Single Index Model for Tensor Data: Theory and Application

by

RUI WANG

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
December 11, 2021

Keywords: Tensor Regression, Single-Index Model, Nuclear Norm

Approved by

Peng Zeng, Associate Professor
Ash Abebe, Professor
Nedret Billor, Professor
Guanqun (Vivian) Cao, Associate Professor

Abstract

Modern scientific applications are frequently producing data sets where the data are not in the form of vectors but instead higher order tensors. For instance, multi-channel MEG signals in biomedical engineering, gene expression data in bioinformatics and so on. In this dissertation, we combine the semi-parametric model (single index model) with nuclear norm regularization to fit the data with order-2 tensor (matrix). An efficient estimation algorithm is developed. Furthermore, we proved that this algorithm has a good asymptotic property that the estimator of the true parameter B is root-n consistent. In addition to theoretical results, we demonstrate the efficiency of the new method through simulation. One real data set is analyzed by this new method and traditional logistic regression, then the results show that the performance of the new proposed method is better than the performance of logistic regression.

Acknowledgments

First and foremost, I would like to express my sincerest gratitude to my advisor, Professor Peng Zeng for his careful guidance, incredible patience and priceless advice throughout my entire Ph.D time. This dissertation would not have been completed without his endless support and constant encouragement and caring. Prof. Zeng has taught me so much and has been incredibly kind. I appreciate all his contributions of time and ideas to make my Ph.D study productive and stimulating. Prof. Zeng's wide knowledge, brilliant insight and perpetual enthusiasm in statistics have greatly motivated me. I feel very fortunate to be his Ph.D student and I am forever grateful!

I would also like to sincerely thank Professor Ash Abebe, Professor Nedret Billor and Professor Guanqun Cao for serving on my Ph.D committee and providing constant help and guidance along the way. I would also like to sincerely thank Professor Guofu Niu for being the university reader.

My sincere thanks also go to all my friends, for always offering me help and support in my study and life in Auburn. Last but not least, I would like to thank my parents for their unconditional love and support. Finally, I am immensely grateful for everyone who has been a part of my educational experience throughout the years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Tensor Data

Tensor is a type of data structure used in matrix theory/linear algebra. In the physics and traditional mathematics field, the definition of tensor is that a tensor is a generalized concept of scalars (that have no indices), vectors (that have exactly one index), and matrices (that have exactly two indices) to an arbitrary number of indices. For example, a scalar is an order 0 tensor, a vector is an order 1 tensor, a 2D-matrix is an order 2 tensor, and so on. We can create a higher order tensor if we add more dimensions. Kolda and Bader (2009) gave a formal definition of tensor that is:

Definition 1.1.1 (Tensor)**.** An N-way or Nth-order tensor is an element of the tensor product of N vector spaces, each of which has its own coordinate system.

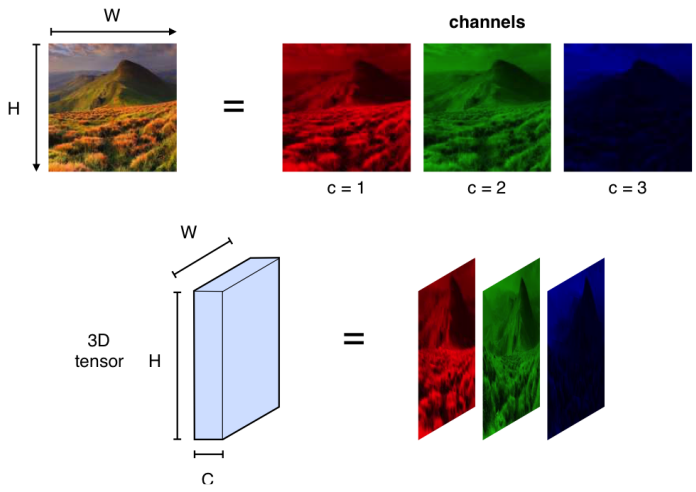


Figure 1.1: A color photo is order-3 tensor data

With the advances of data collection and storage capabilities, higher order tensor data are being generated on a daily basis in a wide range of emerging applications, for instance, order 2 tensor data include gray-level images in computer vision and pattern recognition (Yan et al., 2006; Lu et al., 2003), multichannel EEG signals in biomedical engineering (Li et al., 2008), etc. Order 3 tensor data include 3D objects in generic object recognition (Sahambi and Khorasani, 2003), hyperspectral cube in remote sensing (Renard and Bourennane, 2009), and gray-level video sequences in activity or gesture recognition for surveillance or human-computer interaction (Chellappa et al., 2005; Green and Guan, 2004). In Figure 1.1, a color image is also an order 3 tensor data, because each color image has three 2D arrays, blue, green and red.

The algorithms for extracting information from these data become more and more important. For example, the field of computer vision, concerning machines being able to understand images and videos, is one of the hottest topics in the tech industry. There is one popular database, MNIST database (Modified National Institute of Standards and Technology database), treating images (order-2 tensor data) as independent variable, X. This database contains 60,000 training image cases and 10,000 testing image cases. In each case, the size of independent variable data is a 28×28 image, which is a grayscale handwritten digit and the image is a matrix in which each value represents a pixel and describes the intensity in this pixel. For example, 0 represents the darkest and 255 represents the brightest. In Figure 1.2, there are some images about handwritten digits in the database. We need to find an algorithm that the computer can recognize the handwritten digits in these images correctly.

Now our problem is a sequence of $\{(y_i, X_i),\ i = 1, \cdots, n\}$, in which $y_i \in \mathbb{R}$ is a scalar and $X_i \in \mathbb{R}^{p \times q}$ is an order 2 tensor. There is a relationship between y and X

$$y = m(X) + \epsilon$$

where function $m()$ is an unknown smooth function and ϵ is the random error that is not relate to X, we need to develop some methods to fit the model.

Figure 1.2: The Images of Handwritten Digits in the MNIST

1.2 Linear Matrix Regression

Zhou and Li (2014) used the easiest and simplest way, linear matrix regression, to fit the model. We consider the model

$$y = \langle X, B \rangle + \epsilon$$

where B is the coefficient matrix of the same size as X. The inner product between two matrices is defined as $\langle B, X \rangle = \langle vecB, vecX \rangle = \sum_{r,c} \beta_{rc} x_{rc}$, where $vec(\cdot)$ is the vectorization operator that stacks the columns of a matrix into a vector, β_{rc} means the element in the matrix B whose row index is r and column index is c. ϵ represents the random error whose conditional expectation is 0 and variance is a constant, which means $E(\epsilon|X) = 0$ and $Var(\epsilon) = \sigma^2$.

Then, we get the estimator of B by solving the optimization problem

$$\min_B \frac{1}{n} \sum_{i=1}^{n} (y_i - \langle B, X_i \rangle)^2 \tag{1.1}$$

1.3 Regularization

During the estimation process of B, we need to avoid over-fitting and improve model interpretability. A regularization term is generally imposed upon the equation (1.1). Regularization can regularize or shrink the coefficient estimates towards zero. Regularization term is added to equation (1.1).

$$\min_B \frac{1}{n}\sum_{i=1}^{n}(y_i - \langle B, X_i\rangle)^2 + \lambda R(B)$$

in which $R(B)$ is the regularization function, the λ is the regularization parameter, which is a hyper-parameter that controls how severe the regularization term is. The value of λ varies from 0 to ∞. When $\lambda = 0$, the regularization term does not affect the loss function. As the λ gets larger, the impact of the shrinkage penalty grows, more parameters tend to zero in order to avoid the model over-fitting.

There are several commonly used regularizations.

- Power family (Frank and Friedman, 2002)

$$R_{\eta,\lambda}(w) = \lambda\|w\|_\eta,$$

where w is a vector $w \in \mathbb{R}^p$, $\|\cdot\|_\eta$ means the η-norm of w. There are two important special cases for this family, namely the lasso penalty when $\eta = 1$ (Tibshirani, 1996; Chen et al., 2001), and the ridge penalty when $\eta = 2$ (Hoerl and Kennard, 1970).

- l_1-norm regularization: It is $\|w\|_1 = \sum_{i=1}^{p}|w_i|$. l_1-norm regularization limits the size of the coefficients, some coefficients can become zero and eliminated. l_1-norm regularization encourages the parameter space to be sparse. l_1-regularization can reduce the curse of dimensionality problem, multi-collinearity problem.

- l_2-norm regularization: It is $\|w\|_2 = \sum_{i=1}^{p} w_i^2$. l_2-norm regularization encourages to keep all parameters instead of a subset of it. It will not yield sparse models and all coefficients are shrunk. Figure 1.3 is plotted by Hastie et al. (2009), which shows the constraint

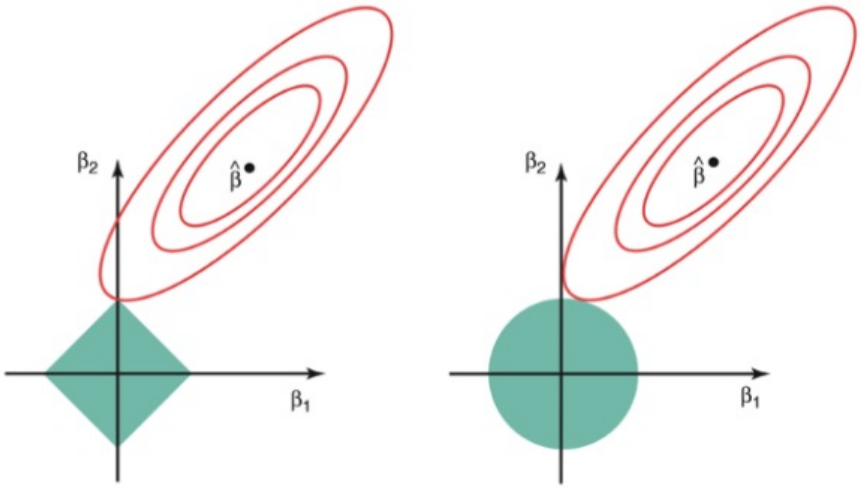


Figure 1.3: The difference between l_1-norm and l_2-norm

function, for l_1-norm (left) and l_2-norm (right), along with contours for residual sum of squares (RSS). It is easy to find that the l_2-norm has a round feasible region without sharp corners. If RSS and feasible area have intersection points that are not on the axis, it means that the l_2-norm coefficient estimates will be non-zero. However, the l_1-norm feasible region has corners, then the intersections points between RSS and constraint area will be likely on the axis. If this situation happens, the coefficients will equal zero.

- Elastic net (Zou and Hastie, 2005)

$$R_{\eta,\lambda}(w) = \lambda[(1-\eta)\|w\|_2 + \eta\|w\|_1]$$

The elastic net, which combines l_1 and l_2 norm regularization methods, simultaneously does automatic variable selection and continuous shrinkage, and it can select groups of correlated variables. Varying η from 0 to 1 bridges the lasso to the ridge penalty functions. l_1-norm regularization (LASSO penalty) and l_2-norm regularization (Ridge penalty) both have some disadvantages. For example, l_2-norm is a little difficult to be interpreted, because the final model will include all predictors. The disadvantage of l_1-norm is that it can not do group selection. If there is a group of variables among which

the pairwise correlations are very high, then the l_1-norm tends to arbitrarily select only one variable from the group.

- Log penalty (Armagan et al., 2011)

$$R_{\eta,\lambda}(w) = \lambda ln(\eta + |w|)$$

Armagan et al. (2011) proposed this penalty function and proved this penalty has three advantages:

1. Nearly unbiased when the true unknown parameter is large
2. A thresholding rule, which automatically sets small estimated coefficients to zero to reduce the model complexity
3. Continuous in data to avoid instability in model prediction.

- SCAD (Fan and Li, 2001a), in which the penalty is defined via its partial derivative

$$\frac{\partial}{\partial w} R_{\eta,\lambda}(w) = \lambda\{1_{\{|w|\leq\lambda\}} + \frac{(\eta\lambda - |w|)_+}{(\eta-1)\lambda} 1_{\{|w|>\lambda\}}\}$$

With $\eta > 2$, the penalty corresponds to a quadratic spline function with knots at λ and $\eta\lambda$. Explicitly, the penalty is

$$R_{\lambda,\eta}(w) = \begin{cases} \lambda|w|, & \text{if } |w| \leq \lambda \\ \frac{2\eta\lambda|w| - w^2 - \lambda^2}{2(\eta-1)}, & \text{if } \lambda < |w| \leq a\lambda \\ \frac{\lambda^2(\eta+1)}{2}, & \text{otherwise} \end{cases}$$

These penalty functions are fit for the vector covariates. For linear matrix regressions, a direct approach is to first vectorize the matrix covariates then apply the classical penalization function. But if we did it, there would be a severe problem caused. Vectorization destroys the wealth of structural information inherently possessed in the matrix.

We want to explore for models that incorporate the structural information of matrices. Low-rank is a property that we want to consider and it will lead to a parsimonious model. The

low-rank matrix is that matrices have fewer degrees of freedom than its ambient dimensions $p \times q$. If the rank of the matrix is r, $r \leq min(p, q)$.

We need to find the penalty function that fits for the matrix covariate. Zhou and Li (2014) suggests the regularization function to be $R(B) = f \circ \delta(B)$, where $f : \mathbb{R}^q \to \mathbb{R}$ is a function of the singular values of B. They choose $f(w) = \lambda \sum_{i=1}^{q} |w_i|$, which is the l_1 norm of the singular values of B, as the regularization function. $f(w) = \lambda \sum_{i=1}^{q} |w_i|$ corresponds to the nuclear norm regularization of B, $\|B\|_*$. The nuclear norm of B is $\sum_{i=1} \delta_i$ where δ_i's are the singular values of the matrix B. Zhou and Li (2014) listed two reasons why the nuclear norm $\|B\|_*$ is a suitable measure of the size of a matrix. The first one is that the nuclear norm a convex relaxation of rank(B)=$\|\delta(B)\|_0$. The second reason is that nuclear norm is analogous to the l_1 norm for a vector, because the l_1 ball in high dimensions is extremely "pointy" – the extreme values of a linear function on this ball are very likely to be attained on the faces of low dimensions, those that consist of sparse vectors. When applied to matrices, the sparseness of the set of singular values means low rank. Furthermore some researches (Luo et al., 2015; Chen et al., 2015; Qian et al., 2015; Yang et al., 2014) find that if the nuclear norm is used as a criterion, it can measure the low-rank structural information. Because the nuclear norm only sets the singular values whose absoluted values are small to zero, but not change the eigenvectors corresponding to these zero singular values. For this reason, the nuclear norm can keep the important information (eigenvalues whose absoluted values are large) of this matrix. Because of these reasons, we select the nuclear norm as the regularization term and add it to function (1.1).

$$\min_B \frac{1}{n} \sum_{i=1}^{n} (y_i - \langle X_i, B \rangle)^2 + \lambda \|B\|_* \tag{1.2}$$

1.4 Single Index Model

There is at least one disadvantage about linear matrix regression, that is, the linear matrix regression assumes a linear relationship between dependent variables and predictors. It means

that it assumes that there is a straight-line relationship between them. It may be a misspecification. Because there are many other relationships between X and y beside linear. For this reason, single index model can be considered.

$$y = g(\langle X, B \rangle) + \epsilon \tag{1.3}$$

in which the item contained in the function $g()$ is $\langle X, B \rangle$. $X \in \mathbb{R}^{r \times c}$ is a matrix of explanatory variables, $B \in \mathbb{R}^{r \times c}$ is the matrix of regression coefficients and $\|B\|_F = 1$, $\| \cdot \|_F$ represents the Frobenius norm, $\|B\|_F = \sqrt{\langle B, B \rangle} = \sqrt{\sum_i \sum_j b_{ij}^2}$, where b_{ij} represents each element in matrix B. The Frobenius norm is differentiable with respect to the individual entries of B, and Frobenius norm is induced by a natural matrix inner product. $y \in \mathbb{R}$ is the response variable, $\epsilon \in \mathbb{R}$ is the noise that $E(\epsilon|X) = 0$ almost surely. Let $g : \mathbb{R}^1 \to \mathbb{R}^1$ is an unknown smooth function. The single index model has many advantages, one of them is that it mitigates the risk of misspecifying the link function, Horowitz and Härdle (1996) have shown that misleading results are obtained if a binary probit model is estimated by specifying the cumulative normal distribution function as the link function rather than estimating $g(\cdot)$ by nonparametric methods. The other advantages are listed in Horowitz (2012), including the ability to overcome the curse of dimensionality. It is known that estimating the regression function is especially difficult whenever the dimension p of explanatory vector variable $vec(\mathbf{X})$ in the regression function becomes large. The convergence rate of the estimation of a k-times differentiable regression function's optimal mean square is $n^{-2k/(2k+p)}$, which will converge to zero dramatically slowly if the dimension p is large compared to k. In single index model, Gaïffas et al. (2007) proved that the optimal rate of convergence over the single-index model class is $n^{-2k/(2k+1)}$.

In the single index model, our priority target is estimating the parameter B matrix. Currently there is no method that can be used to estimate matrix parameter. But there are several popular methods that can be used to estimate the vectorized matrix B, $vec(B)$. Such as the average derivative estimation (ADE) method (Härdle and Stoker, 1989), the minimum average conditional variance estimation (MAVE) method (Xia et al., 2002) and the sliced inverse regression (Li, 1991). The majority of these methods can estimate the true parameter $vecB$ at the

root-n rate of convergence, but they are different from each other in terms of estimation and computation efficiencies (Xia et al., 2002). Xia and Tong (2006) showed that MAVE can be more advantageous than those in the other two categories in terms of estimation efficiency.

1.4.1 Average Derivative Method

This average derivative method was introduced in Powell et al. (1989). This approach is to estimate a specific set of coefficients, $vecB$, termed average derivatives. This method is based on the fact that the derivative of conditional expectation is proportional to $vecB$:

$$\frac{\partial E(y|X)}{\partial X} = \bigtriangledown m(X) = g^{'}(\langle X, B\rangle)vecB$$

Thus any weighted average of the derivatives $\bigtriangledown m(X)$ will also be proportional to $vecB$. Then a natural estimator for $vecB$ is $\widehat{vecB} = N^{-1}\sum_{i=1}^{N}\widehat{\bigtriangledown m(X_i)}/\|N^{-1}\sum_{i=1}^{N}\widehat{\bigtriangledown m(X_i)}\|$ with $\|\cdot\|$ being the Euclidean norm.

The advantage of the ADE approach is that ADE allows estimating $vecB$ directly. Xia et al. (2002) mentioned the limitations of ADE. To estimate $vecB$, the condition $E(g^{'}(vecX^{T}vecB)) \neq 0$ is needed. This condition is violated when $g(\cdot)$ is an even function and X is symmetrically distributed. As far as we know, there is no successful extension to the case of more than one EDR direction. The high-dimensional kernel smoothing used for computing $\widehat{\bigtriangledown m(X)}$ suffers from the *curse of dimensionality* if the model dimension is large.

1.4.2 Minimum Average Conditional Variance (MAVE)

It is proposed by Xia et al. (2002). Consider the model (1.3). By the local linear smoothing techniques, the MAVE method estimate $vecB$ by solving the following minimization problem

$$\min_{vecB, a_j, b_j, j=1,\cdots,n} n^{-2}\sum_{i=1}^{n}\sum_{j=1}^{n}[y_i - (a_j + b_j vecB^{T}vec(X_i - X_j))]^2 w_{ij} \tag{1.4}$$

with respect to $a_j \in \mathbb{R}$, $b_j \in \mathbb{R}$ and $vecB^T vecB = 1$. $w_{ij} \geq 0$ are some weights with $\sum_{i=1}^{n} w_{ij} = 1$. MAVE minimizes (1.4) with respect to $(a_j, b_j), j = 1, \cdots, n$ and $vecB$ iteratively, with an explicit solution of each optimization.

Xia et al. (2002) mentioned the advantages of MAVE. A faster consistency rate can be achieved by the MAVE even without undersmoothing the nonparametric link function estimator. The MAVE method is applicable to a wide range of models, with fewer restrictions on the distribution of the covariates.

1.4.3 Sliced Inverse Regression (SIR)

This method is introduced by Li (1991). The name of SIR comes from computing the inverse regression (IR) curve. Which means that instead of working on $E(Y|X = x)$, we investigate $E(X|Y = y)$. There are a sequence of $(X_i, y_i),\ i = 1, \cdots, n$. SIR method estimates the so-called sufficient dimension reduction directions $\beta_1, \cdots, \beta_d$. The corresponding algorithm is outlined as follows.

1. Divide the range of the y_i into H disjoint intervals, denoted as $S_1, \cdots, S_H$;
2. Compute for $h = 1, \cdots, H, \bar{X}_h = H^{-1} \sum_{y_i \in S_h} X_i$, where n_h is the number of $y_i's$ in S_h.
3. Estimate $Cov(E(X|y))$ by $\widehat{M} = H^{-1} \sum_{h=1}^{H} n_h (\bar{X}_h - \bar{X})(\bar{X}_h - \bar{X})^T$ and $Cov(X)$ by the sample covariance matrix $\widehat{\Sigma}$.
4. SIR uses the first K eigenvectors of $\widehat{\Sigma}^{-1}\widehat{M}$ to estimate the SDR directions, where K is an estimate of d based on the data.

The limitation of SIR is that SIR need a distributional assumption on predictors, which may not be satisfied in practice.

Currently, the single index model is a very useful semiparametric regression model. Compared to linear matrix regression, the single index model works better in the situations where the linear regression model may not perform well.

1.5 Contribution

In this dissertation, the data is not vector any more, the parameter is a matrix B and our primary purpose is to estimate the parameter matrix B of the single index model. Although these forementioned methods can still be used to estimate the coefficient matrix B by vectorizing B to $vec(B)$, Zhou and Li (2014) stated some reasons why those classical penalty functions do not incorporate the matrix structural information. One is that sparsity is in terms of the rank of the matrix parameters and it is different from sparsity in the number of nonzero entries. The other is if the matrix is vectorized, some important information about the matrix is vanished,for example eigenvalues, eigenvectors, the structure and so on.

These limits may restrict the effectiveness of these regression methods in practical application. Therefore a new single index model is needed to estimate the matrix parameter. A penalty function that is different from l_1 and l_2 norm is also needed to add to the new single index model method, because the penalty function is very important for estimating the parameter matrix B as mentioned above, but l_1-norm and l_2-norm fit for the parameter vector θ, not for the order 2 tensor B.

The contribution of this dissertation is that we develop a new estimation approach for single index model with order 2 tensor data as covariates. And in the loss function of (1.3), we add a nuclear norm as penalty function. The new method is called as single index model with nuclear norm, or SIM-nuclear. We proposed an algorithm by combing a fast iterative shrinkage-thresholding (FISTA) algorithm with Nesterov algorithm to estimate the matrix parameter B in the SIM-nuclear. We proved that the new method has good property that it can estimate B at the rate of root-n. Furthermore we also discuss how to select the hyperparameters, such as λ and bandwidth h and kernel function in order to get the optimal estimator $\hat{B}$.

Chapter 2

Single Index Model with Nuclear Norm Penalty

2.1 Formulation

In this chapter we will introduce the new single index model mentioned at the end of the previous chapter and the algorithm can be used to solve this new model. Our priority task is to estimate the matrix parameter B by solving the loss equation of (1.3)

$$\min_{B,\|B\|_F=1} \frac{1}{n}\sum_{i=1}^{n}(y_i - g(\langle X_i, B\rangle))^2 + \lambda\|B\|_*$$

in this function, both $g(\cdot)$ and B are unknown. We propose to use the local linear approximation idea.

$$g(u) \approx g(v) + g^{'}(v)(u-v) = a + b(u-v)$$

for v in a neighborhood of u, where $a = g(v)$ and $b = g^{'}(v)$ are local constants. Then the estimate of B is obtained by solving the minimization problem,

$$\min_{a,b,B,\|B\|_F=1} \frac{1}{2n}\sum_{j=1}^{n}\sum_{i=1}^{n}[y_i - a_j - b_j\langle B, X_i - X_j\rangle]^2 w_{ij} + \lambda\|B\|_* \tag{2.1}$$

where $a = (a_1, a_2, \cdots, a_n)^T$, $b = (b_1, b_2, \cdots, b_n)^T$ and $w_{ij}(\cdot)$ is the weight function that $w_{ij}(\cdot) \geq 0$, $\sum_{i=1}^{n} w_{ij}(\cdot) = 1$ and

$$w_{ij} = K_h(\langle B, x_i - x_j\rangle) / \sum_{l=1}^{n} K_h(\langle B, x_l - x_j\rangle) \tag{2.2}$$

where $K_h = h^{-d}K(\cdot/h)$ and d is the dimension of $K(\cdot)$. The equation (2.1) involves two summations. The inner summation is

$$\sum_{i=1}^{n}[y_i - a_j - b_j\langle B, X_i - X_j\rangle]^2 w_{ij} \tag{2.3}$$

which is the loss function for the local linear smoothing at X_j and B. a_j and b_j can be estimated by minimizing the equation (2.3). Summing (2.3) over all the X_j and adding the nuclear norm about B as the penalty function leads to the following penalized minimization problem:

$$\min_{a,b,B,\|B\|_F=1} \frac{1}{2n}\sum_{j=1}^{n}\sum_{i=1}^{n}[y_i - a_j - b_j\langle B, X_i - X_j\rangle]^2 w_{ij} + \lambda\|B\|_* \tag{2.4}$$

We refer to (2.4) as the SIM-nuclear minimization problem. Next we will give the asymptotic properties about SIM-nuclear method.

2.2 Asymptotic properties

We will show the asymptotic properties about the single index model for tensor data in this section. We assume the true parameter matrix B is unknown. The value of B is not given but there exists an initial estimator of B with root-n rate. Actually if average derivative estimator(ADE) is applied, its estimator is root-n consistent. We can use ADE to get an initial estimator named $\hat{B}_{initial}$ of the true parameter matrix B, and $\|\hat{B}_{initial} - B\| = O(n^{-1/2})$. The root-n neighborhood assumption is a common assumption in a single-index model, refer to Carroll et al. (1997) and Hardle et al. (1993) for more information.

We establish the asymptotic properties of the SIM nuclear model. At first, let define some notations that will be used in the proof

$$Q(g, B) = \sum_{i=1}^{n}[y_i - g(\langle X_i, B\rangle)]^2 + \lambda\|B\|_*$$

and

$$m(\cdot) = \frac{1}{2n}(\cdot)^2$$

Denote B_0 as the true value of B in the model and $\|B_0\|_F = 1$. Then we impose the following regularity conditions:

A. $B_0 \in \mathbb{R}^{p\times q}$ is the parameter matrix. The marginal density of $\langle x, B_0\rangle$ is positive and uniformly continuous.

B. For the unknown smooth function $g(\langle X, B\rangle)$, its second derivative $g^{''}(.)$ is continuous and bounded in D.

C. X is bounded and its density function has a continuous second derivative.

D. The kernel function $K(\cdot)$ is a symmetric density function with compact and bounded support and bounded first derivative. It satisfies $\int_{-\infty}^{\infty} K(z)dz = 1$, $\int_{-\infty}^{\infty} zK(z)dz = 0$, $\int_{-\infty}^{\infty} z^2K(z)dz < \infty$.

E. $E(\epsilon_i|x_i) = 0$, $E(\epsilon_i^2|x_i) = \sigma^2$ and $E(\epsilon_i^4|x_i)$ exists.

F. $nh^3 \to \infty$ and $nh^4 \to 0$

G. The conditional density function of y given $u = \langle X, B_0\rangle$, $f(y|u)$ is continuous in u for each y. Moreover, there exist positive constants ϵ and δ and a positive function $G(y|u)$ such that $sup_{|u_n-u|\le\epsilon} f_y(y|u_n) \le G(y|u)$ and that $\int |m^{'}(y - g_0(u))|^{2+\delta} G(y|u)d\mu(y) < \infty$, and $\int (m(y-t) - m(y) - m^{'}(y)t)^2 G(y|u)d\mu(y) =o(t^2)$ as $t \to 0$.

These conditions are commonly used in the literature. Conditions (A) and (B) are regular conditions for the single index model. Condition (C) is imposed to facilitate the technical arguments though it is somewhat complex. Condition (D) simply requires that the kernel function is a proper density with finite second moment which is required for the asymptotic variance of estimators. Condition (F) is used for the rate of bandwidth. The bandwidth h is selected to satisfy the condition, and assumed to be constant during the whole computation process. Condition (G) is weaker than the Lipschitz continuity of the function $m^{'}()$, which is required by the dominated convergence theorem and moment calculation in proving the asymptotic normality.

Theorem 2.2.1. *Let $x_1, \cdots, x_n$ be independent and identically distributed with a density f(x) that satisfies conditions (A)-(F). Under the assumption that $\lambda_n = O(n^{-1/2})$, there exists a local minimizer $\hat{B}$ of $Q(g_B, B)$ such that $\|\hat{B} - B_0\| = O_p(n^{-1/2})$, where $\|\hat{B}\|_F = \|B_0\|_F = 1$, and g_B is the local linear estimate of the link function g(.) given B.*

By this theorem, there exists a root-n consistent penalized least squares estimate for B_0 if we can properly select the tunning parameter λ_n.

Theorem 2.2.2. *Under conditions above, if $n \to \infty$, $h \to 0$, and $nh \to \infty$, then for an interior point u.*

$$\hat{g}(u; \hat{B}) - g(u) - \frac{1}{f_U(u)\phi''(0|u)}(nh)^{-1} \sum_{i=1} m'(y_i^*) K_i \to o((nh)^{-1/2})$$

where $g()$ is the true function, $y^* = y_i - g(u) - g'(u)(\langle X_i, \hat{B}\rangle - u)$, $\phi(t|u) = E(m(y - g(u) + t)|U = u)$ and $f_U()$ is the density function of $u = \langle X, B_0\rangle$.

In this theorem, we consider the difference between the estimated link function and true link function. Both the estimated link function and true link function are evaluated at the same covariate value u. But the pointwise accuracy is based on the quantity $\hat{g}(\langle X, \hat{B}\rangle) - g(\langle X, B\rangle)$. Both the estimated link function and true link function are evaluated at the same covariate value X. The scaled pointwise error term can be written as $\hat{g}(\langle X, \hat{B}) - g(\langle X, B\rangle) = \hat{g}(\langle X, \hat{B}\rangle) - \hat{g}(\langle X, B\rangle) + \hat{g}(\langle X, B\rangle) - g(\langle X, B\rangle)$. We can use the next corollary to prove the scaled pointwise between $\hat{g}(\langle X, \hat{B}\rangle)$ and $g(\langle X, B\rangle)$ tends to $o(1)$.

Corollary 2.2.2.1. *Under the same conditions as in theorem 2.2.2, we can have*

$$\hat{g}(\langle X, \hat{B}\rangle) - g(\langle X, B_0\rangle) - \frac{g''(u)h^2 \int t^2 k(t)}{2} \to O(n^{-1/2})$$

Before the next theorem, the singular value decomposition (SVD) of a matrix B should be introduced.

Definition 2.2.1. A singular value decomposition (SVD) of $B \in \mathbb{R}^{p\times q}$ is a factorization

$$B = USV^T$$

where:

- U is an $p \times p$ orthogonal matrix.
- V is an $q \times q$ orthogonal matrix.
- S is an $p \times q$ matrix whose ith diagonal entry equals the ith singular value δ_i for $i = 1, \cdots, r$. All other entries of S are zero.

Theorem 2.2.3. *If $n^{1/2}\lambda_n$ tends to a limit $\lambda_0 \geq 0$, then $B_* = n^{1/2}(\hat{B} - B_0)$ converges in distribution to the unique global minimizer of*

$$\sqrt{n}(\hat{B} - B_0) \rightarrow_d argmin(T)$$

where

$$T(B_*) = \frac{1}{2}vec(B_*)C_1vec(B_*) - C_0vec(B_*) + \lambda_0[trU^TB_*V + \|U_\perp B_* V_\perp\|_*]$$

where U and V are from the singular value decomposition of $B_0 = USV^T$, $U_\perp$ and $V_\perp$ are any orthonormal complements of U and V, $\phi(t|\langle X, B_0\rangle) = Em(y - \hat{g}(\langle X, \hat{B}\rangle) + t|\langle X, B_0\rangle)$,

$$C_1 = E\{f_y(0|\langle X, B_0\rangle)g(\langle X, B_0)^2[X - E(X|\langle X, B_0\rangle)][X - E(X|\langle X, B_0\rangle)]^T\},$$

and

$$C_0 = ZE\{g^{'}(\langle X, B\rangle)(X - E(X|\langle X, B_0\rangle))\}$$

. In which $Z \rightarrow_D N(0, \sigma^2)$. The function $T(\cdot)$ can be proved to be convex, and $T(\cdot)$ has unique solution.

2.3 Proofs

In order to prove the theorem 2.2.1, 2.2.2 and 2.2.3, we will prove prelimiaries that would be needed.

2.3.1 Prelimiaries knowledge

Let $S_n^j = \sum_{i=1}^n K_h(U_i - u)(U_i - u)^j$ in which $K_h(t) = \frac{K(t/h)}{h}$, $K()$ is a kernel function that satisfies condition(D). Then it is very easy to prove that $S_n^j = E(S_n^j) + O(\sqrt{var(S_n^j)})$.

$$\begin{aligned} S_n^j =& E(S_n^j) + O(\sqrt{var(S_n^j)}) \\ \leq& nh^j f(u) \int K(t)t^j dt + O(\sqrt{nEK_h(U_i - u)(U_j - u)^{2j}}) \\ \leq& nh^j [f(u) \int K(t)t^j dt + O(\sqrt{1/nh})] \end{aligned} \tag{2.5}$$

We define

$$s_n = \begin{bmatrix} s_n^0 & s_n^1 \\ s_n^1 & s_n^2 \end{bmatrix}$$

From (2.5), s_n can be rewritten as

$$s_n = \begin{bmatrix} n(f(u) + O(\sqrt{1/nh})) & nh(O(\sqrt{1/nh})) \\ nh(O(\sqrt{1/nh})) & nh^2(f(u) \int k(t)t^2 + O(\sqrt{1/nh})) \end{bmatrix}$$

And the inverse of s_n is

$$s_n^{-1} = \begin{bmatrix} \frac{s_n^2}{s_n^0 s_n^2 - (s_n^1)^2} & \frac{-s_n^1}{s_n^0 s_n^2 - (s_n^1)^2} \\ \frac{-s_n^1}{s_n^0 s_n^2 - (s_n^1)^2} & \frac{s_n^0}{s_n^0 s_n^2 - (s_n^1)^2} \end{bmatrix}$$

From (2.5), the inverse matrix can be rewritten as

$$s_n^{-1} = \begin{bmatrix} \frac{\int k(t)t^2 + O(\sqrt{1/nh})}{nf(u)(\int k(t)t^2 + O(1/nh))} & \frac{O(\sqrt{1/nh})}{nhf(u)(\int k(t)t^2 + O(1/nh))} \\ \frac{O(\sqrt{1/nh})}{nhf(u)(\int k(t)t^2 + O(1/nh))} & \frac{1 + O(\sqrt{1/nh})}{nh^2 f(u)(\int k(t)t^2 + O(1/nh))} \end{bmatrix}$$

We can get the value of (a, b) by minimizing the equation $\sum_{i=1}^{n}(y_i - a - b(U_i - u))^2 K_h(U_i - u)$. Then we get $a = \sum_{i=1}^{n} s_n^{-1} K_h(U_i - u) y_i$. We define the function $W(\cdot)$

$$\begin{aligned} W(\frac{U_i - u}{h}) &= [1, 0] s_n^{-1} K_h(U_i - u)[1, U_i - u]^T \\ &= \frac{\int k(t)t^2 + O(\sqrt{1/nh})}{nf(u)(\int k(t)t^2 + O(1/nh))} K_h(U_i - u) \\ &+ \frac{O(\sqrt{1/nh})}{nhf(u)(\int k(t)t^2 + O(1/nh))} K_h(U_i - u)(U_i - u) \\ &= \frac{\int k(t)t^2 + O(\sqrt{1/nh})}{nf(u)(\int k(t)t^2 + O(1/nh))} K_h(U_i - u) + O(\sqrt{1/nh}) \end{aligned} \tag{2.6}$$

2.3.2 Proof of Theorem 2.2.1

Let

$$Q(\hat{g}_B, B) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{g}(\langle X_i, B\rangle))^2 + \lambda_n \|B\|_*$$

And

$$Q(\hat{g}_B, B_0) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{g}(\langle X_i, B_0\rangle))^2 + \lambda_n \|B_0\|_*$$

where $\hat{g}$ is the local linear regression of the link function g() with B.

In this theorem, we need only to show that for any given ϵ there exists a large C such that

$$P\{\sup_{\|B - B_0\| = Cn^{-1/2}, \|B\|_F = 1} Q(\hat{g}, B) > Q(\hat{g}, B_0)\} \geq 1 - \epsilon$$

This implies that with probability tending to 1 there is a local minimum B in the ball $\{B : \|B - B_0\| = Cn^{-1/2}, \|B\|_F = 1\}$, hence B_0 is the true parameter. We have

$$Q(\hat{g}, B) - Q(\hat{g}, B_0) \geq \frac{1}{n}\sum_{i=1}^{n}\{(y_i - \hat{g}(\langle x_i, B\rangle))^2 - (y_i - \hat{g}(\langle x_i, B_0\rangle))^2\} + \lambda_n(\|B\|_* - \|B_0\|_*)$$

There are two steps for this proof:

1. $\frac{1}{n}\sum_{i=1}^{n}[(y_i - \hat{g}(\langle x_i, B\rangle))^2 - (y_i - \hat{g}(\langle x_i, B_0\rangle))^2]$ is dominated by $O(\|B - B_0\|)$.

2. $\lambda_n(\|B\|_* - \|B_0\|_*)$ is dominated by $O(\|B - B_0\|)$.

Next we show that the equation is bounded by the first term. Denote $\hat{U}_i = \langle x_i, B\rangle$ and $U_i = \langle x_i, B_0\rangle$. We then can write the first term as

$$\begin{aligned}
&\frac{1}{2n}\sum_{i=1}^{n}[(y_i - \hat{g}(\langle x_i, B\rangle))^2 - (y_i - \hat{g}(\langle x_i, B_0\rangle)^2] \\
=&\frac{1}{2n}\sum_{i=1}^{n}[(y_i - \sum_{j=1}^{n} W(\frac{\hat{U}_j - \hat{U}_i}{h})y_j)^2 - (y_i - \sum_{j=1}^{n} W(\frac{U_j - U_i}{h})y_j)^2] \\
=&\frac{1}{n}\sum_{i=1}^{n}(y_i - \sum_{j=1}^{n} W(\frac{U_j - U_i}{h})y_j)(\sum_{j=1}^{n} W(\frac{U_j - U_i}{h})y_j - \sum_{j=1}^{n} W(\frac{\hat{U}_j - \hat{U}_i}{h})y_j) \qquad (2.7)\\
+&\frac{1}{2n}\sum_{i=1}^{n}(\sum_{j=1}^{n} W(\frac{U_j - U_i}{h})y_j - \sum_{j=1}^{n} W(\frac{\hat{U}_j - \hat{U}_i}{h})y_j)^2 \\
=&\frac{1}{n}I_1 + \frac{1}{2n}I_2
\end{aligned}$$

First consider I_2 in (2.7). It is easy to know that

$$\begin{aligned}
&\sum_{j=1}^{n} W(\frac{U_j - U_i}{h})y_j - \sum_{j=1}^{n} W(\frac{\hat{U}_j - \hat{U}_i}{h})y_j \\
=&\sum_{j=1}^{n} W(\frac{U_j - U_i}{h})g(U_j) - \sum_{j=1}^{n} W(\frac{\hat{U}_j - \hat{U}_i}{h})g(U_j) \qquad (2.8)\\
+&\sum_{j=1}^{n} W(\frac{U_j - U_i}{h})\epsilon_j - \sum_{j=1}^{n} W(\frac{\hat{U}_j - \hat{U}_i}{h})\epsilon_j
\end{aligned}$$

So I_2 can be divided into three parts

$$
\begin{aligned}
&\sum_{i=1}^{n}(\sum_{j=1}^{n}W(\frac{U_j-U_i}{h})y_j-\sum_{j=1}^{n}W(\frac{\hat{U}_j-\hat{U}_i}{h})y_j)^2\\
=&\sum_{i=1}^{n}[\sum_{j=1}^{n}W(\frac{U_j-U_i}{h})g(U_j)-\sum_{j=1}^{n}W(\frac{\hat{U}_j-\hat{U}_i}{h})g(U_j)\\
+&\sum_{j=1}^{n}W(\frac{U_j-U_i}{h})\epsilon_j-\sum_{j=1}^{n}W(\frac{\hat{U}_j-\hat{U}_i}{h})\epsilon_j]^2\\
=&\sum_{i=1}^{n}[\sum_{j=1}^{n}W(\frac{U_j-U_i}{h})g(U_j)-\sum_{j=1}^{n}W(\frac{\hat{U}_j-\hat{U}_i}{h})g(U_j)]^2\\
+&\sum_{i=1}^{n}[\sum_{j=1}^{n}W(\frac{U_j-U_i}{h})\epsilon_j-\sum_{j=1}^{n}W(\frac{\hat{U}_j-\hat{U}_i}{h})\epsilon_j]^2\\
+&2\sum_{i=1}^{n}[\sum_{j=1}^{n}W(\frac{U_j-U_i}{h})g(U_j)-\sum_{j=1}^{n}W(\frac{\hat{U}_j-\hat{U}_i}{h})g(U_j)]\cdot[\sum_{j=1}^{n}W(\frac{U_j-U_i}{h})\epsilon_j-\sum_{j=1}^{n}W(\frac{\hat{U}_j-\hat{U}_i}{h})\epsilon_j]\\
=&p_1+p_2+p_3
\end{aligned}
\tag{2.9}
$$

We will find the bound of these parts. For p_1 in (2.9), we have

$$
\begin{aligned}
&\sum_{j=1}^{n}W(\frac{U_j-U_i}{h})g(U_j)-\sum_{j=1}^{n}W(\frac{\hat{U}_j-\hat{U}_i}{h})g(U_j)\\
=&\sum_{j=1}^{n}W(\frac{U_j-U_i}{h})(g(U_i)+g^{'}(U_i)(U_j-U_i)+O(h^2))\ ①\\
-&\sum_{j=1}^{n}W(\frac{\hat{U}_j-\hat{U}_i}{h})(g(\hat{U}_i)+g^{'}(\hat{U}_i)(U_j-\hat{U}_i)+O(h+n^{-1/2})^2)\ ②
\end{aligned}
\tag{2.10}
$$

For ①, it be proved by $① = g(U_i)+O(h^2)$

$$
\begin{aligned}
&\sum_{j=1}^{n}W(\frac{U_j-U_i}{h})(g(U_i)+g^{'}(U_i)(U_j-U_i)+O(h^2)\\
=&\sum_{j=1}^{n}[1,0]s_n^{-1}K_h(U_j-U_i)[1,U_j-U_i]^T[g(U_i)+g^{'}(U_i)(U_j-U_i)+O(h^2)]\\
=&g(U_i)+O(h^2)
\end{aligned}
\tag{2.11}
$$

where the first equation holds because of the definition of W() in (2.6).

For ②,

$$\begin{aligned}
&\sum_{j=1}^{n} W(\frac{\hat{U}_j - \hat{U}_i}{h})[g(\hat{U}_i) + g^{'}(\hat{U}_i)(U_j - \hat{U}_i) + O(U_j - \hat{U}_i)^2] \\
=&g(\hat{U}_i) + \sum_{j=1}^{n} W(\frac{\hat{U}_j - \hat{U}_i}{h})[g^{'}(\hat{U}_i)(U_j - \hat{U}_j + \hat{U}_j - \hat{U}_i) + O(U_j - U_i + U_i - \hat{U}_i)^2] \\
=&g(\hat{U}_i) + EW(\frac{\hat{U}_j - \hat{U}_i}{h})g^{'}(\hat{U}_i)E(U_j - \hat{U}_j|U = \hat{U}_j) + O(n^{-1/2} + h)^2 \\
=&g(\hat{U}_i) + g^{'}(\hat{U}_i)vecE(x_j|U = \hat{U}_j)^T vec(B_0 - B) + O(n^{-1/2} + h)^2
\end{aligned} \tag{2.12}$$

We combine the results of (2.11) and (2.12), then we get

$$\begin{aligned}
&\sum_{j=1}^{n} W(\frac{U_j - U_i}{h})g(U_j) - \sum_{j=1}^{n} W(\frac{\hat{U}_j - \hat{U}_i}{h})g(U_j) \\
=&g(U_i) - g(\hat{U}_i) - g^{'}(\hat{U}_i)vecE(x_j|U = \hat{U}_j)^T vec(B_0 - B) + O(h + n^{-1/2})^2 + O(h^2) \\
=&g^{'}(\hat{U}_i)vec(x_i - E(x_j|U = \hat{U}_j))^T vec(B_0 - B) + O(h^2) + O(h + n^{-1/2})^2
\end{aligned} \tag{2.13}$$

Then part p_1 can be expressed by

$$\begin{aligned}
&\sum_{i=1}^{n}(\sum_{j=1}^{n} W(\frac{\hat{U}_j - \hat{U}_i}{h})g(U_j) - \sum_{j=1}^{n} W(\frac{U_j - U_i}{h})g(U_j))^2 \\
=&nvec(B - B_0)^T \Sigma vec(B - B_0) + nO(\|B - B_0\|h^2) + O(nh^4)
\end{aligned} \tag{2.14}$$

where $\Sigma = E[g^{'}(\hat{U}_j)]^2 vec(x_j - E(x_j|U = \hat{U}_j))vec(x_j - E(x_j|U = \hat{U}_j))^T$. For the part p_2 of (2.9)

$$\sum_{i=1}^{n}(\sum_{j=1}^{n} W(\frac{\hat{U}_j - \hat{U}_i}{h})\epsilon_j - \sum_{j=1}^{n} W(\frac{U_j - U_i}{h})\epsilon_j)^2 = \boldsymbol{\epsilon}^T (S_B - S_{B_0})^T (S_B - S_{B_0})\boldsymbol{\epsilon} = V_1 \tag{2.15}$$

Note that

$$EV_1 = \sigma^2 Etr((S_B - S_{B_0})^T (S_B - S_{B_0})) = \sigma^2 \sum_{i=1}^{n} ES_{ii}$$

where

$$S_{ii} = \sum_{j=1}^{n}(W(\frac{\hat{U}_j - \hat{U}_i}{h}) - W(\frac{U_j - U_i}{h}))^2$$

By Taylor expansion,it can be proved by

$$W(\frac{\hat{U}_j - \hat{U}_i}{h}) = \frac{f^{-1}(U_i) + O(\sqrt{1/nh})}{n}(k_h(U_j - U_i) + k_h'(U_j^* - U_i^*)\frac{(\hat{U}_j - \hat{U}_i) - (U_j - U_i)}{h})$$
$$W(\frac{U_j - U_i}{h}) = \frac{f^{-1}(U_i) + O(\sqrt{1/nh})}{n}k_h(U_j - U_i)$$

Where U_i^* is a value between U_i and $\hat{U}_i$, and U_j^* is between U_j and $\hat{U}_j$. Hence

$$\begin{aligned}
&W(\frac{\hat{U}_j - \hat{U}_i}{h}) - W(\frac{U_j - U_i}{h}) \\
=&\frac{O(\sqrt{1/nh})}{n}k_h(U_j - U_i) + \frac{f^{-1}(U_i) + O(\sqrt{1/nh})}{n}k_h'(U_j^* - U_i^*)\frac{(\hat{U}_j - \hat{U}_i) - (U_j - U_i)}{h} \\
=&O(1/(n^3h) + 1/n)
\end{aligned} \tag{2.16}$$

Then the expectation of S_{ii}

$$\begin{aligned}
&ES_{ii} \\
\leq&2\sum_{j=1}^{n}[\frac{O(\sqrt{1/nh})}{n}k_h(U_j - U_i)]^2 + 2\sum_{j=1}^{n}[\frac{f^{-1}(U_i) + O(\sqrt{1/nh})}{n}k_h'(U_j^* - U_i^*)\frac{(\hat{U}_j - \hat{U}_i) - (U_j - U_i)}{h}]^2 \\
=&S_{i1} + S_{i2}
\end{aligned}$$

It is obvious that

$$S_{i1} = O(1/(n^2h^2)) \text{ and } S_{i2} = O(\|B - B_0\|^2/nh^3) \tag{2.17}$$

Then by conditions $nh \to \infty$,$nh^3 \to \infty$ and $\|B - B_0\| = O(n^{-1/2})$, we have

$$\begin{aligned}
&EV_1 \\
=&\sigma^2 E[tr(S_B - S_{B_0})^T(S_B - S_{B_0})] \\
=&n \cdot O(1/(nh)^2) + n \cdot O(\|B - B_0\|^2/nh^3) \\
=&O(1/nh^2) + O(\|B - B_0\|^2/h^3)
\end{aligned} \tag{2.18}$$

and by Gersgorin theorem(Quarteroni et al. (2000)), $|\lambda_i(S_B - S_{B_0})(S_B - S_{B_0})^T| = o(1) < 1$, where $\lambda_i, i = 1, \cdots, p$ are eigenvalues of $(S_B - S_{B_0})(S_B - S_{B_0})^T$. Then we have

$$\begin{aligned} EV_1^2 &\leq 2E\epsilon^4 E[tr(S_B - S_{B_0})^T(S_B - S_{B_0})(S_B - S_{B_0})^T(S_B - S_{B_0})] \\ &\leq 2E\epsilon^4 E[(S_B - S_{B_0})^T(S_B - S_{B_0})] \\ &= o(1) \end{aligned} \tag{2.19}$$

The last equation is held by (2.18). Based on (2.18) and (2.16), it is easy to show that the part p_2 is bounded.

$$\begin{aligned} &\sum_{i=1}^{n}(\sum_{j=1}^{n} W(\frac{\hat{U}_j - \hat{U}_i}{h})\epsilon_j - \sum_{j=1}^{n} W(\frac{U_j - U_i}{h})\epsilon_j)^2 \\ =&\boldsymbol{\epsilon}^T(S_B - S_{B_0})^T(S_B - S_{B_0})\boldsymbol{\epsilon} \\ =&O(1/nh^2) + O(\|B - B_0\|^2/h^3) \end{aligned} \tag{2.20}$$

In next step, we need to use the property $\sum_{j=1}^{n}\epsilon_i = nE(\epsilon) + O(\sqrt{nvar(\epsilon_i)})$, from the condition(E). We have $E(\epsilon) = 0$ and $\sqrt{nvar(\epsilon)} \leq \sqrt{nE(\epsilon)^2} = O(\sqrt{n})$. For the third term p_3 of (2.9)

$$\sum_{i=1}^{n}[\sum_{j=1}^{n} W(\frac{U_j - U_i}{h})g(U_j) - \sum_{j=1}^{n} W(\frac{\hat{U}_j - \hat{U}_i}{h})g(U_j)]\cdot[\sum_{j=1}^{n} W(\frac{U_j - U_i}{h})\epsilon_j - \sum_{j=1}^{n} W(\frac{\hat{U}_j - \hat{U}_i}{h})\epsilon_j]$$

From (2.13) and (2.17), we have

$$\begin{aligned} &\sum_{i=1}^{n} O(\|B - B_0\| + h^2)[\sum_{j=1}^{n} W(\frac{U_j - U_i}{h})\epsilon_j - \sum_{j=1}^{n} W(\frac{\hat{U}_j - \hat{U}_i}{h})\epsilon_j] \\ =&\sum_{i=1}^{n} O(\|B - B_0\| + h^2)\sum_{j=1}^{n}(W(\frac{U_j - U_i}{h}) - W(\frac{\hat{U}_j - \hat{U}_i}{h}))\epsilon_j \\ =&O(\frac{\|B - B_0\| + h^2}{\sqrt{h}} + \frac{\sqrt{n}\|B - B_0\|^2 + \sqrt{n}h^2\|B - B_0\|}{h} + \frac{\|B - B_0\|^2 + h^2\|B - B_0\|}{\sqrt{h}h}) \end{aligned} \tag{2.21}$$

By (2.14), (2.20) and (2.21), we have

$$\frac{1}{2n}I_2 = O(\frac{\|B - B_0\|}{\sqrt{n}h}) \tag{2.22}$$

Now we consider I_1 in (2.7)

$$\begin{aligned} I_1 =& \sum_{i=1}^{n}(y_i - \sum_{j=1}^{n} W(\frac{U_j - U_i}{h})y_j)(\sum_{j=1}^{n} W(\frac{U_j - U_i}{h})y_j - \sum_{j=1}^{n} W(\frac{\hat{U}_j - \hat{U}_i}{h})y_j) \\ =& \sum_{i=1}^{n} T_1 \cdot T_2 \end{aligned} \tag{2.23}$$

For T_2

$$\begin{aligned} & \sum_{j=1}^{n} W(\frac{U_j - U_i}{h})y_j - \sum_{j=1}^{n} W(\frac{\hat{U}_j - \hat{U}_i}{h})y_j \\ =& \sum_{j=1}^{n} W(\frac{U_j - U_i}{h})g(U_j) - \sum_{j=1}^{n} W(\frac{\hat{U}_j - \hat{U}_i}{h})g(U_j) \\ +& \sum_{j=1}^{n} W(\frac{U_j - U_i}{h})\epsilon_j - \sum_{j=1}^{n} W(\frac{\hat{U}_j - \hat{U}_i}{h})\epsilon_j \\ =& g^{'}(\hat{U}_i)vec(x_i - E(x_j|U = \hat{U}_j))^T vec(B_0 - B) + O(\frac{\|B - B_0\|}{\sqrt{h}} + \frac{\sqrt{n}\|B - B_0\|}{h} + \sqrt{\frac{\|B - B_0\|^2}{h^3}}) \end{aligned} \tag{2.24}$$

where the second equation holds because $\sum_{j=1}^{n} W(\frac{U_j - U_i}{h})g(U_j) - \sum_{j=1}^{n} W(\frac{\hat{U}_j - \hat{U}_i}{h})g(U_j) = g^{'}(\hat{U}_i)vec(x_i - E(x_j|U = \hat{U}_j))^T vec(B_0 - B) + O(h^2)$ and $\sum_{j=1}^{n} W(\frac{U_j - U_i}{h})\epsilon_j - \sum_{j=1}^{n} W(\frac{\hat{U}_j - \hat{U}_i}{h})\epsilon_j =$

$O(\frac{\|B-B_0\|}{\sqrt{h}} + \frac{\sqrt{n}\|B-B_0\|}{h} + \sqrt{\frac{\|B-B_0\|}{h^3}})$. For T_1

$$
\begin{aligned}
& y_i - \sum_{j=1}^{n} W(\frac{U_j - U_i}{h}) y_j \\
=& y_i - \sum_{j=1}^{n} W(\frac{U_j - U_i}{h})(g(U_j) + \epsilon_j) \\
=& y_i - \sum_{j=1}^{n} W(\frac{U_j - U_i}{h})(g(U_i) + g^{'}(U_i)(U_j - U_i) + O(U_j - U_i)^2 + \epsilon_j) \\
=& y_i - \sum_{j=1}^{n} W(\frac{U_j - U_i}{h}) g(U_i) - \sum_{j=1}^{n} W(\frac{U_j - U_i}{h}) g^{'}(U_i)(U_j - U_i) \\
& - \sum_{j=1}^{n} W(\frac{U_j - U_i}{h}) O(U_j - U_i)^2 - \sum_{j=1}^{n} W(\frac{U_j - U_i}{h}) \epsilon_j \\
=& y_i - g(U_i) + O(\sqrt{1/nh}) + O(h^2) \\
=& \epsilon_i + O(h^2)
\end{aligned}
\tag{2.25}
$$

By (2.24) and (2.25),

$$\frac{1}{n} I_1 = O(g^{'}(\hat{U}_i) vec(x_i - E(x_j | U = \hat{U}_j))^T vec(B_0 - B)/\sqrt{n})$$

Then $\frac{1}{n} I_1$ and $\frac{1}{2n} I_2$ are dominated by $\|B - B_0\|^2 E g^{'}(\hat{U}_j)^2 \{X_j - E(X_j | U = \hat{U}_j)\}^2$. As shown by Bach (2008). The second term is bounded by

$$\lambda_n(\pm 16 min\{p, q\} \frac{s_1^2}{s_r^2} \|B_0 - B\|_2^2 + tr U^T (B_0 - B) V + \|U_\perp^T (B_0 - B) V_\perp\|_*) \tag{2.26}$$

where s_1 and s_r are the largest and smallest strictly positive singular values of B. Since we know $\|B - B_0\| = O(n^{-1/2})$ and $\lambda_n = O(n^{-1/2})$, it is very easy to prove that (2.26) is bounded by $O(\|B - B_0\|^2)$. The first and second term of (2.7) are bounded by $O(\|B - B_0\|^2)$. Hence this theorem is proved.

2.3.3 Proof of Theorem 2.2.2

The proof is similar to the proof of Fan et al. (1994). One lemma is needed in the proof.

Lemma 2.3.1 (Quadratic Approximation Lemma (Hjort and Pollard, 2011)). *Suppose $A_n(s)$ is convex and can be represented as $\frac{1}{2}s^{'}Vs+U_n^{'}s+C_n+r_n(s)$,where V is symmetric and positive definite, U_n is stochastically bounded, C_n is arbitrary , and $r_n(s)$ goes to zero in probability for each s. Then α_n, the argmin of A_n is only o(1) away from $\beta_n = -V^{-1}U_n$, the argmin of $\frac{1}{2}s^{'}Vs + U_n^{'}s + C_n$. If also $U_n \to U$, then $\alpha_n \to -V^{-1}U$.*

$\hat{g}(u; B_0)$ is a local linear estimator of $g()$ if the index coefficient B_0 is known. We divide $\hat{g}(u; \hat{B}) - g(u)$ into two parts.

$$\hat{g}(u; \hat{B}) - g(u) = \hat{g}(u; \hat{B}) - \hat{g}(u; B_0) + \hat{g}(u; B_0) - g(u) \tag{2.27}$$

There are three steps to finish the proof,

1. The first part on the right-hand side of (2.27), $\hat{g}(u; \hat{B}) - \hat{g}(u; B_0)$ is bounded by $O_p(\|\hat{B} - B_0\|)$.
2. Then we need to prove $\hat{g}(u; B_0) - g(u) - \frac{1}{f_{u_0}(u)\phi^{''}(0|u)} \sum m^{'}(y_i^*)K_i \to o(1/\sqrt{nh})$.
3. At last $\hat{g}(u; \hat{B}) - g(u) - \frac{1}{f_{u_0}(u)\phi^{''}(0|u)}(nh)^{-1} \sum m^{'}(y_i^*) \to O(n^{-1/2})$.

In which $\phi(t|u) = E(m(y - g(u) + t)|U = u)$, For the part $\hat{g}(u; \hat{B}) - \hat{g}(u; B_0)$ in the equation (2.27)

$$\begin{aligned}
&\hat{g}(u; \hat{B}) - \hat{g}(u; B_0) \\
=&\hat{g}^{'}(u)O(\|\hat{B} - B\|) \\
=&\hat{g}^{'}(u) \cdot O(n^{-1/2}) \\
=&O(n^{-1/2})
\end{aligned}$$

where $O(\|\hat{B} - B_0\|) = O(n^{-1/2})$, the first step is proved.

For given u, for notational simplicity, we write $\hat{a}_{B_0} := \hat{g}(u; B_0)$ and $\hat{b}_{B_0} := \hat{g}^{'}(u; B_0)$ which are the solutions of the following minimization problem,

$$min_{a,b} \sum m(y_i - a - b(\langle x_i, B_0\rangle - u))K(\frac{\langle X_i, B_0\rangle - u}{h}) \tag{2.28}$$

Denote

$$\bar{\theta} = (nh)^{1/2}(\hat{a}_{B_0} - g(u), h(\hat{b}_{B_0} - g^{'}(u)))$$

$$z_i = (1, (\langle x_i, B_0\rangle - u)/h)^T$$

$$y_i^* = y_i - g(u) - g^{'}(u)(\langle X_i, B_0\rangle - u)$$

$$m(y_i - a - b(\langle X_i, B_0\rangle - u)) = \frac{1}{2n}(y_i - a - b(\langle X_i, B_0\rangle - u))^2$$

$$K_i = K(\frac{\langle X_i, B_0\rangle - u}{h})$$

Thus $\bar{\theta}$ minimize

$$Q(\theta) = \sum_{i=1}[m(y_i^* - \theta^T z_i/\sqrt{nh}) - m(y_i^*)]K_i,$$

it can be proved that $Q(\theta)$ is convex in θ. We will show

$$Q(\theta) = \frac{1}{2}\theta^T S\theta + W^T\theta + r(\theta), r(\theta) = o_p(1) \tag{2.29}$$

where

$$S = f_{U_0}(u)\phi^{''}(0|u)\begin{pmatrix} 1 & 0 \\ 0 & \int_v K(v)v^2dv \end{pmatrix},$$

$W_n = -(nh)^{-\frac{1}{2}}\sum m^{'}(y_i^*)z_iK_i$. Here $\phi^{''}(0|u)$ is the second derivative of $\phi(t|u) = E(m(y - g(u)+t)|U = u)$ with respect to t evaluated at $t = 0$. The first and second derivatives of $\phi(t|u)$ with respect to t, $\phi^{'}()$ and $\phi^{''}()$, are assumed to exist. And $v \in [-M, M]$, where M is such a real number that $[-M, M]$ contains the support of $K()$, which means $|u_i - u| \leq Mh$ which $u_i = \langle x_i, \hat{B}\rangle$, when we face the situation that B_0 is known, the conclusion still works.

Write

$$Q(\theta) = E(Q(\theta)|u) - (nh)^{-1/2}\sum_{i=1}(m^{'}(y_i^*) - E(m^{'}(y_i^*)|u_i))K_i\theta^T z_i + R(\theta) \tag{2.30}$$

$$\begin{aligned}
E(Q(\theta)|u_i) =& \sum_{i=1}[\phi(g(u_i) - g(u) - g^{'}(u)(u_i - u) - \theta^T z_i/\sqrt{nh}|u_i) \\
&- \phi(g(u_i) - g(u) - g^{'}(u)(u_i - u)|u_i)]K_i \\
=& -(nh)^{-1/2}\sum_{i=1}\phi^{'}(g(u_i) - g(u) - g^{'}(u)(u_i - u)|u_i)(\theta^T z_i)K_i \\
&+ (2nh)^{-1}\theta^T(\sum_{i=1}K_i\phi^{''}(g(u_i) - g(u) - g^{'}(u)(u_i - u)|u_i)z_i z_i^T)\theta(1 + o_p(1)) \\
=& -(nh)^{-1/2}\sum_{i=1}E(m^{'}(y_i^*)|u_i)(\theta^T z_i)K_i \\
&+ (2nh)^{-1}\theta^T(\sum_{i=1}K_i\phi^{''}(g(u_i) - g^{'}(u)(u_i - u)|u_i)z_i z_i^T)\theta(1 + o_p(1))
\end{aligned}$$

and next we will show that

$$\sum K_i\phi^{''}(g_0(u_i) - g_0(u) - g_0^{'}(u)(u_i - u)|u_i)z_i z_i^T = S(1 + O_p(h^2))$$

where

$$S = (nh)^{-1}\sum_{i=1}K_i\phi^{''}(0|u)z_i z_i^T = \begin{pmatrix} s_0 & s_1 \\ s_1 & s_2 \end{pmatrix} \tag{2.31}$$

$$\begin{aligned}
&\sum_{i=1}K_i\phi^{''}(g_0(u_i) - g_0(u) - g_0^{'}(u)(u_i - u)|u_i)z_i z_i^T \\
=& \sum K_i(\phi^{''}(0 + O_p(h^2)|u))z_i z_i^T \\
=& \sum K_i(\phi^{''}(0|u) + O_p(h^2))z_i z_i^T \\
=& \sum K_i\phi^{''}(0|u)(1 + O_p(h^2))z_i z_i^T \\
=& S(1 + O_p(h^2))
\end{aligned}$$

in the matrix S, the matrix components $s_j = (nh)^{-1}\sum_{i=1} K_i\phi^{''}(0|u)((u_i-u)/h)^j$, $j = 0, 1, 2$. Because $s_j = E(s_j) + O(Var(s_j))$, we calculate the expectation and variance of s_j to find the boundness .

$$\begin{aligned}
E(s_j) =& h^{-1}\phi^{''}(0|u)\int K(\frac{U-u}{h})(\frac{U-u}{h})^j f_U(U)dU \\
=& \phi^{''}(0|u)\int K(t)t^j f_U(th+u)dt \\
=& f_U(u)\phi^{''}(0|u)\int K(t)t^j dt(1+o(1)) \\
=& f_U(u)\phi^{''}(0|u)c_j(1+o(1))
\end{aligned}$$

where $c_j = \int K(t)t^j dt$, $c_0 = 1$, $c_1 = 0$ and $c_2 = \int K(t)t^2 dt$ from condition (D). Next,we will discuss the $Var(s_j)$, it can be proved that $Var(s_j) \to o_p(1)$ when $j = 0, 1, 2$.

$$\begin{aligned}
Var(s_j) \leq& (nh)^{-2}E(\sum \phi^{''}(0|u)K(\frac{U-u}{h})(\frac{U-u}{h})^j)^2 \\
\leq& (nh)^{-2}nE(\phi^{''}(0|u)K(\frac{U-u}{h})(\frac{U-u}{h})^j)^2 \\
\leq& O(1/nh) \\
\leq& o_p(1)
\end{aligned}$$

Therefore

$$S = f_{u_0}(u)\phi^{''}(0|u)\begin{pmatrix} 1 & 0 \\ 0 & c_2 \end{pmatrix} + o_p(1) \tag{2.32}$$

At last, we have

$$E(Q(\theta)|u_i) = -(nh)^{-1/2}\sum E(m^{'}(y_i^*)|u_i)(\theta^T z_i)k_i + \frac{1}{2}\theta^T S\theta(1+o_p(1)). \tag{2.33}$$

Next, we show that with $R(\theta)$ defined by (2.30), $R(\theta) = o_p(1)$. Note that $E(R(\theta)) = 0$. Then

$$\begin{aligned} Var(R(\theta)) &\leq E(\sum_{i=1}(m(y_i^* - \theta^T z_i/\sqrt{nh}) - m(y_i^*) - m^{'}(y_i^*)\theta^T z_i/\sqrt{nh})K_i)^2 \\ &\leq \sum_{i=1} E(m(y_i^* - \theta^T z_i/\sqrt{nh}) - m(y_i^*) - m^{'}(y_i^*)\theta^T z_i/\sqrt{nh})^2 K_i^2 \end{aligned} \tag{2.34}$$

Because we assume that every X_i is independent with each other, $cov(X_i, X_j) = 0$, if $i \neq j$. We assume (y_1^*, z_1, K_1) can maximize $(m(y_i^* - \theta^T z_i/\sqrt{nh}) - m(y_i^*) - m^{'}(y_i^*)\theta^T z_i/\sqrt{nh})^2 K_i^2$. So the equation (2.34) can becomes

$$Var(R(\theta)) \leq nE(m(y_1^* - \theta^T z_1/\sqrt{nh}) - m(y_1^*) - m^{'}(y_1^*)\theta^T z_1/\sqrt{nh})^2 K_1^2$$

Hence, by condition (B), the equation (2.34) is

$$E(R^2(\theta)) \leq o(n \int \frac{(\theta_1^T z_1)^2}{nh} K^2(\frac{U_1 - u}{h}) f_{U_1}(U_1) dU_1)$$

And from the relation $R(\theta) = E(R(\theta)) + O(Var(R(\theta)))$. Thus, $R(\theta) = o_p(1)$. Now we have:

$$Q(\theta) = \frac{1}{2}\theta^T S\theta + W_n^T\theta + R(\theta) \tag{2.35}$$

where $W_n = -(nh)^{-1/2}\sum m^{'}(y_i^*) z_i K_i$. We outline the proof for stochastic boundedness of W. By change of variable and existence of $\int K^2(t)t^j dt$, $j = 0, 1, 2$ in condition (D), for some $c > 0$.

$$\begin{aligned} E(W_n W_n^T) =& (nh)^{-1} E(\sum_i \sum_j m^{'}(y_i^*) m^{'}(y_j^*) K_i K_j z_i z_j^T) \\ \leq& (nh)^{-1}[\sum_i m^{'}(y_i^*)^2 K_i^2 z_i z_i^T + \sum_j m^{'}(y_j^*)^2 K_j^2 z_j z_j^T] \\ \leq& c(nh)^{-1} E \sum_i (m^{'}(y_i^*)^2 K_i^2 z_i z_i^T) \\ =& O(h^{-1} E(K_i^2 z_i z_i^T)) \\ =& O(1) \end{aligned}$$

which also implies $E(W_n) = O(1)$ as a result of jensen's inequality. Bounded second moment implies that W_n is stochastically bounded. According to the Quadratic Approximation Lemma, $\bar{\theta}$ converges in probability to the minimizer $\hat{\theta} = -S^{-1}W_n$ of the right-hand side of (2.35).

$$\bar{\theta} - \hat{\theta} = o_p(1)$$

The first component of the above equality is

$$\sqrt{nh}(\hat{g}(u; B_0) - g(u) - V_n) = o_p(1)$$

where $V_n = U_n/[f_{u_0}(u)\phi^{''}(0|u)]$ and

$$U_n = \frac{1}{nh}\sum_{i=1} m^{'}(y_i^*)K_i$$

The second step is finished. At last, we combine the results of the previous two steps then get

$$\hat{g}(u; \hat{B}) - g(u) - \frac{1}{f_{u_0}(u)\phi^{''}(0|u)}(nh)^{-1}\sum m^{'}(y_i^*)K_i \to o_p(\frac{1}{\sqrt{nh}})$$

The theorem is proved.

2.3.4 Proof of Corollary 2.2.2.1

This proof has three steps:

1. Based on theorem 2.2.2, we can prove that $\hat{\theta}(X) - \theta(X) - \frac{1}{f_U(u)\phi^{''}(0|u)}(nh)^{-1}\sum_{i=1} m^{'}(y_i^*)K_i \to O(n^{-1/2})$, where $\hat{\theta}(X) = \hat{g}(\langle X, \hat{B}\rangle)$ and $\theta(X) = g(\langle X, B_0\rangle)$.
2. Next we can prove $\frac{1}{f_U(u)\phi^{''}(0|u)}(nh)^{-1}\sum_{i=1} m^{'}(y_i^*)K_i - \frac{g^{''}(u)h^2\int t^2K(t)}{2} \to o(1)$.
3. At last, we get $\hat{\theta}(X) - \theta(X) - \frac{g^{''}(u)h^2\int t^2k(t)}{2} \to O(n^{-1/2})$.

where $\phi(t|u) = E(m(y - g(u) + t)|U = u)$. For given X

$$\begin{aligned} &\hat{\theta}(X) - \theta(X) \\ =&\hat{g}(\langle X, \hat{B}\rangle; \hat{B}) - \hat{g}(\langle X, B_0\rangle; \hat{B}) + \hat{g}(\langle X, B_0\rangle; \hat{B}) - g(\langle X, B_0\rangle) \\ =&A + B \end{aligned} \tag{2.36}$$

Base on the Taylor theorem, part A converges to 0 at the rate of $n^{-1/2}$

$$A = \hat{g}(\langle X, \hat{B}\rangle, \hat{B}) - \hat{g}(\langle X, B_0\rangle, B_0) = \hat{g}^{'}(\langle X, \hat{B}\rangle, \hat{B})\|\hat{B} - B_0\| = O_p(n^{-1/2})$$

The part B tends to $o_p(\frac{1}{\sqrt{nh}})$ based on theorem 2.2.2. Then we get that

$$\hat{\theta}(X) - \theta(X) - \frac{1}{f_U(u)\phi^{''}(0|u)}(nh)^{-1}\sum_{i=1} m^{'}(y_i^*)K_i \to O(n^{-1/2})$$

and we define $U = (nh)^{-1}\sum m(y_i^*)K_i$ and $U = E(U) + O(var(U))$. Note that $\phi^{'}(0|u) = 0$ by the definition of $\phi()$. From the Taylor expansion, we have

$$\phi^{'}(t|u) = \phi^{''}(0|u)t(1 + o(t)), \text{as } t \to 0$$

$$\begin{aligned} E(U) &= E(E(U|u)) \\ &= \int \phi^{'}(\frac{1}{2}g^{''}(u)(th)^2|u)f(u)K(t)dt \\ &= \phi^{''}(0|u)\frac{1}{2}g^{''}(u)h^2 f(u)\int t^2 K(t)dt \end{aligned}$$

For $Var(U)$. It can be proved $Var(U) \leq E(U^2)$ and $E(U^2) = o(1)$, so that we get $\hat{\theta}(X) - \theta(X) - \frac{g^{''}(u)h^2\int t^2 K(t)}{2} \to O(n^{-1/2})$

At last this corollary is proved.

2.3.5 Proof of Theorem 2.2.3

This proof is similar to Fu and Knight (2000). At first, we define $T_n(B_*)$, and B_* can be estimated by minimizing T_n.

$$T_n(B_*) = \sum_{j=1}^{n}\sum_{i=1}^{n}[m(y_{ij} - \frac{1}{\sqrt{n}}\hat{b}_j\langle X_{ij}, B_*\rangle) - m(y_{ij})]w_{ij} + \lambda_n n^{1/2}(\|B_0 + n^{-1/2}B_*\|_* - \|B_0\|_*)$$

The proof can be divided into several steps:

1. Prove $\sum_{j=1}^{n}\sum_{i=1}^{n}[m(y_{ij} - \frac{1}{\sqrt{n}}\hat{b}_j\langle X_{ij}, B_*\rangle) - m(y_{ij})]w_{ij} \to \frac{1}{2}vec(B_*)C_1vec(B_*) - C_0vec(B_*)$

2. Prove $\lambda_n n^{1/2}(\|B_0 + n^{-1/2}B_*\|_* - \|B_0\|_*) \to \lambda_0[trU^TB_*V] + \|U_\perp B_* V_\perp\|$

3. The theorem is proved by combing the results of previous steps.

At first, we review the definition of function m()

$$m(y_i - \hat{a}_j - \hat{b}_j\langle X_i - X_j, B\rangle) = \frac{1}{2n}(y_i - \hat{a}_j - \hat{b}_j\langle X_i - X_j, B\rangle)^2 \tag{2.37}$$

, where $\hat{a}_j = \hat{g}(\langle X_j, B_0\rangle)$ and $\hat{b}_j = \hat{g}^{'}(\langle X_j, B_0\rangle)$. Write $\hat{B}_* = \sqrt{n}(\hat{B} - B_0)$, then we define

$$T_n(B_*) = \sum_{j=1}^{n}\sum_{i=1}^{n}[m(y_{ij} - \frac{1}{\sqrt{n}}\hat{b}_j\langle X_{ij}, B_*\rangle) - m(y_{ij})]w_{ij} + \lambda_n n^{1/2}(\|B_0 + n^{-1/2}B_*\|_* - \|B_0\|_*) \tag{2.38}$$

where $y_{ij} = y_i - \hat{a}_j - \hat{b}_j\langle X_{ij}, B_0\rangle$ and $X_{ij} = X_i - X_j$. We define $Q(B_*) = \sum_{j=1}^{n}\sum_{i=1}^{n}[m(y_{ij} - \frac{1}{\sqrt{n}}\hat{b}_j\langle X_{ij}, B_*\rangle) - m(y_{ij})]w_{ij}$. It can be shown that

$$Q(B_*) = E(Q(B_*)) - \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\sum_{j=1}^{n}[m^{'}(y_{ij}) - E(m^{'}(y_{ij}))]\hat{b}_j\langle X_{ij}, B_*\rangle w_{ij} + r(B_*) \tag{2.39}$$

where $r(B_*) = o_p(1)$, the similar proof can be found in the proof of theorem 2.2.2. Write $\phi(t|\langle X, B_0\rangle) = Em(y - \hat{g}(\langle X, \hat{B}\rangle) + t|\langle X, B_0\rangle)$

$$
\begin{aligned}
E(Q(B_*)) &= \sum_{i=1}^{n}\sum_{j=1}^{n}[Em(y_{ij} - \frac{1}{\sqrt{n}}\hat{b}_j\langle X_{ij}, B_*\rangle) - Em(y_{ij})]w_{ij} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} Em(y_{ij} - \frac{1}{\sqrt{n}}\hat{b}_j\langle X_{ij}, \hat{B}_*\rangle + \frac{1}{\sqrt{n}}\hat{b}_j\langle X_{ij}, \hat{B}_* - B_*\rangle)w_{ij} \\
&- \sum_{i=1}^{n}\sum_{j=1}^{n} Em(y_{ij} - \frac{1}{\sqrt{n}}\hat{b}_j\langle X_{ij}, \hat{B}_*\rangle + \frac{1}{\sqrt{n}}\hat{b}_j\langle X_{ij}, \hat{B}_*\rangle)w_{ij} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} Em(y_i - \hat{a}_j - \hat{b}_j\langle X_{ij}, \hat{B}\rangle + \frac{1}{\sqrt{n}}\hat{b}_j\langle X_{ij}, \hat{B}_* - B_*\rangle)w_{ij} \\
&- \sum_{i=1}^{n}\sum_{j=1}^{n} Em(y_i - \hat{a}_j - \hat{b}_j\langle X_{ij}, \hat{B}\rangle + \frac{1}{\sqrt{n}}\hat{b}_j\langle X_{ij}, \hat{B}_*\rangle)w_{ij} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} Em(y_i - \hat{g}(\langle X_i, \hat{B}\rangle|\langle X, B_0\rangle) + \frac{1}{\sqrt{n}}\hat{b}_j\langle X_{ij}, \hat{B}_* - B_*\rangle)w_{ij} \\
&- \sum_{i=1}^{n}\sum_{j=1}^{n} Em(y_i - \hat{g}(\langle X_i, \hat{B}\rangle|\langle X, B_0\rangle) + \frac{1}{\sqrt{n}}\hat{b}_j\langle X_{ij}, \hat{B}_*\rangle)w_{ij} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} \phi(\frac{1}{\sqrt{n}}\hat{b}_j\langle X_{ij}, \hat{B}_* - B_*\rangle|\langle X, B_0\rangle)w_{ij} - \sum_{i=1}^{n}\sum_{j=1}^{n} \phi(\frac{1}{\sqrt{n}}\hat{b}_j\langle X_{ij}, \hat{B}_*\rangle|\langle X, B_0\rangle)w_{ij} \\
&= -\sum_{i=1}^{n}\sum_{j=1}^{n} \phi^{'}(\frac{1}{\sqrt{n}}\hat{b}_j\langle X_{ij}, \hat{B}_*\rangle|\langle X, B_0\rangle)\frac{1}{\sqrt{n}}\hat{b}_j\langle X_{ij}, B_*\rangle w_{ij} \\
&+ \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{1}{2n}\phi^{''}(\frac{1}{\sqrt{n}}\hat{b}_j\langle X_{ij}, \hat{B}_*\rangle|\langle X, B_0\rangle)\hat{b}_j^2\langle X_{ij}, B_*\rangle^2 w_{ij} + o_p(1) \\
&= -\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\sum_{j=1}^{n} \phi^{'}(0|\langle X, B_0\rangle)\hat{b}_j\langle X_{ij}, B_*\rangle w_{ij} \\
&+ \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{1}{2n}\phi^{''}(0|\langle X, B_0\rangle)\hat{b}_j^2\langle X_{ij}, B_*\rangle^2 w_{ij} + o_p(1)
\end{aligned}
\tag{2.40}
$$

The last equation holds because of root-n assumption. So we have the following representation,

$$\begin{aligned} Q_n(B_*) = & -\frac{1}{\sqrt{n}}[\sum_{i=1}^{n}\sum_{j=1}^{n} m^{'}(y_{ij})\hat{b}_j vec(x_{ij})^T w_{ij}]vec(B_*) \\ & + \frac{1}{2n} vec(B_*)^T[\sum_{i=1}^{n}\sum_{j=1}^{n} \phi^{''}(0|\langle x, B_0\rangle)\hat{b}_j^2 vec(x_{ij})^T vec(x_{ij}) w_{ij}]vec(B_*) \\ & + o_p(1) \end{aligned} \tag{2.41}$$

Kong and Xia (2012) proved that $\phi^{''}(0|\langle X, B_0\rangle) = f_y(0|\langle X, B_0\rangle)$, $f_y(0|\langle X, B_0\rangle)$ is the conditional density function of y given $\langle X, B_0\rangle$. With a simple calculation and by Slutsky's Theorem, we have the following approximation,

$$Q_n(B_*) = \frac{1}{2} vec(B_*) C_1 vec(B_*) - C_0 vec(B_*) \tag{2.42}$$

, where

$$C_1 = E\{f_y(0|\langle X, B_0\rangle) g(\langle X, B_0)^2 [X - E(X|\langle X, B_0\rangle)][X - E(X|\langle X, B_0\rangle)]^T\}$$

and

$$C_0 = ZE\{g^{'}(\langle X, B\rangle)(X - E(X|\langle X, B_0\rangle))\},$$

where $Z \rightarrow_D N(0, \sigma^2)$. We have discussed the extension form of $Q(B_*)$. Then we will analyze $\lambda_n n^{1/2}(\|B_0 + n^{-1/2}B_*\|_* - \|B_0\|_*)$. From the subdifferential, we get the directional derivative(Borwein and Lewis (2000)) as

$$n^{1/2}\lambda_n(\|B_0 + n^{-1/2}B_*\|_* - \|B_0\|_*) \rightarrow \lambda_0[trU^T B_* V + \|U_\perp^T B_* V_\perp\|_*] \tag{2.43}$$

where U and V are from the singular value decomposition of $B_0 = Udiag(s)V^T$.And $U_\perp$ and $V_\perp$ denote any orthogonal complements of U and V.

If we combine (2.42) with (2.43), we can get

$$T_n(B_*) = \frac{1}{2} vec(B_*) C_1 vec(B_*) - C_0 vec(B_*) + \lambda_0[trU^T B_* V + \|U_\perp^T B_* V_\perp\|_*] \tag{2.44}$$

Bach (2008) proved $T_n(B_*)$ is strictly convex, which implies that it has an unique global minimum. At last this theorem 2.2.3 is proved.

Chapter 3

Implementation

3.1 Algorithm

The algorithm of SIM nuclear can be used the table 3.1 to illuminate. There are two steps in this algorithm. The first step is to use the outer product of gradients (OPG) to get the initial estimator of B, $\hat{B}_0$. The second step is that we use the alternating minimization algorithm to update the estimator $\hat{B}_t$ and $\hat{g}()$ iteratively until both of them converge to fixed points.

Table 3.1: SIM-Nuclear Algorithm

Initialize
Use outer product of gradients (OPG) method to
get the initial B^0, then standardize $\hat{B}_0 = \hat{B}_0/\|\hat{B}_0\|_F$
Repeat
P1: Given $\hat{B}_{t-1}$, use equation (3.2) to calculate $\hat{a_{j_t}}$ and $\hat{b_{j_t}}$
P2: Given $\hat{a}_t$ and $\hat{b}_t$, use equation (3.4), to renew B_t
Until objective value converges.

3.2 The First Step of the Algorithm

It is to find an initial estimator $\hat{B}_0$. A convenient choice for B is $\hat{B}/\|\hat{B}\|_F$ where $\hat{B}$ is the least squares estimate calculated by regressing y on X. But this choice may not be appropriate when the function g() is symmetric. For this reason, in this dissertation we propose to use outer product of gradients method by Xia et al. (2002) and get the resulting estimate of B as the initial matrix parameter.

Let us introduce OPG method briefly. Suppose that

$$y = g(\langle X, B\rangle) + \epsilon$$

with $E(\epsilon|X) = 0$ almost surely. Consider the loss function

$$\min_{a_j, b_j} \sum_{i=1}^{n} (y_i - a_j - b_j vec(B)^T vec(X_i - X_j))^2 w_{ij}$$

and

$$w_{ij} = K_h(X_i - X_j) / \sum_{l=1}^{n} K_h(X_l - X_j)$$

in which X_j is in the neighborhood of X_i in the sample, the mean response $E(Y|X = X_i) = g(\langle B, X_i\rangle)$ can be approximated by $a_j + b_j\langle B, X_i - X_j\rangle$, where $a_j = g(\langle B, X_j\rangle)$ and $b_j = g^{'}(\langle X_j, B\rangle)$. $vec(B)$ represents the vectorized B.

We can obtain an estimate for $\beta_j = g^{'}(\langle X_j, B\rangle)vec(B)$, $\hat{\beta}_j$, by simply solving the following weighted loss function problem:

$$\min_{a_j, \beta_j} \sum_{i=1}^{n} [y_i - a_j - \beta_j vec(X_i - X_j)]^2 w_{ij}$$

Because β_j is proportional to $vec(B)$, $\hat{\beta}_j$ can be standardized to produce an estimate of $vec(B)$. An efficient approach for estimating B is to consider the outer-product $M = \sum_{j=1}^{n} \hat{\beta}_j \hat{\beta}_j^T / n$. We propose to estimate B by transferring the eigen-vector corresponding to the largest singular value of M to a matrix and denote it by $\hat{B}_0$.

3.3 The Second Step of the Algorithm

After we get the initial estimator $\hat{B}_0$. We can estimate B and $g(\cdot)$ iteratively in the second step. This step can be divided into two parts.

1. We need to estimate the unknown function $g()$ under the assumption that we have an estimator of B, $\hat{B}$ from

$$\min_{g,\|B\|_F=1} \frac{1}{n}\sum_{i=1}^{n}(y - g(\langle X, \hat{B}\rangle))^2 \tag{3.1}$$

To solve the equation 3.1. We can replace $g(\langle X_i, \hat{B}\rangle)$ by $a_j + b_j(\langle X_i, \hat{B}\rangle - \langle X_j, \hat{B}\rangle)$. Then the function (3.1) becomes

$$\min_{a_j,b_j,\|B\|_F=1} \frac{1}{2n}\sum_{j=1}^{n}\sum_{i=1}^{n}(y_i - a_j - b_j(\langle X_i, \hat{B}\rangle - \langle X_j, \hat{B}\rangle))^2 w_{ij}(\frac{\langle X_i, \hat{B}\rangle - \langle X_j, \hat{B}\rangle}{h}) \tag{3.2}$$

In the equation (3.2), where $a_j = g(\langle X_j, \hat{B}\rangle)$ and $b_j = g^{'}(\langle X_j, \hat{B}\rangle)$. We assume that the optimal bandwidth is obtained. Then we can get least square estimators $\hat{a}_j$ and $\hat{b}_j$, $j = 1, \cdots, n$ for function (3.2). $\hat{a_j}$ and $\hat{b_j}$ can be expressed by a certain function of $x, y,$ and $\hat{B}$. Use some simple calculation, we can get the expression of $\hat{a_j}$ and $\hat{b_j}$ based on (3.2)

$$\begin{bmatrix} \hat{a}_j \\ \hat{b}_j \end{bmatrix} = (\mathbf{U}^T\mathbf{W}\mathbf{U})^{-1}\mathbf{U}^T\mathbf{W}\mathbf{Y}$$

in which $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$, $\mathbf{U} = \begin{bmatrix} 1 & U_1 - U_j \\ 1 & U_2 - U_j \\ \vdots & \\ 1 & U_n - U_j \end{bmatrix}$, $U_i = \langle X_i, \hat{B}\rangle$ and $\mathbf{W} = \begin{bmatrix} w1j & & \\ & \ddots & \\ & & w_{Nj} \end{bmatrix}$.

In this step, we estimate the link function $g(\cdot)$ as a classical univariate non-parametric regression problem. And the estimator $\hat{a}_j$ represents $\hat{g}(\langle X_j, \hat{B}\rangle)$ and $\hat{b}_j$ represents $\hat{g}^{'}(\langle X_j, \hat{B}\rangle)$.

2. When $\hat{a}_j$ and $\hat{b}_j$ are given, we can update the estimator of B by solving the problem below

$$\min_{B:\|B\|_F=1} \frac{1}{2n}\sum_{j=1}^{n}\sum_{i=1}^{n}[y_i - \hat{a}_j - \hat{b}_j(\langle x_i, B\rangle - \langle x_j, B\rangle)]^2 w(\frac{\langle x_i, B\rangle - \langle x_j, B\rangle}{h}) + \lambda\|B\|_* \tag{3.3}$$

The second step is that after $\hat{a}_j$ and $\hat{b}_j$ are given, we can update the estimator of parameter matrix B by solving equation (3.4)

$$\min_{\|B\|_F=1} \frac{1}{2n} \sum_{j=1}^{n} \sum_{i=1}^{n} (y_i - a_j - b_j(\langle X_i - X_j, B \rangle))^2 w_{ij} + \lambda \|B\|_* \tag{3.4}$$

This step is to update the $\hat{B}$ by solving the function (3.4). In this step, the equation (3.4) can be calculated through a fast iterative shrinkage-thresholding algorithm (FISTA) combing Nesterov method. This method has attracted much attention in recent years due to its efficiency in solving regularization problems (Beck and Teboulle, 2009). Because it resembles the classical gradient descent algorithm in that only the first order gradients of the objective function are utilized to produce next algorithm iterate from current search point. It differs from the gradient descent algorithm by extrapolating the previous two algorithmic iterates to generate the next search point. This extrapolation step incurs trivial computational cost but improves the convergence rate dramatically. A simple iterative shrinkage-thresholding has a worst-case complexity result of $O(1/k)$, where k represents the iteration steps. However the fast iterative shrinkage thresholding has complexity result of $O(1/k^2)$.

3.3.1 The Nesterov Method

Next, we will introduce the FISTA algorithms. At first ,we define the loss function with regularization $F(X)$.

$$F(X) = f(X) + g(X)$$

.

$f(X) : \mathbb{R}^n \to \mathbb{R}$ is a smooth convex function of type $C^{1,1}$, i.e., continuously differentiable with Lipschitz continuous gradient $H(f)$:

$$\|\nabla f(x) - \nabla f(y)\| \leq H(f)\|x - y\| \text{ for every } x, y \in R^n$$

where $\|\cdot\|$ denotes the standard Euclidean norm and $H(f) > 0$ is the Lipschitz constant of ∇f.
$g(X) : \mathbb{R}^n \to \mathbb{R}$ is a continuous convex function which is possibly non-smooth.

For example, when $g(X) = 0$, $F(X)$ is the general unconstrained smooth convex minimization problem. For any $L > 0$, consider the following quadratic approximation of $F(X) = f(X) + g(X)$ at a given point y:

$$Q_L(X, y) = f(y) + \langle X - y, \nabla f(y) \rangle + \frac{L}{2}\|X - y\|^2 + g(x) \tag{3.5}$$

which admits a unique minimizer

$$P_L(y) = argmin\{Q_L(x, y) : x \in R^n\} \tag{3.6}$$

in detail, we can have

$$p_L(y) = argmin_x\{g(x) + \frac{L}{2}\|x - (y - \frac{1}{L}\nabla f(y))\|^2\}$$

Beck and Teboulle (2009) has proved that key recursive relation for the sequence $\{F(x_k) - F(x_*)\}$, x_* is the true parameter, will imply the better complexity rate $O(1/k^2)$. We just need to replace order 1 tensor data (vector) $X \in \mathbb{R}^n$ in the algorithm to order 2 tensor data (matrix) $X \in \mathbb{R}^{m \times n}$ and $B_t \in R^{m \times n}$ in order to apply this algorithm to SIM-nuclear.

Define these notations that are used in algorithm

$$\begin{aligned}
l(B) &= \sum_{j=1}^{n}\sum_{i=1}^{n}\frac{1}{2n}(y_j - a_j - b_j\langle B, x_i - x_j\rangle)^2 w_{ij}, \\
\nabla l(B) &= \frac{\partial l(B)}{\partial B} = -\sum_{j=1}^{n}\sum_{i=1}^{n}\frac{1}{n}(y_j - a_j - b_j\langle B, x_i - x_j\rangle)b_j(x_i - x_j)w_{ij}, \\
J(B) &= \lambda\|B\|_*, \\
h(B) &= l(B) + J(B), \\
g(B|S_t, \delta) &= l(S_t) + \langle \nabla l(S_t), B - S_t\rangle + \frac{1}{2\delta}\|B - S_t\|_F^2 + J(B).
\end{aligned}$$

Then the process of Nesterov method is as follows. There are three important steps in Nesterov method for estimating the parameter B.

Table 3.2: The Algorithm of Estimating B

Initialize
  $B_0 = B_1$
  $\delta > 0$
  $\alpha_0 = 0$
  $\alpha_1 = 1$
Repeat
  $S_t = B_t + \frac{a_{t-1}}{a_t}(B_t - B_{t-1})$
  $A_{temp} = S_t - \delta \nabla l(S_t)$
  compute SVD $A_{temp} = Udiag(a)V^T$
  a_i is the ith eigenvalue of A_{temp}
  $b = \text{argmin} \frac{1}{2\delta}\|X - A_{temp}\|_2^2 + \lambda\|X\|_*$
  So the value of $b_i = (a_i - \lambda\delta)^+$
  $B_{temp} = Udiag(b)V^T$
  While$(h(B_{temp}) > g(B_{temp}|S_t, \delta))$
    $\delta = \frac{\delta}{2}$
    $A_{temp} = S_t - \delta \nabla l(S_t)$
    $A_{temp} = Udiag(a)V^T$
    $b = \text{argmin} \frac{1}{2\delta}\|X - A_{temp}\|_2^2 + \lambda\|X\|_*$
    $b_i = (a_i - \lambda\delta)^+$
    $B_{temp} = Udiag(b)V^T$
  if $h(B_{temp}) \leq h(B_t)$
    $B_{t+1} = B_{(temp)}$
  else
    $B_{t+1} = B_t$
  $\alpha_{t+1} = (1 + \sqrt{1 + (2\alpha_t)^2}/2)$
Until objective value converges.

1. Predict a search point **S** by a linear extrapolation from previous two iterates (Line 7 of Table 3.2).

2. Perform gradient descent from the search point **S** possibly with Armijo type line search (Lines 8-20 of Table 3.2).

3. Force the descent property of the next iterate (Lines 21-24 of Table 3.2).

In step 1, α_t is a crucial hyperparameter in the extrapolation. a_t is updated by each iteration by the equation $a_t = (1 + \sqrt{1 + (2\alpha_{t-1})^2}/2)$ (Line 25 of Table 3.2), other sequences, for example $\alpha_t = (t-1)/(t+2)$, can also be used.

In step 2, the gradient descent is based on the first order approximation to the loss function at the current search point S_t

$$\begin{aligned} g(B_{temp}|S_t, \delta) &= l(S_t) + \langle \nabla l(S_t, B - S_t) \rangle + \frac{1}{2\delta}\|B - S_t\|_F^2 + J(B) \\ &= \frac{1}{2\delta}\|B - [S_t - \delta \nabla l(S_t)]\|_F^2 + J(B) + c, \end{aligned} \tag{3.7}$$

where the variable δ is determined during the loop while $h(B_{temp}) > g(B_{temp}|S_t, \delta)$ is true and the constant c contains terms irrelevant to the optimization. $(2\delta)^{-1}\|B - S_t\|_F^2$ can shrink the next iterate towards S_t. If the loss function $l(\cdot)$ denotes the class of functions that are convex, continuously differentiable and the gradient satisfies $\|\nabla l(u) - \nabla l(v)\| \leq L(l)\|u - v\|$ with a unknown gradient Lipschitz constant $L(l)$. Then δ is updated dynamically to capture the unknown $l()$ by using the classical Armijo line search rule. $g(\cdot)$ is the surrogate function and Zhou and Li (2014) proved that the solution of $g(\cdot)$ is the same as the solution of function

$$\min_b \frac{1}{2\delta}\|b - a\|_2^2 + f(b), \tag{3.8}$$

where b's are the matrix B's singular values, a's are the matrix $[S_t - \delta \nabla l(S_t)]$'s singular values and $f(b) = \lambda \sum_j |b_j|$, the summation of the singular values of B because the penalty is nuclear norm. Then the solution of function (3.8) is $b_i = (a_i - \lambda\delta)_+$. For this reason, single value decomposition is performed on the intermediate matrix $A_{temp} = S_t - \delta \nabla l(S_t)$. The next iterate

B_{t+1} shares the same singular vectors as A_{temp} and the singluar values b_{t+1} are determined by minimizing $\frac{1}{2\delta}\|b-a\|_2^2 + f(b)$.

For minimization of a smooth convex function l. It is well-known that the Nesterov method is optimal with the convergence rate at order $O(t^{-2})$, where t indicates the iterate number. In contrast, the gradient descent has a slower convergence rate of $O(t^{-1})$. The SIM-nuclear is non-smooth, but the same convergence result can be established.

3.3.2 The Advantages of Nesterov Method

There are some common questions in gradient descent.

The first question is non-convex loss function. The Nesterov method and its convergence properties hinge upon convexity of the loss l. But sometimes the loss function may be non-convex. The solution is the iteratively reweighted least squares strategy can be applied in this scenario. At each IWLS step, the Nesterov method is used to solve the penalized weighted least squares problem, which is convex.

The second problem is whether the monotonicity of the objective function during iterations. Because of the extrapolation step, the objective values of algorithmic iterates $f(B_t)$ are not guaranteed to be monotonically decreasing. The solution is that we added one checking point in the algorithm, only if the objective function of B_{temp} smaller than the objective function of B_t, then we updated B_{t+1} by B_t.

The last question is to estimate the Lipschitz constant L for the GLM loss. Each step halving the line search part of algorithm involves an expensive singular value decomposition. Therefore even a rought initial estimate of L potentially cuts the computational cost significantly. Recall based on the table 3.2, the estimate of parameter matrix B can be updated by standardizing the result, $\hat{B}_t = \hat{B}_t/\|\hat{B}_t\|_F$, then $\hat{B}_t$ is plugged back in equation (2.4), then just repeat the process until the estimator of B converges.

3.4 Implementation

3.4.1 Selection of Kernel Function

Before we select the weight, we need to select the kernel function at first. In nonparametric statistics, a kernel is a weighting function used in non-parametric estimation. In general there are several requirements that every kernel function needs to satisfy.

- Normalization:

$$\int_{-\infty}^{\infty} K(u)du = 1$$

- Symmetry about the origin:

$$\int xK(x)dx = 0 \text{ for all values of } x$$

- Finite second moment:

$$\int x^2K(x)dx < \infty$$

The first requirement ensures that the method of kernel density estimation results in a probability density function. The second requirement ensures that the kernel function is an odd function. Various kernel functions can be used as $K(\cdot)$, we just introduce two of them.
Epanechnikov kernel function:

$$K(u) = \frac{3}{4}(1-u^2) \qquad |u| \leq 1$$

Gaussian kernel function:

$$K(u) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}u^2} \qquad |u| \leq \infty$$

The Gaussian kernel function is chosen in this dissertation. There are some advantages for selecting the guassian kernel function.

- One advantage is the Epanechnikov kernel has 3 derivatives before it's identically zero, unlike the Gaussian which has infinitely many (nonzero) derivatives. Furthermore, the first derivative of the Epanechnikov kernel is not continuous where the function crosses the kernel's own bounds.

- The other advantage is Gaussian kernel function is more efficiency. The performance of kernel is measured by mean integrated squared error (MISE) or asymptotic mean integrated squared error (AMISE). The smaller these values are, the better the kernel function performs. If we restrict attention to kernels that are probability density functions, the optimal kernel is the Epanechnikov kernel. We can define the relative efficiency of other kernels compared with the Epanechnikov kernel as the ration of their values of $\sqrt{\int u^2 K(u)du} \int K(u)^2 du$. The efficiency of the Gaussian is about 95%, the relative efficiency of the other kernel function whose domain is $(-\infty, \infty)$ are lower than 95%, for example, the relative efficiency of the logistic kernel function is about 88.7%, the relative efficiency of the sigmoid kernel function is about 84.3%.

In this dissertation, we select the Gaussian kernel function.

3.5 Selection of hyper-parameters

In this part, we discuss how to select the hyper-parameters in the model. Hyper-parameter is a terminology in the machine learning, which represents the parameters in the model that can not be updated by the loss function automatically. There are two hyper-parameters in the model, one is the bandwidth, h. The other is the tuning parameter of the penalty function, λ. We discuss how to select these two hyper-parameters one by one.

In general, the m-fold cross validation procedure can be used for simultaneously selecting the bandwidth h and the tuning parameter of the penalty, λ.

3.5.1 Selection of Bandwidth h

Selecting a proper bandwidth h for an observed data sample is a crucial problem, because of the effect of the bandwidth on the performance of the corresponding estimator. If the bandwidth is

small, the estimator will be under smooth, the variability will be very high. On the contrary, if the value of h is big, the resulting estimator will be over smooth and farther from the function that we are trying to estimate. Guidoum (2015) use Figure 3.1 to help us understand the effect

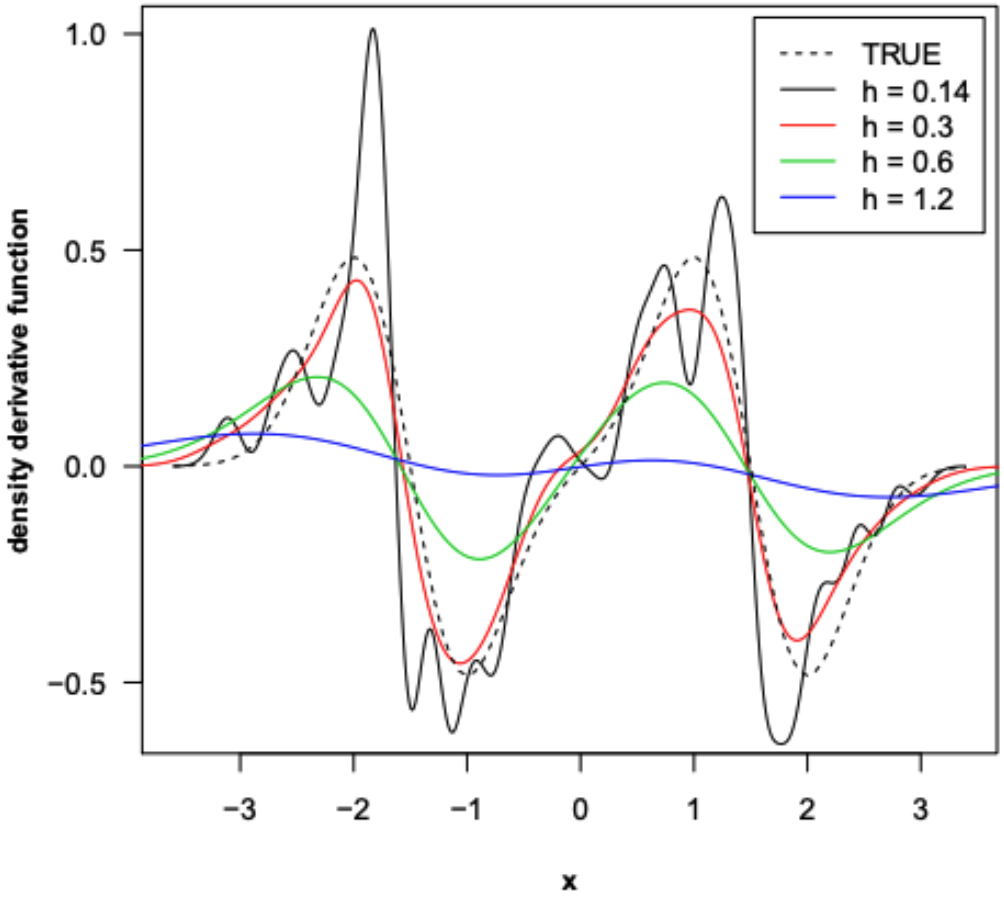


Figure 3.1: Effect of the bandwidth on the kernel estimator

of the bandwidth on the kernel estimator. If the value of h is small, then the graph of the kernel function will have more fluctuations or tides. If the value of h is big, the graph will be more flat. The choice of bandwidth h is much more important than the choice of kernel function $K(\cdot)$. There is a trade-off of selecting the value of h. If h is small, the bias can be reduced. If h is large, the curve will be smoother. We need to select an optimal bandwidth. A natural metric to use is mean squared error (MSE), the sum of squared bias and variance. Intuitively MSE is minimized by choosing h. Since our proposed estimation procedure is iterative in nature, GCV type of bandwidth selection methods can be computationally intensive when estimating

B. From extensive simulation study, we have found that the rule of thumb for bandwidth selection suggested by Silverman (1986) works reasonably well for SIM-nuclear. Let s be the median of the standard deviations of the predictor. The rule of thumb suggests to choose h as $h = 1.06sN^{-1/5}$. We recommend to use this bandwidth when applying sim-nuclear.

3.5.2 Selection of tuning parameter in the penalty function

The λ is contained in the penalty part of the single index model for the tensor data. The penalty part of the function can control the complexity of the model. Our goal is to make a trade-off between how well the data is fit and how complex the model is. The λ parameter controls how much emphasis is given to the penalty term. The larger the λ value, the more coefficients in the model will be pushed towards zero. So the model will be simplified. But with large bias and underfitting problem. The smaller the λ value, the model will fit the data better, but with high variance and overfitting problem.

After h is specified, because of the iterative nature of the proposed estimation method, standard methods, such as CV and GCV, are computationally intensive. By the discussion of Fan and Li (2001b), when $\lambda = \sqrt{Clogn/n\sigma}$, the nonconcave penalized least squares method has the so-called oracle property, namely, the zero coefficients are set to zero automatically and the nonzero coefficients can be estimated efficiently as if the true model was known. We will set $\sqrt{Clogn/n\sigma}$ as the initial value of λ and use the 10-CV procedure to select λ. In general,the 10-CV procedure results in an optimal penalty parameter, denoted by λ_0, for the estimation of B.

Chapter 4

Simulation

In this chapter, we illustrate the performance of the SIM-nuclear by several simulation studies. In this chapter, we define the parameter B as a sparse and low-rank order 2 tensor parameter. We will conduct three simulations.

The first one is we test whether SIM-nuclear is asymptotic consistent or not and whether the recommendation about selecting the bandwidth h and λ are valid. The second one is comparing the estimator $\hat{B}$ and the true matrix B graphically, and the last one is comparing SIM-nuclear with other methods such as linear regression, linear lasso and SIM-lasso.

4.1 The Criteria

To evaluate the results of different experiments, we use two criterions.

One criterion is $A(\hat{B}, B) = (180/\pi)\arccos(|vec(\hat{B})^T vec(B))|)$, which is the angle (in degree) between the vectorized estimator $\hat{B}$ and the vectorized true parameter B. Note that $A(\hat{B}, B) \in [0, 90]$, and $A(\hat{B}, B)$ is equal to 90 when $\hat{B}$ and B are perpendicular to each other. The performance of an estimate $\hat{B}$ is measured by $A(\hat{B}, B)$, with small values indicating good performance.

The other criterion is $L(\hat{B}, B) = \|\hat{B} - B\|_F$, in which $\|\cdot\|_F$ is the Frobenius norm of a matrix. The Frobenius norm is the l_2-norm of matrix norm, the smaller value, the better performance.

4.2 Demonstration of proposed methods

In this section, we demonstrate the single index model for nuclear norm works. The model we consider joined by

$$y = 3 \cdot \sin(0.25\langle x, B\rangle) + \epsilon \tag{4.1}$$

in which, $\epsilon \sim N(0, 1)$ and $B \in R^{5\times5}$, which is

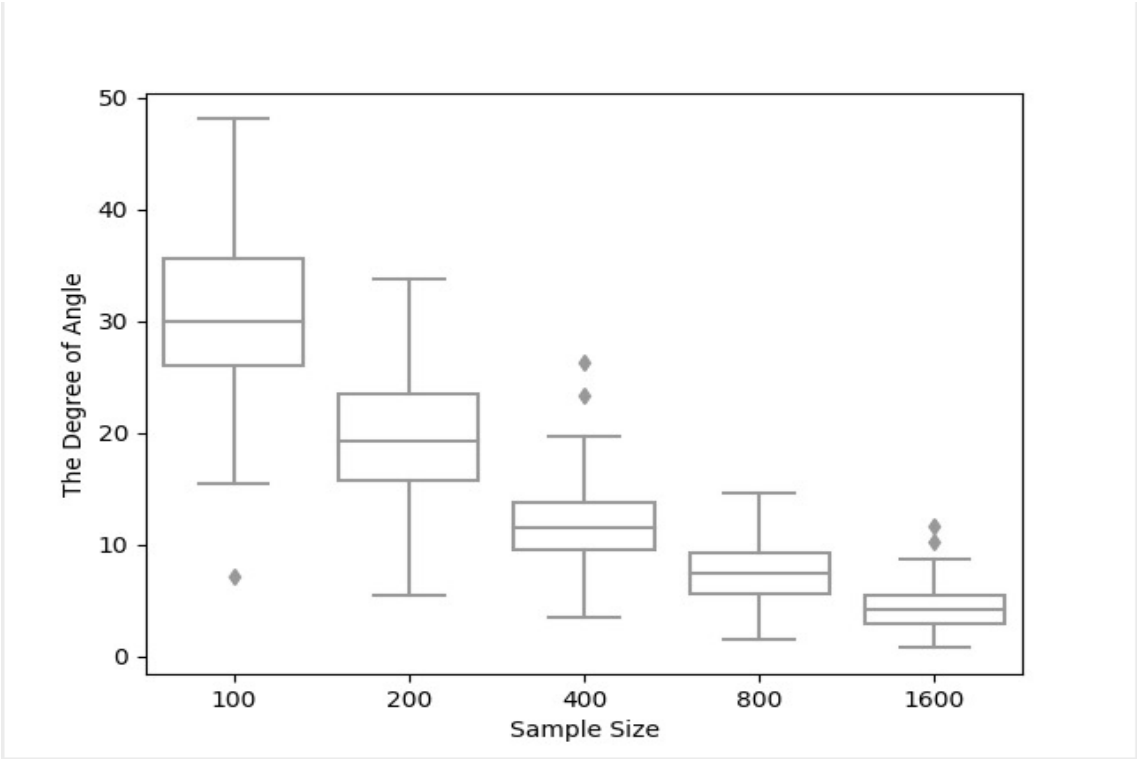
$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}_{5\times5}$$

In our first simulation the parameter tensor size is 5×5, the rank of it is just 1. So it is a low-rank matrix. We generated $X \in R^{5\times5}$ by getting random values from the distribution $N(10, 1)$ to be elements in the X.

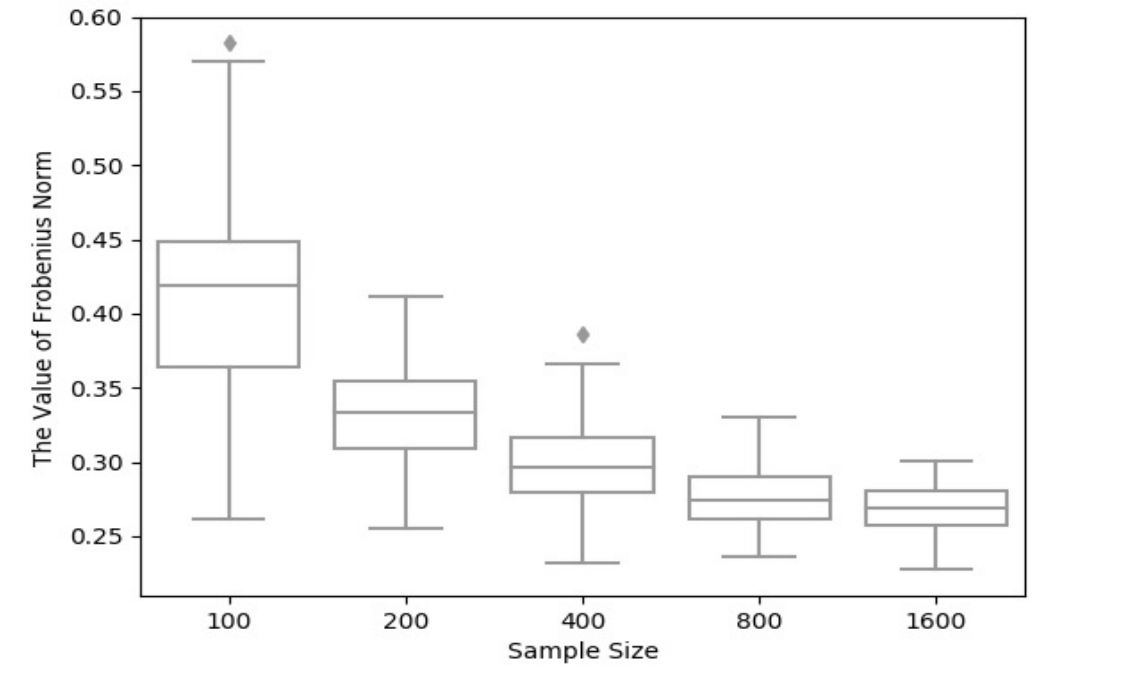
4.2.1 Sample size increasing

In this subsection, we use the function (4.1) to generate the data in several simulations. Each simulation contains 100 experiments, but the data size of the experiments from these simulations is different from each other. There are 5 simulations, the experiment from different simulation uses different sample size. For example, in the first simulation, its experiment's sample size is 100. In the second simulation, its experiment's sample size is 200, and so on. When we got the estimator $\hat{B}$ in a certain experiment, we calculated the degree of angle between B and $\hat{B}$ and the Frobenius norm of $\hat{B} - B$. Then we used the boxplot to describe these 100 results(because we have 100 experiments in each simulation). The results are below:

In Figure 4.1, there are 5 types of sample-size, from 100 to 1600. Both graphs told us that as the sample size increases, the height of boxplot will go down and tends to zero, in other words, the values of degree of angle and the values of Frobenius norm tend to decrease from



(a)



(b)

Figure 4.1: a) $\|B - \hat{B}\|_F$ and b) The Angle between B and $\hat{B}$

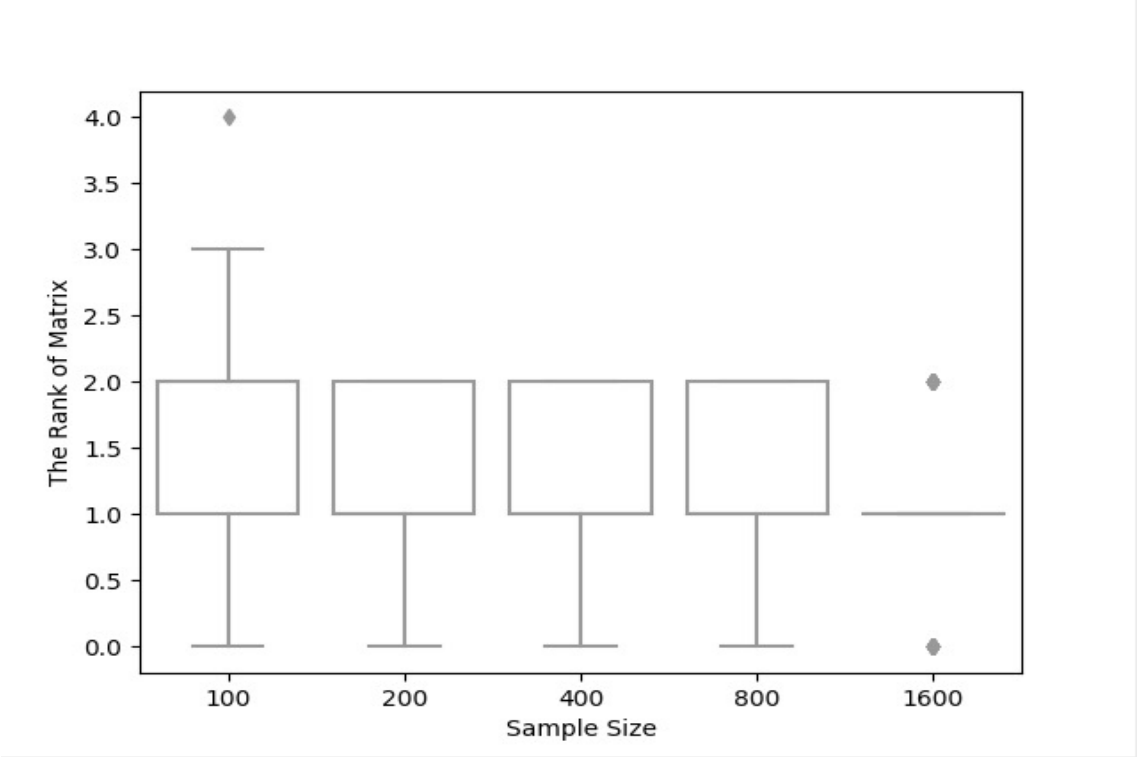


Figure 4.2: The Rank of the Matrix $\hat{B}$

left to right. Figure 4.2 indicates the rank of $\hat{B}$ as the simple size increases, we can see as the simple size increases the rank of $\hat{B}$ is lower and tending to 1. It means the single index model with nuclear norm is asymptotic consistent.

4.2.2 The Impact of Different Bandwidth Values

In this simulation, we need to test whether the bandwidth selected by the rule of thumb can make the degree of angle between $\hat{B}$ and B and the values of Frobenius norm of $\hat{B} - B$ minimum or not. Because the rule of thumb suggests $h = O(n^{-1/5})$, under the condition $n = 200$, the value of bandwidth h is around $0.3 \sim 0.4$.

From Figure 4.3, we can see that the value of angle degree and the value of Frobenius norm are the smallest when $h = 0.3$. Based on this simulation, we got the practical optimal bandwidth is $h = 0.3$. It is very close to the theoretical optimal bandwidth.

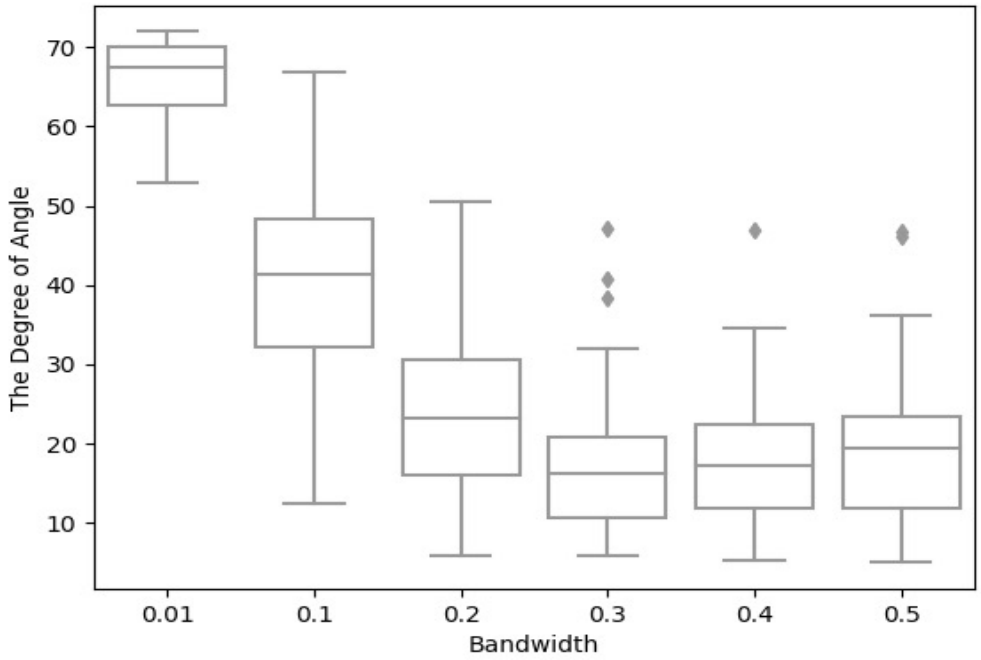
4.2.3 The impact of λ values

In this section, we want to demonstrate the impact of λ. Theoretically the value of λ can affect the parameter, the larger the λ is, the more parameters tends to zero. Smaller λ will have high variance, but low bias. On the contrary, larger λ will lead to low variance and high bias. So selecting a proper λ value is very important. But we need to determine one bandwidth, h, at first. In this subsection, the bandwidth is selected by the rule-of-thumb, which is $O(n^{-1/5})$. Because our sample size is 200, we choose $h = 0.3$.

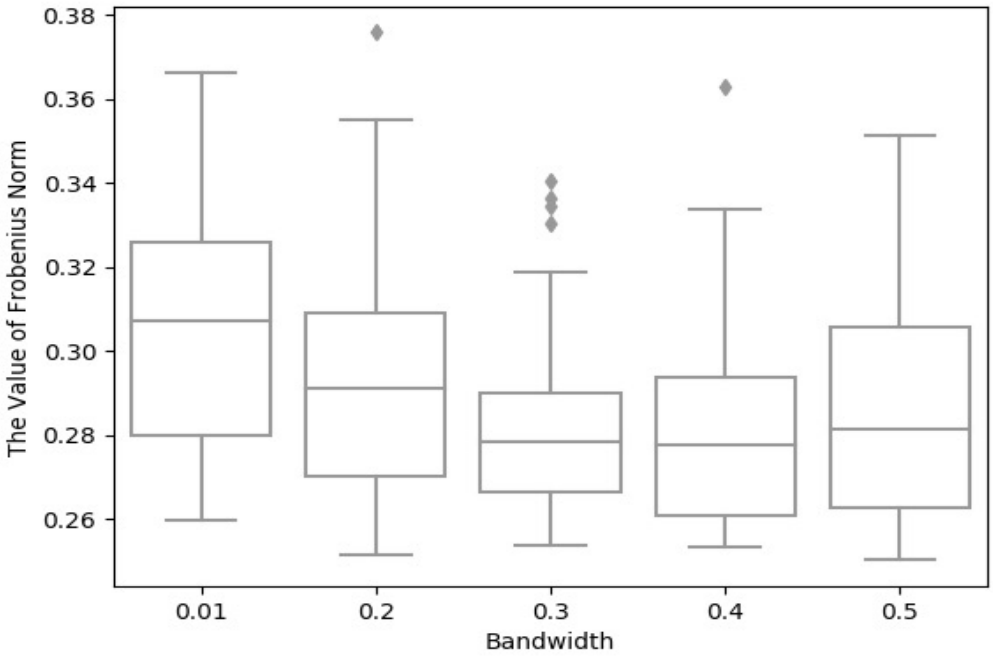
We got Figure 4.4. From Figure 4.4, we can find when the value of λ is around 5, the box-plot is lowest, which means the degree of angle is smallest around this value. We can get the conclusion, the optimal λ value is near 5. The Frobenius norm of $\hat{B} - B$ also shows that when the λ is 5, the average is minimum. So in this section, we found the optimal λ based on a fixed bandwidth $h = 0.3$.

4.3 Comparison of $\hat{B}$ and B

In previous section, we discussed the effect of different hyper-parameters, such as sample size, bandwidth and tuning parameter of penalty function, to the performance of the algorithm for the SIM-nuclear. We use the degree of angle and Frobenius norm to evaluate the performance of the estimator. This method is too abstract that we can not check the performance of the estimator directly. In this section, we use the property of order 2 tensor to create the pictures

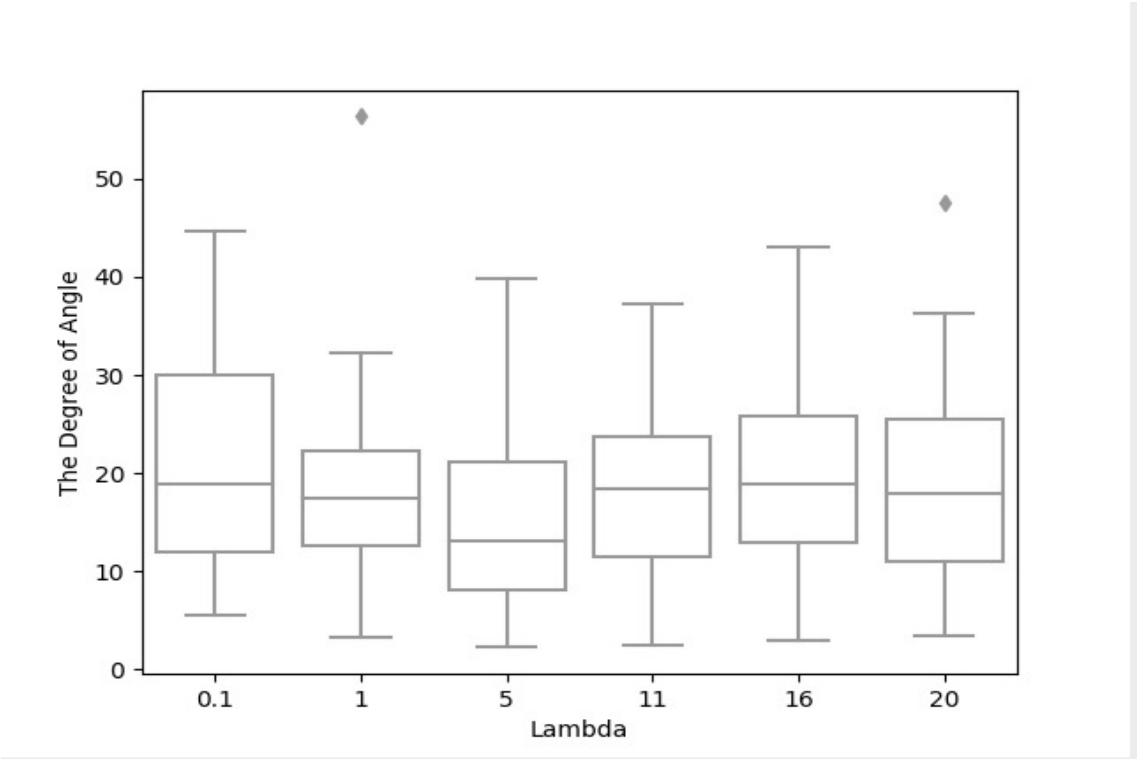


(a)

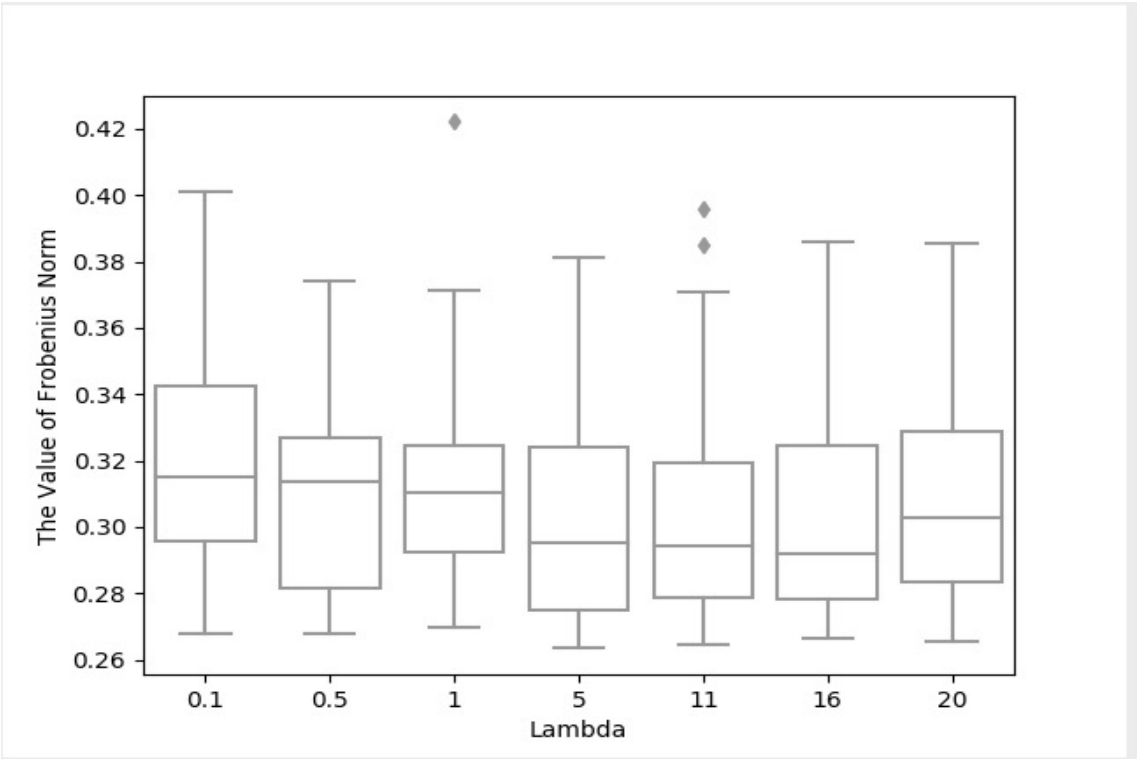


(b)

Figure 4.3: a) $\|B - \hat{B}\|_F$ and b) The Angle between B and $\hat{B}$



(a)



(b)

Figure 4.4: a) $\|B - \hat{B}\|_F$ and b) The Angle between B and $\hat{B}$

corresponding to the estimator, $\hat{B}$, and the true parameter matrix, B. Then we can compare with them visually.

There are several parameter matrices B, 9×9 and low-rank. We use three plots to illuminate the performance of the SIM-nuclear algorithm. On each row, the first plot is the true parameter matrix, the second one is the scatter plot created by the points $(y_i, \langle x_i, B \rangle)$, $i = 1, \cdots, 100$. In the third, we show the estimate of parameter matrix estimated by SIM-nuclear.

In this example, we have four parameter matrices. There are number, simple shape and Roman letters. From the scatter plots, the relationship between y and $\langle x, B \rangle$ are demonstrated to be nonlinear. Under this situation, we compared the estimator with the true, although the estimate is a little blurred, it can still describe the correct shape.

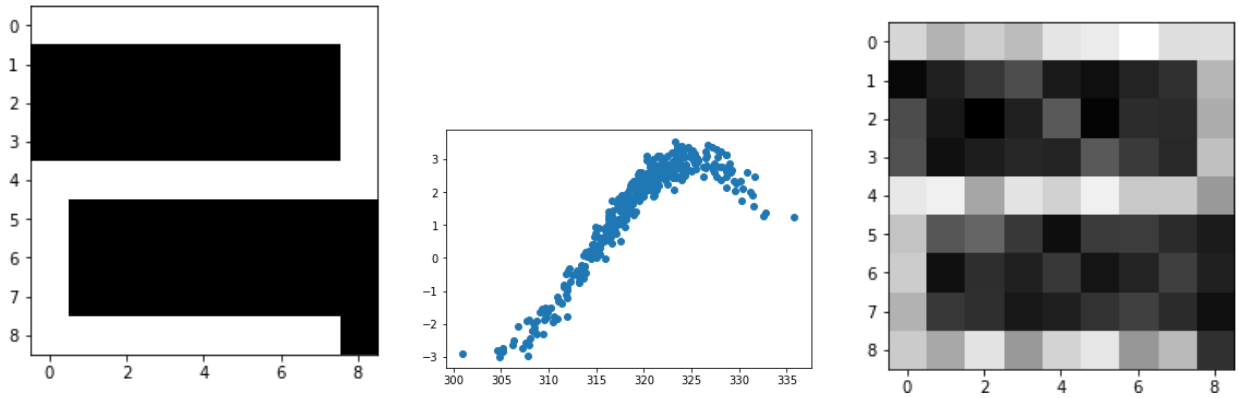


Figure 4.5: Parameter matrix of shape 2 (R=3, Angle Value=21.5965)

In Figure 4.5, the true parameter matrix is number 2, and the estimator of parameter is still 2, even it is a little blurred. In Figure 4.6, the true parameter matrix is letter 'F', and the estimator is 'F'. So we get a good estimator. In Figure 4.7, the true parameter matrix is letter 'H'. Although the scatter plot is non-linear, the estimator is a letter 'H' obviously. In Figure 4.8, the scatter plot shows there is a nearly linear relationship between y and $\langle X, B \rangle$. And the estimator is almost the same as the true one.

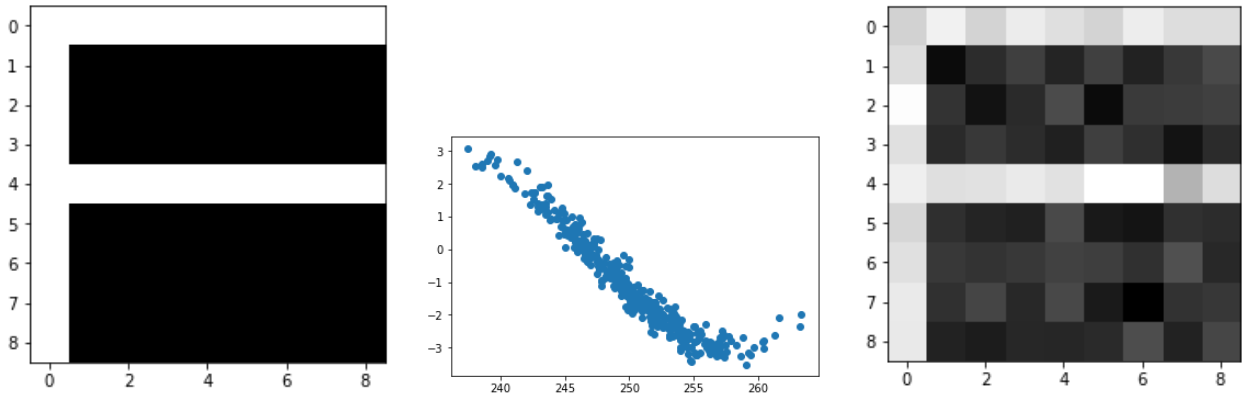


Figure 4.6: Parameter matrix of letter F (R=2, Angle Value=18.506)

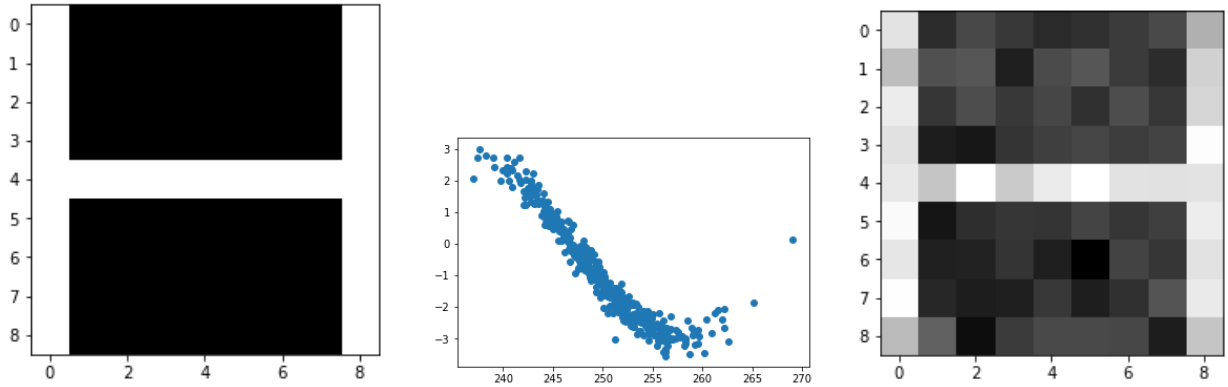


Figure 4.7: Parameter matrix of letter H (R=2, Angle Value=16.414)

4.4 Comparison of Different Models

In this section, we will compare the performances of sim-lasso, sim-nuclear, linear matrix regression and linear lasso. To eliminate the influence of the selection of bandwidth, h, and λ. We compare the best possible performances of these methods under the optimal h and λ. In these models, we use the same B,

$$B = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}_{4\times 4}$$

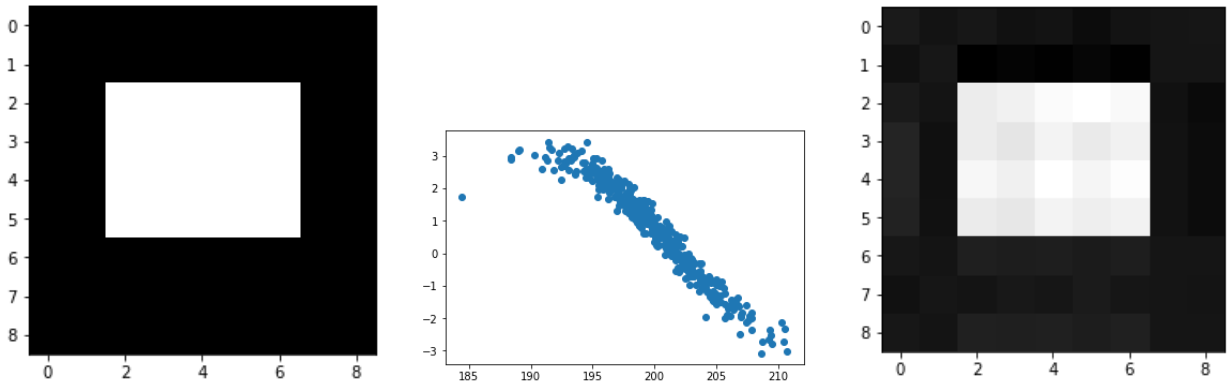


Figure 4.8: Parameter matrix of Shape Rectangle (R=1, Angle Value=8.104)

Each element of X is generated by N(10,1) randomly, and ϵ follows normal distribution whose mean is 0 and variance is 1. Randomly generate 100 experiments for each model, each experiment has a size of $n = 200$ samples.

$$\text{Model 1} : y = 3 \cdot \sin(0.25\langle x, B\rangle) + \epsilon$$

$$\text{Model 2} : y = 2(\langle x, B\rangle)^2 + exp(\langle x, B\rangle) + \epsilon$$

$$\text{Model 3} : y = 3(\langle x, B\rangle)cos(\langle x, B\rangle) + \epsilon$$

$$\text{Model 4} : y = 3(\langle x, B\rangle)^2 + (\langle x, B\rangle) + \epsilon$$

The linear matrix regression is proposed by Zhou and Li (2014), they refer the equation (1.2) as the loss function of linear matrix regression problem with nuclear norm penalty function. Then we introduce the SIM-lasso briefly.

4.4.1 SIM-lasso

It is proposed by Zeng et al. (2012). Suppose $\{(y_i, x_i), i = 1, \cdots, n\}$ is a random sample drawn from a single index model

$$y = g(\theta^T x) + \epsilon$$

Table 4.1: The degree of angle under $n = 200$

Method	Mean	Std. dev.	h	λ
Model 1				
SIM-nuclear	11.253	2.712	0.3	11
SIM-lasso	16.797	4.957	0.6	0.02
Linear Nuclear	19.0906	0.902	NA	24
Linear Lasso	19.722	3.114	NA	0.001
Model 2				
SIM-nuclear	4.114	0.038	0.3	11
SIM-lasso	3.085	0.019	0.690	0.02
Linear Nuclear	29.452	0.813	NA	30
Linear Lasso	24.314	6.878	NA	0.02
Model 3				
SIM-nuclear	21.307	2.554	0.4	12
SIM-lasso	24.945	3.957	0.705	0.04
Linear Nuclear	56.344	9.647	NA	34
Linear Lasso	76.349	10.071	NA	0.001
Model 4				
SIM-nuclear	5.196	0.178	0.3	11
SIM-lasso	3.789	0.15	0.675	0.1
Linear Nuclear	16.839	0.372	NA	25
Linear Lasso	14.397	0.408	NA	0.002

where $\theta \in \mathbb{R}^p$, $x \in \mathbb{R}^p$. Zeng proposed the penalized minimization problem about this single index model is:

$$\min_{a,b,\theta,\|\theta\|=1} \sum_{j=1}^{n}\sum_{i=1}^{n}[y_i - a_j - b_j\theta^T(x_i - x_j)]^2 w_{ij} + \lambda \sum_{j=1}^{n}|b_j| \sum_{k=1}^{p}|\theta_k|. \tag{4.2}$$

Where all the notations are mentioned in previous chapters. The penalized loss function (4.2) is refered as the sim-lasso model. This method is fit for the vector parameter. If we want to use it, we should vectorize the matrix parameter B firstly.

Table 4.2: The degree of angle under $n = 400$

Method	Mean	Std. dev.	h	λ
Model 1				
SIM-nuclear	8.204	1.796	0.3	11
SIM-lasso	8.967	1.785	0.674	0.02
Linear Nuclear	9.382	0.8613	NA	25
Linear Lasso	10.159	1.881	NA	0.01
Model 2				
SIM-nuclear	3.271	0.733	0.4	12
SIM-lasso	1.024	0.294	0.668	0.02
Linear Nuclear	24.152	1.24	NA	35
Linear Lasso	20.464	5.864	NA	0.02
Model 3				
SIM-nuclear	11.679	1.517	0.3	12
SIM-lasso	13.049	1.383	0.604	0.01
Linear Nuclear	46.176	6.125	NA	38
Linear Lasso	67.993	7.328	NA	0.01
Model 4				
SIM-nuclear	2.344	0.076	0.3	11
SIM-lasso	1.045	0.018	0.665	0.02
Linear Nuclear	13.771	0.370	NA	42
Linear Lasso	10.092	0.326	NA	0.002

From Table 4.1 and 4.2, the degree of angle between the estimator $\hat{B}$ and the true B is smaller as the sample size increases. Focused on either one of two tables, we can see the performances of SIM-nuclear and SIM-lasso are better than the traditional methods, linear nuclear and linear lasso in these four models. If the data are generated from model 1 and 3, the estimation result of SIM-nuclear is better than the estimation result of SIM-lasso. On the contrary, if the data are generated from the rest two models, the SIM-lasso is better than the SIM-nuclear.

Chapter 5

Real Data

In the chapter, we use SIM-nuclear method to solve a real world problem, and to test whether SIM-nuclear performs better than the traditional statistical method, logistical regression.

5.1 Background of Application

In this experiment, the data came from the magnetoencephalography (MEG). Magnetoencephalography (MEG) is a functional neuroimaging technique for mapping brain activity by recording magnetic fields produced by electrical currents occurring naturally in the brain, using very sensitive magnetometers. When a participant is scanned by MEG, the gantry can be moved so as to position the MEG scanner over the participant. The chair on which the participant sits is then raised up so that the participants head rests inside the helmet of the scanner(see Figure 5.1). Magnetoencephalography (MEG) can detect and record magnetic fields outside the head that are generated by electrical activity in the participant brain. The magnitude of these magnetic fields is of the order of femtotesla (10-15 T), which can be sensed by Magnetometers.

Arrays of SQUIDs (superconducting quantum interference devices) are currently the most common magnetometer, while the SERF (spin exchange relaxation-free) magnetometer is being investigated for future machines. Applications of MEG include basic research into perceptual and cognitive brain processes, localizing regions affected by pathology before surgical removal, determining the function of various parts of the brain, and neurofeedback. This can be applied in a clinical setting to find locations of abnormalities as well as in an experimental setting to simply measure brain activity.

Figure 5.1: A Person Undergoing An MEG

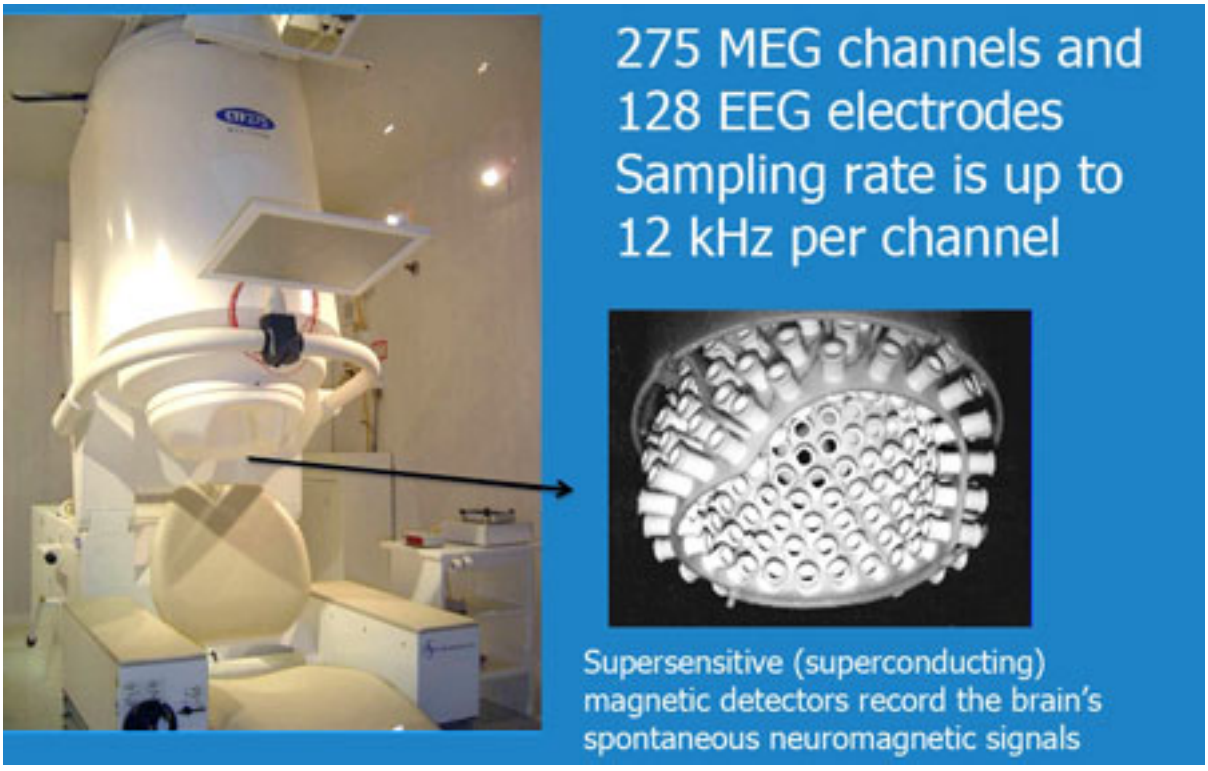


Figure 5.2: Sensor tunnels in the MEG

In the helmet of MEG, there are many sensor tunnels like Figure 5.2, which are supersensitive magnetic detectors that can record the brain's spontaneous neuromagnetic signals. This experiment whose data were recorded by MEG. The experiment lasted 277.77 $secs$. During this period of time, the participants were seated in front of screen and worn headphone, and given some visual and audio stimuli, such as Checkerboard patterns were presented into the left and right visual field, interspersed by tones to the left or right ear. So there are four types of stimuli. The interval between the stimuli was 750 ms. A listing of the corresponding trigger codes in provided.

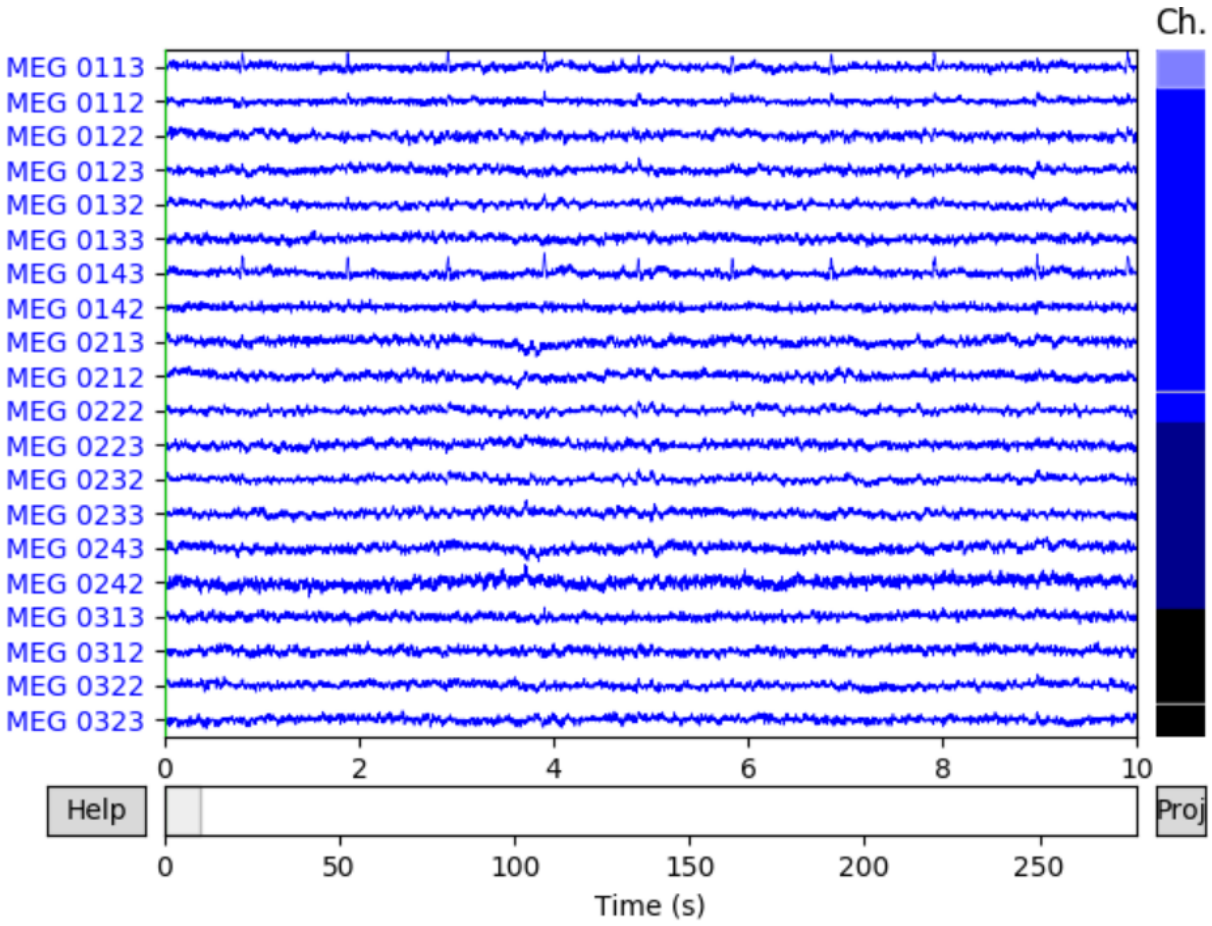


Figure 5.3: Raw data for Channels

The sample data set is recorded using a 306-channel Neuromag vectorview system (Figure 5.3). In Figure 5.3, we can see a timeslot of the experiment, the label values are color-coded. Each row represents the data recorded from a certain sensor in MEG. And these stimuli can be treated as order-2 tensor data. The stimuli are plotted as vertical lines so you can see how they align with the raw data (Figure 5.4) . On top of Figure 5.4, you can see the different types of

Table 5.1: Trigger codes for the sample data set

Name		Contents
A/L	1	Response to left-ear auditory stimulus
A/R	2	Response to right-ear auditory stimulus
V/L	3	Response to left visual field stimulus
V/R	4	Response to right visual field stimulus

stimulus affected the brain activity. When the stimulus occur, the curves of each row would fluctuate.

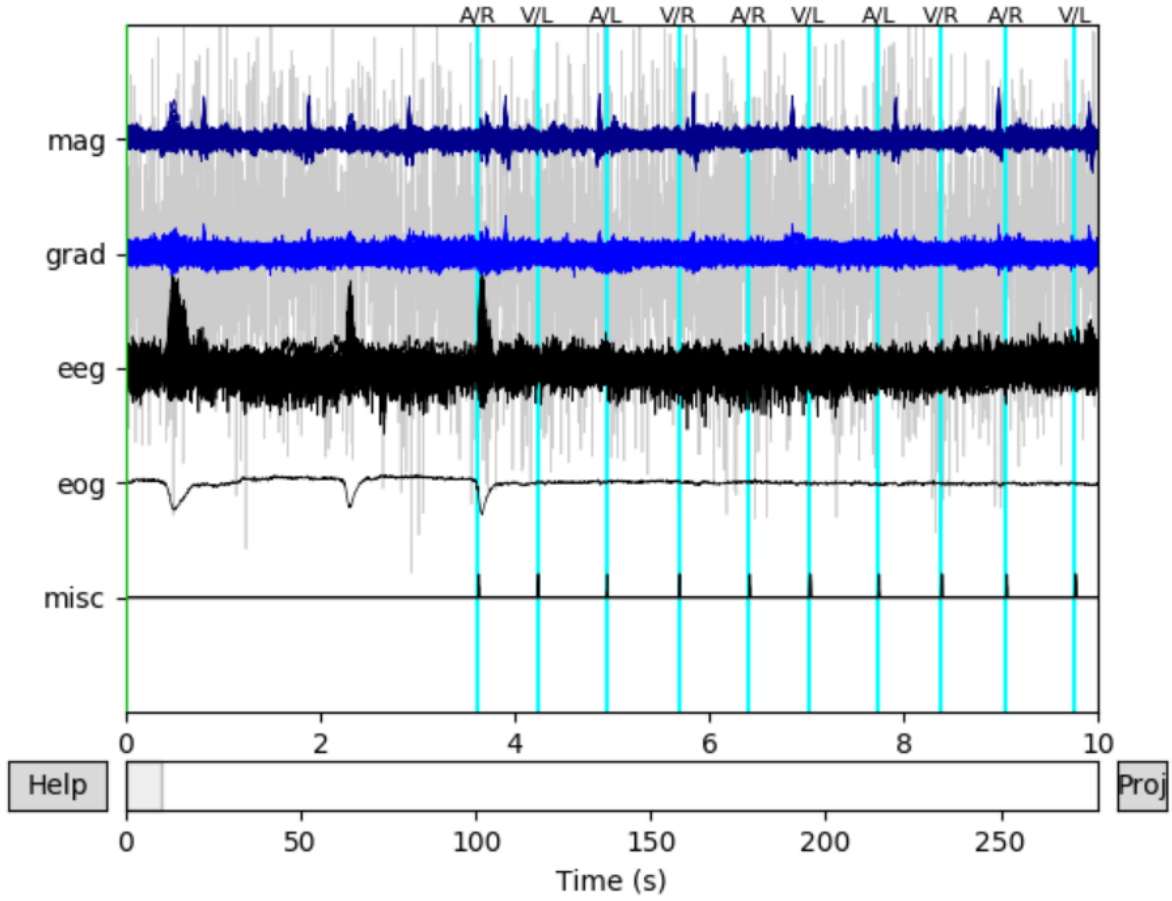


Figure 5.4: Apply the stimuli to Raw data

We began to record the brain activity started at 0.1s earlier than the stimulus and stopped recording 0.4s after it. Based on these data, we could plot the topomaps the magnetometers(figure 5.5) to restore the brain activity when the stimulus came. From the Figure 5.5, 0.093s means the topomap of the brain activity was 0.093s before the stimulus. In the 0.071s after the stimulus, the brain activity became more active obviously. After 0.234s of the stimulus, the brain returned to normal again. This is how the brain responsed to a stimulus

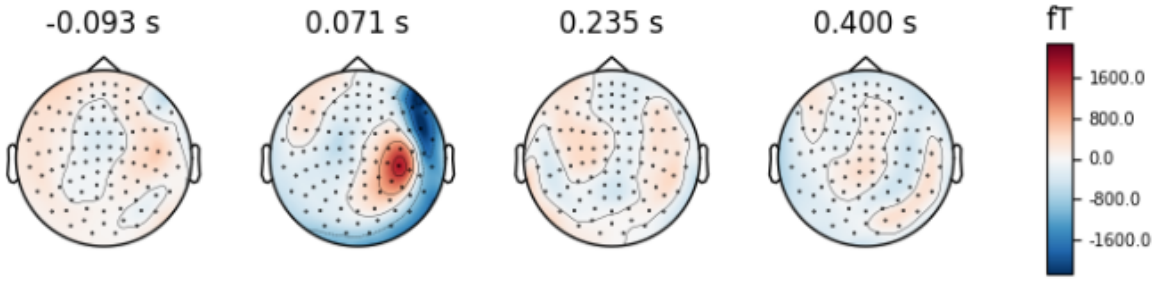


Figure 5.5: The brain response for a stimulus

Table 5.2: Misclassification error rate for MEG data

Method	The Mean of Error Rate
Logistic regression	0.052
SIM-nuclear	0.0392

In each stimulus, for every channel, we can record 38 values. For this reason, the resulting covariates x_i are 38×305 matrices, not 38×306, because there are some 'bad' channels(for example 'MEG 2443') and abandoned. 'Bad' means that the segment will be discorded when epoching the data. There are some reasons that can cause the discord, such as if a channel does not show a signal at all (flat), or if a channel as a noise level significantly higher than the other channels. The response y_i is a binary variable indicating whether the i-th subject is response to left-ear auditory stimulus ($y_i = 1$) or response to left visual field stimulus ($y_i = 3$), $i = 1, \cdots, 145$. There is a single index model between binary variable y_i and matrix variable $X_i \in R^{38\times305}$.

5.2 Data and Analysis

There are two methods applied to analyse the data. One is linear logistic regression, the other is SIM-nuclear. We applied the famous python machine learning module scikit-learn. The data is divided into two groups, training group and testing group by using the command sklearn.model_selection.train_test_split. The training group takes up 80% of the whole data, the test group takes up the rest 20%. We use the training data to select the optimal parameters and hyper-parameters. And the testing group is used to test the performance of each method.

But the SIM-nuclear is built for the quantative response variables, not for the binary response variables. In this experiment, the left-ear stimulus was treated as -1 and the left visual field stimulus was treated as 1. If $\hat{y}_i$ that was estimated by SIM-nuclear was greater than 0, this stimulus was treated as left visual field stimulus. On the contrary, a negative estimator $\hat{y}_i$ was put into the left-ear stimulus group. We use the misclassification error rate as the criterion.

$$\text{misclassification error rate} = \frac{\text{the number of wrong classifications}}{\text{the total number of data}}$$

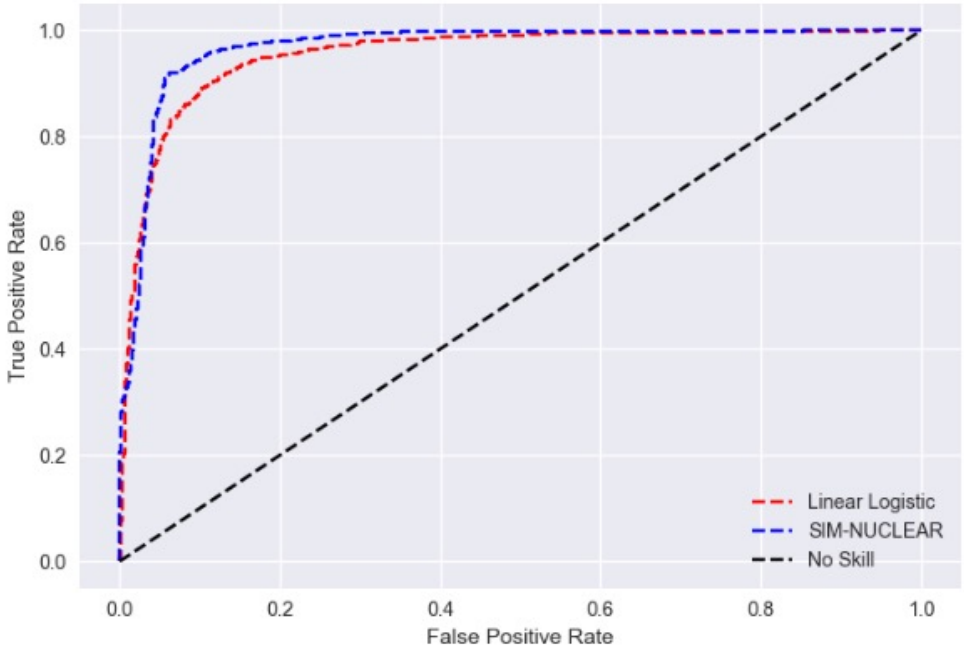


Figure 5.6: The ROC Curves of Two Methods

Table 5.2 showed the result. We can see the misclassification error rate of Logistic regression is about 0.05, the misclassification error rate of SIM-nuclear is about 0.04. And we draw the ROC curves about these two methods. From the figure 5.6, it is obvious that the area that the curve of SIM-nuclear covers larger area than the area that the curve of logisitic regression obtains. And no-skill classifier is one that cannot discriminate between the two classes and would predict a random class in all cases. A model with no skill is represented at the point (0.5,

0.5). A model with no skill at each threshold is represented by a diagonal line from the bottom left of the plot to the top right and has an AUC of 0.5. In this real data example, SIM-nuclear performs better

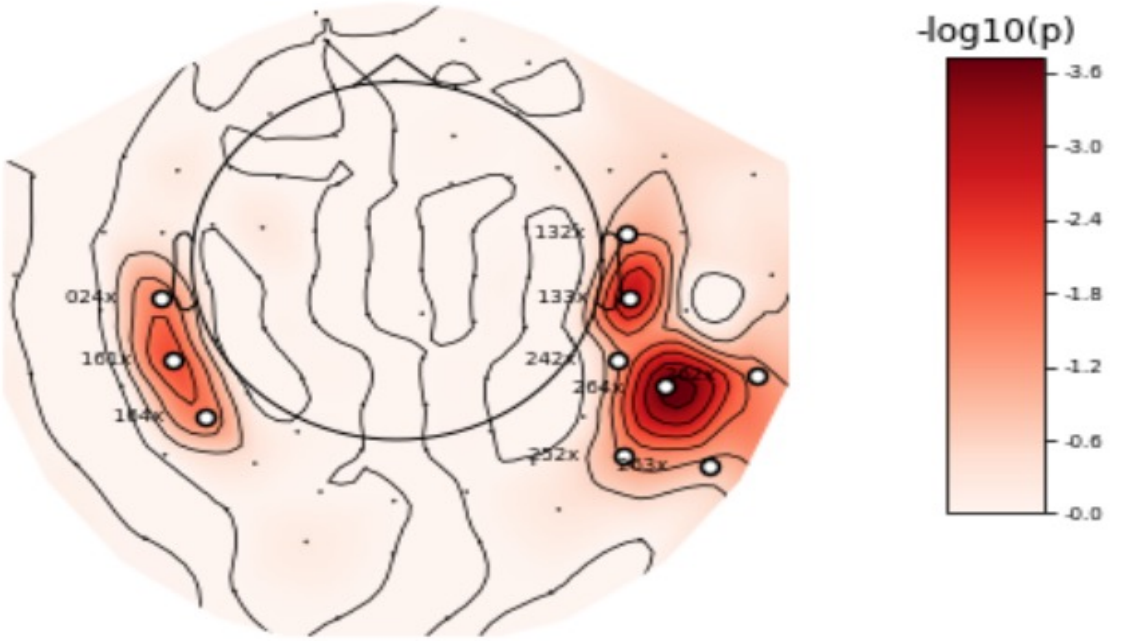


Figure 5.7: The T-test values

In the figure 5.7, a t-test is done to each element in the estimator matrix $\hat{B}$ to find which elements that deviates from 0 significantly.

- H_0: The value β_i corresponding to the ith sensor is equals to 0
- H_1: The value β_i corresponding to the ith sensor is not equals to 0

The p value of a sensor's t-test is applied to $-log_{10}$. The larger $-log_{10}(p)$, the darker red color. We can find the darkest red area in the figure 5.7, this part is the most important part in the matrix $\hat{B}$.

Chapter 6

Conclusion

6.1 Summary

In this dissertation, we proposed a new single index model in which the parameter is order 2 tensor and the penalty function is nuclear norm. This new single index model is named as single index model with nuclear norm penalty. Compared to the traditional single index model whose parameter θ is an order 1 tensor, the parameter of the SIM-nuclear is not an order 1 tensor, but an order 2 tensor.

For our proposed method, we simply use an alternating minimization algorithm to find the $\hat{B}$ and $\widehat{g(\cdot)}$ iteratively. Given an estimator $\hat{B}$, estimate $\hat{g}(\langle X, \hat{B}\rangle)$ and $\hat{g}^{'}(\langle X, \hat{B}\rangle)$ by minimizing a simple least square function. After $\hat{g}(\langle X, \hat{B}\rangle)$ and $\hat{g}^{'}(\langle X, \hat{B}\rangle)$ are got, we used an algorithm which combined the fast iterative shrinkage thresholding algorithm(FISTA) and Nesterov method to get the estimator of B faster and more accurately.

And we discuss the asymptotic properties about the SIM-nuclear method. Under some mild assumptions, we proved that the estimator of B can converge to the true B at root-n rate. Simultaneously, the estimator of B can keep low-rank and sparse because of the property of nuclear norm penalty function.

Then we applied this method to some simulations. Firstly, we found as the simple size increases, the performance of SIM-nuclear would be better, secondly we found there would be an optimal hyper-parameter combination. In this combination, we could find the value of λ and bandwidth h.

At last, we apply the SIM-nuclear method to real data, MEG data. In the MEG data, the independent variable X is order 2 tensor data, and the dependent variable y is binary value. We used two methods, SIM-nuclear and logistic regression, to analyse the data. Then we found the error rate of the result was estimated by SIM-nuclear is lower than the error rate of the result was estimated by logistic regression.

6.2 Future work

At last, we need to talk about the future work about the research. Currently, the main problems of this method is the process of estimating the parameter $\hat{B}$ costs a lot of time. It is not friendly for ordinary people. The reason why the method costs a lot of time is I wrote this code by Python. In future, I will rewrite the code via C++. And during the process of rewriting, I will use orient object programming(OOP) idea to finish this job in order to update the algorithm easily.

The second thing I plan to do is to use amazon web service(AWS) to build a pipeline containing SIM-nuclear. It can be used to handle data stream that is a type of big data. The data stream can renew the method dynamically, then we can use the renewed method to fit the future data stream. By using AWS, we can share this method to more people. These people may have some commercial requests about this method, which means we can make profit from this method. The profit can attract more and more outstanding researchers to join us to develop this method.

The third thing is we will extend the method from matrix-valued data to tensor with arbitrary order by modifying the algorithm and penalty function. We try to get the modified SIM-nuclear's asymptotic properties.

From the real data example, we find that SIM-nuclear is not fit for the binary response variables, but there are a lot of reality data with binary response variable, for example, in the field of computer version, there are a lot of dogs and cats photos, we want the computer can recognize the creature in a photo is a cat or a dog. In this example, the response variables are binary, cat or dog, and the explanatory variables are a order 2 tensor data. For this reason, we will develop the SIM-nuclear model to fit for the binary response, for example, the modified

log-likelihood function can be the loss function of SIM-nuclear. And we found that in the real data, we should use multiple hypothesis test to test which element in the estimator matrix is not equal to 0. Because the sensors may not independent to each other. It is the job I will do in the future.

Bibliography

Armagan, A., D. Dunson, and J. Lee (2011). Generalized double pareto shrinkage. arxiv:1104.0861v1.

Bach, F. (2008). Consistency of trace norm minimization. J.Mach. Learn. Res 9, 1019–1048.

Beck, A. and M. Teboulle (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Img. Sci. 2(1), 183–202.

Borwein, J. M. and A. S. Lewis (2000). Convex Analysis and Nonlinear Optimization. CMS Books in Mathematics. Springer, New York, second edition.Theory and examples.

Carroll, R. J., J. Fan, I. Gijbels, and M. P. Wand (1997). Generalized partially linear single-index models. Journal of the American Statistical Association 92(438), 477–489.

Chellappa, R., A. K. Roy-Chowdhury, and S. K. Zhou (2005). Recognition of humans and their activities using video. Synthesis Lectures on Image, Video & Multimedia Processing 1(1), 1–173.

Chen, J., J. Yang, L. Luo, J. Qian, and W. Xu (2015). Matrix variate distribution induced sparse representation for robust image classification. IEEE Trans, Neural Netw learn Syst 26(10), 2291–2300.

Chen, S. S., D. L. Donoho, and M. A. Saunders (2001). Atomic decomposition by basis pursuit. SIAM Rev 43(1), 129–159.

Fan, J., T.-C. Hu, and Y. K. Truong (1994). Robust nonparametric function estimation. Scandinavian Journal of Statistics 21, 433–446.

Fan, J. and R. Li (2001a). Variable selection via nonconcave penalized likelihood and its oracle properties. J. Amer. Statist. Assoc 96(456), 1348–1360.

Fan, J. and R. Li (2001b). Variable selection via nonconcave penalized likelihood and its oracle properties. Journal of the American statistical Association 96(456), 1348–1360.

Frank, I. E. and J. H. Friedman (2002). A statistical view of some chemometrics regression tools. Technometrics 35(2), 109–135.

Fu, W. and K. Knight (2000). Asymptotics for lasso-type estimators. Annals of Statistics 28, 1356–1378.

Gaïffas, S., G. Lecué, et al. (2007). Optimal rates and adaptation in the single-index model using aggregation. Electronic journal of statistics 1, 538–573.

Green, R. D. and L. Guan (2004). Quantifying and recognizing human movement patterns from monocular video images-part ii: applications to biometrics. IEEE Transactions on Circuits and Systems for Video Technology 14(2), 191–198.

Guidoum, A. C. (2015). Kernel estimator and bandwidth selection for density and its derivatives. The kedd package, version 1.

Hardle, W., P. Hall, and H. Ichimura (1993). Optimal smoothing in single-index models. The annals of Statistics, 157–178.

Härdle, W. and T. M. Stoker (1989). Investigating smooth multple regression by the method of average derivatives. Journal of the American Statistical Association 84(92), 986–995.

Hastie, T., R. Tibshirani, and J. Friedman (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer series in statistics. Springer.

Hjort, N. L. and D. Pollard (2011). Asymptotics for minimisers of convex processes. arXiv preprint arXiv:1107.3806.

Hoerl, A. E. and R. W. Kennard (1970). Ridge regression: biased estimation for nonorthogonal problems. Technometrics 12, 55–67.

Horowitz, J. L. (2012). Semiparametric methods in econometrics, Volume 131. Springer Science & Business Media.

Horowitz, J. L. and W. Härdle (1996). Direct semiparametric estimation of single-index models with discrete covariates. Journal of the American Statistical Association 91(436), 1632–1640.

Kolda, T. G. and B. W. Bader (2009). Tensor decompositions and applications. SIAM review 51(3), 455–500.

Kong, E. and Y. Xia (2012). A single-index quantile regression model and its estimation. Econometric Theory 28(4), 730–768.

Li, J., L. Zhang, D. Tao, H. Sun, and Q. Zhao (2008). A prior neurophysiologic knowledge free tensor-based scheme for single trial eeg classification. IEEE Transactions on Neural Systems and Rehabilitation Engineering 17(2), 107–115.

Li, K. C. (1991). Sliced inverse regression for dimension reduction (with dicussion). Journal of the American Statistical Association 86(93), 316–342.

Lu, J., K. N. Plataniotis, and A. N. Venetsanopoulos (2003). Face recognition using kernel direct discriminant analysis algorithms. IEEE Transactions on Neural Networks 14(1), 117–126.

Luo, L., J. Yang, J. Qian, and Y. Tai (2015). Nuclear-l1 norm joint regression for face reconstruction and recognition with mixed noise. Pattern Recognit 48(12), 3811–3824.

Powell, J. L., J. H. Stock, and T. M. Stoker (1989). Semiparametric estimation of index coefficients. Econometrica: Journal of the Econometric Society, 1403–1430.

Qian, J., L. Luo, J. Yang, F. Zhang, and Z. Lin (2015). Robust nuclear norm regularized regression for face recognition with occlusion. Pattern Recognit 48(10), 3145–3159.

Quarteroni, A., R. Sacco, and F. Saleri (2000). Numerical Mathematics. Springer-Verlag,New York.

Renard, N. and S. Bourennane (2009). Dimensionality reduction based on tensor modeling for classification methods. IEEE Transactions on Geoscience and Remote Sensing 47(4), 1123–1131.

Sahambi, H. S. and K. Khorasani (2003). A neural-network appearance-based 3-d object recognition using independent component analysis. IEEE transactions on neural networks 14(1), 138–149.

Silverman, B. W. (1986). Density Estimation For Statistics And Data Analysis. London: Chapman & Hall.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. J Roy Statist Soc Ser B 58(1), 267–288.

Xia, Y. and H. Tong (2006). On the Efficiency of Estimation for a Single-Index Model, pp. 63–86. Imperial College Press.

Xia, Y., H. Tong, W. K. Li, and L.-X. Zhu (2002). An adaptive estimation of dimension reduction space. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 64(3), 363–410.

Yan, S., D. Xu, Q. Yang, L. Zhang, X. Tang, and H.-J. Zhang (2006). Multilinear discriminant analysis for face recognition. IEEE Transactions on image processing 16(1), 212–220.

Yang, J., L. Luo, J. Qian, Y. Tai, F. Zhang, and Y. Xu (2014). Nuclear norm based matrix regression with applications to face recognition with occlusion and illumination changes. arXiv preprint arXiv:1405.1207.

Zeng, P., T. He, and Y. Zhu (2012). A lasso-type approach for estimation and variable selection in single index model. Journal of computational and graphical statistics, 92–109.

Zhou, H. and L. Li (2014). Regularized matrix regression. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 76(2), 463–483.

Zou, H. and T. Hastie (2005). ”regularization and variable selection via the elastic net. J.R.Stat.Soc.Ser.B Stat.Methodol 67(2), 301–320.