proposed to solve this problem. When the model is misspecified, however, we found the worst-

case bias for a subsample least squares estimator can be inflated to arbitrarily large. To overcome this problem, we aim to find a robust SLS estimator whose variance is bounded and the worst-case bias is relatively small. We found such a goal can be achieved by selecting a subsample whose information matrix has a relatively small condition number. Motivated by this, we proposed the LowCon subsampling algorithm which utilizes the orthogonal Latin hypercube design technique. We proved the proposed estimator has a finite mean squared error. Furthermore, the bias of the proposed estimator has an upper bound which approximately achieves the minimum value of the worst-case bias. We evaluated the performance of the proposed estimator through extensive simulation and real data analysis. Consistent with the theorem, the empirical results showed the proposed method has robust performance.

Chapter 2

An Optimal Transport Approach for Selecting a Representative Subsample

Subsampling methods have drawn significant attention in large-scale data analysis, active learning, and privacy-preserving analysis. Most of the existing subsampling methods are model-based methods, which assume the sample follows a pre-specified model. These methods, however, suffer from deteriorated performance in practice when the model specification is incorrect. Instead, we present a model-free subsampling method, by combining the idea of optimal transport map and star discrepancy, a metric that measures how uniformly-distributed a sample is. Specifically, the proposed method first transforms the given sample to be uniformly distributed, then selects a representative subsample in accordance with the star discrepancy, requiring only (approximately) linear computational time. Moreover, we theoretically demonstrate that the selected subsample can be used for efficient density estimation, by deriving the convergence rate for the subsample kernel density estimator. We explore our findings empirically and illustrate the benefits through two empirical studies: density estimation and active learning.

2.1 Introduction

Subsampling problem can be described as follows: given a sample $\{\boldsymbol{x}_i\}_{i=1}^n \in \mathbb{R}^d$ generated from an unknown probability distribution, the goal is to take a subsample $\{\boldsymbol{x}_i^*\}_{i=1}^r$, $r \ll n$, as a surrogate for the original sample. The

subsampling problem for a large sample with moderate dimension, i.e., "large n moderate d", has drawn significant attention in machine learning, statistics, and computer science. In active learning, there is usually a large sample of unlabeled data, and one is interested in selecting a subsample to label (Settles, 2012). In the privacy-preserving analysis, working on only a subset of data can reveal little confidential information. Subsampling methods have the potential to enhance data security (N. Li et al., 2012; Nissim et al., 2007). In model selection, a validation subsample resembling the full sample enables the effective model selection. Recently, subsampling methods have been used in numerical analysis to alleviate the computational burden in large-scale data analysisTsai et al., 2015; Zhou et al., 2017.

However, most of the existing subsampling methods are model-based subsampling, i.e., the data is assumed to follow a pre-specified model. The model-based subsampling aims to select an *informative subsample* for model-fitting and prediction (Huang et al., 2010). Such subsampling methods have been proposed for different models, which include linear regression (Derezinski et al., 2018; Drineas et al., 2006; Drineas et al., 2011; Ma, Mahoney, et al., 2015; Ma & Sun, 2015), logistic regression, l_p regression (A. Dasgupta et al., 2009), Gaussian mixture model (Feldman et al., 2011) and Bayesian logistic regression (Huggins et al., 2016). While the model-based subsampling methods have already yielded impressive achievements, the key to the success of these methods highly depends on the *correct* model specification. Nevertheless, in practice, model specification is a dynamic process of trial and error. For example, in supervised learning, we start with a high dimensional model with numerous features, and using model selection, we may end up with a low dimensional model with parsimonious features. In another instance, we may start with a linear regression model for a continuous response, by the categorization of the response, we may end up with a classification model. Existing literature suggests model-based subsampling methods may result in subsamples misleading such dynamic processes Tsao and Ling, 2012.

Different from model-based subsampling methods, there are emerging subsampling methods that aim to select a subsample that can capture the overall patterns of the original sample (Settles, 2012). These methods, termed as representative subsampling methods, are not confined by the model assumption, and provide a reasonable surrogate of the full sample regardless of the downstream analyses tasks performed. The representative subsampling methods can be divided into two classes: clustering-based methods and kernel-based methods. The clustering-based methods, which are usually developed for clustering analysis, include k-medoids method (Kaufman & Rousseeuw, 1987; H.-S. Park &

Jun, 2009), k-center method (Feder & Greene, 1988), and Wasserstein barycenter method (Cuturi & Doucet, 2014; Rabin et al., 2011). The k-medoids method is closely related to k-means algorithm, and the k-center method is widely used in fast multipole methods, and fast Gauss transform methods (Greengard & Strain, 1991; Lee & Gray, 2009; White et al., 1994; Yang et al., 2003). Wasserstein barycenter method aims to find the "mean" of a set of empirical probability measures under the optimal transport metric, and this "mean" itself can be regarded as a representative subsample. A similar idea has been explored in (Claici & Solomon, 2018) to selected a Wasserstein coreset. Although these clustering-based methods provide task-invariant subsamples, the empirical distributions of their subsamples may not resemble the probability distribution of the original data. Alternatively in mathematical language, as the subsample size increases, the probability distribution of the selected subsample by these methods does not necessarily converge to the true probability distribution (Mak, Joseph, et al., 2018; Y. Su, 2000).

To address such a limitation, the kernel-based representative subsampling methods utilize the kernel method to estimate the probability density function of the full sample and select a subsample that can effectively estimate the probability distribution. These methods include kernel herding (Chen & Zhang, 2014) and coreset for kernel density estimation (Phillips, 2013; Zheng et al., 2013; Zheng et al., 2017). In practice, however, the kernel-based methods depend on computationally intensive methods to tune kernel-bandwidth parameters and are thus not scalable to large sample analysis. Moreover, kernel-based methods usually suffer from deteriorated performance when they are applied to high-dimensional data.

Our contributions. We present an innovative model-free representative subsampling method, termed as space-filling design after optimal transport (SDAOT). Compared with existing representative subsampling methods, the proposed SDAOT method is computationally efficient, works well for high-dimensional data, and theoretically guaranteed to select a subsample that can effectively represent the underlying probability distribution. In this article, we demonstrate a method to quantify the representativeness of a subsample in accordance with the star discrepancy (Fang et al., 2005; Pukelsheim, 2006) after transforming the sample to be uniformly distributed. Utilizing the techniques of space-filling design (Pukelsheim, 2006) and optimal transport map (Villani, 2008), SDAOT provides a computational-efficient way to select a "relatively" representative subsample within the computational time at the order of $O(d^2 n \log(n))$. Theoretically, we derive the convergence rate for the SDOAT density estimator under general regularity conditions. We evaluate the empirical

performance for the proposed method by the applications of density estimation and active learning. The proposed method outperforms several state-of-the-art representative subsampling methods using extensive experiments on various synthetic and real-world datasets.

2.2 Definition and methodology

2.2.1 Star discrepancy

As an extensively adopted metric in experimental design and quasi-Monte Carlo methods (Fang & Wang, 1993; Niederreiter, 1992), star discrepancy measures the "discrepancy" between a set of discrete data points and $U[0,1]^d$, i.e., the uniform distribution on the d-dimensional unit hypercube $[0,1]^d$. Let $\mathcal{U}_r = \{\boldsymbol{u}_i\}_{i=1}^r$ be a set of r data points in $[0,1]^d$, $[\mathbf{0}, \boldsymbol{a}) = \prod_{j=1}^d [0, a_j)$ be a hyper-rectangle in $\mathbb{R}^d$, and $1\{\cdot\}$ be the indicator function. The star discrepancy are defined as follows (Fang et al., 2005; Pukelsheim, 2006).

Definition 1. (Star discrepancy) Given $\mathcal{U}_r$ and a hyper-rectangle $[\mathbf{0}, \boldsymbol{a})$, where $\boldsymbol{a} \in [0,1]^d$, the corresponding local discrepancy is defined as,

$$D(\mathcal{U}_r, \boldsymbol{a}) = |\frac{1}{r}\sum_{i=1}^{r} 1\{\boldsymbol{u}_i \in [\mathbf{0}, \boldsymbol{a})\} - \prod_{j=1}^{d} a_j|.$$

The star discrepancy is defined as,

$$D^*(\mathcal{U}_r) = \sup_{\boldsymbol{a}\in[0,1]^d} D(\mathcal{U}_r, \boldsymbol{a}).$$

We illustrate the idea of local discrepancy in Fig. 2.1. Three of ten data points locate in the rectangular $[\mathbf{0}, \boldsymbol{a})$, where $\boldsymbol{a} = (0.4, 0.5)^T$. The corresponding local discrepancy is thus calculated as $|3/10 - 0.4 \times 0.5| = 0.1$. The star discrepancy $D^*(\mathcal{U}_r)$ calculates the supreme of all the local discrepancy over $\boldsymbol{a} \in [0,1]^d$. As a result, a small value of $D^*(\mathcal{U}_r)$ indicates $\mathcal{U}_r$ is representative to the uniform distribution $U[0,1]^d$ and vice versa.

One limitation for star discrepancy, however, is that it can only be applied to the data points located in the unit hypercube. The limitation can be addressed by transforming the data points to the unit hypercube before calculating the star discrepancy. Denote F as the cumulative distribution function (CDF) of a sample $\{\boldsymbol{x}_i\}_{i=1}^n$. The transformed sample $\{F(\boldsymbol{x}_i)\}_{i=1}^n$ follows the uniform distribution $U[0,1]^d$. Star discrepancy thus can be employed to measure the *representativeness* of the sample $\{\boldsymbol{x}_i\}_{i=1}^n$ using $D^*(\{F(\boldsymbol{x}_i)\}_{i=1}^n)$. That is, the

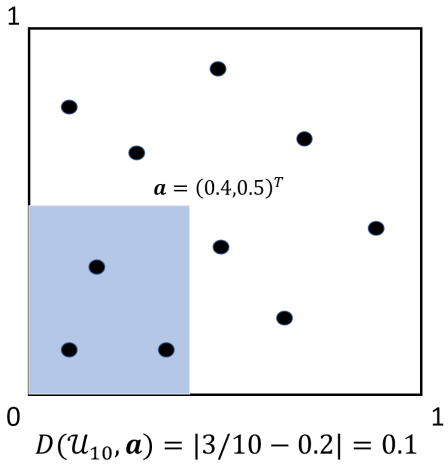

Figure 2.1: Illustration for the local discrepancy.

smaller the value of $D^*(\{F(\boldsymbol{x}_i)\}_{i=1}^n)$ is, the more representative the sample $\{\boldsymbol{x}_i\}_{i=1}^n$ will be.

In practice, however, such a measurement cannot be directly used to select the most representative subsample from the given sample. This is because (i) the explicit form of F is usually unknown in practice and estimating the empirical CDF may be computationally intensive, especially when the dimension d is large; and (ii) even when F is known and the transformed sample $\{F(\boldsymbol{x}_i)\}_{i=1}^n$ can be obtained, it is computationally infeasible to search the subsample with the smallest star discrepancy exhaustively. To surmount the computational challenges, we develop a new method integrating both space-filling design and optimal transport map.

2.2.2 Space-filling design

There are methods to generate the design points using directly minimizing the star discrepancy, e.g., the uniform design methods (Fang et al., 2005). Nevertheless, such methods are computationally expensive and are applicable only for small datasets on $U[0,1]^d$. To alleviate such a computational burden, the method which yields a set of data points with relatively small star discrepancy could be used as an alternative for uniform design methods. These alternatives include the space-filling design methods (Fang et al., 2005; Pukelsheim, 2006; Wu & Hamada, 2011) and the low-discrepancy sequence method(Owen, 2003). The former aims to generate a set of design points that spread out over the domain as uniformly as possible. The latter generates the design points sequen-

tially, and the generated points achieve an asymptotically fast decay rate for star discrepancy. For a Sobol sequence $\mathcal{S}_r = \{\boldsymbol{s}_i\}_{i=1}^r$, one kind of low-discrepancy sequence, $D^*(\mathcal{S}_r)$ converges to zero at the rate of $O(\log(r)^d/r)$ (Owen, 2003). In other words, the convergence rate of $D^*(\mathcal{S}_r)$ is of the order $O(r^{-(1-\delta)})$ for an arbitrary small $\delta > 0$ and fixed d, as r goes to infinity. Some space-filling design techniques can achieve even smaller convergence rate (Fang et al., 2005; Owen, 2003). For comparison, when a set of data points $\mathcal{X}_r = \{\boldsymbol{x}_i\}_{i=1}^r$ is randomly generated from $U[0,1]^d$, the convergence rate of $D^*(\mathcal{X}_r)$ is only of the order $O((\log\log(r)/r)^{1/2})$, which is much slower than $O(r^{-(1-\delta)})$ (Chung, 1949). Without loss of generality, in this chapter, we always assume the star discrepancy $D^*(\mathcal{S}_r)$ converges to zero with the rate $O(r^{-(1-\delta)})$.

The space-filling design techniques provide powerful tools to generate a set of representative design points in $U[0,1]^d$. Besides, these techniques could also be used to select a representative subsample from a given sample generated from $U[0,1]^d$, with the help of one-nearest-neighbor approximation. The detailed algorithm follows.

Algorithm 2 Select a subsample by approximating a space-filling design

Input: a sample $\mathcal{U}_n = \{\boldsymbol{u}_i\}_{i=1}^n$ generated from $U[0,1]^d$.
Generate a set of space-filling design points $\mathcal{S}_r = \{\boldsymbol{s}_i\}_{i=1}^r \in [0,1]^d$.
for $i = 1$ to r **do**
  Select the nearest neighbor for $\boldsymbol{s}_i$ from $\{\boldsymbol{u}_i\}_{i=1}^n$, using l_2 distance.
end for
Output: the final subsample is $\mathcal{U}_r^* = \{\boldsymbol{u}_i^*\}_{i=1}^r$.

The following Lemma characterizes the approximation errors of the subsample selected by Algorithm 2. The proof is relegated to the Appendix.

Lemma 2.2.1. *Let $\mathcal{S}_r = \{\boldsymbol{s}_i\}_{i=1}^r \in [0,1]^d$ be a set of design points which satisfy $D^*(\mathcal{S}_r) = O(r^{-(1-\delta)})$ for any arbitrary small $\delta > 0$, as $r \to \infty$. Suppose d is fixed, when $r = O(n^{1/d})$, as $n \to \infty$, we have $D^*(\mathcal{U}_r^*) = O_p(r^{-(1-\delta)})$.*

Lemma 2.2.1 states that the approximation error yielded by one-nearest-neighbour approximation is negligible, when d is fixed and $r = O(n^{1/d})$. In particular, Lemma 2.2.1 suggests the selected subsample $\mathcal{U}_r^*$ can effectively approximate the design points $\mathcal{S}_r$, in the sense that the convergence rate of $D^*(\mathcal{U}_r^*)$ is almost the identical as that of $D^*(\mathcal{S}_r)$.

To extend Algorithm 2 to the case when the cumulative distribution function F is non-uniform but has an explicit form, one can employ the following two-step strategy similar to the inverse transform sampling method (Devroye, 1986; Mosegaard & Tarantola, 1995). First, select a subsample $\{\boldsymbol{u}_i^*\}_{i=1}^r$ from

the transformed sample $\{F(\boldsymbol{x}_i)\}_{i=1}^n$ using Algorithm 2. Notice that the transformed sample is uniformly distributed on $[0,1]^d$ and the selected subsample $\{\boldsymbol{u}_i^*\}_{i=1}^r$ thus is relatively representative to $U[0,1]^d$. Second, take the desired representative subsample $\{F^{-1}(\boldsymbol{u}_i^*)\}_{i=1}^r$ using the look-up table. Such a two-step strategy, however, is inapplicable when F is unknown or is computational-intensive to obtain, especially when d is large[1]. To overcome such an obstacle, in the following, we introduce the optimal transport map, which serves as a surrogate for F.

[1] One exception is that when all the covariates of the sample are independent with each other, in which case one can directly calculate the multivariate CDF as the product of all the one-dimensional marginal CDF. Nevertheless, independent covariates are rarely the case in practice.

2.2.3 Optimal transport map

Optimal transport map (OTM) has been extensively used as a standard technique to transform one probability distribution to another. Recently, OTM has received significant attention due to its close relationship with generative models, including generative adversarial nets (Goodfellow et al., 2014), the "decoder" network in variational autoencoders (Kingma & Welling, 2013; Meng et al., 2019), among others. Moreover, OTM also plays essential roles in various machine learning applications, e.g., color transfer (Ferradans et al., 2014; Rabin et al., 2014), shape match (Z. Su et al., 2015), transfer learning (Courty et al., 2017), and natural language processing (Peyré, Cuturi, et al., 2019).

Instead of introducing the general definition of the OTM, we now present a specific map of our interest, and we refer to (Cuturi, 2013; X. D. Gu & Yau, 2008; Santambrogio, 2015; Villani, 2008) for more details. Let u be the uniform probability distribution on $[0,1]^d$. Let p_X and $\Omega \subseteq \mathbb{R}^d$ be the probability distribution and the domain of the random variable X, respectively. For all $B \subset \Omega$, denote $\phi_{\#}(p_X)(B) = p_X(\phi^{-1}(B))$. Among all the maps $\phi : \Omega \to [0,1]^d$ such that $\phi_{\#}(p_X) = u$ and $\phi_{\#}^{-1}(u) = p_X$, the OTM ϕ^* of our interest is the one which minimizes the L_2 cost, $\int_\Omega \|X - \phi(X)\|^2 \mathrm{d}p_X$, where $\|\cdot\|$ denotes the Euclidean norm.

As a special case, when $\Omega = \mathbb{R}$ and $d = 1$, the OTM ϕ^* is equivalent to the CDF F (Villani, 2008). This fact motivates us to use the OTM ϕ^* as a surrogate for F in high-dimensional cases.

Calculating the exact OTM on a large-scale sample may yield an enormous computational cost. In practice, ϕ^* can be approximated using the iterative method (Meng et al., 2019; Pitié et al., 2007) or the sliced method (Bonneel et al., 2015; Rabin et al., 2011). These methods tackle the problem of estimating a d-dimensional OTM iteratively by breaking down the problem into a series of subproblems, each of which finds a one-dimensional OTM using projected samples. Note that the one-dimensional OTM can be easily solved through

sorting algorithms. In this work, we opt to use the iterative approach for approximating ϕ^*, summarized in Algorithm 2.

Algorithm 3 Approximating the optimal transport map

Input: A sample matrix $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T \in \mathbb{R}^{n\times d}$, a space-filling design matrix $\mathbf{S} \in [0,1]^{n\times d}$.
$k \leftarrow 0$, $\mathbf{X}^{[0]} \leftarrow \mathbf{X}$, let $\mathbf{X}_{(j)}$ be the jth column of $\mathbf{X}$.
repeat
  generate a random orthogonal matrix $\mathbf{B} \in \mathbb{R}^{d\times d}$.
  $\mathbf{S}' \leftarrow \mathbf{SB}$ and $\mathbf{X}' \leftarrow \mathbf{X}^{[k]}\mathbf{B}$.
  for j in $1:d$ **do**
    find the 1-dimensional OTM ϕ_j^* that match $\mathbf{X}'_{(j)}$ to $\mathbf{S}'_{(j)}$ using sorting algorithms.
    $\mathbf{X}'_{(j)} \leftarrow \phi_j^*(\mathbf{X}'_{(j)})$.
  end for
  $\mathbf{X}^{[k+1]} \leftarrow \mathbf{X}'\mathbf{B}^{-1}$; $k \leftarrow k+1$.
until converge.
Output: $\widehat{\phi}(\boldsymbol{x}_i)$ is given by: $\widehat{\phi}(\boldsymbol{x}_i) = \boldsymbol{x}_i^{[k]}$, $i = 1, \ldots, n$.

2.2.4 SDAOT algorithm

We are now ready to present our main algorithm, named as Space-filling Design After Optimal Transport (SDAOT). In particular, SDAOT first transforms a given sample to be uniformly distributed, using Algorithm 3. Subsequently, SDOAT uses Algorithm 2 to select a subsample from the transformed sample, and the final subsample is obtained using the look-up table. The details are presented in Algorithm 4.

We illustrate the SDAOT algorithm in Figure 2.2. The synthetic data points (grey dots) in Figure 2(a) are generated from a two-dimensional donuts-shape distribution. Using Algorithm 3, the data points are transformed to be uniformly distributed in $[0,1]^2$, as shown in Figure 2(b). In Figure 2(c), we generate 32 design points (black triangles) using the max projection design, one kind of space-filling design methods (Joseph et al., 2015). For each design point, we find its nearest neighbor (red dots) from the transformed data points (grey dots). Notice that both the design points and the selected subsample are "space-filling." The red dots in Figure 2(d) are the final subsample obtained using the look-up table. One can observe that the subsample can effectively represent the population.

SDAOT algorithm involves both calculating the "forward step" from the probability distribution to the uniform distribution, and the "backward step"

that map the selected subsample to the original data points. Such a two-step procedure may look unnecessarily complicated at first glance. Indeed, one can directly calculate the "backward step" and map the design points to the probability distribution, then select the nearest neighbor for each mapped design point as the subsample point. One limitation of one-step "backward" approach, however, is that finding the nearest neighbor in the original sample, instead of the uniformly-distributed sample, may yield ultra-large approximation error. In contrast, our approach, which utilizes the "forward step," ensures that the approximation error is bounded, and the selected subsample can effectively approximate the design points, as stated in Lemma 1. Furthermore, the "backward" step in SDAOT algorithm can be easily implemented using a look-up table with negligible computation. Thus SDAOT algorithm is not as complicated as it seems.

Algorithm 4 Space-filling Design After Optimal Transport (SDAOT)

Input: A sample $\mathcal{X}_n = \{\boldsymbol{x}_i\}_{i=1}^n$.
Generate a set of space-filling design points $\mathcal{S}_r = \{\boldsymbol{s}_i\}_{i=1}^r \in [0,1]^d$.
Step 1: calculate the transformed sample $\{\widehat{\phi}(\boldsymbol{x}_i)\}_{i=1}^n$ (use Algorithm 3).
Step 2: select a subsample $\{\boldsymbol{u}_i^*\}_{i=1}^r$ from $\{\widehat{\phi}(\boldsymbol{x}_i)\}_{i=1}^n$ (use Algorithm 2).
Output: the final subsample is given by $\mathcal{X}_r^* = \{\widehat{\phi}^{-1}(\boldsymbol{u}_i^*)\}_{i=1}^r$.

2.2.5 Implementation details and computational cost

There are two steps in the Algorithm 4. In Step 1, we approximate the desired OTM using Algorithm 3, and the computational complexity for Algorithm 3 is of the order $O(Kdn\log(n))$, where K denotes the number of iterations.

Empirically, K is usually set to be at the order of $O(d)$ (Meng et al., 2019; Pitie et al., 2005; Pitié et al., 2007), and we find $K = 10d$ works well in most of our experiments. In this case, the computational complexity becomes $O(d^2 n\log(n))$. In Step 2, we select the subsample from the transformed sample using Algorithm 2, which includes two sub-steps: generating the design points and searching the nearest neighbor. Sobol sequence method(Owen, 2003) is conducted to generate the design points in the simulation. From the empirical perspective, we find other space-filling design techniques, e.g., Latin hypercube design (Pukelsheim, 2006), uniform design (Fang et al., 2000), and max projection design (Joseph et al., 2015), also yield results similar to ours. It is worth noting that the design points in our algorithm are generated beforehand; thus, the computational cost for generating the design points is not included in the computational cost for the SDAOT algorithm. In terms of searching the

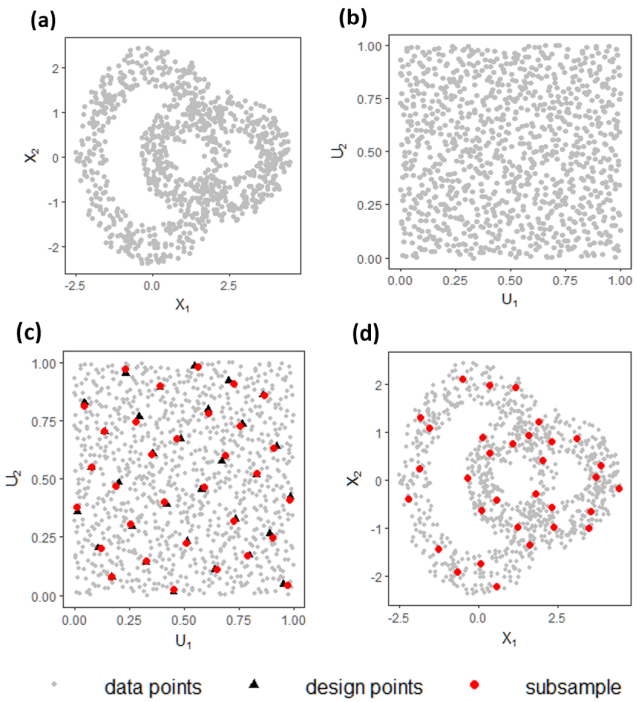

Figure 2.2: Illustration for Algorithm 3.

nearest neighbor, we opt to use the k-d tree method, whose computation cost is $O(n\log(n))$ (Bentley, 1975; Wald & Havran, 2006). In summary, the overall computational complexity for the SDAOT algorithm is $O(d^2 n\log(n))$. The memory costs for both Step 1 and Step 2 are of the order $O(nd)$. The overall memory cost is thus of the order $O(nd)$.

To demonstrate the computational cost for SDOAT empirically, we evaluate the CPU time for SDAOT according to various sample sizes n, dimensions d and subsample sizes r, respectively. The CPU time is obtained using an Intel 2.6GHz processor. Currently, the code is implemented in R, and a faster running time could be achieved after implementing the code in Python.

The results are shown in Fig. 2.3, in which the 95% confidence bands (gray shadows) are calculated using 100 replicates. In the left panel, we observe that the CPU time roughly linearly increases as the sample size n increases. The middle panel shows the CPU time has a quadratic growth when the dimension d increases, however, the overall computational time is still reasonable and moderate. If further computation saving is desired, one may use dimension reduction methods, e.g., principal component analysis, to reduce the number

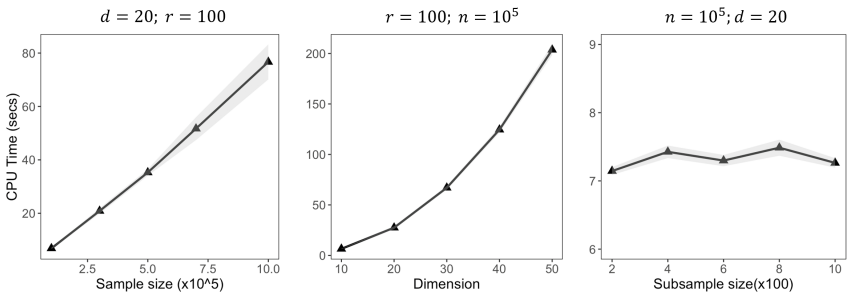

Figure 2.3: CPU time of SDAOT with various sample sizes (left), dimensions (middle), and subsample sizes (right).

of dimensions before using SDAOT. In the right panel, one can observe that the CPU time remains stable regardless of the changes in subsample size r.

2.3 Theoretical results

In this section, we demonstrate the use of SDAOT for efficient density estimation. Here, we focus on the kernel density estimation method (KDE), which is a widely-used nonparametric density estimation method, see Scott, 2015 for more reference.

Regularity conditions. Let $K(\cdot)$ be a non-negative real-valued integrable function, which satisfies the following conditions: (a) $\int_{-\infty}^{\infty} K(z)\mathrm{d}z = 1$; (b) $K(-z) = K(z)$ for all $z \in \mathbb{R}$; (c) $\int_{-\infty}^{\infty} z^2 K(z)\mathrm{d}z < \infty$; (d) $\int_{-\infty}^{\infty} K^2(z)\mathrm{d}z < \infty$; (e) $\int_{-\infty}^{\infty} (K'(z))^2 \mathrm{d}z < \infty$. One example which satisfies these conditions is the Gaussian kernel function. Denote x_{ij} as the jth entry of $\boldsymbol{x}_i$, $h > 0$ as the bandwidth, and $K_h(t)$ as an abbreviation for $K(t/h)/h$.

Recall that $\mathbf{X}$ denotes the full-sample matrix. For $\boldsymbol{z} \in \mathbb{R}^d$, the full-sample product kernel density estimator can be written as

$$\widehat{p}(\boldsymbol{z}|\mathbf{X}) = \sum_{i=1}^{n} \{ \prod_{j=1}^{d} K_h \left(z_j - x_{ij} \right) \} / n.$$

The asymptotic integrated mean squared error (AIMSE) of an estimator $\widehat{p}$ is defined as

$$\mathrm{AIMSE}(\widehat{p}) = \int E\{(\widehat{p}(\boldsymbol{z}) - p(\boldsymbol{z}))^2\} \mathrm{d}\boldsymbol{z}.$$

Two regularity conditions for p are required : for $j = 1, \ldots, d$, (1) $\partial^2 p / \partial x_j^2$ is absolutely continuous; (2) $\partial^3 p / \partial x_j^3 \in L_2$.

Let $\mathbf{X}^* \in \mathbb{R}^{r\times d}$ be the subsample matrix yielded by the proposed SDAOT algorithm, and $\widetilde{p}(\boldsymbol{z}|\mathbf{X}^*)$ be the subsample estimator. We derive the following convergence rate of AIMSE$(\widetilde{p}(\boldsymbol{z}|\mathbf{X}^*))$. The proof is relegated to the Appendix.

Theorem 2.3.1. *Suppose $K(\cdot)$ satisfies the regularity conditions (a)-(e) and the population density function p satisfies the regularity conditions (1) and (2). Suppose p has a compact convex domain $\Omega \subset \mathbb{R}^d$. For any arbitrary small $\delta > 0$, if $r = O(n^{1/d})$ and $h = O(r^{-2(1-\delta)/(d+6)})$, we have*

$$AIMSE(\widetilde{p}(\boldsymbol{z}|\mathbf{X}^*)) \leq O_p(r^{-8(1-\delta)/(d+6)}). \tag{2.1}$$

Theorem 2.3.1 reveals that the SDAOT can be used for efficient density estimation. To see this, let $\mathbf{X}^+ \in \mathbb{R}^{r\times d}$ be a randomly selected subsample matrix and $\widetilde{p}(\boldsymbol{z}|\mathbf{X}^+)$ as the corresponding subsample estimator. According to Theorem 6.4 of Scott, 2015, as $r = o(n)$ and $n \to \infty$, if $h = O(r^{-1/(4+d)})$, AIMSE$(\widetilde{p}(\boldsymbol{z}|\mathbf{X}^+))$ achieves the optimal convergence rate $O(r^{-4/(d+4)})$. Note that such a convergence rate is much slower than the convergence rate in (2.1). Theorem 2.3.1 thus implies the proposed estimator $\widetilde{p}(\boldsymbol{z}|\mathbf{X}^*)$ converges to the true density function faster than $\widetilde{p}(\boldsymbol{z}|\mathbf{X}^+)$.

An essential regularity condition in Theorem 2.3.1 is that the domain of p is compact convex. Empirically, we find the proposed estimator still works reasonably well even when such a condition does not hold, as shown in Section 4. How to determine the bandwidth h, or in general, the bandwidth matrix $\mathbf{H} \in \mathbb{R}^{d\times d}$, is essential for the performance of the kernel density estimators[2]. Let $\widehat{\boldsymbol{\Sigma}}$ be the empirical variance-covariance matrix. In practice, for the random subsample estimator $\widetilde{p}(\boldsymbol{z}|\mathbf{X}^+)$, we opt to choose $\mathbf{H} = r^{-1/(d+4)} \times \widehat{\boldsymbol{\Sigma}}^{1/2}$, according to the general Scott's rule (Scott, 2015). In terms of the proposed estimator $\widetilde{p}(\boldsymbol{z}|\mathbf{X}^*)$, we choose $\mathbf{H} = r^{-2/(d+6)} \times \widehat{\boldsymbol{\Sigma}}^{1/2}$, according to Theorem 1.

[2] As a special case, in Theorem 1, the bandwidth matrix $\mathbf{H} = h \cdot \mathbf{I}_p$, where $\mathbf{I}_p$ denotes the identity matrix.

2.4 Numerical experiments

2.4.1 SDAOT subsample visualization

We show the performance of the subsamples (red dots) selected by SDAOT (lower row) versus the subsamples selected by uniform sampling (upper row).

The samples (grey dots) are generated from three different bivariate distributions: standard Gaussian distribution (left), mixture beta distribution (middle), and mixture Gaussian distribution (right). In figures of the left column, one can observe that the uniformly selected subsample is far from symmetric. In figures of the middle and the right columns, one can see that some of the peaks in the

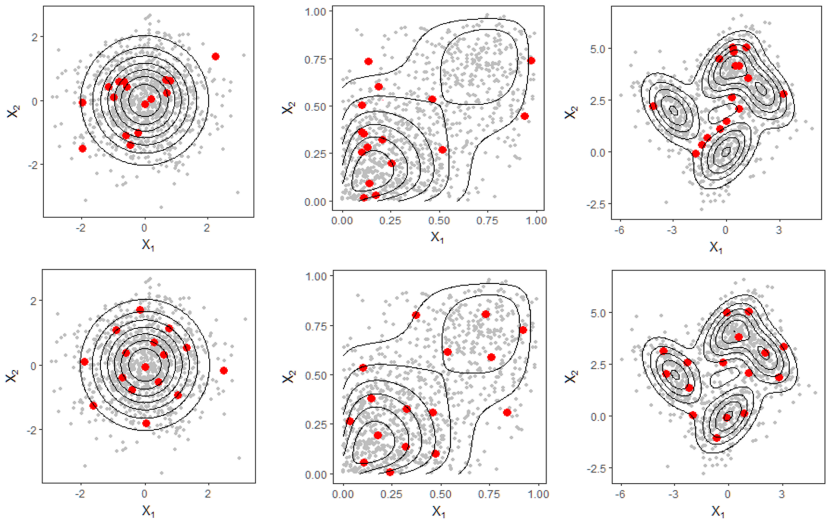

Figure 2.4: Subsamples (red dots) selected by SDAOT (lower) versus randomly selected subsamples (upper). Contour curves (black) are superimposed.

probability distribution are largely overlooked by the uniform random subsampling method. In summary, in all these three cases, the subsamples identified by SDAOT have a more appealing visual representation of the corresponding probability distribution than that selected by the uniform subsampling method.

2.4.2 SDAOT for density estimation

We now investigate the performance of SDAOT for density estimation in comparison with uniform subsampling (UNIF), k-medoids (KM), support point (SP), and kernel herding (KH). We simulate the data with $n = 10^4$, $d = \{5, 10, 20\}$ and $r = \{32, 64, 128, 256\}$ from a multivariate standard Gaussian distribution and a "dumbbell-like" mixture Gaussian distribution (Duong et al., 2007), $N(\mathbf{1}, \mathbf{I}_d)/4 + N(-\mathbf{1}, \mathbf{I}_d)/4 + N(\mathbf{0}, \mathbf{\Sigma})/4$, where $\mathbf{\Sigma} = 0.8^{|i-j|}$, $i, j = 1, ..., d$. The Gaussian kernel is used in the simulation, and we opt to use the general Scott's rule for selecting the bandwidth for the kernel function (Scott, 2015). Specifically, we choose the bandwidth matrix $r^{-2/(d+6)} \times \widehat{\mathbf{\Sigma}}^{1/2}$ for the SDAOT estimator (according to Theorem 1) and $r^{-1/(d+4)} \times \widehat{\mathbf{\Sigma}}^{1/2}$ for the other estimators, where $\widehat{\mathbf{\Sigma}}$ denotes the empirical variance-covariance matrix. To measure the estimation accuracy, we use the empirical Hellinger distance (D. Li et al., 2016), $1 - \frac{1}{n}\sum_{i=1}^{n}\sqrt{\widehat{p}(\boldsymbol{x}_i)/p(\boldsymbol{x}_i)}$, where $\widehat{p}(\boldsymbol{x})$ is the subsample kernel density estimate of the density $p(\boldsymbol{x})$.

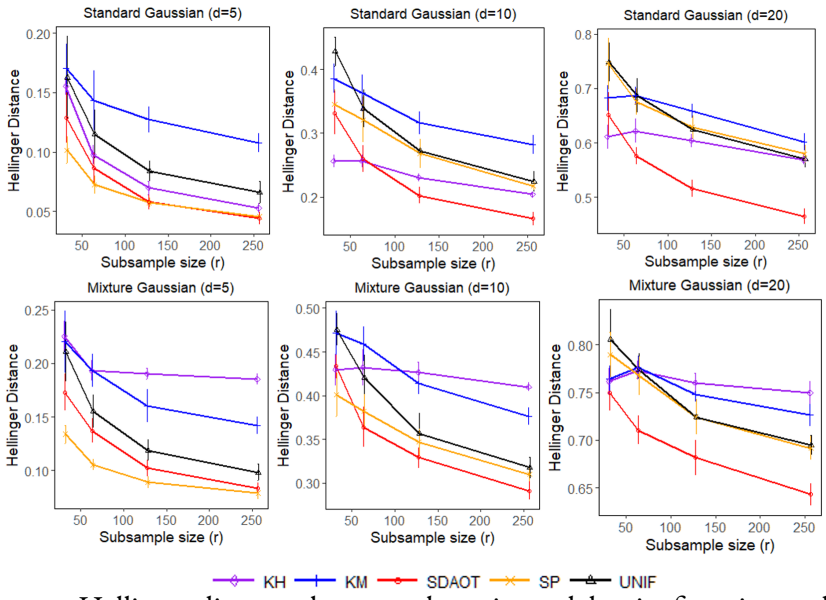

Figure 2.5: Hellinger distances between the estimated density function and true density function.

Figure 2.5 shows the result for the estimation accuracy, where the lines are the mean Hellinger distance, and the vertical bars are the standard error obtained across ten replicates. The upper and lower row show the result for the standard Gaussian case and the mixture Gaussian cases, respectively.

We observe that the UNIF (black line) yields a decent performance. The KM (blue line) does not perform as well as the UNIF in most cases since the subsamples selected by the KM are not representative of the probability distribution. We also observe that the KH method does not perform well, which may be attributed to the fact that the performance of the KH is sensitive to the choice of its parameter. The SP method (orange line) performs reasonably well when d is small, and its performance deteriorates as d increases. Finally, we observe that the proposed SDAOT method (red line) outperforms the other methods significantly in most of the cases. Such success can be attributed to the fact that the selected subsamples are representative of the probability distribution, and the SDAOT algorithm is adaptive to the high-dimensional cases.

2.4.3 SDAOT for active learning

Active learning approaches aim to make an accurate prediction, with the number of labeled training data points as small as possible (Cohn et al., 1996; Krogh & Vedelsby, 1995). These approaches are essential for numerous sophisticated

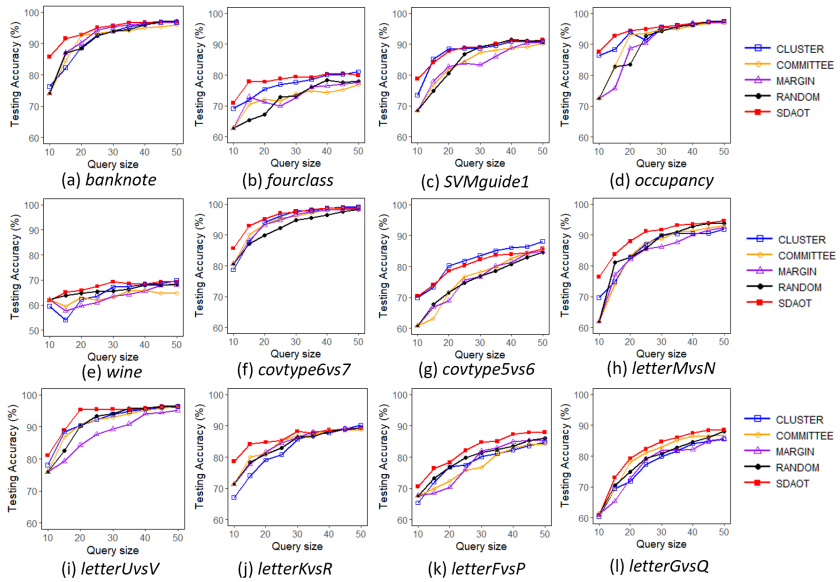

Figure 2.6: The mean classification accuracy on testing sets across ten replicates.

supervised learning tasks, where the labeled instances are challenging, time-consuming, or expensive to obtain . Take speech recognition as an example; accurate labeling of speech utterances is extremely time-consuming and requires trained linguists. It is reported that annotation at the level of the phoneme can take 400 times longer than the actual audio (Settles, 2012).

In general, active learning approaches select the data points (also termed as the query points) iteratively and interactively. In each iteration, one query the oracle to obtain the label at a new query point, based on certain criteria. The SDAOT algorithm can be cast as an active learning approach. In particular, we generate Sobol sequence $\mathcal{S}_r$ (Owen, 2003). in Algorithm 1 and select the query points sequentially in Algorithm 3. We now compare SDAOT in active learning and with the following baseline methods: (1) random sampling (RANDOM), (2) query by committee (COMMITTEE) which select query points that maximize the "disagreement" among different models (Settles, 2012), (3) margin-based method (MARGIN) which choose query points that lie on the margin of the decision line (Schohn & Cohn, 2000), and (4) hierarchical-clustering-based method (CLUSTER) which select the center nodes in each cluster (S. Dasgupta & Hsu, 2008).

We consider twelve binary-class real-world datasets: *Fourclass* Ho and Kleinberg, 1996, *SVMguide1* Hsu et al., 2003, *banknote*, *occupancy*, *wine*, and *covtype*,

the last four of which are downloaded from UCI machine learning repository (Asmussen & Glynn, 2007): Since *covtype* is a multi-class data set, we construct two binary-class datasets: data with labels 5 and 6, data with labels 6 and 7, respectively. *Letter* is also a multi-class data set, and we construct a binary class data set for each of the following pairs of letters that are relatively difficult to distinguish, i.e., M *vs* N, U *vs* V, K *vs* R, F *vs* P, and G *vs* Q. For each dataset, we replicate the experiment ten times. In each replication, each dataset is randomly divided into the training set testing set of equal sizes.

We evaluate the classification model by its mean classification accuracy on the testing set. The classification accuracy is defined as $(TP + FN)/n$, where n denotes the size of the testing set, and TP and FN denote true positive and false negative, respectively. We opt to use the support vector machine (SVM) (implemented by the R package "e1071" (Meyer et al., 2015)) for classification. The RBF kernel with default parameters is used in SVM, and the parameter λ is selected using cross-validation. To mimic the real scenarios where the labeling cost is expensive, we select only a small number of query points, ranging from 10 to 50, in each dataset. For COMMITTEE and MARGIN where several initial labeled data points are required, ten data points are randomly selected and labeled.

Figure 2.6 shows the mean classification accuracy of different active learning methods versus different numbers of query points. We first observe that RANDOM (black line) yields decent performance even though such a method is expected to be less efficient than others. A similar observation was also found in Guyon et al., 2011, where the random sampling strategy is a runner-up in the active learning challenge. It is suggested that the advantages of state-of-the-art active learning methods over the random sampling methods decreases as the full sample classification accuracy decreases. In specific, (Mussmann & Liang, 2018) indicated that when the full-sample classification accuracy is lower than 90%, the random sampling method is non-significantly worse than active learning methods. Note that most of the datasets we considered in the experiments have the full sample classification accuracy no more than 90%, and such a fact may be the cause of why the naive random sampling method has decent performance in most of our experiments.

MARGIN (purple line) and COMMITTEE (orange line) do not perform well when the number of the query points is small. These two methods quickly catch up with other methods as the number of query points increases. Such a phenomenon is attributed to the fact that the performance of these two methods highly depends on the accuracy of the learned classification model. Nevertheless, with only a few training data points, the learned classification models

tend to be inaccurate, resulting in unsatisfactory results. Compared with the two aforementioned "semi-supervised" methods, CLUSTER (blue line) is an "unsupervised" method in the sense that it selects the next query points without using the information from the available labels. We observe that the performance of the CLUSTER is inconsistent, i.e., it performs well on some datasets and performs poorly on others. We attribute such an observation to the fact that the cluster structure of the training set may not always be informative to learn the desired classification model. Finally, we observe that the proposed SDAOT method (red line) outperforms the other baseline methods significantly in most of the cases, especially when the number of query data points is small. Such success can be attributed to the fact that the query points selected by SDAOT are representative of the probability distribution. Intuitively, the classification models learned from a set of representative query points are very close to the classification models learned from the whole training set.

2.5 Concluding remarks

We consider the problem of identifying a subsample that can effectively represent the underlying probability distribution of a given sample. The key step in the proposed SDAOT algorithm is the optimal transport map, which is employed to transform the sample to be uniformly distributed. We demonstrate the selected subsample has an appealing visual representation of the full sample, and the computational cost is only of the (approximately) linear order of the sample size. The proposed algorithm provides a systematic way of selecting representative subsamples and can be used for efficient density estimation and active learning.

Chapter 3

More Efficient Approximation of Smoothing Splines using Space-filling Basis Selection

We consider the problem of approximating smoothing spline estimators in a nonparametric regression model. When applied to a sample of size n, the smoothing spline estimator can be expressed as a linear combination of n basis functions, requiring $O(n^3)$ computational time when the number of predictors $d \geq 2$. Such a sizable computational cost hinders the broad applicability of smoothing splines. In practice, the full sample smoothing spline estimator can be approximated by an estimator based on q randomly-selected basis functions, resulting in a computational cost of $O(nq^2)$. It is known that these two estimators converge at the identical rate when q is of the order $O\{n^{2/(pr+1)}\}$, where $p \in [1, 2]$ depends on the true function η, and $r > 1$ depends on the type of spline. Such q is called the essential number of basis functions. In this article, we develop a more efficient basis selection method. By selecting the ones corresponding to roughly equal-spaced observations, the proposed method chooses a set of basis functions with a large diversity. The asymptotic analysis shows our proposed smoothing spline estimator can decrease q to roughly $O\{n^{1/(pr+1)}\}$, when $d \leq pr + 1$. Applications on synthetic and real-world datasets show the proposed method leads to a smaller prediction error compared with other basis selection methods.

3.1 Introduction

Consider the nonparametric regression model $y_i = \eta(x_i) + \epsilon_i (i = 1, \ldots, n)$, where $y_i \in \mathbb{R}$ denotes the ith observation of the response, η represents an unknown function to be estimated, $x_i \in \mathbb{R}^d$ is the ith observation of the predictor variable, and $\{\epsilon_i\}_{i=1}^n$ are independent and identically distributed random errors with zero mean and unknown variance σ^2. The function η can be estimated by minimizing the penalized least squares criterion,

$$\frac{1}{n}\sum_{i=1}^{n}\{y_i - \eta(x_i)\}^2 + \lambda J(\eta), \tag{3.1}$$

where $J(\eta)$ denotes a quadratic roughness penalty (C. Gu, 2013; Wahba, 1990; X. Wang et al., 2011). The smoothing parameter λ here administrates the trade-off between the goodness-of-fit of the model and the roughness of the function η. In this chapter, expression (3.1) is minimized in a reproducing kernel Hilbert space $\mathcal{H}$, which leads to a smoothing spline estimate for η.

Although univariate smoothing splines can be computed in $O(n)$ time (Reinsch, 1967), it takes $O(n^3)$ time to find the minimizer of (3.1) when $d \geq 2$. Such a computational cost hinders the use of smoothing splines for large samples. To reduce the computational cost for smoothing splines, extensive efforts have been made to approximate the minimizer of (3.1) by restricting the estimator $\hat{\eta}$ to a subspace $\mathcal{H}_E \subset \mathcal{H}$. Let the dimension of the space $\mathcal{H}_E$ be q and the restricted estimator be $\hat{\eta}_E$. Compared with the $O(n^3)$ computational cost of calculating $\hat{\eta}$, the computational cost of $\hat{\eta}_E$ can be reduced to $O(nq^2)$. Along this line of thinking, numerous studies have been developed in recent decades. Luo and Wahba, 1997 and H. H. Zhang et al., 2004 approximated the minimizer of (3.1) using variable selection techniques. Pseudosplines (Hastie, 1996) and penalized splines (Ruppert et al., 2009) were also developed to approximate smoothing splines.

Although these methods have already yielded impressive algorithmic benefits, they are usually *ad hoc* in choosing the value of q. The value of q regulates the trade-off between the computational time and the prediction accuracy. One fundamental question is how small q can be in order to ensure the restricted estimator $\hat{\eta}_E$ converge to the true function η at the identical rate as the full sample estimator $\hat{\eta}$. To answer this question, C. Gu and Kim, 2002; Ma, Huang, et al., 2015 developed random sampling methods for selecting the basis functions and established the coherent theory for the convergence of the restricted estimator $\hat{\eta}_E$. To ensure that $\hat{\eta}_E$ has the same convergence rate as $\hat{\eta}$, both methods in C. Gu and Kim, 2002 and Ma, Huang, et al., 2015 require q be of the order

$O\{n^{2/(pr+1)+\delta}\}$, where δ is an arbitrary small positive number, $p \in [1, 2]$ depends on the true function η, and r depends on the fitted spline. It is shown that fewer basis functions are needed to warrant the aforementioned convergence rate if we select the basis functions $\{R(z_j, \cdot)\}_{j=1}^q$, where $\{z_j\}_{j=1}^q$ are roughly equal-spaced. However, they only provide the theory in the univariate predictor case, and their method cannot be directly applied to multivariate cases.

In this chapter, we develop a more efficient computational method to approximate smoothing splines. The distinguishing feature of the method is that it considers the notion of diversity of the selected basis functions. We propose the space-filling basis selection method, which chooses the basis functions with a large diversity, by selecting the ones that correspond to roughly uniformly-distributed observations. When $d \leq pr + 1$, we show that the smoothing spline estimator proposed here has the same convergence rate as the full sample estimator, and the order of the essential number of basis function q is reduced to $O\{n^{(1+\delta)/(pr+1)}\}$.

3.2 Backgrounds

Let $\mathcal{H} = \{\eta : J(\eta) < \infty\}$ be a reproducing kernel Hilbert space, where $J(\cdot)$ is a squared semi-norm. Let $\mathcal{N}_J = \{\eta : J(\eta) = 0\}$ be the null space of $J(\eta)$ and assume that $\mathcal{N}_J$ is a finite-dimensional linear subspace of $\mathcal{H}$ with basis $\{\xi_i\}_{i=1}^m$ in which m is the dimension of $\mathcal{N}_J$. Let $\mathcal{H}_J$ be the orthogonal complement of $\mathcal{N}_J$ in $\mathcal{H}$ such that $\mathcal{H} = \mathcal{N}_J \oplus \mathcal{H}_J$. The space $\mathcal{H}_J$ is a reproducing kernel Hilbert space with $J(\cdot)$ as the squared norm. The reproducing kernel of $\mathcal{H}_J$ is denoted by $R_J(\cdot, \cdot)$. The well-known representer theorem (Wahba, 1990) states that there exist vectors $d = (d_1, \ldots, d_m)^T \in \mathbb{R}^m$ and $c = (c_1, \ldots, c_n)^T \in \mathbb{R}^n$, such that the minimizer of (3.1) in $\mathcal{H}$ is given by $\eta(x) = \sum_{k=1}^m d_k \xi_k(x) + \sum_{i=1}^n c_i R_J(x_i, x)$. Let $Y = (y_1, \ldots, y_n)^T$ be the vector of response observations, S be the $n \times m$ matrix with the (i, j)th entry $\xi_j(x_i)$, and R be the $n \times n$ matrix with the (i, j)th entry $R_J(x_i, x_j)$. Solving the minimization problem in (3.1) thus is equivalent to solving

$$(\hat{d}, \hat{c}) = \underset{d,c}{\operatorname{argmin}} \frac{1}{n}(Y - Sd - Rc)^T(Y - Sd - Rc) + \lambda c^T R c, \qquad (3.2)$$

where the smoothing parameter λ can be selected based on the generalized cross-validation criterion (Wahba & Craven, 1978). In a general case where $n \gg m$ and $d \geq 2$, the computation cost for calculating $(\widehat{d}, \widehat{c})$ in equation (3.2) is of the order $O(n^3)$, which is prohibitive when the sample size n is considerable.

To reduce this computational burden, one can restrict the full sample estimator $\hat{\eta}$ to a subspace $\mathcal{H}_E \subset \mathcal{H}$, where $\mathcal{H}_E = \mathcal{N}_J \oplus \text{span}\{R_J(x_i^*, \cdot), i = 1, \ldots, q\}$. Here, $\mathcal{H}_E$, termed as the effective model space, can be constructed by selecting a subsample $\{x_i^*\}_{i=1}^q$ from $\{x_i\}_{i=1}^n$. Such an approach is thus called the basis selection method.

Denote a matrix $R_* \in \mathbb{R}^{n\times q}$ with the (i,j)th entry $R_J(x_i, x_j^*)$ and $R_{**} \in \mathbb{R}^{q\times q}$ with the (i,j)th entry $R_J(x_i^*, x_j^*)$. The minimizer of (3.1) in the effective model space $\mathcal{H}_E$ thus can be written as

$$\eta_E(x) = \sum_{k=1}^{m} d_k \xi_k(x) + \sum_{i=1}^{q} c_i R(x_i^*, x)$$

and the coefficients, $d_E = (d_1, \ldots, d_m)^T$ and $c_E = (c_1, \ldots, c_q)^T$ can be obtained through solving

$$(\hat{d}_E, \hat{c}_E) = \underset{d_E, c_E}{\text{argmin}} \frac{1}{n}(Y - Sd_E - R_* c_E)^T (Y - Sd_E - R_* c_E) \tag{3.3}$$

$$+\lambda c_E^T R_{**} c_E. \tag{3.4}$$

The evaluation of the restricted estimator $\hat{\eta}_E$ at sample points thus satisfies $\hat{\eta}_E = S\hat{d}_E + R_*\hat{c}_E$, where $\hat{\eta}_E = \{\hat{\eta}_E(x_1), \ldots, \hat{\eta}_E(x_n)\}^T$. In a general case where $m \ll q \ll n$, the overall computational cost for the basis selection method is $O(nq^2)$, which is a significant reduction compared with $O(n^3)$. Recall that the value of q controls the trade-off between the computational time and the prediction accuracy. To ensure that $\hat{\eta}_E$ converges to the true function η at the same rate as $\hat{\eta}$, researchers showed that the essential number of basis functions q needs to be of the order $O\{n^{2/(pr+1)+\delta}\}$, where δ is an arbitrary small positive number (Kim & Gu, 2004; Ma, Huang, et al., 2015). In the next section, we present the proposed space-filling basis selection method, which reduces such an order to $O\{n^{(1+\delta)/(pr+1)}\}$.

3.3 Space-filling basis selection

3.3.1 Motivation and Notations

To motivate the development of the proposed method, we first re-examine the ensemble learning methods, which are well-known in statistics and machine learning (Dietterich, 2002; Rokach, 2010). To achieve better predictive performance than a single learner [3], ensemble learning methods first build a committee which consists of a number of different learners, then aggregate the predictions

[3] A learner is either a model or a learning algorithm.

of these learners in the committee. The aggregation is usually achieved by employing the majority vote or by calculating a weighted average. The diversity among the learners in the committee holds the key to the success of the ensemble learning methods. A large diversity in the committee yields a better performance of ensemble learning methods (Kuncheva & Whitaker, 2003).

The restricted smoothing spline estimator $\hat{\eta}_E$ can be considered as an ensemble learning method. In particular, the prediction of $\hat{\eta}_E$ is conducted by taking a weighted average of the predictions of the selected basis functions $R_J(x_i^*, \cdot)$, $i \in \{1, \ldots, q\}$, in addition to the basis functions in the null space $\mathcal{N}_J$. Inspired by Kuncheva and Whitaker, 2003, we propose to select a subsample $\{x_i^*\}_{i=1}^q$, such that the diversity among the basis functions $\{R_J(x_i^*, \cdot)\}_{i=1}^q$ is as large as possible. One crucial question is how to measure the diversity among a set of basis functions. Notice that adjacent data points, i.e., $x_i^* \approx x_j^* (i, j \in \{1, \ldots, q\})$ yields similar basis functions, i.e., $R_J(x_i^*, \cdot) \approx R_J(x_j^*, \cdot)$. On the other hand, if x_i^* is far away from x_j^*, the basis function $R_J(x_i^*, \cdot)$ tends to be different from $R_J(x_j^*, \cdot)$. These observations inspire us to select a set of basis functions $\{R_J(x_i^*, \cdot)\}_{i=1}^q$ where $\{x_i^*\}_{i=1}^q$ are as uniformly-distributed as possible. The uniformly-distributed property, usually termed as the space-filling property in the experimental design literature (Pukelsheim, 2006), can be systematically measured by the star discrepancy.

Since the star discrepancy is defined for data in $[0, 1]^d$, in practice, we need to map the data with arbitrary distribution to this domain. Suppose $\mathcal{X}_n = \{x_i\}_{i=1}^n$ are independent and identically distributed observations generated from the cumulative distribution function F with bounded support $\mathcal{D} \subset \mathbb{R}^d$. Suppose τ is a transformation, such that $\{\tau(x_i)\}_{i=1}^n$ has the uniform distribution on $[0, 1]^d$. In a simple case where $d = 1$ and F is known, we can find the transformation τ by setting $\tau = F$. In a more general case where $d > 1$ and F is unknown, the transformation τ can be calculated using the optimal transport theory (Villani, 2008). However, finding the exact solution using the optimal transport theory can be time-consuming. Instead, one may approximate the transformation τ using the iterative transformation approach (Pukelsheim, 2006) or the sliced optimal transport map approach (Rabin et al., 2011). The computational cost of these two approaches is of the order $O\{Kn \log(n)\}$, where K denotes the number of iterations (Bonneel et al., 2015; Cuturi & Doucet, 2014; Kolouri et al., 2018). Such a computational cost is negligible compared with the computational cost of the proposed method. In practice, the data can always be preprocessed using the transformation of τ. Without loss of generality, $\mathcal{X}_n$ may be assumed to be independent and identically distributed observations generated from the uniform distribution on $[0, 1]^d$.

3.3.2 Star discrepancy and space-filling design

Let $a = (a_1, \ldots, a_d)^T \in [0,1]^d$, $[0, a) = \prod_{j=1}^{d}[0, a_j)$ be a hyper-rectangle and $1\{\cdot\}$ be the indicator function. The local discrepancy and the star discrepancy are defined as follows (Fang et al., 2005; Pukelsheim, 2006).

Definition 3.3.1. Given $\mathcal{X}_q = \{x_1, \ldots, x_q\}$ in $[0,1]^d$ and a hyper-rectangle $[0, a)$, the corresponding local discrepancy is defined as

$$D(\mathcal{X}_q, a) = |\frac{1}{q}\sum_{i=1}^{q} 1\{x_i \in [0, a)\} - \prod_{j=1}^{d} a_j|.$$

The star discrepancy corresponding to $\mathcal{X}_q$ is defined as

$$D^*(\mathcal{X}_q) = \sup_{a \in [0,1]^d} D(\mathcal{X}_q, a).$$

The local discrepancy $D(\mathcal{X}_q, a)$ measures the difference between the volume of the hyper-rectangle $[0, a)$ and the fraction of the data points located in $[0, a)$. The local discrepancy is illustrated in the left panel of Fig. 3.1. The star discrepancy $D^*(\mathcal{X}_q)$ calculates the supreme of all the local discrepancy over $a \in [0,1]^d$. In other words, the smaller the $D^*(\mathcal{X}_q)$ is, the more space-filling the data points $\mathcal{X}_q$ are (Fang et al., 2005).

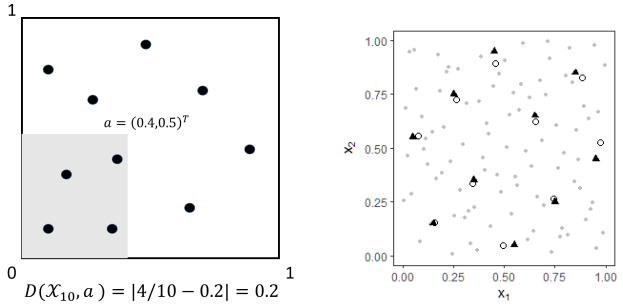

Figure 3.1: Left panel: A toy example for local discrepancy. Right panel: An illustration for the proposed basis selection method.

Chung, 1949 showed that when $\mathcal{X}_q$ is generated from the uniform distribution in $[0,1]^d$, the convergence rate for $D^*(\mathcal{X}_q)$ is $O[\{\log\log(q)/q\}^{1/2}]$. Faster convergence rate can be achieved using space-filling design methods and the low-discrepancy sequence method (Halton, 1960; Owen, 2003; Sobol, 1967). The space-filling design methods, developed in the experimental design literature, aim to generate a set of design points that can cover the space as uniformly

as possible. For further details, please refer to Fang et al., 2005; Pukelsheim, 2006; Wu and Hamada, 2011. The low-discrepancy sequence method Such a method is frequently applied in the field of quasi-Monte Carlo and is extensively employed for numerical integration. For a Sobol sequence $\mathcal{S}_q$, one type of low-discrepancy sequence, it is known that $D^*(\mathcal{S}_q)$ is of the order $O\{\log(q)^d/q\}$, which is roughly the square order $D^*(\mathcal{X}_q)$ for fixed d. For more in-depth discussions on the quasi-Monte Carlo methods, see e.g., Dick et al., 2013; Lemieux, 2009; Leobacher and Pillichshammer, 2014 or Chapter 5 in Glasserman, 2013 and references therein.

Existing studies suggested that space-filling property can be exploited to achieve a fast convergence rate for numerical integration and response surface estimation (Fang et al., 2005; Pukelsheim, 2006). These results inspire us to select the space-filling basis functions in smoothing splines. Unfortunately, the existing techniques of space-filling design cannot be applied to our basis selection problem directly due to the following fact. The design space in space-filling design methods is usually continuous, whereas our sample space $\{x_i\}_{i=1}^n$ is finite and discrete. We propose an algorithm to overcome the barrier.

3.3.3 Main algorithm

We shall develop a space-filling basis selection method, in which we select the space-filling data points in a computationally-attractive manner. First, a set of design points $\mathcal{S}_q = \{s_i\}_{i=1}^q \in [0,1]^d$ are generated, either using low-discrepancy sequence or space-filling design methods. Subsequently, the nearest neighbor x_i^* is selected for each s_i, from the sample points $\{x_i\}_{i=1}^n$. Thus, $\{x_i^*\}_{i=1}^q$ can approximate the design points $\mathcal{S}_q$ well, if x_i^* is sufficiently close to s_i, for $i = 1, \ldots q$. The proposed method is summarized as in Algorithm 5.

Algorithm 5 Space-filling basis selection algorithm

Input: A sample $\{\boldsymbol{x}_i\}_{i=1}^n$.
Step 1: Generate a set of design points $\{s_i\}_{i=1}^q$ from $[0,1]^d$.
Step 2: Select the nearest neighbor for each design point s_i from $\{x_i\}_{i=1}^n$. Let the selected data points be $\{x_i^*\}_{i=1}^q$.
Step 3: Minimize the penalized least squares criterion (3.1) over the following effective model space $\mathcal{H}_E = \mathcal{N}_J \oplus \text{span}\{R_J(x_i^*, \cdot), i = 1, \ldots, q\}$.

The proposed algorithm is illustrated through a toy example in the right panel of Fig. 3.1. One hundred data points (gray dots) are generated from the uniform distribution in $[0,1]^2$, and a set of design points (black triangles) are generated through the max projection design (Joseph et al., 2015), a recently

developed space-filling design method. The nearest neighbor to each design point is selected (circle). It is observed that the selected subsample is space-filling since it can effectively approximate the design points.

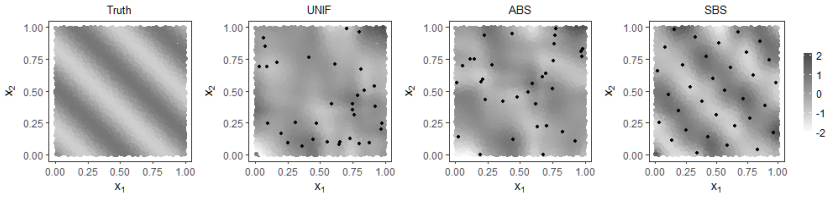

Figure 3.2: Comparison of different basis selection methods.

In Fig. 3.2, the proposed space-filling basis selection method is compared with the uniform basis selection method (C. Gu & Kim, 2002) and the adaptive basis selection method (Ma, Huang, et al., 2015) using a two-dimensional toy example. We generate 5000 data points from the uniform distribution in $[0, 1]^2$. The leftmost panel in Fig. 3.2 presents the heat map for the true response surface $y = \sin\{5(x_1 + x_2)\}$. The dimension of the effective model space q is set to be $5 \times (5000)^{2/9} \approx 33$, for all basis selection methods. The selected basis functions are labeled as solid dots in each panel. The right three panels of Fig. 3.2 plot the heat maps of the spline estimates of all three basis selection methods. In the uniform basis selection method, the default random number generator in R is employed to select the basis functions. It is observed that the selected points are not uniformly distributed. This is a very common phenomenon for uniform basis selection since the randomly selected points do not necessarily look uniformly-distributed, especially when the number of selected points is small. In contrast, it is observed that the basis functions selected by the proposed method are space-filling. Using the space-filling design techniques, the proposed method overcomes the pitfall of uniform basis selection method and uniformly-distribute the selected points. The true response can be better estimated using the proposed method than using other methods.

Now we calculate the computational cost of the proposed method. In Step 1, the design points can be generated beforehand; thus, the computational cost in Step 1 can be ignored. For the nearest neighbor search in Step 2, we employ the k-d tree method, which takes $O\{n \log(n)\}$ flops (Bentley, 1975; Wald & Havran, 2006). The computational cost of this step can be further reduced if we are willing to sacrifice the accuracy of the searching results, e.g., using those approximate nearest neighbor searching algorithms (Altman, 1992; Arya et al., 1994). For Step 3, computing the smoothing spline estimates in the restricted space $\mathcal{H}_E$ is of the order $O(nq^2)$, as discussed in Section 2.2. In summary, the

overall computational cost for the space-filling basis selection method is of the order $O(nq^2)$.

3.4 Convergence rates for function estimation

Recall that the data points are assumed to be generated from the uniform distribution on $[0,1]^d$. Thus, for each coordinate x, the corresponding marginal density $f_X(\cdot) = 1$. We define that $V(g) = \int_{[0,1]^d} g^2 \mathrm{d}x$. The following four regularity conditions are required for the asymptotic analysis, and the first three are the identical conditions employed by Ma, Huang, et al., 2015, in which one can find more technical discussions.

Condition 1. The function V is completely continuous with respect to J;

Condition 2. for some $\beta > 0$ and $r > 1$, $\rho_\nu > \beta\nu^r$ for sufficiently large ν;

Condition 3. for all μ and ν, $\mathrm{var}\{\phi_\nu(x)\phi_\mu(x)\} \leq C_1$, where ϕ_ν, ϕ_μ are the eigenfunctions associated with V and J in $\mathcal{H}$, C_1 denotes a positive constant;

Condition 4. for all μ and ν, $\mathcal{V}(g_{\nu,\mu}) \leq C_2$, where $\mathcal{V}(\cdot)$ denotes the total variation, $g_{\nu,\mu}(x) = \phi_\nu(x)\phi_\mu(x)$, and C_2 represents a positive constant. The total variation here is defined in the sense of Hardy and Krause (Owen, 2003). As a specific case when $d = 1$, the total variation $\mathcal{V}(g) = \int |g'(x)| \mathrm{d}x$, revealing that a smooth function displays a small total variation. Intuitively, the total variation measures how wiggly the function g is.

Condition 1 indicates that there exist a sequence of eigenfunctions $\phi_\nu \in \mathcal{H}$ and the associated sequence of eigenvalues $\rho_\nu \uparrow \infty$ satisfying $V(\phi_\nu, \phi_\mu) = \delta_{\nu\mu}$ and $J(\phi_\nu, \phi_\mu) = \rho_\nu \delta_{\nu\mu}$, where $\delta_{\nu\mu}$ is the Kronecker delta. The growth rate of the eigenvalues ρ_ν dictates how fast λ should approach to 0, and further what the convergence rate of smoothing spline estimates is (C. Gu, 2013). Notice that the eigenfunctions ϕ_νs have a close relationship with the Demmler-Reinsch basis, which are orthogonal vectors representing l_2 norm (Ruppert, 2002). The eigenfunctions ϕ_νs can be calculated explicitly in several specific scenarios. For instance, ϕ_νs are the sine and cosine functions when $J(\eta) = \int_0^1 (\eta'')^2 \mathrm{d}x$, where η denotes a periodic function on $[0,1]$. For more details on the construction of ϕ_νs can be found in Section 9.1 of C. Gu, 2013.

We now present our main theoretical results, and all the proofs are relegated to the Appendix. For a set of design points $\mathcal{S}_q$ of size q, we now assume the star discrepancy $D^*(\mathcal{S}_q)$ converges to zero at the rate of $O\{\log(q)^d/q\}$, or $O\{q^{-(1-\delta)}\}$ for an arbitrary small positive number δ. Such a convergence rate is warranted if $\mathcal{S}_q$ is generated from a low-discrepancy sequence or space-filling design methods, as discussed in Section 3.1. Recall that the proposed method aims to select a subsample that is space-filling, and the key to success depends

on whether the chosen subsample $\mathcal{X}_q^*$ can effectively approximate the design points $\mathcal{S}_q$. The following lemma bounds the difference between $\mathcal{X}_q^*$ and $\mathcal{S}_q$ in terms of star discrepancy.

Lemma 3.4.1. *Suppose d is fixed and $D^*(\mathcal{S}_q) = O\{q^{-(1-\delta)}\}$, for any arbitrary small $\delta > 0$. If $q = O(n^{1/d})$, as $n \to \infty$, we have $D^*(\mathcal{X}_q^*) = O_p\{q^{-(1-\delta)}\}$.*

Lemma 3.4.1 states that the selected subsample $\mathcal{X}_q^*$ can effectively approximate the design points $\mathcal{S}_q$ in the sense that the convergence rate of $D^*(\mathcal{X}_q^*)$ is similar to that of $D^*(\mathcal{S}_q)$. The following theorem is the Koksma–Hlawka inequality, which will be used in proving our main theorem. See Kuipers and Niederreiter, 2012 for a proof.

Theorem 3.4.1 (Koksma–Hlawka inequality)**.** *Let $\mathcal{T}_q = \{t_1, \ldots, t_q\}$ be a set of data points in $[0,1]^d$, and h be a function defined on $[0,1]^d$ with bounded total variation $\mathcal{V}(h)$. We have $|\int_{[0,1]^d} h(x)dx - \sum_{i=1}^q h(t_i)/q| \leq D^*(\mathcal{T}_q)\mathcal{V}(h)$.*

Combining Lemma 3.4.1 and Theorem 3.4.1 and set $h = g_{\nu,\mu}$, $\mathcal{T}_q = \mathcal{X}_q^*$ yields the following lemma.

Lemma 3.4.2. *If $q = O(n^{1/d})$, under Condition 4, for all μ and ν, we have*

$$\left| \int_{[0,1]^d} \phi_\nu \phi_\mu dx - \frac{1}{q} \sum_{j=1}^{q} \phi_\nu(x_j^*)\phi_\mu(x_j^*) \right| = O_p\{q^{-(1-\delta)}\}.$$

Lemma 3.4.2 shows the advantage of $\{x_i^*\}_{i=1}^q$, the subsample selected by the proposed method, over a randomly selected subsample $\{x_i^+\}_{i=1}^q$. To be specific, as a direct consequence of Condition 3, we have $E[\int_{[0,1]^d} \phi_\nu \phi_\mu \mathrm{d}x - \sum_{j=1}^q \phi_\nu(x_j^+)\phi_\mu(x_j^+)/q]^2 = O(q^{-1})$, for all μ and ν. Lemma 2 therefore implies the subsample $\mathcal{X}_q^*$ can more efficiently approximate the integration $\int_{[0,1]^d} \phi_\nu \phi_\mu \mathrm{d}x$, for all μ and ν. We now present our main theoretical result.

Theorem 3.4.2. *Suppose $\sum_i \rho_i^p V(\eta_0, \phi_i)^2 < \infty$ for some $p \in [1,2]$ and δ is an arbitrary small positive number. Under Conditions 1-4, $q = O(n^{1/d})$, as $\lambda \to 0$ and $q^{1-\delta}\lambda^{1/r} \to \infty$, we have $(V + \lambda J)(\hat{\eta}_E - \eta_0) = O_p(n^{-1}\lambda^{-1/r} + \lambda^p)$. In particular, if $\lambda \asymp n^{-r/(pr+1)}$, the estimator achieves the optimal convergence rate $(V + \lambda J)(\hat{\eta}_E - \eta_0) = O_p\{n^{-pr/(pr+1)}\}$.*

It is shown in Theorem 9.17 of C. Gu, 2013 that the full sample smoothing spline estimator $\hat{\eta}$ has the convergence rate,

$$(V + \lambda J)(\hat{\eta} - \eta_0) = O_p(n^{-pr/(pr+1)})$$

under some regularity conditions. Theorem 3.4.2 thus states that proposed estimator $\hat{\eta}_E$ achieves the same convergence rate as the full sample estimator $\hat{\eta}$, under two extra conditions imposed on q (a) $q = O(n^{1/d})$, and (b) $q^{1-\delta}\lambda^{1/r} \to \infty$ as $\lambda \to 0$. Moreover, Theorem 3.4.2 indicates that to achieve the identical convergence rate as the full sample estimator $\hat{\eta}$, the proposed approach requires a much smaller number of basis functions, in the case when $\lambda \asymp n^{-r/(pr+1)}$. $q^{1-\delta}\lambda^{1/r} \to \infty$ indicates an essential choice of q for the proposed estimator should satisfy $q = O\{n^{(1+\delta)/(pr+1)}\}$, when $\lambda \asymp n^{-r/(pr+1)}$. For comparison, both the random basis selection (C. Gu & Kim, 2002) and the adaptive basis selection method (Ma, Huang, et al., 2015) require the essential number of basis functions to be $q = O\{n^{2/(pr+1)+\delta}\}$. As a result, the proposed estimator is more efficient since it reduces the order of the essential number of basis functions.

Given $q = O(n^{1/d})$, when $d \leq pr + 1$, it follows when $q^{1-\delta}\lambda^{1/r} \to \infty$ is satisfied. Otherwise, when $d > pr + 1$, $q = O(n^{1/d})$ becomes sufficient but not necessary for $q^{1-\delta}\lambda^{1/r} \to \infty$. We thus stress that the essential number of basis functions for the proposed method, $q = O\{n^{(1+\delta)/(pr+1)}\}$, can only be achieved when $d \leq pr + 1$. The parameter p in Theorem 2 is closely associated with the true function η_0 and will affect the convergence rate of the proposed estimator. Intuitively, the larger the p is, the smoother the function η_0 will be. For $p \in [1, 2]$, the optimal convergence rate of $(V + \lambda J)(\hat{\eta}_E - \eta_0)$ falls in the interval $[O_p(n^{-r/(r+1)}), O_p(n^{-2r/(2r+1)})]$. To the best of our knowledge, the problem of selecting the optimal p has rarely been studied, and one exception is Serra and Krivobokova, 2017, where the author studied such a problem in one-dimensional cases. Serra and Krivobokova, 2017 provided a Bayesian approach for selecting the optimal parameter, named β, which is known to be proportional to p. Nevertheless, since the constant β/p is usually unknown, such an approach still cannot be used to select the optimal p in practice. Furthermore, whether such an approach can be extended to the high-dimensional cases remains unclear.

For the dimension of the effective model space q, a suitable choice is $q = n^{(1+\delta)/(4p+1)+\delta}$ in the following two cases. Case 1. Univariate cubic smoothing splines with the penalty $J(\eta) = \int_0^1 (\eta'')^2$, $r = 4$ and $\lambda \asymp n^{-4/(4p+1)}$; Case 2. Tensor-product splines with $r = 4 - \delta^*$, where $\delta^* > 0$. For $p \in [1, 2]$, the dimension roughly lies in the interval $(O(n^{1/9}), O(n^{1/5}))$.

3.5 Simulation Results

To assess the performance of the proposed space-filling basis selection method, we carry out extensive analysis on simulated datasets. We report some of them in this section. We compare the proposed method with uniform basis selection and adaptive basis selection, and report both prediction error and running time.

The following four functions on $[0,1]$ (Lin & Zhang, 2006) are used as the building blocks in our simulation study, $g_1(t) = t$, $g_2(t) = (2t-1)^2$, $g_3(t) = \sin(2\pi t)/\{2-\sin(2\pi t)\}$, and $g_4(t) = 0.1\sin(2\pi t)+0.2\cos(2\pi t)+0.3\sin(2\pi t)^2+0.4\cos(2\pi t)^3+0.5\sin(2\pi t)^3$. In addition, we also use the following functions on $[0,1]^2$ (Wood, 2003) as the building blocks,

$$h_1(t_1,t_2) = \{0.75/(\pi\sigma_1\sigma_2)\} \times \exp\{-(t_1-0.2)^2/\sigma_1^2 - (t_2-0.3)^2/\sigma_2^2\},$$
$$h_2(t_1,t_2) = \{0.45/(\pi\sigma_1\sigma_2)\} \times \exp\{-(t_1-0.7)^2/\sigma_1^2 - (t_2-0.8)^2/\sigma_2^2\},$$

where $\sigma_1 = 0.3$ and $\sigma_2 = 0.4$. The signal-to-noise ratio (SNR), defined as $\text{var}\{\eta(X)\}/\sigma^2$, is set to be at two levels: 5 and 2. We generate replicated samples with sample sizes $n = \{2^{10}, 2^{11}, \ldots, 2^{14}\}$ and dimensions $d = \{2,4,6\}$ uniformly on $[0,1]^p$ from the following four regression settings,

(1) A 2-d function $g_1(x_1x_2)+g_2(x_2)+g_3(x_1)+g_4(x_2)+g_3\{(x_1+x_2)/2\}$;

(2) A 2-d function $h_1(x_1,x_2)+h_2(x_1,x_2)$;

(3) A 4-d function $g_1(x_1)+g_2(x_2)+g_3(x_3)+2g_1\{(x_1+x_4)/2\}+2g_2\{(x_2+x_3)/2\}+2g_3\{(x_1+x_3)/2\}$;

(4) A 6-d function $h(x_1,x_2)+h(x_1,x_5)$.

In the simulation, q is set to be $5n^{2/9}$ and $10n^{1/9}$, based on the asymptotic results. To combat the curse of dimensionality, we fit smoothing spline analysis of variance models with all main effects and two-way interactions. The prediction error is measured by the mean squared error (MSE), defined as $[\sum_{i=1}^{n_0}\{\hat{\eta}_E(t_i)-\eta_0(t_i)\}^2]/n_0$, where $\{t_i\}_{i=1}^{n_0}$ denotes an independent testing dataset uniformly generated on $[0,1]^p$ with $n_0 = 5000$. The max projection design (Joseph et al., 2015) is used to generate design points in Step 1 of the proposed method. Our empirical studies suggest that the Sobol sequence and other space-filling techniques, e.g., the Latin hypercube design (Pukelsheim, 2006) and the uniform design (Fang et al., 2000), also yield similar performance.

Figure 3.3 shows the MSE against the sample size on the log-log scale. Each column presents the results of a function setting as described above. We set the signal-to-noise ratio to be five and two in the upper row and the lower row, respectively. We use solid lines for the proposed method, dotted lines for adaptive basis selection method, and dashed lines for uniform basis selection method.

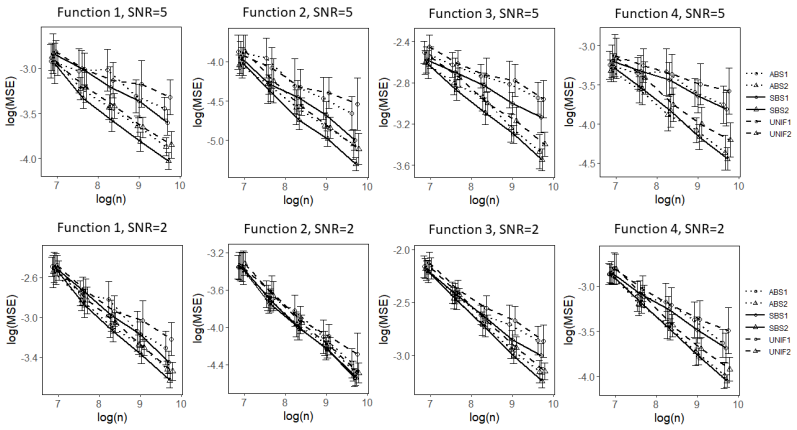

Figure 3.3: Simulation under different settings (from left to right) with SNR being five (the upper row) and two (the lower row).

The number of the basis functions q is $5n^{2/9}$ and $10n^{1/9}$ for the lines with triangles and the lines with circles, respectively. The vertical bars represent standard error bars obtained from 20 replicates. The full sample estimator is omitted here due to the high computation cost. It is observed that the space-filling basis selection method provides more accurate smoothing spline predictions than the other two methods in almost all settings. Notably, the lines with circles for the space-filling basis selection method displays a linear trend similar to the lines with triangles for the other two methods. This observation reveals the proposed estimator yields a faster convergence rate than the other two methods.

More simulation results can be found in the Appendix, in which we consider the regression functions that exhibit several sharp peaks. In those cases, the results suggest that both the space-filling basis selection method and the adaptive basis selection method outperform the uniform basis selection, whereas neither the space-filling basis selection method nor the adaptive basis selection method dominates each other. Moreover, the proposed space-filling basis selection method outperforms the adaptive basis selection method as the sample size n gets larger.

Table 3.1 summarizes the computing times of model-fitting using all methods on a synthetic dataset with $n = 2^{14}$ and $q = 5n^{2/9}$. The simulation is replicated for 20 runs using a computer with an Intel 2.6 GHz processor. In Table 1, UNIF, ABS, and SBS represent the uniform basis selection method, the adaptive basis selection method, and the proposed space-filling basis selection

Table 3.1: Means and standard errors (in parentheses) of the computational time, in CPU seconds, for multivariate cases, based on 20 replicates.

True function	SNR	UNIF	ABS	SBS
Function 1	5	0.97(0.15)	0.90(0.05)	0.90(0.04)
	2	0.92(0.10)	0.87(0.04)	0.87(0.06)
Function 2	5	0.88(0.04)	0.87(0.03)	0.90(0.06)
	2	0.86(0.05)	0.85(0.02)	0.90(0.06)
Function 3	5	3.92(0.24)	3.95(0.24)	4.04(0.19)
	2	4.08(0.30)	4.51(0.66)	4.27(0.39)
Function 4	5	12.95(0.61)	15.10(3.20)	15.45(3.04)
	2	14.33(1.44)	13.72(1.02)	14.25(1.09)

method, respectively. The time for calculating the smoothing parameter is not included. The result for the full basis smoothing spline estimator is omitted here due to the huge computational cost. The computational time for generating a set of design points, i.e., Step 1 in the proposed algorithm, is not included since the design points can be generated beforehand. It is observed that the computing time of the proposed method is comparable with that of the other two basis selection methods under all settings. Combining such an observation with the result in Fig. 3.3, it is concluded that the proposed method can achieve a more accurate prediction, without requiring much more computational time.

3.6 Real data example

The problem of measuring total column ozone has attracted significant attention for decades. Ozone depletion facilitates the transmission of ultraviolet radiation (290–400 nm wavelength) through the atmosphere and causes severe damage to DNA and cellular proteins that are involved in biochemical processes, affecting growth and reproduction. Statistical analysis of total column ozone data has three steps. In the first step, the raw satellite data (level 1) are retrieved by NASA. Subsequently, NASA calibrates and preprocesses the data to generate spatially and temporally irregular total column ozone measurements (level 2). Finally, the level 2 data are processed to yield the level 3 data, which are the daily and spatially regular data product released extensively to the public.

We fit the nonparametric model $y_{ij} = \eta(x_{\langle 1\rangle i}, x_{\langle 2\rangle j}) + \epsilon_{ij}$ to a level 2 total column ozone dataset (n=173,405) compiled by Cressie and Johannesson, 2008. Here, y_{ij} is the level 2 total column ozone measurement at the ith longitude,

i.e., $x_{\langle 1\rangle i}$, and the jth latitude, i.e., $x_{\langle 2\rangle i}$, and ϵ_{ij} represent the independent and identically distributed random errors. The heat map of the raw data is presented in the Appendix. The thin-plate smoothing spline is used for the model-fitting, and the proposed method is employed to facilitate the estimation. The number of basis functions is set to $q = 20n^{2/9} \approx 292$. The design points employed in the proposed basis selection method are yielded from a Sobol sequence. The heat map of the predicted image on a $1^\circ \times 1^\circ$ regular grid is presented in Fig. 3.4. It is seen that the total column ozone value decreases dramatically to form the ozone hole over the South Pole, around -55° latitudinal zone.

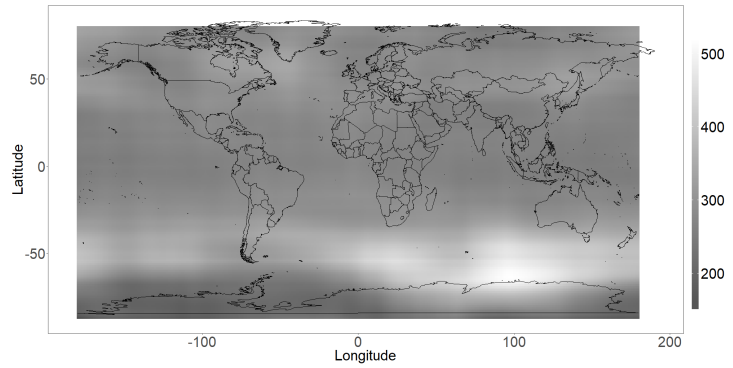

Figure 3.4: Smoothing spline prediction of total column ozone value for 10/01/1988, in Dobson units

We now report computing times of the model-fitting that are performed on the identical laptop computer for the simulation studies. The computational times, in CPU seconds, are presented in parentheses, including basis selection (0.1s), model fitting (129s), and prediction (21s). Further comparison between the proposed method and other basis selection methods on this dataset can be found in the Appendix.

Chapter 4

Large-scale Optimal Transport Map Estimation using Projection Pursuit

This chapter studies the large-scale optimal transport maps (OTM), which is a well known challenging problem owing to the curse of dimensionality. Existing literature approximates the large-scale OTM by a series of one-dimensional OTM problems through iterative random projection. Such methods, however, suffer from slow or none convergence in practice due to the nature of randomly selected projection directions. Instead, we propose an estimation method of large-scale OTM by combining the idea of projection pursuit regression and sufficient dimension reduction. The proposed method, named projection pursuit Monge map (PPMM), adaptively selects the most "informative" projection direction in each iteration. We theoretically show the proposed dimension reduction method can consistently estimate the most "informative" projection direction in each iteration. Furthermore, the PPMM algorithm weakly convergences to the target large-scale OTM in a reasonable number of steps. Empirically, PPMM is computationally easy and converges fast. We assess its finite sample performance through the applications of Wasserstein distance estimation and generative models.

4.1 Introduction

Recently, optimal transport map (OTM) draws great attention in machine learning, statistics, and computer science due to its close relationship to gen-

erative models, including generative adversarial nets (Goodfellow et al., 2014), the "decoder" network in variational autoencoders (Kingma & Welling, 2013), among others. In a generative model, the goal is usually to generate a "fake" sample, which is indistinguishable from the genuine one. This is equivalent to find a transport map ϕ from random noises with distribution p_X (e.g., Gaussian distribution or uniform distribution) to the underlying population distribution p_Y of the genuine sample, e.g., the MNIST or the ImageNet dataset. Nowadays, generative models have been widely-used for generating realistic images (Dosovitskiy & Brox, 2016; Liu et al., 2017), songs (Blaauw & Bonada, 2016; Engel et al., 2017) and videos (Liang et al., 2017; Vondrick et al., 2016). Besides generative models, OTM also plays essential roles in various machine learning applications, say color transfer (Ferradans et al., 2014; Rabin et al., 2014), shape match (Z. Su et al., 2015), transfer learning (Courty et al., 2017; Peyré, Cuturi, et al., 2019) and natural language processing (Peyré, Cuturi, et al., 2019).

Despite its impressive performance, the computation of OTM is challenging for a large-scale sample with massive sample size and/or high dimensionality. Traditional methods for estimating the OTM includes finding a parametric map and using ordinary differential equations (Benamou et al., 2002; Brenier, 1997). To address the computational concern, recent developments of OTM estimation have been made based on solving linear programs (Pele & Werman, 2009; Rubner et al., 1997). Let $\{\boldsymbol{x}_i\}_{i=1}^n \in \mathbb{R}^d$ and $\{\boldsymbol{y}_i\}_{i=1}^n \in \mathbb{R}^d$ be two samples from two continuous probability distributions functions p_X and p_Y, respectively. Estimating the OTM from p_X to p_Y by solving a linear program requiring $O(n^3 \log(n))$ computational time for fixed d (Peyré, Cuturi, et al., 2019; Seguy et al., 2017). To alleviate the computational burden, some literature (Arjovsky et al., 2017; Cuturi, 2013; Genevay et al., 2016; Gulrajani et al., 2017) pursued fast computation approaches of the OTM objective, i.e., the Wasserstein distance. Another school of methods aims to estimate the OTM efficiently when d is small, including multi-scale approaches (Gerber & Maggioni, 2017; Mérigot, 2011) and dynamic formulations (Papadakis et al., 2014; Solomon et al., 2014). These methods utilize the space discretization, thus are generally not applicable in high-dimensional cases.

The random projection method (or known as the radon transformation method) is proposed to estimate OTMs efficiently when d is large Pitie et al., 2005; Pitié et al., 2007. Such a method tackles the problem of estimating a d-dimensional OTM iteratively by breaking down the problem into a series of subproblems, each of which finds a one-dimensional OTM using projected samples. Denote $\mathbb{S}^{d-1}$ as the d-dimensional unit sphere. In each iteration, a random direction $\boldsymbol{\theta} \in \mathbb{S}^{d-1}$ is picked, and the one-dimensional OTM is then calculated

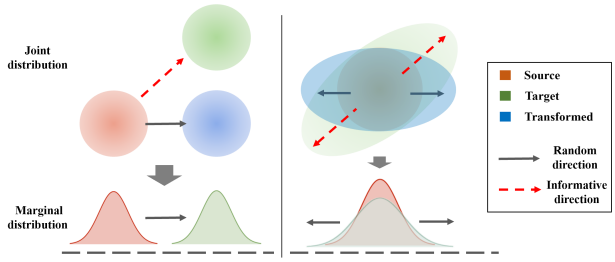

Figure 4.1: Illustration for the "informative" projection direction

between the projected samples $\{\boldsymbol{x}_i^\top \boldsymbol{\theta}\}_{i=1}^n$ and $\{\boldsymbol{y}_i^\top \boldsymbol{\theta}\}_{i=1}^n$. The collection of all the one-dimensional maps serves as the final estimate of the target OTM. The sliced method modifies the random projection method by considering a large set of random directions from $\mathbb{S}^{d-1}$ in each iteration (Bonneel et al., 2015; Rabin et al., 2011). The "mean map" of the one-dimensional OTMs over these random directions is considered as a component of the final estimate of the target OTM. We call the random projection method, the sliced method, and their variants as the *projection-based approach*. Such an approach reduces the computational cost of calculating an OTM from $O(n^3 \log(n))$ to $O(Kn \log(n))$, where K is the number of iterations until convergence. However, there is no theoretical guideline on the order of K. In addition, the existing projection-based approaches usually require a large number of iterations to convergence or even fail to converge. We speculate that the slow convergence is because a randomly selected projection direction may not be "informative", leading to a one-dimensional OTM that failed to be a decent representation of the target OTM. We illustrate such a phenomenon through an illustrative example as follows.

An illustrative example. The left and right panels in Figure 1 illustrates the importance of choosing the "informative" projection direction in OTM estimation. The goal is to obtain the OTM ϕ^* which maps a source distribution p_X (colored in red) to a target distribution p_Y (colored in green). For each panel, we first randomly pick a projection direction (black arrow) and obtain the marginal distributions of p_X and p_Y (the bell-shaped curves), respectively. The one-dimensional OTM then can be calculated based on the marginal distributions. Applying such a map to the source distribution yields the transformed distribution (colored in blue). One can observe that the transformed distributions are significantly different from the target ones. Such an observation indicates that the one-dimensional OTM with respect to a random projection

direction may fail to well-represent the target OTM. This observation motivates us to select the "informative" projection direction (red arrow), which yields a better one-dimensional OTM.

Our contributions. To address the issues mentioned above, this chapter introduces a novel statistical approach to estimate large-scale OTMs. The proposed method, named projection pursuit Monge map (PPMM), improves the existing projection-based approaches from two aspects. First, PPMM uses a sufficient dimension reduction technique to estimate the most "informative" projection direction in each iteration. Second, PPMM is based on projection pursuit (Friedman & Stuetzle, 1981). The idea is similar to boosting that search for the next optimal direction based on the residual of previous ones. Theoretically, we show the proposed method can consistently estimate the most "informative" projection direction in each iteration, and the algorithm weakly convergences to the target large-scale OTM in a reasonable number of steps. The finite sample performance of the proposed algorithm is evaluated by two applications: Wasserstein distance estimation and generative model. We show the proposed method outperforms several state-of-the-art large-scale OTM estimation methods through extensive experiments on various synthetic and real-world datasets.

4.2 Problem setup and methodology

Optimal transport map and Wasserstein distance. Denote $X \in \mathbb{R}^d$ and $Y \in \mathbb{R}^d$ as two continuous random variables with probability distribution functions p_X and p_Y, respectively. The problem of finding a transport map $\phi : \mathbb{R}^d \to \mathbb{R}^d$ such that $\phi(X)$ and Y have the same distribution, has been widely-studied in mathematics, probability, and economics, see (Ferradans et al., 2014; Reich, 2013; Z. Su et al., 2015) for examples of some new developments. Note that the transport map between the two distributions is not unique. Among all transport maps, it may be of interest to define the "optimal" one according to some criteria. A standard approach, named Monge formulation (Villani, 2008), is to find the OTM[4] ϕ^* that satisfies

$$\phi^* = \inf_{\phi \in \Phi} \int_{\mathbb{R}^d} \|X - \phi(X)\|^p \mathrm{d}p_X,$$

[4] Such a map is thus also called the Monge map.

where Φ is the set of all transport maps, $\|\cdot\|$ is the vector norm and p is a positive integer. Given the existence of the Monge map, the Wasserstein distance of

order p is defined as

$$W_p(p_X, p_Y) = \left(\int_{\mathbb{R}^d} \|X - \phi^*(X)\|^p \mathrm{d}p_X \right)^{1/p}.$$

Denote $\widehat{\phi}$ as an estimator of ϕ^*. Suppose one observe $\mathcal{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^{\mathrm{T}} \in \mathbb{R}^{n \times d}$ and $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)^{\mathrm{T}} \in \mathbb{R}^{n \times d}$ from p_X and p_Y, respectively. The Wasserstein distance $W_p(p_X, p_Y)$ thus can be estimated by

$$\widehat{W}_p(\mathcal{X}, \boldsymbol{y}) = \left(\frac{1}{n} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \widehat{\phi}(\boldsymbol{x}_i)\|^p \right)^{1/p}.$$

Projection pursuit method. Projection pursuit regression (Huber, 1985; Ifarraguerri & Chang, 2000) is widely-used for high-dimensional nonparametric regression models which takes the form.

$$z_i = \sum_{j=1}^{s} f_j(\boldsymbol{\beta}_j^{\mathrm{T}} \boldsymbol{x}_i) + \epsilon_i, \quad i = 1, \ldots, n,$$

where s is a hyper-parameter, $\{z_i\}_{i=1}^n$ is the univariate response, $\{\boldsymbol{x}_i\}_{i=1}^n$ are d-dimensional covariates, and $\{\epsilon_i\}_{i=1}^n$ are i.i.d. normal errors. The goal is to estimate the unknown link functions $\{f_j\}_{j=1}^s : \mathbb{R} \to \mathbb{R}$ and the unknown coefficients $\{\boldsymbol{\beta}_j\}_{j=1}^s \in \mathbb{R}^d$.

The additive model can be fitted in an iterative fashion. In the kth iteration, $k = 2, \ldots, s$, denote $\{(\widehat{f}_j, \widehat{\boldsymbol{\beta}}_j)\}_{j=1}^{k-1}$ the estimate of $\{(f_j, \boldsymbol{\beta}_j)\}_{j=1}^{k-1}$ obtained from previous $k-1$ iterations. Denote $R_i^{[k]} = z_i - \sum_{j=1}^{k-1} \widehat{f}_j(\widehat{\boldsymbol{\beta}}_j^{\mathrm{T}} \boldsymbol{x}_i)$, $i = 1, \ldots, n$, the residuals. Then $(f_k, \boldsymbol{\beta}_k)$ can be estimated by solving the following least squares problem

$$\min_{f_k, \boldsymbol{\beta}_k} \sum_{i=1}^{n} \left[R_i^{[k]} - f_k(\boldsymbol{\beta}_k^{\mathrm{T}} \boldsymbol{x}_i) \right]^2.$$

The above iterative process explains the intuition behind the projection pursuit regression. Given the model fitted in previous iterations, we fit a one dimensional regression model using the current residuals, rather than the original responses. We then add this new regression model into the fitted function in order to update the residuals. By adding small regression models to the residuals, we gradually improve fitted model in areas where it does not perform well.

The intuition of projection pursuit regression motivates us to modify the existing projection-based OTM estimation approaches from two aspects. First,

in the kth iteration, we propose to seek a new projection direction for the one-dimensional OTM in the subspace spanned by the residuals of the previously $k-1$ directions. On the contrary, following a direction that is in the span of used ones can lead to an inefficient one dimensional OTM. As a result, this "move" may hardly reduce the Wasserstein distance between p_X and p_Y. Such inefficient "moves" can be one of the causes of the convergence issue in existing projection-based OTM estimation algorithms. Second, in each iteration, we propose to select the most "informative" direction with respect to the current residuals rather than a random one. Specifically, we choose the direction that explains the highest proportion of variations in the subspace spanned by the current residuals. Intuitively, this direction addresses the maximum marginal "discrepancy" between p_X and p_Y among the ones that are not considered by previous iterations. We propose to estimate this most "informative" direction with sufficient dimension reduction techniques introduced as follows.

Sufficient dimension reduction. Consider a regression problem with univariate response Z and a d-dimensional predictor X. Sufficient dimension reduction for regression aims to reduce the dimension of X while preserving its regression relation with Z. In other words, sufficient dimension reduction seeks a set of linear combinations of X, say $\mathbf{B}^{\mathrm{T}}X$ with some $\mathbf{B} \in \mathbb{R}^{d\times q}$ and $q \leq d$, such that Z depends on X only through $\mathbf{B}^{\mathrm{T}}X$, i.e., $Z \perp\!\!\!\perp X|\mathbf{B}^{\mathrm{T}}X$. Then, the column space of $\mathbf{B}$, denoted as $\mathcal{S}(\mathbf{B})$ is called a dimension reduction space (DRS). Furthermore, if the union of all possible DRSs is also a DRS, we call it the central subspace and denote it as $\mathcal{S}_{Z|X}$. When $\mathcal{S}_{Z|X}$ exists, it is the minimum DRS. We call a sufficient dimension reduction method exclusive if it induces a DRS that equals to the central subspace. Some popular sufficient dimension reduction techniques include sliced inverse regression (SIR) (K.-C. Li, 1991), principal Hessian directions (PHD) (K.-C. Li, 1992), sliced average variance estimator (SAVE) (Cook & Weisberg, 1991), directional regression (DR) (B. Li & Wang, 2007), among others.

Estimation of the most "informative" projection direction. Consider estimating an OTM between a source sample and a target sample. We first form a regression problem by adding a binary response, which equals zero for the source sample and one for the target sample. We then utilize the sufficient dimension reduction technique to select the most "informative" projection direction. To be specific, we select the projection direction $\boldsymbol{\xi} \in \mathbb{R}^d$ as the eigenvector corresponds to the largest eigenvalue of the estimated $\mathbf{B}$. The direction $\boldsymbol{\xi}$ is most "informative" in the sense that, the projected samples $\mathcal{X}\boldsymbol{\xi}$ and $\boldsymbol{y}\boldsymbol{\xi}$ have the most substantial " discrepancy." The metric of the "discrepancy" depends on the choice of the sufficient dimension reduction technique. Figure 1.3 gives a

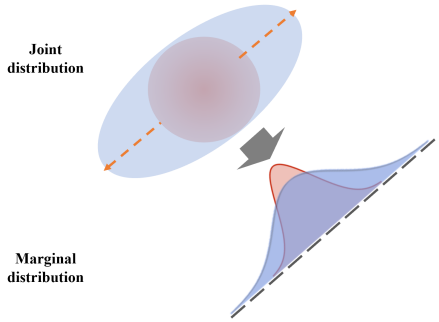

Figure 4.2: The most "informative" projection direction ensures the projected samples (illustrated by the distributions colored in red and blue, respectively) have the largest "discrepancy".

toy example to illustrate this idea. In this chapter, we opt to use SAVE for calculating $\mathbf{B}$, and hence the "discrepancy" metric is the difference between $\mathrm{Var}(\mathcal{X}\boldsymbol{\xi})$ and $\mathrm{Var}(\boldsymbol{y}\boldsymbol{\xi})$. Empirically, we find other sufficient dimension reduction techniques, like PHD and DR, also yield similar performance. The SIR method, however, yields inferior performance, since it only considers the first moment. The Algorithm 6 below introduces our estimation method of "informative" projection direction in detail.

Algorithm 6 Select the most "informative" projection direction using SAVE

Input: two standardized matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ and $\mathbf{Y} \in \mathbb{R}^{n\times d}$
Step 1: calculate $\widehat{\mathcal{S}} \in \mathbb{R}^{d\times d}$, i.e., the sample variance-covariance matrix of $\binom{\mathbf{X}}{\mathbf{Y}}$
Step 2: calculate the sample variance-covariance matrices of $\mathcal{X}\widehat{\mathcal{S}}^{-1/2}$ and $\boldsymbol{y}\widehat{\mathcal{S}}^{-1/2}$, denoted as $\widehat{\mathcal{S}}_1 \in \mathbb{R}^{d\times d}$ and $\widehat{\mathcal{S}}_2 \in \mathbb{R}^{d\times d}$, respectively
Step 3: calculate the eigenvector $\boldsymbol{\xi} \in \mathbb{R}^d$, which corresponding to the largest eigenvalue of the matrix $((\widehat{\mathcal{S}}_1 - I_d)^2 + (\widehat{\mathcal{S}}_2 - I_d)^2)/4$
Output: the final result is given by $\widehat{\mathcal{S}}^{-1/2}\boldsymbol{\xi}/||\widehat{\mathcal{S}}^{-1/2}\boldsymbol{\xi}||$, where $||\cdot||$ denotes the Euclidean norm

Projection pursuit Monge map algorithm. Now, we are ready to present our estimation method for large-scale OTM. The detailed algorithm, named projection pursuit Monge map, is summarized in Algorithm 7 below. In each iteration, the PPMM applies a one-dimensional OTM following the most "informative" projection direction selected by the Algorithm 1.

Computational cost of PPMM. In Algorithm 7, the computational cost mainly resides in the first two steps within each iteration. In step (a), one calcu-

Algorithm 7 Projection pursuit Monge map (PPMM)

Input: two matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ and $\mathbf{Y} \in \mathbb{R}^{n\times d}$
$k \leftarrow 0$, $\mathbf{X}^{[0]} \leftarrow \mathbf{X}$
repeat
(a) calculate the projection direction $\boldsymbol{\xi}_k \in \mathbb{R}^d$ between $\mathbf{X}^{[k]}$ and $\mathbf{Y}$ (using Algorithm 1)
(b) find the one-dimensional OTM $\phi^{(k)}$ that matches $\mathbf{X}^{[k]}\boldsymbol{\xi}_k$ to $\mathbf{Y}\boldsymbol{\xi}_k$ (using look-up table)
(c) $\mathbf{X}^{[k+1]} \leftarrow \mathbf{X}^{[k]} + (\phi^{(k)}(\mathbf{X}^{[k]}\boldsymbol{\xi}_k) - \mathbf{X}^{[k]}\boldsymbol{\xi}_k)\boldsymbol{\xi}_k^{\mathrm{T}}$ and $k \leftarrow k+1$
until converge
The final estimator is given by $\widehat{\phi} : \mathbf{X} \to \mathbf{X}^{[k]}$

lates $\boldsymbol{\xi}_k$ using Algorithm 1, whose computational cost is of order $O(nd^2)$. In step (b), one calculates a one-dimensional OTM using the look-up table, which is simply a sorting algorithm (Peyré, Cuturi, et al., 2019; Pitié et al., 2007).

The computational cost for step (b) is of order $O(n\log(n))$. Suppose that the algorithm converges after K iterations. The overall cost of Algorithm 2 is of order $O\left(Knd^2 + Kn\log(n)\right)$. Empirically, we find $K = O(d)$ works reasonably well. When $\log(n)^{1/2} \leq d \ll n^{2/3}$, the order of computational cost of PPMM is $o\left(n^3\log(n)\right)$ which is smaller than the computational cost of the naive method for calculating OTMs. When $d \leq \log(n)^{1/2}$, the order of computational cost reduces to $O\left(Kn\log(n)\right)$ which is faster than the exiting projection-based methods given PPMM converges faster. The memory cost for Algorithm 2 mainly resides in the step (a), which is of the order $O(Knd^2)$.

4.3 Theoretical results

Exclusiveness of SAVE. For mathematical simplicity, we assume $E[X] = E[Y] = \mathbf{0}_d$. When $E[X] \neq E[Y]$, one can use a first-order dimension reduction method like SIR to adjust means before applying SAVE.

Denote $W = (X+Y)/2$, $\Sigma_W = \mathrm{Var}(W)$, and $Z = W\Sigma_W^{-1/2}$. For a univariate continuous response variable R, one can approximate the central subspace $\mathcal{S}_{R|Z}$ by $\mathcal{S}_{\mathrm{SAVE}}$, which is the population version of the dimension re-

duction space of SAVE. To be specific, $\mathcal{S}_{\text{SAVE}}$ is the column space of matrix

$$E[\text{Var}(Z|R) - I_d]^2 = \frac{1}{4}\left\{E[\text{Var}(X\Sigma_W^{-1/2}|R) - I_d]^2 + E[\text{Var}(Y\Sigma_W^{-1/2}|R) - I_d]^2\right\},$$

where the above equation used the fact that $X \perp\!\!\!\perp Y$.

Assumption 4.3.1. *Let P be the projection onto the central space $\mathcal{S}_{R|Z}$ with respect to the inner project $a \cdot b = a^{\mathrm{T}}b$. For any nonzero vectors $u, v \in \mathbb{R}^d$, such that u is orthogonal to $\mathcal{S}_{R|Z}$ and $v \in \mathcal{S}_{R|Z}$, we assume*

(a) $E(u^{\mathrm{T}}Z|PZ)$ is a linear function of Z;

(b) $\text{Var}(u^{\mathrm{T}}Z|PZ)$ is a nonrandom number;

(c) Let $(\widetilde{Z}, \widetilde{R})$ be an independent copy of (Z, R). $E\left[v^{\mathrm{T}}(Z - \widetilde{Z})^2|R, \widetilde{R})\right]$ is non degenerate; that is, it is not equal almost surely to a constant.

Theorem 4.3.1. *Let R be a univariate continuous response variable. Under Assumption 1, the dimension reduction space induced by SAVE is exclusive. In other words, $\mathcal{S}_{SAVE} = \mathcal{S}_{R|Z}$.*

Consistency of the most "informative" projection direction. Let $\widehat{\Sigma}_1$ and $\widehat{\Sigma}_2$ be the sample covariance matrix estimator of Σ_1 and Σ_2, respectively. Denote

$$\Sigma_{\text{SAVE}} = \frac{1}{4}\left[(\Sigma_1 - I_d)^2 + (\Sigma_2 - I_d)^2\right]$$
$$\widehat{\Sigma}_{\text{SAVE}} = \frac{1}{4}\left[(\widehat{\Sigma}_1 - I_d)^2 + (\widehat{\Sigma}_2 - I_d)^2\right].$$

Denote $\boldsymbol{\xi}_1$ and $\widehat{\boldsymbol{\xi}}_1$ the eigenvectors correspond to the largest eigenvalues of Σ_{SAVE} and $\widehat{\Sigma}_{\text{SAVE}}$, respectively. Further, denote $r = \text{Rank}(\Sigma_{\text{SAVE}})$, the rank of Σ_{SAVE}.

Assumption 4.3.2. *Let $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^n$ be an i.i.d. sample of (X, Y). We assume that*

(a) Denote x_{ij} and y_{ik} the jth and kth component of $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$, respectively. $E(x_{ij}y_{ik}) = 0$ for all $1 \le i \le n$ and $1 \le j, k \le d$;

(b) There are $r_1, r_2 > 0$ and $b_1, b_2 > 0$ such that, for any $s > 0$, $1 \le i \le n$ and $1 \le j \le d$,

$$P(|x_{ij}| > s) \le \exp\left\{-(s/b_1)^{r_1}\right\},$$
$$P(|y_{ij}| > s) \le \exp\left\{-(s/b_2)^{r_2}\right\};$$

(c) Let $\lambda_1 \ldots, \lambda_d$ be the eigenvalues of Σ_{SAVE} in descending order. There exist positive constants c_l c_u and c_3 such that

$$c_l \leq \min_{1 \leq l \leq r-1} (\lambda_l - \lambda_{l+1}) d^{-1/2} \leq c_u, \quad \textit{and} \quad 0 \leq \lambda_{r+1} < c_3.$$

Theorem 4.3.2 shows that Algorithm 1 can consistently estimate the most "informative" projection direction. The O_p in Theorem 4.3.2 stands for order in probability, which is similar to O but for random variables.

Theorem 4.3.2. *Under Assumption 2, the SAVE estimator of most "informative" projection direction satisfies,*

$$\|\widehat{\boldsymbol{\xi}}_1 - \boldsymbol{\xi}_1\|_\infty = O_p(r^4 \sqrt{\frac{\log d}{n}} + r^4 \sqrt{d} \frac{\log d}{n}), \quad \textit{as} \quad n, d \to \infty.$$

Convergence of PPMM algorithm. Denote ϕ^* as the d-dimensional optimal transport map from p_X to p_Y and $\phi^{(K)}$ as the PPMM estimator after K iterations, i.e. $\phi^{(K)}(\mathcal{X}) = \mathcal{X}^{[K]}$. The following theorem gives the weak convergence results of the PPMM algorithm.

Theorem 4.3.3. *Suppose Assumption 1 and Assumption 2 hold. Let $K \geq Cd$ for some large enough positive constant C, one has*

$$\widehat{W}_p\Big(\phi^{(K)}(\mathcal{X}), \mathcal{X}\Big) \to W_p\Big(\phi^*(X), X\Big),$$
$$\textit{and} \quad \phi^{(K)}(\mathcal{X}) \to \phi^*(X) \quad \textit{as} \quad n \to \infty.$$

Works are proving the convergence rates of the empirical optimal transport objectives (Boissard et al., 2011; Boissard & Le Gouic, 2014; Sriperumbudur et al., 2012; Weed & Bach, 2017). The convergence rate of the OTM has rarely been studied except for a recent chapter (Hütter & Rigollet, 2019). We believe Theorem 4.3.3 is the first step in this direction.

4.4 Numerical experiments

4.4.1 Estimation of optimal transport map

Suppose that we observe i.i.d. samples $\mathcal{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^{\mathrm{T}}$ from $\mathcal{N}_d(\boldsymbol{\mu}_X, \mathcal{S}_X)$ and $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)^{\mathrm{T}}$ from $\mathcal{N}_d(\boldsymbol{\mu}_Y, \mathcal{S}_Y)$, respectively. We set $n = 10,000$, $d = \{10, 20, 50\}$, $\boldsymbol{\mu}_X = -\mathbf{2}, \boldsymbol{\mu}_Y = \mathbf{2}, \mathcal{S}_X = 0.8^{|i-j|}$, and $\mathcal{S}_Y = 0.5^{|i-j|}$, for $i, j = 1, \ldots, d$.

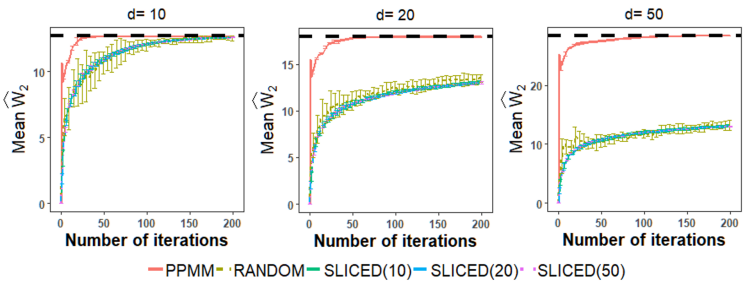

Figure 4.3: The black dashed line is the true value of the Wasserstein distance. The colored lines represent the sample mean of the estimated Wasserstein distances over 100 replications, and the vertical bars represent the standard deviations.

We apply PPMM to estimate the OTM between p_X and p_Y from $\{\boldsymbol{x}_i\}_{i=1}^n$ and $\{\boldsymbol{y}_i\}_{i=1}^n$. In comparison, we also consider the following two projection-based competitors: (1) the random projection method (RANDOM) as proposed in (Pitie et al., 2005; Pitié et al., 2007); (2) the sliced method as proposed in (Bonneel et al., 2015; Rabin et al., 2011). The number of slices L is set to be 10, 20, and 50. We assess the convergence of each method by the estimated Wasserstein distance of order 2 after each iteration, i.e. $\widehat{W}_2\left(\phi^{(k)}(\mathcal{X}), \mathcal{X}\right)$, where $\phi^{(k)}(\cdot)$ is the estimator of OTM after kth iteration. For all three methods, we set the maximum number of iterations to be 200. Notice that, the Wasserstein distance between p_X and p_Y admits a closed form,

$$\begin{aligned} & W_2^2(p_X, p_Y) \\ = & ||\boldsymbol{\mu}_X - \boldsymbol{\mu}_Y||_2^2 + trace\left(\mathcal{S}_X + \mathcal{S}_Y - 2(\mathcal{S}_X^{1/2}\mathcal{S}_Y\mathcal{S}_X^{1/2})^{1/2}\right), \end{aligned}$$

which serves as the ground-truth. The results are presented in Figure 4.3.

In all three scenarios, PPMM (red line) converges to the ground truth within a small number of iterations. The fluctuations of the convergence curves observed in Figure 4.3 are caused by the non-equal sample means. This can be adjusted by applying a first-order dimension reduction method (e.g., SIR). We do not pursue this approach as the fluctuations do not cover the main pattern in Figure 4.3. When $d = 10$, RANDOM and SLICED converge to the ground truth but in a much slower manner. When $d = 20$ and 50, neither RANDOM nor SLICED manages to converge within 200 iterations. We also find a large number of slices L does not necessarily lead to a better estimation for the

Table 4.1: The mean CPU time (sec) per iteration, with standard deviations presented in parentheses

	PPMM	RANDOM	SLICED(10)	SLICED(20)	SLICED(50)
$d = 10$	0.019 (0.008)	0.011 (0.008)	0.111 (0.019)	0.213 (0.024)	0.529 (0.031)
$d = 20$	0.027 (0.011)	0.014 (0.008)	0.125 (0.027)	0.247 (0.033)	0.605 (0.058)
$d = 50$	0.059 (0.036)	0.015 (0.008)	0.171 (0.037)	0.338 (0.049)	0.863 (0.117)

Table 4.2: The mean convergence time (sec) for estimating the Wasserstein distance, with standard deviations presented in parentheses. The symbol "-" is inserted when the algorithm fails to converge.

d	PPMM	RANDOM	SLICED(10)	AUCTION	REVSIM	SHORT
10	0.6 (0.1)	4.8 (1.7)	23.0 (2.6)	99.7 (10.4)	40.2 (4.0)	42.5 (3.2)
20	2.1 (0.3)	24.4 (3.2)	230.2 (28.4)	109.4 (12.5)	42.6 (5.3)	50.2 (6.6)
50	5.5 (0.4)	-	-	125.5 (13.3)	46.5 (5.6)	56.5 (7.1)

SLICED method. As we can see, PPMM is the only one among three that is adaptive to large-scale OTM estimation problems.

In Table 1 below, we compare the computational cost of three methods by reporting the CPU time per iteration over 100 replication.[5] As we expected, the RANDOM method has the lowest CPU time per iteration due to it does not select projection direction. We notice that the CPU time per iteration of the SLICED method is proportional to the number of slices L. Last but not least, the CPU time per iteration of PPMM is slightly larger than RANDOM but much smaller than SLICED.

[5] The experiments are implemented by an Intel 2.6 GHz processor.

In the Table 2 below, we report the mean convergence time over 100 replications for PPMM, RANDOM, SLICED, the refined auction algorithm (AUCTIONBF)(Bertsekas, 1992), the revised simplex algorithm (REVSIM) (Luenberger, Ye, et al., 1984) and the shortlist method (SHORT) (Gottschlich & Schuhmacher, 2014).[6] Table 2 shows that the PPMM is the most computationally efficient method thanks to its cheap per iteration cost and fast convergence.

[6] AUCTION, REVSIM and SHORT are implemented by the R package "transport" (Schuhmacher et al., 2019).

4.4.2 Application to generative models

A critical issue in generative models is the so-called mode collapse, i.e., the generated "fake" sample fails to capture some modes present in the training data (Guo et al., 2019; Salimans et al., 2018). To address this issue, recent studies (Arjovsky et al., 2017; Guo et al., 2019; Kolouri et al., 2018) incorporated genera-

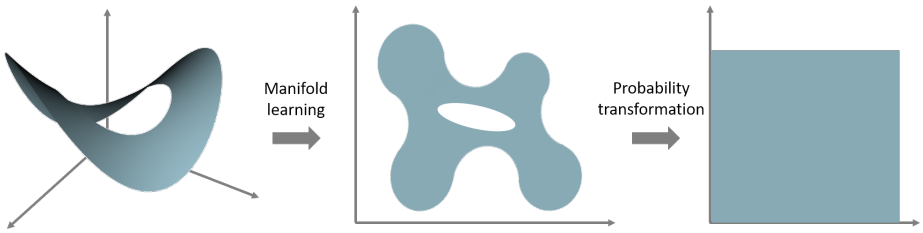

Figure 4.4: Illustration for the generative model using manifold learning and optimal transport

tive models with the optimal transportation theory. As illustrated in Figure 4.4, one can decompose the problem of generating fake samples into two major steps: (1) manifold learning and (2) probability transformation. The step (1) aims to discover the manifold structure of the training data by mapping the training data from the original space $\mathcal{X} \subset \mathbb{R}^d$ to a latent space $\mathcal{Z} \subset \mathbb{R}^{d^*}$ with $d^* \ll d$. Notice that the probability distribution of the transformed data in $\mathcal{Z}$ may not be convex, leading to the problem of mode collapse. The step (2) then addresses the mode collapse issue through transporting the distribution in $\mathcal{Z}$ to the uniform distribution $U([0,1]^{d^*})$. Then, the generative model takes a random input from $U([0,1]^{d^*})$ and sequentially applies the inverse transformations in step (2) and step (1) to generate the output. In practice, one may implement the step (1) and (2) using variational autoencoders (VAE) and OTM, respectively. As we can see, the estimation of OTM plays an essential role in this framework.

In this subsection, we apply PPMM as well as RANDOM and SLICED to generative models to study two datasets: MINST and Google doodle dataset. For the SLICED method, we set the number of slices to be 10, 20, and 50. For all three methods, we set the number of iterations is set to be $10d^*$. We use the squared Euclidean distance as the cost for the VAE model.

MNIST. We first study the MNIST dataset, which contains 60,000 training images and 10,000 testing images of hand written digits. We pull each 28×28 image to a 784-dimensional vector and rescale the grayscale values from $[0, 255]$ to $[0, 1]$. Following the method in (Arjovsky et al., 2017), we apply VAE to encode the data into a latent space $\mathcal{Z}$ of dimensionality $d^* = 8$. Then, the OTM from the distribution in $\mathcal{Z}$ to $U([0,1]^8)$ is estimated by PPMM as well as RANDOM and SLICED.

Figure 4.5: Left: random samples generated by PPMM. Right: linear interpolation between random pairs of images.

First, we visually examine the fake sample generated with PPMM. In the left-hand panel of Figure 4.5, we display some random images generated by PPMM. The right-hand panel of Figure 4.5 shows that PPMM can predict the continuous shift from one digit to another. To be specific, let $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^{784}$ be the sample of two digits (e.g. 3 and 9) in the testing set. Let $T : \mathcal{X} \to \mathcal{Z}$ be the map induced by VAE and $\widehat{\phi}$ the OTM estimated by PPMM. Then, $\widehat{\phi}(T(\cdot))$ maps the sample distribution to $U([0,1]^8)$. We linearly interpolate between $\widehat{\phi}(T(\boldsymbol{a}))$ and $\widehat{\phi}(T(\boldsymbol{b}))$ with equal-size steps. Then we transform the interpolated points back to the sample distribution to generate the middle columns in the right panel of Figure 4.5.

We use the *"Fréchet Inception Distance"* (FID) (Heusel et al., 2017) to quantify the similarity between the generated fake sample and the training sample. Specifically, we first generate 1,000 random inputs from $U([0,1]^8)$. We then apply PPMM, RANDOM, and SLICED to this input sample, yields the fake samples in the latent space $\mathcal{Z}$. Finally, we calculate the FID between the encoded training sample in the latent space and the generated fake samples, respectively. A small value of FID indicates the generated fake sample is similar to the training sample and vice versa. The sample mean and sample standard deviation (in parentheses) of FID over 50 replications are presented in Table 4.3. Table 4.3 indicates PPMM significantly outperforms the other two methods in terms of estimating the OTM.

Google doodle dataset. The Google Doodle dataset contains over 50 million drawings created by users with a mouse under 20 secs. We analyze a preprocessed version of this dataset from the quick draw Github account. In the

Table 4.3: The FID for the generated samples (lower the better), with standard deviations presented in parentheses

	PPMM	RANDOM	SLICED(10)	SLICED(20)	SLICED(50)
MNIST	**0.17** (0.01)	4.62 (0.02)	2.98 (0.01)	3.04 (0.01)	3.12 (0.01)
Doodle (face)	**0.59** (0.09)	8.78 (0.04)	5.69 (0.01)	6.01 (0.01)	5.52 (0.01)
Doodle (cat)	**0.24** (0.03)	8.93 (0.03)	5.99 (0.01)	5.26 (0.01)	5.33 (0.01)
Doodle (bird)	**0.36** (0.03)	7.81 (0.03)	5.44 (0.01)	5.50 (0.01)	4.98 (0.01)

dataset we use, the drawings are centered and rendered into 28×28 grayscale images. We pull each 28×28 image to a 784-dimensional vector and rescale the grayscale values from $[0, 255]$ to $[0, 1]$. In this experiment, we study the drawings from three different categories: smile face, cat, and bird. These three categories contain 161,666, 123,202, and 133,572 drawings, respectively. Within each category, we randomly split the data into a training set and a validation set of equal sample sizes.

We apply VAE to the training set with a stopping criterion selected by the validation set. The dimension of the latent space is set to be 16. Let $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^{784}$ be two vectors in the validation set, $T : \mathcal{X} \to \mathcal{Z}$ be the map induced by VAE and $\widehat{\phi}$ be the OTM estimated by PPMM. Note that $\widehat{\phi}(T(\cdot))$ maps the sample distribution to $U([0, 1]^{16})$. We then linearly interpolate between $\widehat{\phi}(T(\boldsymbol{a}))$ and $\widehat{\phi}(T(\boldsymbol{b}))$ with equal-size steps. The results are presented in Figure 4.6.

Figure 4.6: Linear interpolation between random pairs of images from the dataset of smile face (left), cat (center), and bird (right).

Then, we quantify the similarity between the generated fake samples and the truth by calculating the FID in the latent space. The sample mean and sample standard deviation (in parentheses) of FID over 50 replications are presented in Table 4.3. Again, the results in Table 4.3 justify the superior performance of PPMM over existing projection-based methods.

Chapter 5

Conclusion

Data reduction method, as an effective and general solution for big data problem, becomes more and more attractive. In this thesis, we focus on design-based subsampling methods and sufficient dimension reduction methods for solving large non-parametric regression and optimal transport problems. These methods are recently-proposed popular data reduction methods, shown to be efficient in keeping essential information from the data. In particular, in Chapter I, we show the design-based subsampling methods can be used to select subsamples that are robust to the model. In Chapter II and Chapter III, we show the design-based subsampling methods can be used to select informative subsamples. Finally, we discuss how to use sufficient dimension reduction to solve the computation burden of large-scale optimal transport problems. Although not covered in this chapter, the data reduction methods can also be applied to generalized linear models, time series models, variable selections, etc. A further refinement of the current methods and even brand new algorithms are under intensive development.

Appendix A

Proof for Chapter 1

A.1 Proof of Lemma 1.2.1

Proof. The inequality (1.2) yields $||\boldsymbol{h}||^2 \leq \alpha^2 \sum_{i=1}^{r} ||\boldsymbol{x}_i^*||^2 = \alpha^2 \text{tr}(\mathbf{R}^T\mathbf{R})$. One thus has

$$\boldsymbol{h}^T\mathbf{Q}^T\mathbf{Q}\boldsymbol{h} \leq \lambda_{max}(\mathbf{Q}^T\mathbf{Q})||\boldsymbol{h}||^2 \leq \lambda_{max}(\mathbf{Q}^T\mathbf{Q}) \cdot \alpha^2 \text{tr}(\mathbf{R}^T\mathbf{R}) \tag{A.1}$$

$$= \lambda_{max}(\mathbf{R}^T\mathbf{R}^{-1}) \cdot \alpha^2 \text{tr}(\mathbf{R}^T\mathbf{R}) = \frac{\alpha^2 \text{tr}(\mathbf{R}^T\mathbf{R})}{\lambda_{min}(\mathbf{R}^T\mathbf{R})}. \tag{A.2}$$

Recall that $\boldsymbol{\mu}_{max}(\cdot)$ as the corresponding eigenvector to $\lambda_{max}(\cdot)$. The first equation in (A.1) holds when $\boldsymbol{h} = c \cdot \boldsymbol{\mu}_{max}(\mathbf{Q}^T\mathbf{Q})$ for some real number c, and the second equation in (A.1) holds when $||\boldsymbol{h}||^2 = \alpha^2 \text{tr}(\mathbf{R}^T\mathbf{R})$. As a result, both equations in (A.1) hold when $\boldsymbol{h} = \sqrt{\alpha^2 \text{tr}(\mathbf{R}^T\mathbf{R})} \cdot \boldsymbol{\mu}_{max}(\mathbf{Q}^T\mathbf{Q})$. The desired result follows directly after plugging the inequality (A.2) in the equation (1.4). □

A.2 Proof of Theorem 2.3.1

The following Weyl's inequalities are needed in the proof.

Theorem A.2.1. ***Weyl's inequalities*** *(Horn & Johnson, 1990) Let* $\mathbf{A} \in \mathbb{R}^{n \times d}$ *and* $\mathbf{B} \in \mathbb{R}^{n \times d}$ *be two matrices and* $t = \min\{n, d\}$*. Let* $s_1(\mathbf{A}) \geq s_2(\mathbf{A}) \geq \ldots \geq s_t(\mathbf{A}) \geq 0$*,* $s_1(\mathbf{B}) \geq s_2(\mathbf{B}) \geq \ldots \geq s_t(\mathbf{B}) \geq 0$ *and* $s_1(\mathbf{A}+\mathbf{B}) \geq s_2(\mathbf{A}+\mathbf{B}) \geq \ldots \geq s_t(\mathbf{A}+\mathbf{B}) \geq 0$ *be the singular values of* $\mathbf{A}$*,* $\mathbf{B}$ *and* $\mathbf{A}+\mathbf{B}$*, respectively. Then*

$$|s_i(\mathbf{A}+\mathbf{B}) - s_i(\mathbf{A})| \leq s_1(\mathbf{B}), \quad i = 1, \ldots, t.$$

Proof of Theorem 2.3.1. Let $i = 1$, the Weyl's inequalities yield

$$s_1(\mathbf{R}_L) = s_1(\mathbf{L} + \mathbf{D}) \leq s_1(\mathbf{L}) + s_1(\mathbf{D}). \tag{A.3}$$

Let $i = p$, the Weyl's inequalities yield

$$s_p(\mathbf{R}_L) = s_p(\mathbf{L} + \mathbf{D}) \geq s_p(\mathbf{L}) - s_1(\mathbf{D}). \tag{A.4}$$

Recall that in Theorem 2.3.1, we assume $s_p(\mathbf{L}) - s_1(\mathbf{D}) > 0$. Combining inequality (A.3) and inequality (A.4) thus yields

$$\kappa(\mathbf{R}_L^T \mathbf{R}_L) = \left(\frac{s_1(\mathbf{R}_L)}{s_p(\mathbf{R}_L)}\right)^2 \leq \left(\frac{s_1(\mathbf{L}) + s_1(\mathbf{D})}{s_p(\mathbf{L}) - s_1(\mathbf{D})}\right)^2. \tag{A.5}$$

Performing a Taylor expansion of the right-hand side of the inequality (A.5), which can be viewed as a function of $s_1(\mathbf{D})$, around the point 0 yields

$$\begin{aligned}\left(\frac{s_1(\mathbf{L}) + s_1(\mathbf{D})}{s_p(\mathbf{L}) - s_1(\mathbf{D})}\right)^2 &\leq \kappa(\mathbf{L}^T\mathbf{L}) + 4\frac{s_1(\mathbf{L})^2}{s_p(\mathbf{L})^3} s_1(\mathbf{D}) + W_1 \\ &= \kappa(\mathbf{L}^T\mathbf{L}) + 4\frac{\kappa(\mathbf{L}^T\mathbf{L})}{s_p(\mathbf{L})} s_1(\mathbf{D}) + W_1, \end{aligned} \tag{A.6}$$

where $W_1 = o(s_1(\mathbf{D}))$ is the remainder. Plugging the inequality (A.6) back into (A.5) yields

$$\kappa(\mathbf{R}_L^T \mathbf{R}_L) \leq \kappa(\mathbf{L}^T\mathbf{L}) + 4\frac{\kappa(\mathbf{L}^T\mathbf{L})}{s_p(\mathbf{L})} s_1(\mathbf{D}) + W_1. \tag{A.7}$$

We now derive an upper bound for the first term on the right-hand side of inequality (1.4). Note that

$$\mathrm{tr}[(\mathbf{R}_L^T \mathbf{R}_L)^{-1}] \leq p\lambda_{max}((\mathbf{R}_L^T \mathbf{R}_L)^{-1}) = \frac{p}{s_p(\mathbf{R}_L)^2} \leq \frac{p}{(s_p(\mathbf{L}) - s_1(\mathbf{D}))^2}, \tag{A.8}$$

where the inequality (A.4) is used in the last step.

By performing a Taylor expansion of the right-hand side of the inequality (A.8) around the point 0, one has

$$\frac{p}{(s_p(\mathbf{L}) - s_1(\mathbf{D}))^2} = \frac{p}{s_p(\mathbf{L})^2} + 2\frac{\sqrt{p}}{s_p(\mathbf{L})^2} s_1(\mathbf{D}) + W_2, \tag{A.9}$$

where $W_2 = o(s_1(\mathbf{D}))$ is the remainder. Plugging the inequality (A.9) back into (A.8) yields

$$\text{tr}[(\mathbf{R}_L^T\mathbf{R}_L)^{-1}] \leq \frac{p}{s_p(\mathbf{L})^2} + 2\frac{\sqrt{p}}{s_p(\mathbf{L})^2}s_1(\mathbf{D}) + W_2. \tag{A.10}$$

Finally, plugging both the inequality (A.7) and (A.10) in the inequality (1.4) yields

$$MSE(\widetilde{\boldsymbol{\beta}}_{\mathbf{R}}) \leq \sigma^2 p^2 \frac{\kappa(\mathbf{L}^T\mathbf{L})}{tr(\mathbf{L}^T\mathbf{L})} + \alpha^2 p\kappa(\mathbf{L}^T\mathbf{L}) + O(s_1(\mathbf{D})).$$

The fact that $tr(\mathbf{L}^T\mathbf{L})) \leq p\lambda_{max}(\mathbf{L}^T\mathbf{L})) = p\kappa(\mathbf{L}^T\mathbf{L}))s_p(\mathbf{L})^2$ is used in the last step. This completes the proof. □

Appendix B

Proof for Chapter 2

B.1 Proof of an ancillary lemma

Following the notations in Algorithm 1, we let $\mathbf{S} = (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_r)^T$, $\mathbf{U}^* = (\mathbf{u}_1^*, \ldots, \mathbf{u}_r^*)^T$, and $\Delta\mathbf{S} = \mathbf{U}^* - \mathbf{S} = (\Delta\boldsymbol{s}_1, \ldots, \Delta\boldsymbol{s}_r)^T$. The following lemma derives the distribution of the random variable $||\Delta\boldsymbol{s}_i||$, where $||\cdot||$ denotes the Euclidean norm. Notice that this lemma also coincides with void probabilities of the binomial point process (Moltchanov, 2012).

Lemma B.1.1. *Let κ_d be the volume of the d-dimensional unit Euclidean ball, which is a constant when d is fixed. Suppose $\mathcal{U}_n = \{\mathbf{u}_i\}_{i=1}^n$ are i.i.d. observations generated from the uniform distribution $U[0,1]^d$. For $i = 1, \ldots r$, $||\Delta\boldsymbol{s}_i||$s are identically distributed with the cumulative density function, denoted by $F(\rho)$, where*

$$F(\rho) \geq 1 - \left(1 - \kappa_d(\rho/2)^d\right)^n, \text{ when } 0 < \rho < 2(1/\kappa_d)^{1/d}.$$

Proof of Lemma B.1.1. Let the count $N(\mathcal{B})$ ne the number of points in the domain $\mathcal{B}$, and $b(\boldsymbol{s}_i, \rho)$ be a Euclidean ball centered at $\boldsymbol{s}_i$ with radius ρ. The volume of $b(\boldsymbol{s}_i, \rho)$ is thus $\kappa_d\rho^d$. Now we consider two scenarios:

Scenario 1. When $b(\boldsymbol{s}_i, \rho) \subset [0,1]^d$, as a Bernoulli trial, one has

$$\begin{aligned}
&\text{pr}\left(N(b(\boldsymbol{s}_i, \rho)) = 0 | N([0,1]^d) = n\right) \\
&= \frac{\text{pr}\left(N(b(\boldsymbol{s}_i, \rho)) = 0, N([0,1]^d \backslash b(\boldsymbol{s}_i, \rho)) = n\right)}{\text{pr}\left(N([0,1]^d) = n\right)} \\
&= (1 - \kappa_d\rho^d)^n.
\end{aligned}$$

Scenario 2. When $b(\boldsymbol{s}_i, \rho) \not\subset [0,1]^d$, the volume of the intersection between $b(\boldsymbol{s}_i, \rho)$ and $[0,1]^d$ is no larger than $\kappa_d \rho^d$ and no smaller than $\kappa_d(\rho/2)^d$. This is due to the fact that the design point $\boldsymbol{s}_i \in [0,1]^d$.

Combining these two scenarios, one thus has

$$\begin{aligned}(1-\kappa_d\rho^d)^n \quad &\leq \text{pr}\left(N(b(\boldsymbol{s}_i,\rho))=0 | N([0,1]^d)=n\right) \\ &\leq (1-\kappa_d(\rho/2)^d)^n. \end{aligned} \tag{B.1}$$

Notice that $||\Delta \boldsymbol{s}_i||$ measures the distance between $\boldsymbol{s}_i$ and the its nearest neighbor $\mathbf{u}_i^*$. Thus $N(b(\boldsymbol{s}_i,\rho)) = 0$ is equivalent to $||\Delta \boldsymbol{s}_i|| > \rho$. From the inequality (B.1), one thus has

$$\text{pr}(||\Delta \boldsymbol{s}_i|| > \rho) \leq \left(1-\kappa_d(\rho/2)^d\right)^n. \tag{B.2}$$

The lemma thus follows. □

B.2 Proof of Lemma 1

We need the following definitions and notations for the proof. The projection function for a scalar θ is defined as

$$\text{proj}(\theta) = \begin{cases} 0 & \text{if} \quad \theta < 0 \\ 1 & \text{if} \quad \theta > 1 \\ \theta & \text{otherwise.} \end{cases}$$

For a d-dimensional vector $\boldsymbol{\Theta} = (\theta_1, \ldots, \theta_d)^T$, define

$$\text{Proj}(\boldsymbol{\Theta}) = \big(\text{proj}(\theta_1), \ldots, \text{proj}(\theta_d)\big)^T.$$

We also define two operators $\uparrow$ and $\downarrow$, between a vector $\boldsymbol{\Theta}$ and a scalar θ

$$\begin{aligned}\boldsymbol{\Theta} \uparrow \theta &= (\theta_1 + \theta, \ldots, \theta_d + \theta)^T \\ \boldsymbol{\Theta} \downarrow \theta &= (\theta_1 - \theta, \ldots, \theta_d - \theta)^T.\end{aligned}$$

For any $\boldsymbol{a} \in [0,1]^d$ and $\rho > 0$, we denote

$$\begin{aligned}\mathcal{A}_{a,\rho} &= [0, \text{Proj}(\boldsymbol{a} \uparrow \rho)) \quad \setminus \quad [0, \text{Proj}(\boldsymbol{a} \downarrow \rho)), \\ \mathcal{A}^c_{a,\rho} &= [0,1]^d \quad \setminus \quad \mathcal{A}_{a,\rho}.\end{aligned}$$

Figure B.1 gives an toy example of $\mathcal{A}_{a,\rho}$ (grey area) when $d = 2$.

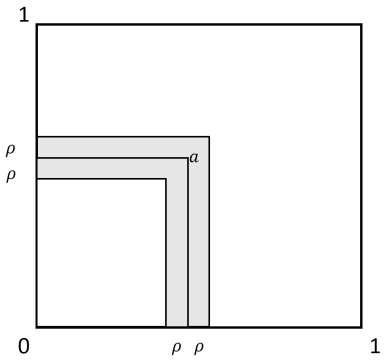

Figure B.1: The grey area illustrates $\mathcal{A}_{a,\rho}$, when $d = 2$

Proof. We now bound the difference between $D^*(\mathcal{U}_r^*)$ and $D^*(\mathcal{S}_r)$.

$$
\begin{aligned}
&|D^*(\mathcal{U}_r^*) - D^*(\mathcal{S}_r)| \\
&= \left| \sup_{\boldsymbol{a}\in[0,1]^d} D(\mathcal{U}_r^*, \boldsymbol{a}) - \sup_{\boldsymbol{b}\in[0,1]^d} D(\mathcal{S}_r, \boldsymbol{b}) \right| \\
&= \left| \sup_{\boldsymbol{a}\in[0,1]^d} \Big(D(\mathcal{U}_r^*, \boldsymbol{a}) - \sup_{\boldsymbol{b}\in[0,1]^d} D(\mathcal{S}_r, \boldsymbol{b}) \Big) \right| \\
&\leq \left| \sup_{\boldsymbol{a}\in[0,1]^d} \Big(D(\mathcal{U}_r^*, \boldsymbol{a}) - D(\mathcal{S}_r, \boldsymbol{a}) \Big) \right| \\
&\leq \left| \sup_{\boldsymbol{a}\in[0,1]^d} \left(\left| \frac{1}{r} \sum_{i=1}^{r} 1\{\boldsymbol{s}_i \in [\mathbf{0}, \boldsymbol{a})\} - \frac{1}{r} \sum_{i=1}^{r} 1\{\mathbf{u}_i^* \in [\mathbf{0}, \boldsymbol{a})\} \right| \right) \right| \\
&= \sup_{\boldsymbol{a}\in[0,1]^d} \left(\left| \frac{1}{r} \sum_{i=1}^{r} \Big(1\{\boldsymbol{s}_i \in [\mathbf{0}, \boldsymbol{a})\} - 1\{(\boldsymbol{s}_i + \Delta\boldsymbol{s}_i) \in [\mathbf{0}, \boldsymbol{a})\} \Big) \right| \right) \\
&\leq \sup_{\boldsymbol{a}\in[0,1]^d} \left(\frac{1}{r} \sum_{i=1}^{r} |1\{\boldsymbol{s}_i \in [\mathbf{0}, \boldsymbol{a})\} - 1\{(\boldsymbol{s}_i + \Delta\boldsymbol{s}_i) \in [\mathbf{0}, \boldsymbol{a})\}| \right), \quad \text{(B.3)}
\end{aligned}
$$

where the property of supreme and the triangle inequality are used.

Denote $\tilde{\delta}_i = 1\{\boldsymbol{s}_i \in [\mathbf{0}, \boldsymbol{a})\} - 1\{(\boldsymbol{s}_i + \Delta\boldsymbol{s}_i) \in [\mathbf{0}, \boldsymbol{a})\}$. Using the notations presented before our proof and inequality (B.3), one has, $\forall \rho > 0$,

$$
\begin{aligned}
&|D^*(\mathcal{U}_r^*) - D^*(\mathcal{S}_r)| \\
\leq &\sup_{\boldsymbol{a}\in[0,1]^d} \left(\frac{1}{r} \sum_{i=1}^{r} \left| \tilde{\delta}_i \right| \right) \\
\leq &\sup_{\boldsymbol{a}\in[0,1]^d} \left(\frac{1}{r} \sum_{i=1}^{r} \left| \tilde{\delta}_i \right| \times \left(1\{\boldsymbol{s}_i \in \mathcal{A}_{a,\rho}\} \right) \right) \\
&+ \sup_{\boldsymbol{a}\in[0,1]^d} \left(\frac{1}{r} \sum_{i=1}^{r} \left| \tilde{\delta}_i \right| \times \left(1\{\boldsymbol{s}_i \in \mathcal{A}_{a,\rho}^c\} \right) \right) \qquad \text{(B.4)}
\end{aligned}
$$

Let the first term and the second term on the right-hand side of the inequality (B.4) be T_1 and T_2, respectively. For T_1, one has

$$
\begin{aligned}
T_1 &\leq \sup_{\boldsymbol{a}\in[0,1]^d} \left(\frac{1}{r} \sum_{i=1}^{r} 1\{\boldsymbol{s}_i \in \mathcal{A}_{a,\rho}\} \right) \\
&= \sup_{\boldsymbol{a}\in[0,1]^d} (I_1(a) - I_2(a)), \qquad \text{(B.5)}
\end{aligned}
$$

where

$$
\begin{aligned}
I_1(a) &= \frac{1}{r} \sum_{i=1}^{r} 1\left\{ \boldsymbol{s}_i \in \left[0, \text{Proj}\big(a \uparrow \rho\big) \right) \right\}, \\
I_2(a) &= \frac{1}{r} \sum_{i=1}^{r} 1\left\{ \boldsymbol{s}_i \in \left[0, \text{Proj}\big(a \downarrow \rho\big) \right) \right\}.
\end{aligned}
$$

For any $\boldsymbol{a} \in [0,1]^d$ and $\rho > 0$, one has

$$\begin{aligned} D^*(\mathcal{S}_r) &= \sup_{\boldsymbol{a}\in[0,1]^d} D(\mathcal{S}_r, a) \\ &\geq D\Big(\mathcal{S}_r, \text{Proj}\big(a \uparrow \rho\big)\Big) \\ &= \left| \frac{1}{r}\sum_{i=1}^{r} 1\Big\{ \boldsymbol{s}_i \in \big[0, \text{Proj}(\boldsymbol{a} \uparrow \rho)\big)\Big\} - \prod_{j=1}^{d} \text{proj}(a_j + \rho) \right| \\ &= \left| I_1(a) - \prod_{j=1}^{d} \text{proj}(a_j + \rho) \right| \\ &\geq I_1(a) - \prod_{j=1}^{d} \text{proj}(a_j + \rho). \end{aligned} \tag{B.6}$$

The inequality (B.6) yields, $\forall \rho > 0$,

$$I_1(a) \leq D^*(\mathcal{S}_r) + \prod_{j=1}^{d} \text{proj}(a_j + \rho). \tag{B.7}$$

Analogous to inequality (B.6), $\forall \rho > 0$, one has

$$D^*(\mathcal{S}_r) \geq \left| I_2(a) - \prod_{j=1}^{d} \text{proj}(a_j - \rho) \right|,$$

which yields

$$I_2(a) \geq \prod_{j=1}^{d} \text{proj}(a_j - \rho) - D^*(\mathcal{S}_r). \tag{B.8}$$

Plug (B.7) and (B.8) back into (B.5) yields, $\forall \rho > 0$,

$$\begin{aligned} T_1 &\leq \sup_{\boldsymbol{a}\in[0,1]^d} \Big(2D^*(\mathcal{S}_r) + \big(\prod_{j=1}^{d} \text{proj}(a_j + \rho) - \prod_{j=1}^{d} \text{proj}(a_j - \rho) \big) \Big) \\ &\leq \sup_{\boldsymbol{a}\in[0,1]^d} \Big(2D^*(\mathcal{S}_r) + \big(\prod_{j=1}^{d} (a_j + \rho) - \prod_{j=1}^{d} (a_j - \rho) \big) \Big). \end{aligned} \tag{B.9}$$

Next, we show $\text{pr}(T_2 = 0) \to 1$ for a properly specified ρ and r as $n \to \infty$. Notice that for any $\rho > 0$ and $\boldsymbol{s}_i \in \mathcal{A}^c_{a,\rho}$, $||\Delta \boldsymbol{s}_i|| \leq \rho$ is a sufficient condition

for

$$1\{\boldsymbol{s}_i \in [\boldsymbol{0}, \boldsymbol{a})\} - 1\{(\boldsymbol{s}_i + \Delta\boldsymbol{s}_i) \in [\boldsymbol{0}, \boldsymbol{a})\} = 0,$$

since $(\Delta\boldsymbol{s}_i)_j \leq (\sum_{j=1}^{d}((\Delta\boldsymbol{s}_i)_j)^2)^{1/2} = ||\Delta\boldsymbol{s}_i||$.

As a result, when $0 < \rho < 2(1/\kappa_d)^{1/d}$, one has

$$\begin{aligned}
&\text{pr}(T_2 = 0) \\
&\geq \text{pr}\Big(\bigcap_{\{i|\boldsymbol{s}_i \in \mathcal{A}^c_{a,\rho}\}} \big(||\Delta\boldsymbol{s}_i|| \leq \rho\big)\Big) \\
&\geq \text{pr}\Big(\bigcap_{i=1}^{r} \big(||\Delta\boldsymbol{s}_i|| \leq \rho\big)\Big) \\
&= 1 - \text{pr}\Big(\bigcup_{i=1}^{r} \big(||\Delta\boldsymbol{s}_i|| > \rho\big)\Big) \\
&\geq 1 - \sum_{i=1}^{r} \text{pr}\Big(||\Delta\boldsymbol{s}_i|| > \rho\Big) \\
&\geq 1 - r\Big(1 - \kappa_d(\rho/2)^d\Big)^n, \qquad \text{(B.10)}
\end{aligned}$$

where the inequality (B.2) is used in the last step.

Plug in the result in (B.9) and (B.10) back into (B.4) yields, when $0 < \rho < 2(1/\kappa_d)^{1/d}$, one has

$$\begin{aligned}
&|D^*(\mathcal{U}_r^*) - D^*(\mathcal{S}_r)| \\
&\leq \sup_{\boldsymbol{a}\in[0,1]^d} \big(2D^*(\mathcal{S}_r) + \big(\textstyle\prod_{j=1}^{d}(a_j + \rho) - \prod_{j=1}^{d}(a_j - \rho)\big)\big), \qquad \text{(B.11)}
\end{aligned}$$

with probability at least $1 - r\Big(1 - \kappa_d(\rho/2)^d\Big)^n$.

Let $\rho = r^{-(1-\delta)}$ for any arbitrary small $\delta > 0$. Notice that such ρ satisfies the condition $0 < \rho < 2(1/\kappa_d)^{1/d}$ when r is large enough. Using the fact that $\boldsymbol{a} \in [0,1]^d$ and $\sum_{j=1}^{d} a_j \leq d$, one has

$$\begin{aligned}
&\sup_{\boldsymbol{a}\in[0,1]^d} \big(2D^*(\mathcal{S}_r) + \big(\prod_{j=1}^{d}(a_j + \rho) - \prod_{j=1}^{d}(a_j - \rho)\big)\big) \\
&= \sup_{\boldsymbol{a}\in[0,1]^d} \big(2D^*(\mathcal{S}_r) + 2\sum_{j=1}^{d}(a_j)r^{-(1-\delta)} + o(r^{-(1-\delta)})\big) \\
&= O(r^{-(1-\delta)}), \qquad \text{(B.12)}
\end{aligned}$$

where the fact that $D^*(\mathcal{S}_r) = O(r^{-(1-\delta)})$ is used in the last step.

Also, notice that when $\rho = r^{-(1-\delta)}$ and $r = O(n^{1/d})$, one has

$$\begin{aligned}
&\lim_{n\to\infty}\left(1 - r\Big(1 - \kappa_d(\rho/2)^d\Big)^n\right) \\
&= 1 - \lim_{n\to\infty} r\left(1 - \kappa_d\Big(\frac{1}{2r^{1-\delta}}\Big)^d\right)^n \\
&= 1. \qquad \text{(B.13)}
\end{aligned}$$

Combining the results in (B.12) and (B.13) together and plug into the equation (B.11) yields, when $r = O(n^{1/d})$, one has $D^*(\mathcal{U}_r^*) = O_p(r^{-(1-\delta)})$. This completes the proof. □

B.3 Proof of Theorem 1

The following Koksma-Hlawka inequality (Kuipers & Niederreiter, 2012) is needed in the proof.

Theorem B.3.1. ***(Koksma-Hlawka inequality)*** *Denote $\mathcal{S}_r = \{\boldsymbol{s}_1, ..., \boldsymbol{s}_r\}$ as a set of data points in $[0,1]^d$ and f is a function on $[0,1]^d$ with bounded total variation $\mathcal{V}(f)$. The total variation is defined in the sense of Hardy and Krause (Owen, 2003). Then,*

$$\left|\int_{[0,1]^d} f(\boldsymbol{x})d\boldsymbol{x} - \frac{1}{r}\sum_{i=1}^{r} f(\boldsymbol{s}_i)\right| \leq D^*(\mathcal{S}_r)\mathcal{V}(f).$$

Proof for Theorem 1. For any fixed point $\boldsymbol{z} \in \mathbb{R}^d$, the full sample estimator can be written as

$$\widehat{p}(\boldsymbol{z}|\mathbf{X}) = \frac{1}{n}\sum_{i=1}^{n}\left\{\prod_{j=1}^{d} K_h\left(z_j - x_{ij}\right)\right\}. \qquad \text{(B.14)}$$

Let X be the random variable with probability distribution function p. Let ϕ^* be the optimal transport map such that $\phi^*(X)$ has the uniform distribution on $[0,1]^d$, i.e., $U[0,1]^d$. One thus can calculate the expectation of the equation (B.14),

$$\mathrm{E}(\widehat{p}(\boldsymbol{z}|\mathbf{X})) = \int_{[0,1]^d} g_{\boldsymbol{z}}(\mathbf{u})d\mathbf{u}, \qquad \text{(B.15)}$$

where $g_{\boldsymbol{z}}(\mathbf{u}) = \prod_{j=1}^{d} K_h\left(z_j - ((\phi^*)^{-1}(\mathbf{u}))_j\right)$.

Recall $\mathcal{U}_r^* = \{\mathbf{u}_i^*\}_{i=1}^r$ in Algorithm 1, the proposed subsample estimator can be written as

$$\begin{aligned}\widetilde{p}(\boldsymbol{z}|\mathbf{X}^*) &= \frac{1}{r}\sum_{i=1}^{r}\left\{\prod_{j=1}^{d} K_h\left(z_j - ((\phi^*)^{-1}(\mathbf{u}_i^*))_j\right)\right\} \\ &= \frac{1}{r}\sum_{i=1}^{r} g_{\boldsymbol{z}}(\mathbf{u}_i^*). \end{aligned} \tag{B.16}$$

Combining the result in (B.15) and (B.16) and using the Koksma-Hlawka inequality, one has,

$$\begin{aligned}|\widetilde{p}(\boldsymbol{z}|\mathbf{X}^*) - \mathrm{E}\left(\widehat{p}(\boldsymbol{z}|\mathbf{X})\right)| &= \left|\frac{1}{r}\sum_{i=1}^{r} g_{\boldsymbol{z}}(\mathbf{u}_i^*) - \int_{[0,1]^d} g_{\boldsymbol{z}}(\mathbf{u})d\mathbf{u}\right| \\ &\leq D^*(\mathcal{U}_r^*)\mathcal{V}(g_{\boldsymbol{z}}). \end{aligned} \tag{B.17}$$

To analyze $\mathcal{V}(g_{\boldsymbol{z}})$, we consider two cases, $d \geq 2$ and $d = 1$. When $d \geq 2$, one has

$$\mathcal{V}(g_{\boldsymbol{z}}) = \int_{[0,1]^d} || \bigtriangledown g_{\boldsymbol{z}}(\mathbf{u})||d\mathbf{u},$$

where $||\cdot||$ is the $\mathcal{L}_2$ norm, and $\bigtriangledown g_{\boldsymbol{z}}(\mathbf{u}) = \left(\frac{\partial g_{\boldsymbol{z}}(\mathbf{u})}{\partial u_1}, ..., \frac{\partial g_{\boldsymbol{z}}(\mathbf{u})}{\partial u_d}\right)^T$. To simplify the expression of $g_{\boldsymbol{z}}(\mathbf{u})$, we set

$$\mathcal{K}(\boldsymbol{x}) = \prod_{j=1}^{d} K(x_j), \quad \boldsymbol{x} \in \mathbb{R}^d.$$

One thus has $g_{\boldsymbol{z}}(\mathbf{u}) = \frac{1}{h^d}\mathcal{K}\left(\frac{\boldsymbol{z}-(\phi^*)^{-1}(\mathbf{u})}{h}\right)$. Let $\boldsymbol{\omega} = \frac{\boldsymbol{z}-(\phi^*)^{-1}(\mathbf{u})}{h}$, we have

$$\bigtriangledown g_{\boldsymbol{z}}(\mathbf{u}) = \frac{1}{h^d} J_{\boldsymbol{\omega}\to\mathbf{u}}^T \bigtriangledown \mathcal{K}(\boldsymbol{\omega}),$$

where

$$J_{\boldsymbol{\omega}\to\mathbf{u}} = \begin{bmatrix} \frac{\partial\omega_1}{\partial u_1} & \cdots & \frac{\partial\omega_1}{\partial u_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial\omega_d}{\partial u_1} & \cdots & \frac{\partial\omega_d}{\partial u_d} \end{bmatrix}.$$

Let

$$J_{\mathbf{u}\to\boldsymbol{\omega}} = \begin{bmatrix} \frac{\partial u_1}{\partial\omega_1} & \cdots & \frac{\partial u_1}{\partial\omega_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial u_d}{\partial\omega_1} & \cdots & \frac{\partial u_d}{\partial\omega_d} \end{bmatrix}, J_{\phi^*} = \begin{bmatrix} \frac{\partial(\phi^*(\boldsymbol{x}))_1}{\partial x_1} & \cdots & \frac{\partial(\phi^*(\boldsymbol{x}))_1}{\partial x_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial(\phi^*(\boldsymbol{x}))_d}{\partial x_1} & \cdots & \frac{\partial(\phi^*(\boldsymbol{x}))_d}{\partial x_d} \end{bmatrix},$$

and

$$J_{(\phi^*)^{-1}} = \begin{bmatrix} \frac{\partial((\phi^*)^{-1}(\mathbf{u}))_1}{\partial u_1} & \cdots & \frac{\partial((\phi^*)^{-1}(\mathbf{u}))_1}{\partial u_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial((\phi^*)^{-1}(\mathbf{u}))_d}{\partial u_1} & \cdots & \frac{\partial((\phi^*)^{-1}(\mathbf{u}))_d}{\partial u_d} \end{bmatrix}.$$

Notice that $J_{\boldsymbol{\omega}\to\mathbf{u}} = -\frac{1}{h} J_{(\phi^*)^{-1}}$, one thus has

$$\bigtriangledown g_{\boldsymbol{z}}(\mathbf{u}) = \frac{1}{h^{d+1}} J^T_{(\phi^*)^{-1}} \bigtriangledown \mathcal{K}(\boldsymbol{\omega}).$$

Using the Jensen's inequality, one has

$$\begin{aligned} \mathcal{V}^2(g_{\boldsymbol{z}}) &\leq \int_{[0,1]^d} || \bigtriangledown g_{\boldsymbol{z}}(\mathbf{u})||^2 d\mathbf{u} \\ &= \int_{[0,1]^d} (\bigtriangledown g_{\boldsymbol{z}}(\mathbf{u}))^T \bigtriangledown g_{\boldsymbol{z}}(\mathbf{u}) d\mathbf{u} \\ &= \frac{1}{h^{2d+2}} \int_{[0,1]^d} (\bigtriangledown \mathcal{K}(\boldsymbol{\omega}))^T J_{(\phi^*)^{-1}} J^T_{(\phi^*)^{-1}} \bigtriangledown \mathcal{K}(\boldsymbol{\omega}) d\mathbf{u} \\ &= \frac{1}{h^{2d+2}} \int_{\Omega} (\bigtriangledown \mathcal{K}(\boldsymbol{\omega}))^T J_{(\phi^*)^{-1}} J^T_{(\phi^*)^{-1}} \bigtriangledown \mathcal{K}(\boldsymbol{\omega}) |\det(J_{\mathbf{u}\to\boldsymbol{\omega}})| d\boldsymbol{\omega} \\ &= \frac{1}{h^{d+2}} \int_{\Omega} (\bigtriangledown \mathcal{K}(\boldsymbol{\omega}))^T J_{(\phi^*)^{-1}} J^T_{(\phi^*)^{-1}} \bigtriangledown \mathcal{K}(\boldsymbol{\omega}) |\det(J_{\phi^*})| d\boldsymbol{\omega}, \end{aligned} \tag{B.18}$$

where that fact that $\mathbf{u} = \phi^*(\boldsymbol{z} - h\boldsymbol{\omega})$, $J_{\mathbf{u}\to\boldsymbol{\omega}} = -hJ_{\phi^*}$, and $|\det(J_{\mathbf{u}\to\boldsymbol{\omega}})| = h^d |\det(J_{\phi^*})|$ are used in the last equation.

Notice that

$$\begin{aligned} &(\bigtriangledown \mathcal{K}(\boldsymbol{\omega}))^T J_{(\phi^*)^{-1}} J^T_{(\phi^*)^{-1}} \bigtriangledown \mathcal{K}(\boldsymbol{\omega}) \qquad \text{(B.19)} \\ &\quad = \operatorname{tr}\left((\bigtriangledown \mathcal{K}(\boldsymbol{\omega}))^T J_{(\phi^*)^{-1}} J^T_{(\phi^*)^{-1}} \bigtriangledown \mathcal{K}(\boldsymbol{\omega}) \right) \\ &\quad = \operatorname{tr}\left(\bigtriangledown \mathcal{K}(\boldsymbol{\omega}) (\bigtriangledown \mathcal{K}(\boldsymbol{\omega}))^T J_{(\phi^*)^{-1}} J^T_{(\phi^*)^{-1}} \right) \\ &\quad \leq \operatorname{tr}\left(\bigtriangledown \mathcal{K}(\boldsymbol{\omega}) (\bigtriangledown \mathcal{K}(\boldsymbol{\omega}))^T \right) \operatorname{tr}\left(J_{(\phi^*)^{-1}} J^T_{(\phi^*)^{-1}} \right). \qquad \text{(B.20)} \end{aligned}$$

For the first term in the right-hand-side of (B.20), one has

$$\begin{aligned} \operatorname{tr}\left(\bigtriangledown \mathcal{K}(\boldsymbol{\omega}) (\bigtriangledown \mathcal{K}(\boldsymbol{\omega}))^T \right) &= \operatorname{tr}\left((\bigtriangledown \mathcal{K}(\boldsymbol{\omega}))^T \bigtriangledown \mathcal{K}(\boldsymbol{\omega}) \right) \\ &= (\bigtriangledown \mathcal{K}(\boldsymbol{\omega}))^T \bigtriangledown \mathcal{K}(\boldsymbol{\omega}) \\ &= \sum_{k=1}^{d} \left(\left\{ \prod_{j\neq k} K^2(\omega_j) \right\} (K'(\omega_k))^2 \right). \end{aligned} \tag{B.21}$$

For the second term in the right-hand-side of (B.20), one has

$$\mathrm{tr}\left(J_{(\phi^*)^{-1}} J_{(\phi^*)^{-1}}^T\right) \leq C,$$

for a positive constant C. This is because, as p is defined on a compact convex domain $\Omega \subset \mathbb{R}^d$, the optimal transport map ϕ^* ensures that all the entries in $J_{(\phi^*)^{-1}}$ are finite (Villani, 2008). Thus, $\mathrm{tr}\left(J_{(\phi^*)^{-1}} J_{(\phi^*)^{-1}}^T\right)$ has a bounded value. As a result, plugging the equation (B.21) and the inequality (B.20) back into (B.18) yields

$$\begin{aligned}
\mathcal{V}^2(g_{\boldsymbol{z}}) &\leq \frac{1}{h^{d+2}} C \int \cdots \int \sum_{k=1}^{d} \left(\left\{ \prod_{j \neq k} K^2(\omega_j) \right\} \left(K'(\omega_k)\right)^2 \right) d\omega_1 \cdots d\omega_d \\
&= \frac{1}{h^{d+2}} C \sum_{k=1}^{d} \left\{ \prod_{j \neq k} \int_{\Omega_j} K^2(\omega_j) d\omega_j \int_{\Omega_k} \left(K'(\omega_k)\right)^2 d\omega_k \right\} \\
&= O\left(\frac{1}{h^{d+2}}\right), \qquad \text{(B.22)}
\end{aligned}$$

where the regularity condition (d) and (e) are used in the last step.

When $d = 1$, one has $\mathcal{V}(g_z) = \int_0^1 |g_z'(u)| du$, where $g_z(u) = K_h(z - (\phi^*)^{-1}(u))$ and

$$g_z'(u) = \frac{1}{h} \frac{d}{du} K\left(\frac{z - (\phi^*)^{-1}(u)}{h}\right).$$

Let $\omega(u) = \frac{z - (\phi^*)^{-1}(u)}{h}$, we have $g_z'(u) = \frac{1}{h} K'(\omega)\omega'(u)$. Using the Jensen's inequality, one has

$$\begin{aligned}
\mathcal{V}^2(g_z) &\leq \int_0^1 |g_z'(u)|^2 du \\
&= \frac{1}{h^2} \int_0^1 \left(K'(\omega)\right)^2 \left(\omega'(u)\right)^2 du \\
&= \frac{1}{h^2} \int_0^1 \left(K'(\omega)\right)^2 \omega'(u) d\omega(u) \\
&\leq \frac{1}{h^2} \int_0^1 \frac{1}{h} \left(K'(\omega)\right)^2 \left|((\phi^*)^{-1})'(u)\right| d\omega(u) \\
&= O\left(\frac{1}{h^3}\right).
\end{aligned}$$

Such a result is consistent with the inequality (B.22) that $\mathcal{V}^2(g_z) \leq O\left(\frac{1}{h^{d+2}}\right)$. Combining the above result, Lemma 1, and inequality (B.17), we have

$$\left(\widetilde{p}(\boldsymbol{z}|\mathbf{X}^*) - \mathrm{E}\left(\widehat{p}(\boldsymbol{z}|\mathbf{X})\right)\right)^2 \leq \left(D^*(\mathcal{U}_r^*)\right)^2 \mathcal{V}^2(g_{\boldsymbol{z}}) \tag{B.23}$$

$$\leq O_p\left(\frac{1}{r^{2(1-\delta)}h^{d+2}}\right). \tag{B.24}$$

We now quantify the AIMSE of the proposed estimator,

$$\begin{aligned}
\mathrm{AIMSE}(\widetilde{p}(\boldsymbol{z}|\mathbf{X}^*)) &= \left(\widetilde{p}(\boldsymbol{z}|\mathbf{X}^*) - p(\boldsymbol{z})\right)^2 \\
&= \left(\widetilde{p}(\boldsymbol{z}|\mathbf{X}^*) - \mathrm{E}(\widehat{p}(\boldsymbol{z}|\mathbf{X})) + \mathrm{E}(\widehat{p}(\boldsymbol{z}|\mathbf{X})) - p(\boldsymbol{z})\right)^2 \\
&\leq 2\left|\widetilde{p}(\boldsymbol{z}|\mathbf{X}^*) - \mathrm{E}(\widehat{p}(\boldsymbol{z}|\mathbf{X}))\right|^2 + 2\left|\mathrm{E}(\widehat{p}(\boldsymbol{z}|\mathbf{X})) - p(\boldsymbol{z})\right|^2.
\end{aligned} \tag{B.25}$$

It is known that

$$\left|\mathrm{E}(\widehat{p}(\boldsymbol{z}|\mathbf{X})) - p(\boldsymbol{z})\right|^2 = O(h^4), \tag{B.26}$$

see (Scott, 2015) for more details. Plugging (B.23) and (B.26) into (B.25) yields

$$\mathrm{AIMSE}(\widetilde{p}(\boldsymbol{z}|\mathbf{X}^*)) \leq O_p\left(\frac{1}{r^{2(1-\delta)}h^{d+2}}\right) + O(h^4).$$

Consequently, when $h = O(r^{-\frac{2(1-\delta)}{6+d}})$, one has

$$\mathrm{AIMSE}(\widetilde{p}(\boldsymbol{z}|\mathbf{X}^*)) \leq O_p(r^{-\frac{8(1-\delta)}{6+d}}).$$

□

Appendix C

Proof for Chapter 3

C.1 Proof of an ancillary lemma

The proof follows Lemma B.2 directly.

C.2 Proof of Lemma 2

Under the regularity condition (4), for all μ and ν, the function $g_{\nu,\mu}(x) = \phi_\nu(x)\phi_\mu(x)$ has bounded total variation $\mathcal{V}(g_{\nu,\mu})$. By Theorem 1, one has

$$\left| \int_{[0,1]^d} \phi_\nu \phi_\mu \mathrm{d}x - \frac{1}{q} \sum_{j=1}^{q} \phi_\nu(x_j^*)\phi_\mu(x_j^*) \right| \leq D^*(\mathcal{X}_q^*)\mathcal{V}(g_{\nu,\mu}). \tag{C.1}$$

By Corollary 1, when $q = O(n^{1/d})$,

$$D^*(\mathcal{X}_q^*) = O_p\{q^{-(1-\delta)}\}. \tag{C.2}$$

Combining the results in Equation (C.1) and (C.2) together yields: when $q = O(n^{1/d})$, for all μ and ν,

$$\left| \int_{[0,1]^d} \phi_\nu \phi_\mu \mathrm{d}x - \frac{1}{q} \sum_{j=1}^{q} \phi_\nu(x_j^*)\phi_\mu(x_j^*) \right| = O_p\{q^{-(1-\delta)}\}.$$

C.3 Proof of Lemma S2

Recall that for basis selection method, we estimate the smoothing spline estimator in the effective model space $\mathcal{H}_E$. Let $\mathcal{H} \ominus \mathcal{H}_E$ be the orthogonal complement

of $\mathcal{H}_E$ in the reproducing kernel Hilbert space $\mathcal{H}$. We have the following lemma which justifies the use of the effective space $\mathcal{H}_E$.

Lemma C.3.1. *Under the regularity conditions* $(1)-(4)$, $q = O(n^{1/d})$, *as* $\lambda \to 0$ *and* $q^{1-\delta}\lambda^{1/r} \to \infty$, *where* δ *is an arbitrary small positive number,* $\forall h \in \mathcal{H} \ominus \mathcal{H}_E$, *one has* $V(h) = o_p\{\lambda J(h)\}$.

Before we prove the lemma, we first introduce an essential lemma as follows, refer to Lemma 9.1 of C. Gu, 2013 for details.

Lemma C.3.2. *Under Condition 2, as* $\lambda \to 0$, *one has*

$$\sum_{\nu} \frac{\lambda\rho_\nu}{(1+\lambda\rho_\nu)^2} = O(\lambda^{-1/r}),$$
$$\sum_{\nu} \frac{1}{(1+\lambda\rho_\nu)^2} = O(\lambda^{-1/r}),$$
$$\sum_{\nu} \frac{1}{1+\lambda\rho_\nu} = O(\lambda^{-1/r}).$$

of Lemma S2. For $h \in \mathcal{H} \ominus \mathcal{H}_E$, one has $h(x_j) = J(R_J(x_j,\cdot), h) = 0$ and $\sum_{j=1}^{q} h^2(x_j) = 0$. Write $h = \sum_\nu h_\nu\phi_\nu$, it follows that

$$\begin{aligned} V(h) &\le \int_{[0,1]^d} h^2 \mathrm{d}x \\ &= \sum_\nu \sum_\mu h_\nu h_\mu \int_{[0,1]^d} \phi_\nu\phi_\mu \mathrm{d}x \\ &= \sum_\nu \sum_\mu h_\nu h_\mu \left\{ \int_{[0,1]^d} \phi_\nu\phi_\mu \mathrm{d}x - \frac{1}{q}\sum_{j=1}^{q} \phi_\nu(x_j^*)\phi_\mu(x_j^*) \right\}. \quad \text{(C.3)} \end{aligned}$$

Denote $\delta_{\phi_\nu,\phi_\mu} = \int_{[0,1]^d} \phi_\nu\phi_\mu \mathrm{d}x - \frac{1}{q}\sum_{j=1}^{q} \phi_\nu(x_j^*)\phi_\mu(x_j^*)$. The Cauchy inequality yields.

$$\begin{aligned} \text{(C.3)} \le &\left\{ \sum_\nu \sum_\mu \frac{1}{1+\lambda\rho_\nu}\frac{1}{1+\lambda\rho_\mu} \left(\delta_{\phi_\nu,\phi_\mu}\right)^2 \right\}^{1/2} \\ &\times \left(\sum_\nu \sum_\mu (1+\lambda\rho_\nu)(1+\lambda\rho_\mu) h_\nu^2 h_\mu^2 \right)^{1/2}. \quad \text{(C.4)} \end{aligned}$$

By Lemma C.3.2, one has

$$\sum_\nu \sum_\mu \frac{1}{1+\lambda\rho_\nu}\frac{1}{1+\lambda\rho_\mu} = O(\lambda^{-2/r}), \quad \text{(C.5)}$$

and Lemma 2 shows, when $q = O(n^{1/d})$,

$$\left(\frac{1}{q}\sum_{j=1}^{q}\phi_\nu(x_j^*)\phi_\mu(x_j^*) - \int_{[0,1]^d}\phi_\nu\phi_\mu \mathrm{d}x\right)^2 = O_p\{q^{-2(1-\delta)}\}. \tag{C.6}$$

One also has

$$\sum_\nu (1+\lambda\rho_\nu)h_\nu^2 = (V+\lambda J)(h), \tag{C.7}$$

since ϕ_ν's simultaneously diagonalize V and J.

Combining the results in (C.4), (C.5), (C.6) and (C.7) together yields, when $q = O(n^{1/d})$,

$$V(h) \le O_p\{q^{-(1-\delta)}\lambda^{-1/r}\}(V+\lambda J)(h). \tag{C.8}$$

As a result, when $q = O(n^{1/d})$ and $q^{2-\delta}\lambda^{2/r} \to \infty$, the inequality (C.8) yields

$$V(h) = o_p\{\lambda J(h)\}.$$

□

C.4 Proof of Theorem 2

Compared with the condition from $q\lambda^{2/r} \to \infty$ in Theorem 9.17 in C. Gu, 2013 and Theorem 1 in Ma, Huang, et al., 2015, the condition for Theorem 2 is $q^{2-\delta}\lambda^{2/r} \to \infty$. As a result, the proposed smoothing spline estimator $\hat{\eta}_E$ can have the same convergence rate as the full-basis estimator with smaller q. Under this condition as well as the regularity conditions (1)-(4), as $\lambda \to 0$, $\forall h \in \mathcal{H} \ominus \mathcal{H}_E$, $V(h)$ is dominated by $\lambda J(h)$, which is guaranteed by Lemma S2. Theorem 2 thus can be proved by following the proof of Theorem 9.17 in C. Gu, 2013 directly.

C.5 Additional Simulation results

We now evaluate the performance of the proposed estimator on the cases when the regression function has several sharp peaks. The function we considered is a bivariate copula function, which is the same as the second function used in Ma, Huang, et al., 2015. The parameters are set as the same as the ones in Ma, Huang, et al., 2015. The regression function has several sharp peaks in

the domain of interest, as shown in the left panel of Fig. 4. We simulated the data with $n = \{2^8, \ldots, 2^{13}\}$ uniformly from $[-2, 2]^2$ and we set the number of basis functions q as $[40n^{1/9}]$, where $[\cdot]$ denotes the rounding function. The prediction error is measured by mean squared error (MSE), defined as $\{\sum_{i=1}^{n_0}(\hat{\eta}_E(t_i) - \eta_0(t_i))^2\}/n_0$, where $\{t_i\}_{i=1}^{n_0}$ is an independent testing dataset uniformly generated on $[-2, 2]^2$ with $n_0 = 5000$. The signal-to-noise ratio is set as 5.

The right panel of Fig. 4 shows the MSE against the sample size on the log-log scale. The vertical bars are standard error bars obtained from 50 replicates. Three significant observations can be made from the right panel of Fig. 4. First, both SBS (the proposed approach) and ABS (the approach in Ma, Huang, et al., 2015) uniformly outperform the naive UNIF approach. Second, neither SBS nor ABS dominates the other. Third, the MSE for the proposed SBS estimator decreases faster than the other two estimators as n increase. Such an observation is consistent with the simulation results in Section 5. In the meanwhile, this observation is also consistent with Theorem 2, which indicates the proposed approach shows merit as $n \to \infty$.

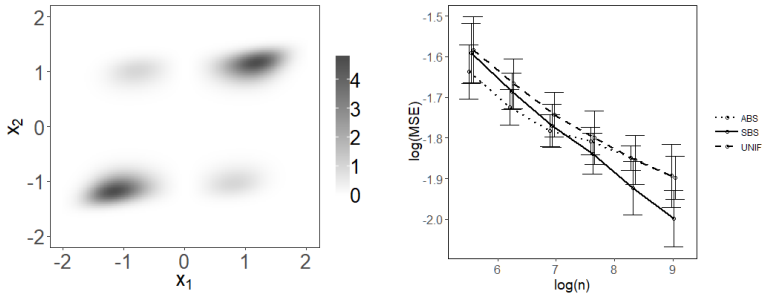

Figure C.1: Left panel: Contour plot of the true function; Right panel: the mean squared error versus the sample size for different estimators.

C.6 Additional details of real data analysis

We plot the heat map of the raw data in Fig. C.2, where some missing data can be observed.

To show the efficiency of the proposed SBS method, we compare it with the uniform basis selection method and the adaptive basis selection method, in terms of the mean squared fitting error (MSE). We set the number of basis functions $q = \{5n^{2/9}, 10n^{2/9}, 15n^{2/9}, 20n^{2/9}\}$, and the standard errors are calculated based on ten replicates for each method, respectively. Figure C.3

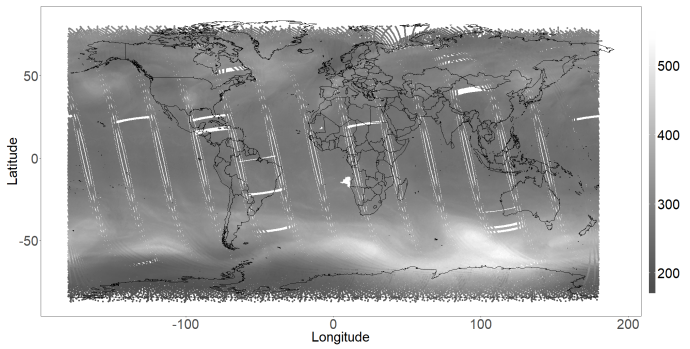

Figure C.2: Level 2 TCO data on October 1st, 1988, in Dobson units.

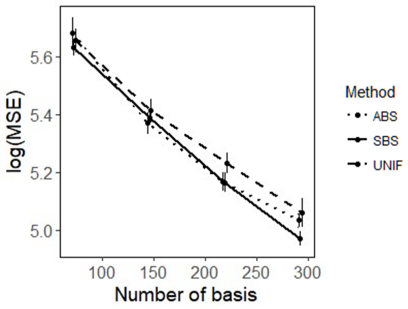

Figure C.3: The lines show the mean predict MSE versus the number of basis for the ozone data. The standard deviations based on ten replicates are shown as vertical bars.

shows the MSE in log scale. We observe that the proposed SBS method yields a smaller MSE.

Appendix D

Proof for Chapter 4

This appendix provides the proofs of the theoretical results for the main document.

D.1 Proof of Theorem 1

First, we presents some Lemmas to facilitate the proof of Theorem 1.

Let $(\widetilde{Z}, \widetilde{R})$ be an independent copy of (Z, R). We denote

$$A(R, \widetilde{R}) = E\left[(Z - \widetilde{Z})(Z - \widetilde{Z})^{\mathrm{T}} | R, \widetilde{R}\right]. \tag{D.1}$$

Let P be the projection onto the central space $\mathcal{S}_{R|Z}$ with respect to the inner project $a \cdot b = a^{\mathrm{T}} b$, and let $Q = I_d - P$. Further, define two quantities

$$C = 2I_d - A(R, \widetilde{R}) \quad \text{and} \quad G = E(C)^2.$$

Lemma D.1.1. *Denote* $\mathrm{span}(G)$ *the column space of matrix G, then* $\mathcal{S}_{SAVE} = span(G)$.

Proof of Lemma D.1.1. Follow the Theorem 2 in B. Li and Wang, 2007 and notice $E(ZZ^{\mathrm{T}}) = I_d$, the matrix G can be re-expressed as

$$\begin{aligned} G = \; & 2E\left[E^2(ZZ^{\mathrm{T}} - I_d|R)\right] + 2E^2\left[E(Z|R)E(Z^{\mathrm{T}}|R)\right] \\ & + 2E\left[E(Z^{\mathrm{T}}|R)E(Z|R)\right] E\left[E(Z|R)E(Z^{\mathrm{T}}|R)\right]. \end{aligned}$$

First, let v be a vector orthogonal to $\mathcal{S}_{\mathrm{SAVE}}$. We have $E(Z^{\mathrm{T}}|R)v = 0$ and $[I_d - \mathrm{Var}(Z|R)]v = 0$ almost surely. Therefore, $G_i v = 0$ for $i = 1, \ldots, 6$. This implies that v is orthogonal to $\mathrm{span}(G)$, and hence $\mathrm{span}(G) \subseteq \mathcal{S}_{\mathrm{SAVE}}$.

On the other hand, let v be a vector orthogonal to $\mathrm{span}(G)$. Then, $v^{\mathrm{T}}Gv = 0$ implies

$$v^{\mathrm{T}}E\left[E^2(ZZ^{\mathrm{T}} - I_d|R)\right]v = 0 \tag{D.2}$$

and

$$v^{\mathrm{T}}E\left[E(Z^{\mathrm{T}}|R)E(Z|R)\right]E\left[E(Z|R)E(Z^{\mathrm{T}}|R)\right]v = 0, \tag{D.3}$$

almost surely.

The second equality implies that $E(Z^{\mathrm{T}}|R) = 0$ almost surely. Furthermore, Using the fact that $E(ZZ^{\mathrm{T}}) = I_d$ and $E(ZZ^{\mathrm{T}}|R) = \mathrm{Var}(Z|R) + E(Z|R)E(Z^{\mathrm{T}}|Y)$, the first inequality can be re-expressed as

$$\begin{aligned} 0 = \quad & v^{\mathrm{T}}E\left[\mathrm{Var}(Z|R) - I_d\right]^2 v \\ + \quad & v^{\mathrm{T}}E\left[(\mathrm{Var}(Z|R) - I_p)E(Z|R)E(Z^{\mathrm{T}}|R)\right]v \\ + \quad & v^{\mathrm{T}}E\left[E(Z|R)E(Z^{\mathrm{T}}|R)(\mathrm{Var}(Z|R) - I_d)\right]v \\ + \quad & v^{\mathrm{T}}E\left[E(Z|R)E(Z^{\mathrm{T}}|R)\right]^2 v. \end{aligned}$$

The second to fourth terms are 0 since $E(Z^{\mathrm{T}}|R) = 0$. Thus the first term must also be 0, almost surely, implying that v is independent of $\mathcal{S}_{\mathrm{SAVE}}$. We complete the proof by showing that $\mathcal{S}_{\mathrm{SAVE}} \subseteq \mathrm{span}(G)$.

□

Lemma D.1.2. *Suppose the Assumption 1 (a) and (b) hold. Denote* $\mathrm{span}(G)$ *the column space of matrix G, then* $\mathcal{S}_{SAVE} = span(G)$.

Proof of Lemma D.1.2. By Lemma 2.1 of B. Li et al., 2005 and Propsition 4.6 of Cook, 2009, $(Z, R) \perp\!\!\!\perp (\widetilde{Z}, \widetilde{R})$ implies that $Z \perp\!\!\!\perp \widetilde{Z}(R, \widetilde{R})$, $Z \perp\!\!\!\perp \widetilde{R}|R$ and $\widetilde{Z} \perp\!\!\!\perp R|\widetilde{R}$. Thus $A(R, \widetilde{R})$ can be re-expressed as

$$\begin{aligned} A(R, \widetilde{R}) = \quad & E(ZZ^{\mathrm{T}}|R) - E(Z|R)E(\widetilde{Z}^{\mathrm{T}}|\widetilde{R}) \\ - \quad & E(\widetilde{Z}|\widetilde{R})E(Z^{\mathrm{T}}|R) + E(\widetilde{Z}\widetilde{Z}^{\mathrm{T}}|\widetilde{R})) \end{aligned} \tag{D.4}$$

Let v be a vector orthogonal to $\mathcal{S}_{R|W}$. By assumption (a), $E(v^{\mathrm{T}}Z|PZ) = \alpha^{\mathrm{T}}PZ$ for some $\alpha \in \mathbb{R}^d$. Multiply both sides by $ZP\alpha$ and then take unconditional expectation to obtain $v^{\mathrm{T}}P\alpha = \alpha^{\mathrm{T}}P\alpha = 0$. Thus $E(v^{\mathrm{T}}Z|PZ) = 0$.

By Assumption 1 (a) and (b), $E\left[(v^{\mathrm{T}}Z)^2|PZ\right] = c + E^2(v^{\mathrm{T}}Z|PZ) = c$, for some constant c. Take unconditional expectations on both sides to obtain $c = v^{\mathrm{T}}v$. Thus $E\left[(v^{\mathrm{T}}Z)^2|PZ\right] = v^{\mathrm{T}}v$.

Because $R \perp\!\!\!\perp Z|PZ$, we have

$$\begin{aligned} E(v^{\mathrm{T}}Z|R) \quad &= E\left[E(v^{\mathrm{T}}Z|PZ|R)\right] = 0, \\ E\left[(v^{\mathrm{T}}Z)^2|R\right] \quad &= E\left\{E\left[(v^{\mathrm{T}}Z)^2|PZ\right]|R\right\} = v^{\mathrm{T}}v. \end{aligned}$$

Substitute the above two lines into D.4, we have

$$v^{\mathrm{T}} A(R, \widetilde{R}) v = 2 v^{\mathrm{T}} v,$$

which implies $v^{\mathrm{T}} G v = 0$. Then, we have $\mathrm{span}(G) \subseteq \mathcal{S}_{R|W}$.

□

Lemma D.1.3. *Let G be a symmetric and positive semi-definite matrix which satisfies $\mathrm{span}(G) \subseteq \mathcal{S}_{R|W}$. Then, $\mathrm{span}(G) = \mathcal{S}_{R|W}$ iff $v^{\mathrm{T}} G v > 0$ for all $v \in \mathcal{S}_{R|W}$, $v \neq 0$.*

Proof of Lemma D.1.3. Suppose $\mathrm{span}(G)$ is a strict subspace of $\mathcal{S}_{R|W}$. Then $v^{\mathrm{T}} G v = 0$ for any $v \neq 0, v \in \mathcal{S}_{R|W} \ominus \mathrm{span}(G)$. Conversely, for $\mathrm{span}(G) = \mathcal{S}_{R|W}$, $v \in \mathcal{S}_{R|W}$, $v \neq 0$, we have $v \in \mathrm{span}(G)$, and hence $v^{\mathrm{T}} G v > 0$. □

Proof of Theorem 1. We first show that $\mathrm{span}(G) = \mathcal{S}_{R|W}$. G is symmetric and positive semi-definite according to its definition. Also, Lemma D.1.2 shows $\mathrm{span}(G) \subseteq \mathcal{S}_{R|W}$ under Assumption 1 (a) and (b).

Let $v \in \mathcal{S}_{R|W}$, $v \neq 0$. Without loss of generality, we assume $\|v\| = 1$. Then

$$v^{\mathrm{T}} G v = v^{\mathrm{T}} E\left[C(I_d - v v^{\mathrm{T}}) C\right] v + E\left[(v^{\mathrm{T}} C v)^2\right]. \tag{D.5}$$

Because $I_d - v v^{\mathrm{T}} \geq 0$, the first term on the right hand side of (D.5) is nonnegative. By Assumption 1 (c), $v^{\mathrm{T}} A(R, \widetilde{R}) v$ is non-degenerate. Therefore, $v^{\mathrm{T}} C v$ is non-degenerate. Then, by Jensen's inequality and notice $E(C) = 0$,

$$E\left[(v^{\mathrm{T}} C v)^2\right] > \left[E(v^{\mathrm{T}} C v)\right]^2 = 0. \tag{D.6}$$

Then, by Lemma D.1.1 and Lemma D.1.3, we complete the proof by showing $\mathcal{S}_{\mathrm{SAVE}} = \mathrm{span}(G) = \mathcal{S}_{R|W}$.

□

D.2 Proof of Theorem 2

Proof of Theorem 2. Suppose Assumption 2 holds. By applying Theorem 3 and Proposition 3, we arrive at

$$\begin{aligned} \|\widehat{\boldsymbol{\xi}}_1 - \boldsymbol{\xi}_1\|_\infty &\leq \max_{1 \leq l \leq r} \|\widehat{\boldsymbol{\xi}}_l - \boldsymbol{\xi}_l\|_\infty \\ &\leq C_1 d^{-3/2} (r^4 \|\widehat{\Sigma}_{\mathrm{SAVE}} - \Sigma_{\mathrm{SAVE}}\|_\infty + r^{3/2} \|\widehat{\Sigma}_{\mathrm{SAVE}} - \Sigma_{\mathrm{SAVE}}\|_2) \\ &\leq C_2 r^4 d^{-1/2} \|\widehat{\Sigma}_{\mathrm{SAVE}} - \Sigma_{\mathrm{SAVE}}\|_{\max}, \end{aligned} \tag{D.7}$$

where C_1 and C_2 are some positive constants.

It can be shown that

$$\begin{aligned}
&\widehat{\Sigma}_{\text{SAVE}} - \Sigma_{\text{SAVE}} \\
&= \tfrac{1}{4}\left[(\widehat{\Sigma}_1 - I_d)^2 - (\Sigma_1 - I_d)^2 + (\widehat{\Sigma}_2 - I_d)^2 - (\Sigma_2 - I_d)^2\right] \\
&= \tfrac{1}{4}\left[(\widehat{\Sigma}_1 + \Sigma_1 - 2I_d)(\widehat{\Sigma}_1 - \Sigma_1) + (\widehat{\Sigma}_2 + \Sigma_2 - 2I_d)(\widehat{\Sigma}_2 - \Sigma_2)\right]
\end{aligned}$$

Then,

$$\begin{aligned}
&\|\widehat{\Sigma}_{\text{SAVE}} - \Sigma_{\text{SAVE}}\|_{\max} \\
&\leq \tfrac{1}{4}[\,\|(\widehat{\Sigma}_1 + \Sigma_1 - 2I_d)(\widehat{\Sigma}_1 - \Sigma_1)\|_{\max} \\
&+\|(\widehat{\Sigma}_2 + \Sigma_2 - 2I_d)(\widehat{\Sigma}_2 - \Sigma_2)\|_{\max}] \\
&\leq \tfrac{1}{4}[\|\widehat{\Sigma}_1 + \Sigma_1 - 2I_d\|_2\|\widehat{\Sigma}_1 - \Sigma_1\|_{\max} \\
&+\|\widehat{\Sigma}_2 + \Sigma_2 - 2I_d\|_2\|\widehat{\Sigma}_2 - \Sigma_2\|_{\max}]
\end{aligned} \tag{D.8}$$

Follow the classic asymptotic result in univariate OLS and use the union bound, we have

$$\|\widehat{\Sigma}_1 - \Sigma_1\|_{\max} = O_p(\sqrt{\tfrac{\log d}{n}}); \tag{D.9}$$

$$\|\widehat{\Sigma}_2 - \Sigma_2\|_{\max} = O_p(\sqrt{\tfrac{\log d}{n}}). \tag{D.10}$$

Then, we bound the first operator norm in (D.8) as

$$\begin{aligned}
&\|\widehat{\Sigma}_1 + \Sigma_1 - 2I_d\|_2 \\
&= \|\widehat{\Sigma}_1 - \Sigma_1 + 2\Sigma_1 - 2I_d\|_2 \\
&\leq \|\widehat{\Sigma}_1 - \Sigma_1\|_2 + 2\|\Sigma_1 - I_d\|_2 \\
&\leq d\|\widehat{\Sigma}_1 - \Sigma_1\|_{\max} + 2\|\Sigma_1 - I_d\|_2 \\
&= O_p(\sqrt{\tfrac{d^2 \log d}{n}}) + O_p(\sqrt{d}),
\end{aligned} \tag{D.11}$$

where the second term of the last equality is due to $\|\Sigma_1\|_2 = O_p(\sqrt{d})$ derived from Assumption 2. Similarly, we have

$$\|\widehat{\Sigma}_2 + \Sigma_2 - 2I_d\|_2 = O_p(\sqrt{\frac{d^2 \log d}{n}} + \sqrt{d}). \tag{D.12}$$

By plugging (D.9), (D.11) and (D.12) back to (D.7), we conclude the proof by showing

$$\|\widehat{\boldsymbol{\xi}}_1 - \boldsymbol{\xi}_1\|_\infty = O_p(r^4\sqrt{\frac{\log d}{n}} + r^4\sqrt{d}\frac{\log d}{n}).$$

□

D.3 Proof of Theorem 3

We will work on the space of probability measures on $X \subset \mathbb{R}^d$ with bounded pth moment, i.e.

$$\mathcal{P}_p(X) \equiv \left\{\mu \in \mathcal{P}(X) : \inf_X |x|^p \mathrm{d}\mu(x) < \infty\right\}.$$

The following Lemma follows the Theorem 5.10 in Santambrogio, 2015, which provides the weak convergence in Wasserstein distance. Hence we omit its proof.

Lemma D.3.1. *Let $X \subset \mathbb{R}^d$ be compact, and $\mu_n, \mu \in \mathcal{P}(X)$. Then $\mu_n \to \mu$ if and only if $W_p(\mu_n, \mu) \to 0$.*

Denote $\widehat{W}_p^*(\mathcal{X}, \boldsymbol{y}) = \left(\frac{1}{n}\sum_{i=1}^n \|\boldsymbol{x}_i - \phi^*(\boldsymbol{x}_i)\|^p\right)^{1/p}$, the empirical Wasserstein distance with true OTM $\phi^*(\cdot)$. The following Lemma follows the Theorem 2.1 in Klein et al., 2017 guarantees that $\widehat{W}_p^*(\mathcal{X}, \boldsymbol{y})$ is a consistent estimator of $W_2(p_x, p_y)$. We refer to Klein et al., 2017 for its proof.

Lemma D.3.2. *Under Assumption 2 (a) and (b), $\widehat{W}_p^*(\mathcal{X}, \boldsymbol{y})$ converges almost surely to $W_2(p_x, p_y)$ as $n \to \infty$.*

Proof of Theorem 3. Notice that, we can decompose the empirical Wasserstein distance as

$$\begin{aligned}
&\widehat{W}_p\Big(\phi^{(K)}(\mathcal{X}), \mathcal{X}\Big) = \Big\{\widehat{W}_p\Big(\phi^{(K)}(\mathcal{X}), \mathcal{X}\Big) - W_p\Big(\phi^{(K)}(X), X\Big)\Big\} \\
&+ \Big\{W_p\Big(\phi^{(K)}(X), X\Big) - W_p\Big(\phi^*(X), X\Big)\Big\} + W_p\Big(\phi^*(X), X\Big) \\
\equiv &I_1 + I_2 + I_3.
\end{aligned}$$

First, under Assumption 2 (a) and (b) and with Lemma D.3.2, one can show that I_1 converges to 0 almost surely as $n \to \infty$.

For any $k \geq 0$, denote $\Delta^{[k]} = \mathcal{X}^{[k+1]} - \mathcal{X}^{[k]}$. Then, we have

$$\begin{aligned}\Delta^{[k]} &= (\phi^{(k)}(\mathcal{X}^{[k]}\boldsymbol{\xi}_k) - \mathcal{X}^{[k]}\boldsymbol{\xi}_k)\boldsymbol{\xi}_k^{\mathrm{T}} \\ &= (\boldsymbol{y}\boldsymbol{\xi}_k - \mathcal{X}^{[k]}\boldsymbol{\xi}_k)\boldsymbol{\xi}_k^{\mathrm{T}} \\ &= (\boldsymbol{y} - \mathcal{X}^{[k]})\boldsymbol{\xi}_k\boldsymbol{\xi}_k^{\mathrm{T}}, \end{aligned} \tag{D.13}$$

where the second inequality used the fact that $\phi^{(k)}(\cdot)$ is the OTM between $\mathcal{X}^{[k]}\boldsymbol{\xi}_k$ and $\boldsymbol{y}\boldsymbol{\xi}_k$.

Therefore, by taking the vector norm to both sides or (D.13), we have

$$\begin{aligned}\|\Delta^{[k]}\|_2 &= \|(\boldsymbol{y} - \mathcal{X}^{[k]})\boldsymbol{\xi}_k\boldsymbol{\xi}_k^{\mathrm{T}}\|_2 \\ &= \mathrm{Tr}\{\boldsymbol{\xi}_k^{\mathrm{T}}(\boldsymbol{y} - \mathcal{X}^{[k]})\boldsymbol{\xi}_k\} \\ &= \lambda_k^2\|\boldsymbol{y} - \mathcal{X}^{[k]}\|_2 \\ &= \lambda_k^2\|(\boldsymbol{y} - \mathcal{X}^{[k+1]}) + \Delta^{[k+1]}\|_2 \\ &\geq \lambda_k^2\left\{\|\boldsymbol{y} - \mathcal{X}^{[k+1]}\|_2 - \|\Delta^{[k+1]}\|_2\right\} \\ &\geq \lambda_k^2\left\{\lambda_{k+1}^{-2}\|\Delta^{[k+1]}\|_2\right\} = \frac{\lambda_k^2}{\lambda_{k+1}^2}\|\Delta^{[k+1]}\|_2. \end{aligned}$$

In other words, we have

$$\|\Delta^{[k+1]}\|_2 \leq \frac{\lambda_{k+1}^2}{\lambda_k^2}\|\Delta^{[k]}\|_2 \leq \frac{\lambda_{k+1}^2}{\lambda_0^2}\|\Delta^{[0]}\|_2, \quad \text{for} \quad k \geq 0.$$

According to Theorem 2, λ_k is a consistent estimator of the leading eigenvalue of Σ_{SAVE} in the kth iteration. Also, according to Theorem 1, λ_k is upper bounded by the kth eigenvalue of Σ, almost surely. Then, under Assumption 2 (c), we have λ_k/λ_1 converges to 0 as $d \to \infty$ and $k \geq Cd$ for some $C > 0$. This implies $\|\Delta^{[k+1]}\|_2 \to 0$ as $d \to \infty$ and $k \geq Cd$.

Then, Lemma D.3.1 guarantees that I_2 weakly converges to 0 as $d \to \infty$ and $k \geq Cd$ and hence completes our proof.

□

Bibliography

Alaoui, A., & Mahoney, M. W. (2015). Fast randomized kernel ridge regression with statistical guarantees, In *Advances in neural information processing systems*.

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, *46*(3), 175–185.

Andersen, R. (2008). *Modern methods for robust regression*. Sage.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks, In *International conference on machine learning*.

Arya, S., Mount, D. M., Netanyahu, N., Silverman, R., & Wu, A. Y. (1994). An optimal algorithm for approximate nearest neighbor searching in fixed dimensions, In *Proc. 5th acm-siam sympos. discrete algorithms*, ACM.

Asmussen, S., & Glynn, P. W. (2007). *Stochastic simulation: Algorithms and analysis* (Vol. 57). Springer Science & Business Media.

Benamou, J.-D., Brenier, Y., & Guittet, K. (2002). The monge–kantorovitch mass transfer and its computational fluid mechanics formulation. *International Journal for Numerical methods in fluids*, *40*(1-2), 21–30.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, *18*(9), 509–517.

Bertsekas, D. P. (1992). Auction algorithms for network flow problems: A tutorial introduction. *Computational optimization and applications*, *1*(1), 7–66.

Blaauw, M., & Bonada, J. (2016). Modeling and transforming speech using variational autoencoders., In *Interspeech*.

Boissard, E. Et al. (2011). Simple bounds for the convergence of empirical and occupation measures in 1-wasserstein distance. *Electronic Journal of Probability*, *16*, 2296–2333.

Boissard, E., & Le Gouic, T. (2014). On the mean speed of convergence of empirical and occupation measures in wasserstein distance, In *Annales de l'ihp probabilités et statistiques*.

Bonneel, N., Rabin, J., Peyré, G., & Pfister, H. (2015). Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, *51*(1), 22–45.

Box, G. E., & Draper, N. R. (1959). A basis for the selection of a response surface design. *Journal of the American Statistical Association*, *54*(287), 622–654.

Brenier, Y. (1997). A homogenized model for vortex sheets. *Archive for Rational Mechanics and Analysis*, *138*(4), 319–353.

Casella, G. (1985). Condition numbers and minimax ridge regression estimators. *Journal of the American Statistical Association*, *80*(391), 753–758.

Chen, C. P., & Zhang, C.-Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on big data. *Information Sciences*, *275*, 314–347.

Chung, K.-L. (1949). An estimate concerning the kolmogroff limit distribution. *Transactions of the American Mathematical Society*, *67*(1), 36–50.

Cioppa, T. M., & Lucas, T. W. (2007). Efficient nearly orthogonal and space-filling latin hypercubes. *Technometrics*, *49*(1), 45–55.

Claici, S., & Solomon, J. (2018). Wasserstein coresets for lipschitz costs. *arXiv preprint arXiv:1805.07412*.

Cochran, W. G. (2007). *Sampling techniques*. John Wiley & Sons.

Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1996). Active learning with statistical models. *Journal of artificial intelligence research*, *4*, 129–145.

Cook, R. D. (2009). *Regression graphics: Ideas for studying regressions through graphics* (Vol. 482). John Wiley & Sons.

Cook, R. D., & Weisberg, S. (1991). Sliced inverse regression for dimension reduction: Comment. *Journal of the American Statistical Association*, *86*(414), 328–332.

Courty, N., Flamary, R., Tuia, D., & Rakotomamonjy, A. (2017). Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, *39*(9), 1853–1865.

Cressie, N., & Johannesson, G. (2008). Fixed rank kriging for very large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *70*(1), 209–226.

Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport, In *Advances in neural information processing systems*.

Cuturi, M., & Doucet, A. (2014). Fast computation of wasserstein barycenters, In *International conference on machine learning*, Journal of Machine Learning Research.

Dasgupta, A., Drineas, P., Harb, B., Kumar, R., & Mahoney, M. W. (2009). Sampling algorithms and coresets for l_p regression. *SIAM Journal on Computing*, *38*(5), 2060–2078.

Dasgupta, S., & Hsu, D. (2008). Hierarchical sampling for active learning, In *Proceedings of the 25th international conference on machine learning*. ACM.

Derezinski, M., Warmuth, M. K., & Hsu, D. J. (2018). Leveraged volume sampling for linear regression, In *Advances in neural information processing systems*.

Devroye, L. (1986). Sample-based non-uniform random variate generation, In *Proceedings of the 18th conference on winter simulation*. ACM.

Dick, J., Kuo, F. Y., & Sloan, I. H. (2013). High-dimensional integration: The quasi-monte carlo way. *Acta Numerica*, *22*, 133–288.

Dietterich, T. G. (2002). Ensemble learning. *The handbook of brain theory and neural networks*, *2*, 110–125.

Dosovitskiy, A., & Brox, T. (2016). Generating images with perceptual similarity metrics based on deep networks, In *Advances in neural information processing systems*.

Drineas, P., Kannan, R., & Mahoney, M. W. (2006). Fast monte carlo algorithms for matrices i: Approximating matrix multiplication. *SIAM Journal on Computing*, *36*(1), 132–157.

Drineas, P., Magdon-Ismail, M., Mahoney, M. W., & Woodruff, D. P. (2012). Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, *13*(Dec), 3475–3506.

Drineas, P., Mahoney, M. W., Muthukrishnan, S., & Sarlós, T. (2011). Faster least squares approximation. *Numerische mathematik*, *117*(2), 219–249.

Duong, T. Et al. (2007). Ks: Kernel density estimation and kernel discriminant analysis for multivariate data in r. *Journal of Statistical Software*, *21*(7), 1–16.

Engel, J., Resnick, C., Roberts, A., Dieleman, S., Norouzi, M., Eck, D., & Simonyan, K. (2017). Neural audio synthesis of musical notes with wavenet autoencoders, In *Proceedings of the 34th international conference on machine learning-volume 70*. JMLR. org.

Fang, K.-T., Li, R., & Sudjianto, A. (2005). *Design and modeling for computer experiments*. CRC Press.

Fang, K.-T., Lin, D. K., Winker, P., & Zhang, Y. (2000). Uniform design: Theory and application. *Technometrics*, *42*(3), 237–248.

Fang, K.-T., Ma, C.-X., & Winker, P. (2002). Centered l2-discrepancy of random sampling and latin hypercube design, and construction of uniform designs. *Mathematics of Computation*, *71*(237), 275–296.

Fang, K.-T., & Wang, Y. (1993). *Number-theoretic methods in statistics* (Vol. 51). CRC Press.

Farr, T. G., Rosen, P. A., Caro, E., Crippen, R., Duren, R., Hensley, S., Kobrick, M., Paller, M., Rodriguez, E., Roth, L., Et al. (2007). The shuttle radar topography mission. *Reviews of geophysics*, *45*(2).

Feder, T., & Greene, D. (1988). Optimal algorithms for approximate clustering, In *Proceedings of the twentieth annual acm symposium on theory of computing*. ACM.

Feldman, D., Faulkner, M., & Krause, A. (2011). Scalable training of mixture models via coresets, In *Advances in neural information processing systems*.

Ferradans, S., Papadakis, N., Peyré, G., & Aujol, J.-F. (2014). Regularized discrete optimal transport. *SIAM Journal on Imaging Sciences*, *7*(3), 1853–1882.

Friedman, J. H., & Stuetzle, W. (1981). Projection pursuit regression. *Journal of the American statistical Association*, *76*(376), 817–823.

Genevay, A., Cuturi, M., Peyré, G., & Bach, F. (2016). Stochastic optimization for large-scale optimal transport, In *Advances in neural information processing systems*.

Gerber, S., & Maggioni, M. (2017). Multiscale strategies for computing optimal transport. *The Journal of Machine Learning Research*, *18*(1), 2440–2471.

Glasserman, P. (2013). *Monte carlo methods in financial engineering* (Vol. 53). Springer Science & Business Media.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets, In *Advances in neural information processing systems*.

Gottschlich, C., & Schuhmacher, D. (2014). The shortlist method for fast computation of the earth mover's distance and finding optimal solutions to transportation problems. *PloS one*, *9*(10), e110214.

Greengard, L., & Strain, J. (1991). The fast gauss transform. *SIAM Journal on Scientific and Statistical Computing*, *12*(1), 79–94.

Gu, C. (2013). *Smoothing spline anova models*. Springer Science & Business Media.

Gu, C., & Kim, Y.-J. (2002). Penalized likelihood regression: General formulation and efficient approximation. *Canadian Journal of Statistics*, *30*(4), 619–628.

Gu, X. D., & Yau, S.-T. (2008). *Computational conformal geometry*. International Press Somerville, MA.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of wasserstein gans, In *Advances in neural information processing systems*.

Guo, Y., An, D., Qi, X., Luo, Z., Yau, S.-T., Gu, X., Et al. (2019). Mode collapse and regularity of optimal transportation maps. *arXiv preprint arXiv:1902.02934*.

Guyon, I., Cawley, G., Dror, G., & Lemaire, V. (2011). Results of the active learning challenge, In *Active learning and experimental design workshop in conjunction with aistats 2010*.

Halton, J. H. (1960). On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, *2*(1), 84–90.

Hastie, T. (1996). Pseudosplines. *Journal of the Royal Statistical Society. Series B (Methodological)*, 379–396.

Hengl, T., Heuvelink, G. B., Kempen, B., Leenaars, J. G., Walsh, M. G., Shepherd, K. D., Sila, A., MacMillan, R. A., de Jesus, J. M., Tamene, L., Et al. (2015). Mapping soil properties of africa at 250 m resolution: Random forests significantly improve current predictions. *PloS one*, *10*(6).

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium, In *Advances in neural information processing systems*.

Ho, T. K., & Kleinberg, E. M. (1996). Building projectable classifiers of arbitrary complexity, In *Pattern recognition, 1996., proceedings of the 13th international conference on*. IEEE.

Horn, R. A., & Johnson, C. R. (1990). *Matrix analysis*. Cambridge University Press.

Hsu, C.-W., Chang, C.-C., Lin, C.-J., Et al. (2003). A practical guide to support vector classification.

Huang, S.-J., Jin, R., & Zhou, Z.-H. (2010). Active learning by querying informative and representative examples, In *Advances in neural information processing systems*.

Huber, P. J. (1985). Projection pursuit. *The annals of Statistics*, 435–475.

Huggins, J., Campbell, T., & Broderick, T. (2016). Coresets for scalable bayesian logistic regression, In *Advances in neural information processing systems*.

Hütter, J.-C., & Rigollet, P. (2019). Minimax rates of estimation for smooth optimal transport maps. *arXiv preprint arXiv:1905.05828*.

Ifarraguerri, A., & Chang, C.-I. (2000). Unsupervised hyperspectral image analysis with projection pursuit. *IEEE Transactions on Geoscience and Remote Sensing*, *38*(6), 2529–2538.

Joseph, V. R., Gul, E., & Ba, S. (2015). Maximum projection designs for computer experiments. *Biometrika*, *102*(2), 371–380.

Joseph, V. R., & Hung, Y. (2008). Orthogonal-maximin latin hypercube designs. *Statistica Sinica*, 171–186.

Kaufman, L., & Rousseeuw, P. (1987). *Clustering by means of medoids*. North-Holland.

Kiefer, J. (1975). Optimal design: Variation in structure and performance under change of criterion. *Biometrika*, *62*(2), 277–288.

Kim, Y.-J., & Gu, C. (2004). Smoothing spline gaussian regression: More scalable computation via efficient approximation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *66*(2), 337–356.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kleijnen, J. P. (2008). *Design and analysis of simulation experiments* (Vol. 20). Springer.

Klein, T., Fort, J.-C., & Berthet, P. (2017). Convergence of an estimator of the wasserstein distance between two continuous probability distributions.

Kolouri, S., Pope, P. E., Martin, C. E., & Rohde, G. K. (2018). Sliced-wasserstein autoencoder: An embarrassingly simple generative model. *arXiv preprint arXiv:1804.01947*.

Kriegel, H.-P., Zimek, A. Et al. (2008). Angle-based outlier detection in high-dimensional data, In *Proceedings of the 14th acm sigkdd international conference on knowledge discovery and data mining*. ACM.

Krogh, A., & Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning, In *Advances in neural information processing systems*.

Kuipers, L., & Niederreiter, H. (2012). *Uniform distribution of sequences*. Courier Corporation.

Kuncheva, L. I., & Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, *51*(2), 181–207.

Lee, D., & Gray, A. G. (2009). Fast high-dimensional kernel summations using the monte carlo multipole method, In *Advances in neural information processing systems*.

Lemieux, C. (2009). *Monte carlo and quasi-monte carlo sampling*. Springer, New York.

Leobacher, G., & Pillichshammer, F. (2014). *Introduction to quasi-monte carlo integration and applications*. Springer.

Li, B., & Wang, S. (2007). On directional regression for dimension reduction. *Journal of the American Statistical Association*, *102*(479), 997–1008.

Li, B., Zha, H., Chiaromonte, F., Et al. (2005). Contour regression: A general approach to dimension reduction. *The Annals of Statistics*, *33*(4), 1580–1616.

Li, D., Yang, K., & Wong, W. H. (2016). Density estimation via discrepancy based adaptive sequential partition, In *Advances in neural information processing systems*.

Li, K.-C. (1991). Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, *86*(414), 316–327.

Li, K.-C. (1992). On principal hessian directions for data visualization and dimension reduction: Another application of stein's lemma. *Journal of the American Statistical Association*, *87*(420), 1025–1039.

Li, N., Qardaji, W., & Su, D. (2012). On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy, In *Proceedings of the 7th acm symposium on information, computer and communications security*. ACM.

Liang, X., Lee, L., Dai, W., & Xing, E. P. (2017). Dual motion gan for future-flow embedded video prediction, In *Proceedings of the ieee international conference on computer vision*.

Lin, Y., & Zhang, H. H. (2006). Component selection and smoothing in multivariate nonparametric regression. *The Annals of Statistics*, *34*(5), 2272–2297.

Liu, Y., Qin, Z., Luo, Z., & Wang, H. (2017). Auto-painter: Cartoon image generation from sketch by using conditional generative adversarial networks. *arXiv preprint arXiv:1705.01908*.

Luenberger, D. G., Ye, Y. Et al. (1984). *Linear and nonlinear programming* (Vol. 2). Springer.

Luo, Z., & Wahba, G. (1997). Hybrid adaptive splines. *Journal of the American Statistical Association*, *92*(437), 107–116.

Ma, P., Huang, J. Z., & Zhang, N. (2015). Efficient computation of smoothing splines via adaptive basis sampling. *Biometrika*, *102*(3), 631–645.

Ma, P., Mahoney, M. W., & Yu, B. (2015). A statistical perspective on algorithmic leveraging. *The Journal of Machine Learning Research*, *16*(1), 861–911.

Ma, P., & Sun, X. (2015). Leveraging for big data regression. *Wiley Interdisciplinary Reviews: Computational Statistics*, *7*(1), 70–76.

Mahoney, M. W. Et al. (2011). Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning*, *3*(2), 123–224.

Mak, S., Joseph, V. R. Et al. (2018). Support points. *The Annals of Statistics*, *46*(6A), 2562–2592.

McKay, M. D., Beckman, R. J., & Conover, W. J. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, *42*(1), 55–61.

Meer, P., Mintz, D., Rosenfeld, A., & Kim, D. Y. (1991). Robust regression methods for computer vision: A review. *International journal of computer vision*, *6*(1), 59–70.

Meng, C., Ke, Y., Zhang, J., Zhang, M., Zhong, W., & Ma, P. (2019). Large-scale optimal transport map estimation using projection pursuit, In *Advances in neural information processing systems*.

Meng, C., Wang, Y., Zhang, X., Mandal, A., Ma, P., & Zhong, W. (2017). Effective statistical methods for big data analytics. *Handbook of Research on Applied Cybernetics and Systems Science*, 280.

Mérigot, Q. (2011). A multiscale approach to optimal transport, In *Computer graphics forum*. Wiley Online Library.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., & Leisch, F. (2015). E1071: Misc functions of the department of statistics, probability theory group (formerly: E1071), tu wien. r package version 1.6-7.

Moltchanov, D. (2012). Distance distributions in random networks. *Ad Hoc Networks*, *10*(6), 1146–1166.

Mosegaard, K., & Tarantola, A. (1995). Monte carlo sampling of solutions to inverse problems. *Journal of Geophysical Research: Solid Earth*, *100*(B7), 12431–12447.

Mussmann, S., & Liang, P. (2018). On the relationship between data efficiency and error for uncertainty sampling, In *International conference on machine learning*.

Niederreiter, H. (1992). *Random number generation and quasi-monte carlo methods*. SIAM.

Nissim, K., Raskhodnikova, S., & Smith, A. (2007). Smooth sensitivity and sampling in private data analysis, In *Proceedings of the thirty-ninth annual acm symposium on theory of computing*. ACM.

Owen, A. B. (2003). Quasi-monte carlo sampling. *Monte Carlo Ray Tracing: Siggraph*, *1*, 69–88.

Papadakis, N., Peyré, G., & Oudet, E. (2014). Optimal transport with proximal splitting. *SIAM Journal on Imaging Sciences*, *7*(1), 212–238.

Park, H.-S., & Jun, C.-H. (2009). A simple and fast algorithm for k-medoids clustering. *Expert systems with applications*, *36*(2), 3336–3341.

Park, J.-S. (1994). Optimal latin-hypercube designs for computer experiments. *Journal of statistical planning and inference*, *39*(1), 95–111.

Pele, O., & Werman, M. (2009). Fast and robust earth mover's distances, In *2009 ieee 12th international conference on computer vision*. IEEE.

Peyré, G., Cuturi, M. Et al. (2019). Computational optimal transport. *Foundations and Trends® in Machine Learning*, *11*(5-6), 355–607.

Phillips, J. M. (2013). ε-samples for kernels, In *Proceedings of the twenty-fourth annual acm-siam symposium on discrete algorithms*. SIAM.

Pitie, F., Kokaram, A. C., & Dahyot, R. (2005). N-dimensional probability density function transfer and its application to color transfer, In *Computer vision, 2005. iccv 2005. tenth ieee international conference on*. IEEE.

Pitié, F., Kokaram, A. C., & Dahyot, R. (2007). Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding*, *107*(1-2), 123–137.

Pukelsheim, F. (2006). *Optimal design of experiments* (Vol. 50). SIAM.

Rabin, J., Ferradans, S., & Papadakis, N. (2014). Adaptive color transfer with relaxed optimal transport, In *2014 ieee international conference on image processing (icip)*. IEEE.

Rabin, J., Peyré, G., Delon, J., & Bernot, M. (2011). Wasserstein barycenter and its application to texture mixing, In *International conference on scale space and variational methods in computer vision*. Springer.

Reich, S. (2013). A nonparametric ensemble transform method for bayesian inference. *SIAM Journal on Scientific Computing*, *35*(4), A2013–A2024.

Reinsch, C. H. (1967). Smoothing by spline functions. *Numerische mathematik*, *10*(3), 177–183.

Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review*, *33*(1-2), 1–39.

Rubner, Y., Guibas, L. J., & Tomasi, C. (1997). The earth mover's distance, multi-dimensional scaling, and color-based image retrieval, In *Proceedings of the arpa image understanding workshop*.

Ruppert, D. (2002). Selecting the number of knots for penalized splines. *Journal of computational and graphical statistics*, *11*(4), 735–757.

Ruppert, D., Wand, M. P., & Carroll, R. J. (2009). Semiparametric regression during 2003–2007. *Electronic journal of statistics*, *3*, 1193.

Sacks, J., & Ylvisaker, D. (1978). Linear estimation for approximately linear models. *The Annals of Statistics*, 1122–1137.

Salimans, T., Zhang, H., Radford, A., & Metaxas, D. (2018). Improving gans using optimal transport. *arXiv preprint arXiv:1803.05573*.

Santambrogio, F. (2015). Optimal transport for applied mathematicians. *Birkäuser, NY*, *55*, 58–63.

Schohn, G., & Cohn, D. (2000). Less is more: Active learning with support vector machines, In *Icml*. Citeseer.

Schuhmacher, D., Bähre, B., Gottschlich, C., Hartmann, V., Heinemann, F., & Schmitzer, B. (2019). *Transport: Computation of optimal transport plans and wasserstein distances* [R package version 0.12-1]. R package version 0.12-1. https://cran.r-project.org/package=transport

Scott, D. W. (2015). *Multivariate density estimation: Theory, practice, and visualization*. John Wiley & Sons.

Seguy, V., Damodaran, B. B., Flamary, R., Courty, N., Rolet, A., & Blondel, M. (2017). Large-scale optimal transport and mapping estimation. *arXiv preprint arXiv:1711.02283*.

Serra, P., & Krivobokova, T. (2017). Adaptive empirical bayesian smoothing splines. *Bayesian Analysis*, *12*(1), 219–238.

Settles, B. (2012). Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *6*(1), 1–114.

Shepherd, K. D., & Walsh, M. G. (2002). Development of reflectance spectral libraries for characterization of soil properties. *Soil science society of America journal*, *66*(3), 988–998.

Sobol, I. (1967). The distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, *7*(4), 86–112.

Solomon, J., Rustamov, R., Guibas, L., & Butscher, A. (2014). Earth mover's distances on discrete surfaces. *ACM Transactions on Graphics (TOG)*, *33*(4), 67.

Sriperumbudur, B. K., Fukumizu, K., Gretton, A., Schölkopf, B., Lanckriet, G. R., Et al. (2012). On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, *6*, 1550–1599.

Stein, M. (1987). Large sample properties of simulations using latin hypercube sampling. *Technometrics*, *29*(2), 143–151.

Steinberg, D. M., & Lin, D. K. (2006). A construction method for orthogonal latin hypercube designs. *Biometrika*, 279–288.

Su, Y. (2000). Asymptotically optimal representative points of bivariate random vectors. *Statistica Sinica*, 559–575.

Su, Z., Wang, Y., Shi, R., Zeng, W., Sun, J., Luo, F., & Gu, X. (2015). Optimal mass transport for shape matching and comparison. *IEEE transactions on pattern analysis and machine intelligence*, *37*(11), 2246–2259.

Tang, B. (1993). Orthogonal array-based latin hypercubes. *Journal of the American statistical association*, *88*(424), 1392–1397.

Thompson, S. K. (2012). Simple random sampling. *Sampling, Third Edition*, 9–37.

Trefethen, L. N., & Bau III, D. (1997). *Numerical linear algebra*. SIAM.

Tsai, C.-W., Lai, C.-F., Chao, H.-C., & Vasilakos, A. V. (2015). Big data analytics: A survey. *Journal of Big data*, *2*(1), 21.

Tsao, M., & Ling, X. (2012). Subsampling method for robust estimation of regression models. *Open Journal of Statistics*, *2*(3), 281.

Villani, C. (2008). *Optimal transport: Old and new* (Vol. 338). Springer Science & Business Media.

Vondrick, C., Pirsiavash, H., & Torralba, A. (2016). Generating videos with scene dynamics, In *Advances in neural information processing systems*.

Wahba, G. (1990). *Spline models for observational data*. SIAM.

Wahba, G., & Craven, P. (1978). Smoothing noisy data with spline functions. estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische Mathematik*, *31*, 377–404.

Wald, I., & Havran, V. (2006). On building fast *k*d-trees for ray tracing, and on doing that in o(n logn), In *Interactive ray tracing 2006, ieee symposium on*. IEEE.

Wang, H., Yang, M., & Stufken, J. (2018). Information-based optimal subdata selection for big data linear regression. *Journal of the American Statistical Association*, 1–13.

Wang, X., Shen, J., & Ruppert, D. (2011). On the asymptotics of penalized spline smoothing. *Electronic Journal of Statistics*, *5*, 1–17.

Wang, Y., Yu, A. W., & Singh, A. (2017). On computationally tractable selection of experiments in measurement-constrained regression models. *Journal of Machine Learning Research*, *18*(143), 1–41.

Weed, J., & Bach, F. (2017). Sharp asymptotic and finite-sample rates of convergence of empirical measures in wasserstein distance. *arXiv preprint arXiv:1707.00087*.

White, C. A., Johnson, B. G., Gill, P. M., & Head-Gordon, M. (1994). The continuous fast multipole method. *Chemical physics letters*, *230*(1-2), 8–16.

Wood, S. N. (2003). Thin plate regression splines. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *65*(1), 95–114.

Wu, C. J., & Hamada, M. S. (2011). *Experiments: Planning, analysis, and optimization* (Vol. 552). John Wiley & Sons.

Yang, C., Duraiswami, R., Gumerov, N. A., & Davis, L. (2003). Improved fast gauss transform and efficient kernel density estimation, In *Null*. IEEE.

Ye, K. Q. (1998). Orthogonal column latin hypercubes and their application in computer experiments. *Journal of the American Statistical Association*, *93*(444), 1430–1439.

Zhang, H. H., Wahba, G., Lin, Y., Voelker, M., Ferris, M., Klein, R., & Klein, B. (2004). Variable selection and model building via likelihood basis pursuit. *Journal of the American Statistical Association*, *99*(467), 659–672.

Zhang, X., Xie, R., & Ma, P. (2018). Statistical leveraging methods in big data, In *Handbook of big data analytics*. Springer.

Zheng, Y., Jestes, J., Phillips, J. M., & Li, F. (2013). Quality and efficiency for kernel density estimates in large data, In *Proceedings of the 2013 acm sigmod international conference on management of data*. ACM.

Zheng, Y., Ou, Y., Lex, A., & Phillips, J. M. (2017). Visualization of big spatial data using coresets for kernel density estimates, In *2017 ieee visualization in data science (vds)*. IEEE.

Zhou, L., Pan, S., Wang, J., & Vasilakos, A. V. (2017). Machine learning on big data: Opportunities and challenges. *Neurocomputing*, *237*, 350–361.

Zhu, X., Lafferty, J., & Rosenfeld, R. (2005). *Semi-supervised learning with graphs* (Doctoral dissertation). Carnegie Mellon University, language technologies institute, school of computer science.

Zimek, A., Schubert, E., & Kriegel, H.-P. (2012). A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *5*(5), 363–387.