Estimation of Sparse Single Index Vector Autoregression Models

ZHONGMEI JI

PhD

UNIVERSITY OF YORK

MATHEMATICS

SEPTEMBER 2018

ABSTRACT

Stimulated by the analysis of a data set about house price variance in the USA, we propose a sparse single-index vector autoregressive model (SSIVARM). In order to solve the model, we develop an iterative algorithm based on least squares estimation procedure (PLSEP) to simultaneously identify the zero components and estimate the non-zero unknown parameters and unknown functions in the model. Not only providing concrete methodology for the implementation of the proposed algorithm, we also conduct intensive simulation studies to investigate the performance of the proposed PLSEP and the iterative algorithm when the sample size is finite. Finally, we apply the proposed SSIVARM together with the proposed PLSEP and iterative algorithm to the data set mentioned above. Our results reveal some interesting connections between some variables and the house price. Although the proposed SSIVARM is stimulated by a data set about house price, our findings suggest it can be applied to any multivariate time series.

CONTENTS

List of Tables

List of Figures

Figure **Page**

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my supervisor Prof.Wenyang Zhang for his continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better supervisor and mentor for my Ph.D study.

Besides my supervisor, I would like to thank the rest of my thesis committee: Prof. Degui Li and Dr. Marina Knight, for their insightful comments and encouragement, but also for the hard questions which incented me to widen my research from various perspectives.

I also thank the YARCC (York Advanced Research Computing Cluster) IT team of our university and support staffs of our department for their unfailing support and assistance during my PhD research.

My sincere thanks also go to Dr. Jen Ning Tan, Mr Eamonn McMurrough, and the whole SRC team in Willis Towers Watson, who provided me an opportunity to join their team as

a part- time intern, which funded part of my study and life. Without their precious support it would not have been possible to conduct this research.

Deeply from my heart with love and faith, I would like to thank my parents for their encouragement and support throughout my life. I am also grateful to my husband for his outstanding and highly appreciated patience day and night throughout the time of my study.

AUTHOR'S DECLARATION

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for an award at this, or any other, university. All sources are acknowledged as references.

The literature review in Chapter 2 provides some key ideas related to this thesis, which includes:

- Section 2.1 provides a summary of the penalised least squares approach, which is mostly based on *Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties* by Fan and Li (2001) and *Sparse High Dimensional Models in Economics* by Fan *et al.*(2011).

- Section 2.2 introduces the Generalised Information Criterion found in *Tuning Parameter Selection in High Dimensional Penalized Likelihood* by Fan and Tang (2013).

- Section 2.3 reviews the framework of local polynomial modelling, which is mainly from the the book: *Local*

Polynomial Modelling and Its Applications by Fan, J. and Gijbels, I. (1996).

- Section 2.4 contains a concise review of varying coefficient models chiefly from *Adaptive Varying-coefficient Linear Models* by Fan *et al.*(2003) and *Statistical Methods with Varying Coefficient Models* by Fan and Zhang (2008).

CHAPTER 1

INTRODUCTION

1.1 Preamble

With the advancement of data collection, we are facing an increasing number of complex and considerable datasets including numerous potential variables. In the analysis of such kinds of datasets, it is often a challenge to specify an efficient model which precisely contains all the significant components. Specifically, the difficulty is twofold: firstly, we are not clear which model is appropriate and which variables are relevant; secondly, we cannot go for a very flexible model which may involve too many variables. Therefore, in modern data analysis, how to deal with those issues in the high dimensional data analysis has become a notably important research area. Among the studies, the penalised likelihood/least squares

approach emerged as a promising approach in the last two decades and much literature is devoted to this method and its application. See Fan and Lv (2008), Fan *et al.*(2009), Bickel *et al.*(2009), Wang and Xia (2009), Stefanski *et al.*(2014), Wang, Peng and Li (2015), Fan *et al.*(2015), Li, Ke and Zhang (2015), Fan and Lv (2016), Zhang *et al.*(2016), and the references therein.

The penalised likelihood/least squares makes the following modelling approach for high dimensional data possible: instead of specifying a particular model, it starts with a model which can be as flexible as possible and includes all the candidate variables. Then the penalised likelihood/least squares approach is applied to select the significant variables and shrink the parameters of irrelevant variables to 0, and thus get the right model. In addition, with a proper penalty function and computational algorithm, the penalised approach would simultaneously select significant variables and estimate coefficients.

In the penalised likelihood/least squares approach, it is of importance to choose an appropriate penalty function. Frank and Friedman (1993) and Fu (1998) proposed the ridge regression, which is associated with the L_2 penalty function. However, the L_2 penalty fails to yield a sparse solution and hence, to overcome this problem, the least absolute shrinkage and selection operator (LASSO), proposed by Tibshirani

(1996, 1997), is the penalised least squares estimate along with the L_1 penalty function. Boyd and Vandenberghe (2004) proposed the proximal gradient descent (PGD) algorithm to solve LASSO and other L_1 based penalised methods. Efron *et al.*(2004) also proposed an effective algorithm, termed as least angle regression (LARS), which can be applied to produce the full set of LASSO solutions. Yuan and Lin (2006) studied and proposed efficient algorithms for the extensions of the LASSO for selecting the grouped variables. However, although LASSO enjoys many attractive properties, it is inconsistent with the variable selection because its resulting estimate is biased. To ameliorate this issue, Zou (2006) proposed the adaptive LASSO and Fan and Li (2001) developed the smoothly clipped absolute deviation (SCAD) penalty. At the same time, Fan and Li (2001) also built a framework for the selection of the penalty function. They suggested a good penalty function should result in an estimator with three properties: unbiasedness, sparsity and continuity, and the proposed SCAD penalty, as an example of a good penalty, enjoys all the three properties. Additionally, Fan and Li (2001) extended the penalised least squares to a general likelihood setting and built a unified algorithm for optimising both the penalised least squares and penalised likelihood via local quadratic approximations. Based on local linear approximation, Zou and Li (2008) developed a one-step sparse estimation procedure for

optimising the penalised likelihood which outperforms the preceding local quadratic approximation.

The penalised approaches has been applied to select a wide range of models. For example, the vector auto-regressive (VAR) model, which serves as a classic technique to deal with the joint evolution of multivariate time series, can deliver a great deal of structural information. However, VAR model has a potential issue in that the parameter space of VAR model increases rapidly with the size of the model, hence the model selection of this class of model is essential. Fan, Lv and Qi (2011) introduced penalised least squares to the VAR model with the addition of neighbourhood variables to examine house-price estimation and prediction. Calomiros *et al.*(2008) performed the panel VAR regression to demonstrate a strong effect of foreclosure on house prices. Rapach and Strauss (2007) considered combinations of individual VAR forecasts, with each equation consisting of only one macroeconomic variable, in forecasting house-price growth in several states. As an illustration, our thesis shall focus on the selection and estimation of a sparse single-index varying coefficient VAR model.

1.2 A motivating example

In recent years, the analysis of house price has attracted much research in statistics, an important topic in which the prediction of house price and the investigation of the connection between house price and some factors associated with house price are of importance. We study a house price index dataset from The USA. The data were collected from $d = 10$ different places in the USA every month for $n = 381$ months from 1987 to 2017. Therefore, the dataset is a 10-dimensional multivariate time series, denoted by $\mathbf{y}_t$, $t = 1, \cdots, n$, with q covariates, denoted by $\mathbf{X}_t$, $t = 1, \cdots, n$. $\mathbf{X}_t$ is a matrix of size $d \times q$, different rows of $\mathbf{X}_t$ are the observations of the q covariates at different places. Traditionally, for such dataset, a multivariate time series model, such as the vector autoregression (VAR) models would be used for prediction.

$$\mathbf{y}_t = \sum_{j=1}^{p} \mathbf{A}_j \mathbf{y}_{t-j} + \mathbf{X}_t \boldsymbol{\alpha} + \boldsymbol{\epsilon}_t, \tag{1.1}$$

However, the problem is we would have $d^2 \times p + q$ unknown parameters to estimate if a VAR(p) model is used, which is too many for this dataset, and would result in a large variance of the prediction. To reduce the unknown parameters, one would be tempted to go for a smaller order p as 1 order less would reduce d^2 unknown parameters. Unfortunately, this would also have a problem of misspecification. Furthermore, even

if $p = 1$, we would still have $d^2 + q$ unknown parameters to estimate, which is still too many for this dataset.

As far as this dataset is concerned, for a given place, it is not the case that the house prices in previous months at every place would affect the house price at this given place, which means some elements in A_j in the model (1.1) may be 0. Therefore, a sensible modelling approach would be staying with model (1.1) but assuming some elements in A_j are 0, which give us a sparse VAR model. Of course, which elements in A_j are 0 should be identified by a data-driven approach.

There is another problem arising: if we apply the model (1.1) to the dataset, we would assume the impact of previous house prices on the current house price is constant, which is not plausible. One way to deal with this problem is to assume A_j depends on the previous house prices $\mathbf{y}_{t-\ell}$. Due to "curse of dimensionality", as we cannot simply assume A_j is a function of $\mathbf{y}_{t-\ell}$, a natural approach would be to assume A_j is a function of $\mathbf{y}_{t-\ell}^{\mathrm{T}}\boldsymbol{\beta}$, this gives us the sparse single-index vector autoregressive models, which we formally define in Section 1.3.

1.3 The sparse single-index vector autoregressive models

To make our definition more generic, let $\mathbf{y}_t$, $t = 1, \cdots, n$, be a d-dimensional stationary time series, and $\mathbf{X}_t$, $t = 1, \cdots, n$, be independent and identically distributed. $\mathbf{X}_t$ is a $d \times q$ matrix. The sparse single-index vector autoregressive model (SSI-VARM) is defined as

$$\mathbf{y}_t = \sum_{j=1}^{p} \mathbf{A}_j(\mathbf{y}_{t-\ell}^{\top}\boldsymbol{\beta})\mathbf{y}_{t-j} + \mathbf{X}_t\boldsymbol{\alpha} + \boldsymbol{\epsilon}_t, \tag{1.2}$$

where $1 \le \ell \le p$, the coefficients $\mathbf{A}_j(\cdot)$, $j = 1, \ldots, p$ are $d \times d$ matrices, the first column of $\mathbf{A}_\ell(\cdot)$ is $\mathbf{0}_d$, $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_d)^{\mathrm{T}}$ and satisfies

$$\|\boldsymbol{\beta}\| = 1, \quad \beta_1 > 0, \tag{1.3}$$

α is a $q-$ dimensional constant vector, $\boldsymbol{\epsilon}_t$, $t = 1, \cdots, n$, are i.i.d. with

$$E(\boldsymbol{\epsilon}_t|\mathscr{F}_t) = \mathbf{0}_d, \qquad \mathrm{cov}(\boldsymbol{\epsilon}_t|\mathscr{F}_t) = \sigma^2\mathbf{I}_d$$

almost surely, where $\mathscr{F}_t = \{(Y_{l-1}, \mathbf{X}_l) : \ l \le t\}$. The unknown parameters that shall be estimated are $\boldsymbol{\beta}$, $\mathbf{A}_j(\cdot)$, $\boldsymbol{\alpha}$ and σ^2. Furthermore, $\boldsymbol{\beta}$ and $\mathbf{A}_j(\cdot)$, $j = 1, \cdots, p$, are sparse.

The conditions (1.3) in the definition are identification conditions, which directly come from the single index vary coefficient linear models. However, unlike the standard models,

the conditions, (1.3), do not guarantee (1.2) is identifiable, see Fan *et al.*(2003), that is why we impose another identification condition, the first column of $\mathbf{A}_\ell(\cdot)$ is $\mathbf{0}_d$, in the definition to make (1.2) identifiable.

The SSIVARM is the model we are going to address in this paper, it has the characteristics of the varying-coefficient models, which have proved to be remarkably useful in the interpretation of dynamic pattern of the relationship between the response and the covariates, see Fan and Zhang (1999), Sun *et al.*(2014), Li *et al.*(2015), Huang *et al.*(2016) and the reference therein, and single index models, which are efficient approaches to ameliorate the "curse of dimensionality" in nonparametric modelling, see Yu and Ruppert (2002), Xia (2008), Guo *et al.*(2016), and the references therein. Moreover, compared with the typical VAR model, model (1.2) substantially enlarges the model capacity. Therefore, as model (1.2) allows the sparsity, the main work in our thesis is to design an efficient algorithm to consistently identify the true sub-model of (1.2) and automatically estimate it.

The thesis is organised as follows. In Chapter 2 we review the existing literature related to the proposed methodology such as penalised least squares and penalised likelihood, Generalised Information Criterion (GIC) for tuning parameter selection, local polynomial modelling and varying coefficient models. Chapter 3 introduces the single index vector autore-

gressive (SIVAR) model and explores an iterative approach for the estimation of the model. In Chapter 4, we propose a model selection and shrinkage method for the SSIVAR model. Chapter 5 discusses the choice of the bandwidth (smoothing parameter) and tuning parameters (complex parameters). Apart from the simulations conducted in previous chapters, in Chapter 6, we use another simulation study to demonstrate the goodness of the proposed approaches in model selection and estimation when the hyper-parameters have been properly selected. In Chapter 7, we apply the SSIVARM along with the proposed penalised iterative procedures to analyse an American housing data set. This real data analysis shall specify how the federal-level data impact the housing market in each city included in the dataset and capture the dynamic pattern of the impacts. Finally, we will concisely conclude the main results of our studies and give the potential future extensions in Chapter 8.

CHAPTER 2

LITERATURE REVIEW

In this chapter, the literature we are going to review is in four diverse areas. The first part presented in Section 2.1 is the penalised least squares with smoothly clipped absolute deviation (SCAD) penalty, which is the key technique we will use in the thesis to select the true sub-model in SSIVAM. Secondly, as appropriately determining how to select the tuning parameters (regularization parameters) involved in the SCAD penalty is of significance to consistently search the true model, in Section 2.2, we refer to Zhang, Li and Tsai (2010) and Fan and Tang (2013) for their research on generalised information criterion (GIC). Thirdly, in Section 2.3, we will provide a brief review of local polynomial modelling, which is the fundamental technique for smoothing the SSIVAM in our thesis. At last, we will review some previous work on varying

coefficient models in Section 2.4.

2.1 Penalised approaches

In this thesis, the penalised least squares are the main technique we employ to select the model. We add the SCAD penalty functions on our square loss functions to do the group selection to the varying coefficients and component-wise selection to the constant parameters, and then, based on the idea of local quadratic approximation (see Fan and Li (2001)), we solve the penalised least squares to obtain the resulting sparse estimates. Hence, in this section, we are going to review penalised least squares, penalised likelihood, the smoothly clipped absolute deviation (SCAD) penalty and the algorithm of local quadratic approximation. We begin with the linear regression model

$$\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\boldsymbol{y} = (y_1, \ldots, y_n)^\top$ is an $n \times 1$ response vector, $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_d)^\top$ is an $n \times d$ design matrix of random variable, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_d)^\top$ is a $d \times 1$ vector of parameters to be estimated and $\boldsymbol{\epsilon}$ is an $n \times 1$ vector of random noise.

We assume the penalised least squares (PLS) as follows,

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^d} \Big\{ \frac{1}{2n} ||\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \sum_{k=1}^{d} p_\lambda(|\beta_k|) \Big\}, \tag{2.1}$$

where $p_\lambda(\cdot)$ is the penalty function allowed to depend on the regularisation parameter $\lambda \geq 0$, which controls the model complexity.

To have clearer insights on the variable selection procedures, we consider a specific case of a canonical linear model with a rescaled orthonormal design matrix, i.e., $\mathbf{X}^\top \mathbf{X} = nI_d$. With this assumption, the penalised least squares (2.1) can be written in the following minimisation problem:

$$\min_{\boldsymbol{\beta}} \Big\{ \frac{1}{2n} ||\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{\beta}}||_2^2 + \frac{1}{2} ||\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}||_2^2 + \sum_{k=1}^{d} p_\lambda(|\beta_k|) \Big\}. \tag{2.2}$$

where $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} = \mathbf{n}^{-1} \mathbf{X}^\top \mathbf{y}$ is the ordinary least squares estimator. As (2.2) can be minimised in a component-wise manner, for the brevity purposes, we consider the minimisation problem of a univariate penalise least squares as follows,

$$\frac{1}{2}(z - \theta)^2 + p_\lambda(|\theta|), \tag{2.3}$$

with respect to the parameter θ, where z is the univariate ordinary least squares estimate. Then, we can obtain the resulting estimator $\hat{\theta}$ by solving

$$\hat{\theta} = \arg\min_{\theta} \Big\{ \frac{1}{2}(z - \theta)^2 + p_\lambda(|\theta|) \Big\}. \tag{2.4}$$

conforming to the rule given by Antoniadis and Fan (2001), the penalty $p_\lambda(\cdot)$ in (2.4) can be regarded as a ideal penalty

function if the corresponding resulting estimate $\hat{\theta}$ can meet the following three requirement:

- Sparsity. If the true parameter $|\theta|$ is small, the corresponding resulting estimate will be $\hat{\theta} = 0$.
- Unbiasedness. When the unknown parameter $|\theta|$ is sufficiently large, the resulting estimate gives $\theta = z$ with high probability.
- Continuity. The resulting estimate $\hat{\theta}$ is continuous.

Fan and Li (2001) also conclude that a penalty function holds the sparsity conditions must be singular at the origin.

Based on these lines, we can evaluate some of the most commonly used penalty functions. As a member in the class of L_q penalties, L_0 penalty

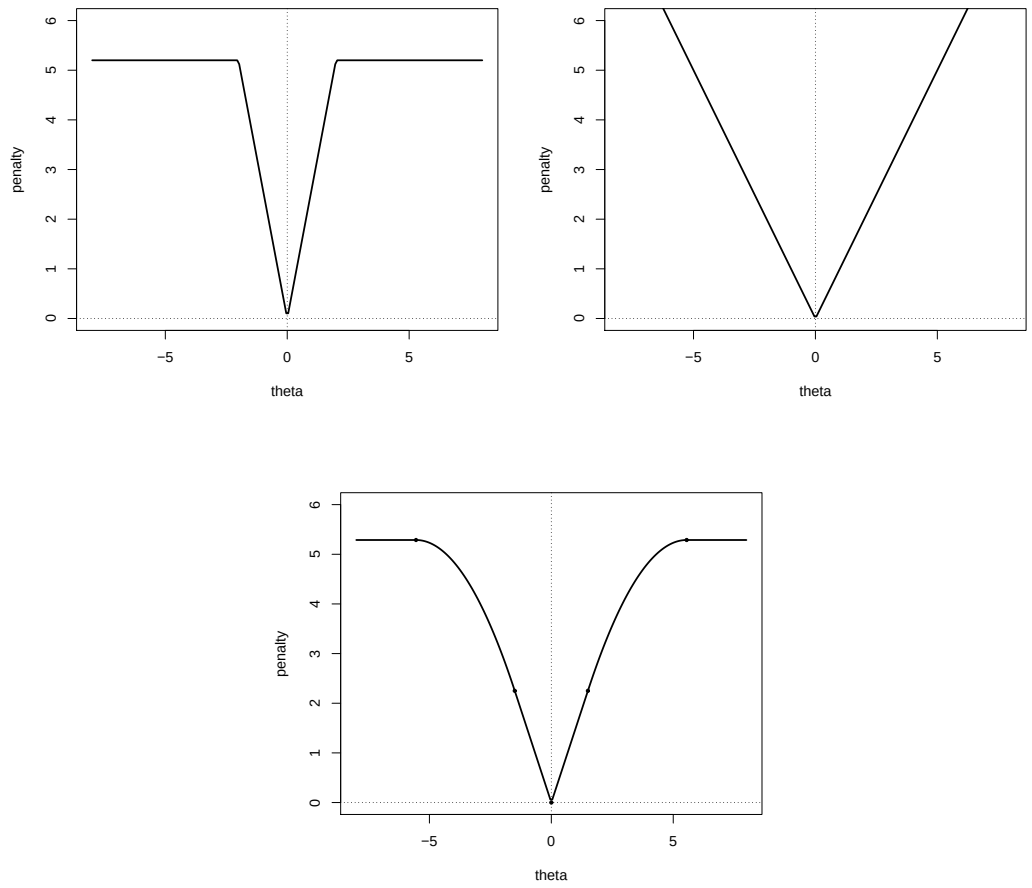
$$p_\lambda(z) = \frac{\lambda^2}{2} I(z \neq 0)$$

yields the hard thresholding estimator $\hat{\theta} = zI(|z| > \lambda)$. Figure 2.1(a) and Figure 2.2(a) visually depicts L_0 penalty. It can be noticed that the penalised estimator does not satisfy the continuity. Another notable penalty is the L_1 penalty (LASSO) (Tibshirani, 1996) $p_\lambda(z) = \lambda|z|$, which leads to the soft thresholding estimator

$$\hat{\theta} = \text{sgn}(z)(|z| - \lambda)_+.$$

We present the thresholding estimate in Figure 2.1(b), from which we can visually find that the resulting estimates consistently produce biased solutions. Additionally, the convex L_p penalties with $p > 1$ are not singular around the origin, and hence they fail to fulfil the condition of sparsity. Thus, None of the L_q penalties can satisfy all three conditions of an ideal penalty at the same time.

Figure 2.1: The penalty functions



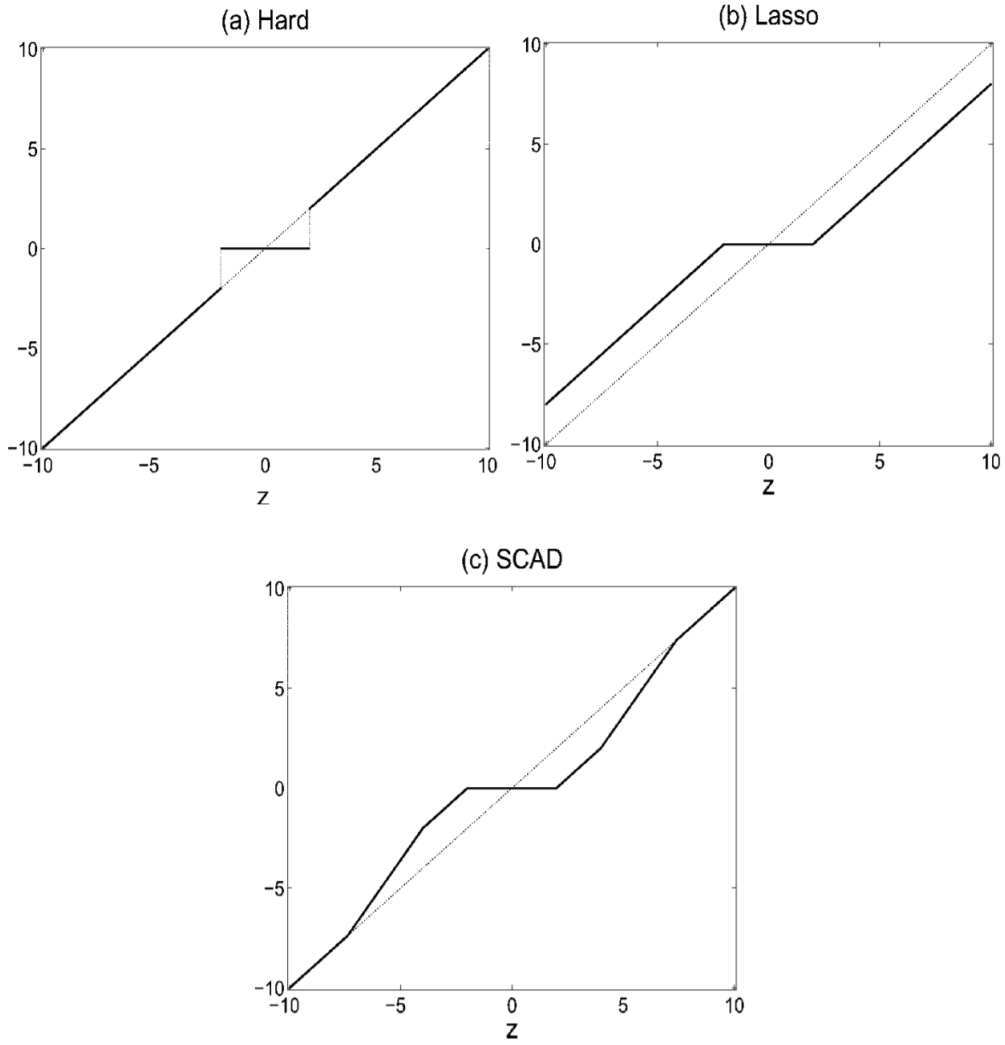
NOTE: *Plot of penalty functions of L_0 penalty, L_1 penalty and SCAD penalty.*

One successful attempt on constructing an ideal penalty is the smoothly clipped absolute deviation (SCAD) penalty, proposed by Fan and Li (2001), whose derivative is given by

$$p'_\lambda(\theta) = \lambda\{I(\theta \le \lambda) + \frac{(a\lambda - \theta)_+}{(a-1)\lambda} I(\theta > \lambda)\}, \quad \text{for some } a > 2 \text{ and } \theta > 0,$$

where $p_\lambda(0) = 0$ and a is suggested to be 3.7. It fulfils the aforementioned three conditions. We gives more insights into this statement by Figure 2.1(c) and 2.2(c).

Figure 2.2: The thresholding functions



NOTE: *Plot of thresholding function for (a) the hard, (b) the soft and (c) the SCAD. The plots are quoted from the Figure 2 in Fan and Li (2001)*

Fan and Li (2001) extended the penalised least squares to likelihood-based models. For generalised linear models, statistical inferences are based on underlying likelihood functions. The penalised maximum likelihood estimator can be used to identify significant variables. Assume that the data $(\mathbf{x}_i, Y_i)$

are collected independently. Given that $\mathbf{x}_i, Y_i$ has a density $f_i(g(\mathbf{x}_i^\top \boldsymbol{\beta}), y_i)$, where g is a link function has been known. Denote the conditional log-likelihood of y_i by $ł_i = \log f_i$. A form of the penalised likelihood is

$$\sum_{i=1}^{n} ł_i(g(\mathbf{x}_i^\top \boldsymbol{\beta}), y_i) - n \sum_{j=1}^{d} p_\lambda(|\beta_j|), \tag{2.5}$$

with respect to $\boldsymbol{\beta}$. Maximising the penalised likelihood function is equivalent to minimising.

$$-\sum_{i=1}^{n} ł_i(g(\mathbf{x}_i^\top \boldsymbol{\beta}), y_i) + n \sum_{j=1}^{d} p_\lambda(|\beta_j|). \tag{2.6}$$

with respect to $\boldsymbol{\beta}$. To get the penalised maximum likelihood estimator of $\boldsymbol{\beta}$, we minimise (2.6) with respect to $\boldsymbol{\beta}$ for some regularisation parameter λ.

Furthermore, Fan and Li (2001) established the asymptotic properties to show that the resulting estimator of SCAD penalty performs as well as the oracle estimator with probability tending to 1. Here, the oracle estimator represents the estimator obtained from the correct sub-model.

Although the SCAD penalty has many appealing properties, solving either the SCAD-type penalised least squares or SCAD-type maximum likelihood is challenging, because the target function is a high-dimensional non-concave function with singularities at the origin. Accordingly, to solve the minimisation problem, Fan and Li (2001) developed a unified algorithm via local quadratic approximations (LQA).

Giving an initial value $\boldsymbol{\beta}^0 = (\beta_1^0, \ldots, \beta_d^0)^\top$ is close to the minimiser of (2.1) and we set $\beta_j = 0$ if β_j^0 close enough to 0, then the penalty function $p_\lambda(\cdot)$ can be locally approximated by a quadratic function as follows

$$p_\lambda(|\beta_j|) \approx p_\lambda(|\beta_j^0|) + \frac{1}{2}\frac{p'_\lambda(|\beta_j^0|)}{|\beta_j^0|}[\beta_j^2 - (\beta_j^0)^2], \quad \text{for } \beta_j \approx \beta_j^0. \quad (2.7)$$

The derivative form of this approximation is given as

$$[p_\lambda(|\beta_j|)]' = p'_\lambda(|\beta_j|)\text{sgn}(\beta) \approx \{p'_\lambda(|\beta_j^0|)/|\beta_j^0|\}\beta_j.$$

With this quadratic approximation (2.7), the penalised least squares problem (2.1) is reduced to a quadratic optimisation problem and admits a closed-form solution. Note that one drawback of LQA is that once a coefficient is shrunken to zero in any iteration, it will remain zero. To overcome this potential issue, Zou and Li (2008) developed a unified algorithm based on the local linear approximation (LLA):

$$p_\lambda(|\beta_j|) \approx p_\lambda(|\beta_j^0|) + p'_\lambda(|\beta_j^0|)[|\beta_j| - |\beta_j^0|], \quad \text{for } \beta_j \approx \beta_j^0.$$

Zou and Li (2008) show that in the algorithm of LLA, it is not necessary to delete any small parameters or select the size of perturbation. Moreover, the LLA can naturally yield sparse estimates through continuous penalisation. Similar to LQA, the LLA algorithm can also largely reduce the computation burden.

2.2 Tuning parameter selection by Generalised information criterion

In employing the nonconcave penalised likelihood in regression analysis, we face two challenges. The first one is to calculate the nonconcave penalised likelihood estimate. This hurdle has been carefully studied in much recent literature, which leads to some efficient algorithms like LQA and LLA. The selection of the regularisation parameter becomes the second challenge, as the performance of penalised likelihood/least squares relies significantly on the choice of a regularisation parameter, which controls the model complexity. In this section, we are going to review some commonly used metrics, particularly the generalised information criterion proposed by Fan and Tang (2013).

In the literature, the criteria of variable selection are usually classified into two categories: the efficient one and the consistent one. Specifically, the efficient criterion, for example, the Akaike information criterion (AIC) and the generalised cross-validation (GCV), identifies the model so that its average squared error is asymptotically equivalent to the minimum offered by the candidate models when the true sub-model is approximated by the candidate models; a con-

sistent criterion select the true sub-model with probability approaching one in large samples when a set of candidate models contains the true sub-model. One typical example of consistent criterion is the Bayesian information criterion (BIC).

Wang, *et al.*. (2007) found that for penalised least squares with the smoothly clipped absolute deviation (SCAD) penalty, two efficient criteria, GCV and AIC, perform similarly in determining the tuning parameters, but the resulting model selected by either of them tends to having a higher variance, namely, it leads to overfitting results. However, Wang, *et al.*. (2007) also indicated that BIC is able to identify the finite-dimensional true linear and partial linear models consistently. Zhang *et al.*(2010) extended the study of regularisation parameter selection to penalised likelihood-based models with a nonconcave penalised function. They found that the BIC selector is still able to select the true model consistently, and the resulting estimator keeps the oracle property in the terminology of Fan and Li (2001).

Although Zhang *et al.*(2010) showed that a modified BIC can work successfully in the diverging dimensionality, when the dimension of the variable space is larger than the sample size, it may fail to identify the true sub-model consistently. To deal with this issue, the study of Fan and Tang (2013) allows the dimensionality d increase exponentially with the sample

size n and developed their generalised information criterion (GIC) to select the tuning parameter in high dimensional penalised approach.

To gain more insights on GIC, we adopt Nishii's (1984) generalised information criterion (GIC) to choose regularisation parameters in nonconcave penalised likelihood functions. This criterion not only contains AIC and BIC as its special cases but also connects the classical variable selection criteria and the nonconcave penalised likelihood methods. In Nishii (1984), a generalised information criterion can be constructed as follows:

$$\text{measure of model fitting} + a_n \times \text{measure of model complexity}, \tag{2.8}$$

where a_n is some sequence that controls the regularisation on model complexity, and thus the choice of a_n is of importance for searching the optimal tuning parameter. Measure of model fitting assess how predictive our model is. Common choices of it are the mean squared error and logistic loss. The measure of model complexity controls the complexity of the model, which helps us to shrink the trivial variables and avoid overfitting. In AIC and BIC, a_n in criterion (2.8) is 2 and $\log(n)$, respectively. Fan and Tang (2013) proposed a range of a_n for consistent and effective model selection and provided a uniform choice for practical implementation, which is given

as

$$a_n = \log\{\log(n)\}\log(d)$$

in GIC (2.8) .

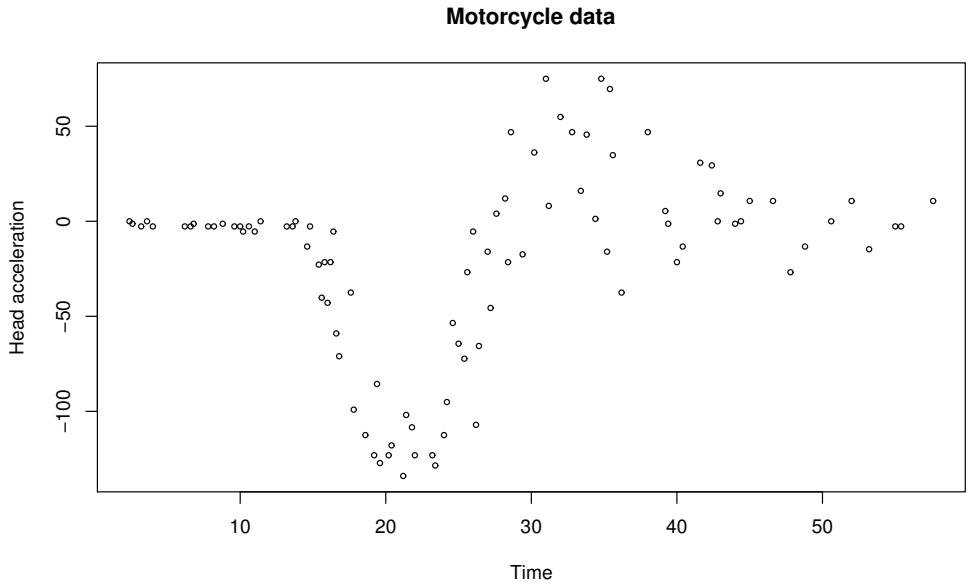
2.3 Framework of local polynomial modelling

In this section, we will review the framework of local polynomial modelling. This non-parametric approach is useful for getting a clear description of an unknown function, which could indict whether a parametric choice is appropriate or not. In this section, we briefly summarise the methodology for local polynomial regression by reviewing Fan and Gijbels (1996).

Firstly, we would like to introduce an example to show the limitations of global polynomial regression, which motivates us to explore a better approach to fit the data. We use the data from Schmidt *et al.*(1981). Two variables are included in the data set: the explanatory variable X represents the time (in milliseconds) after a simulated impact and the response Y stands for head acceleration. Figure 2.3 gives the scatter plot diagram of this dataset.

Then, we apply the linear, quadratic, cubic and quartic global polynomial regressions respectively to fit this data, the

Figure 2.3: Scatter plot for motor data

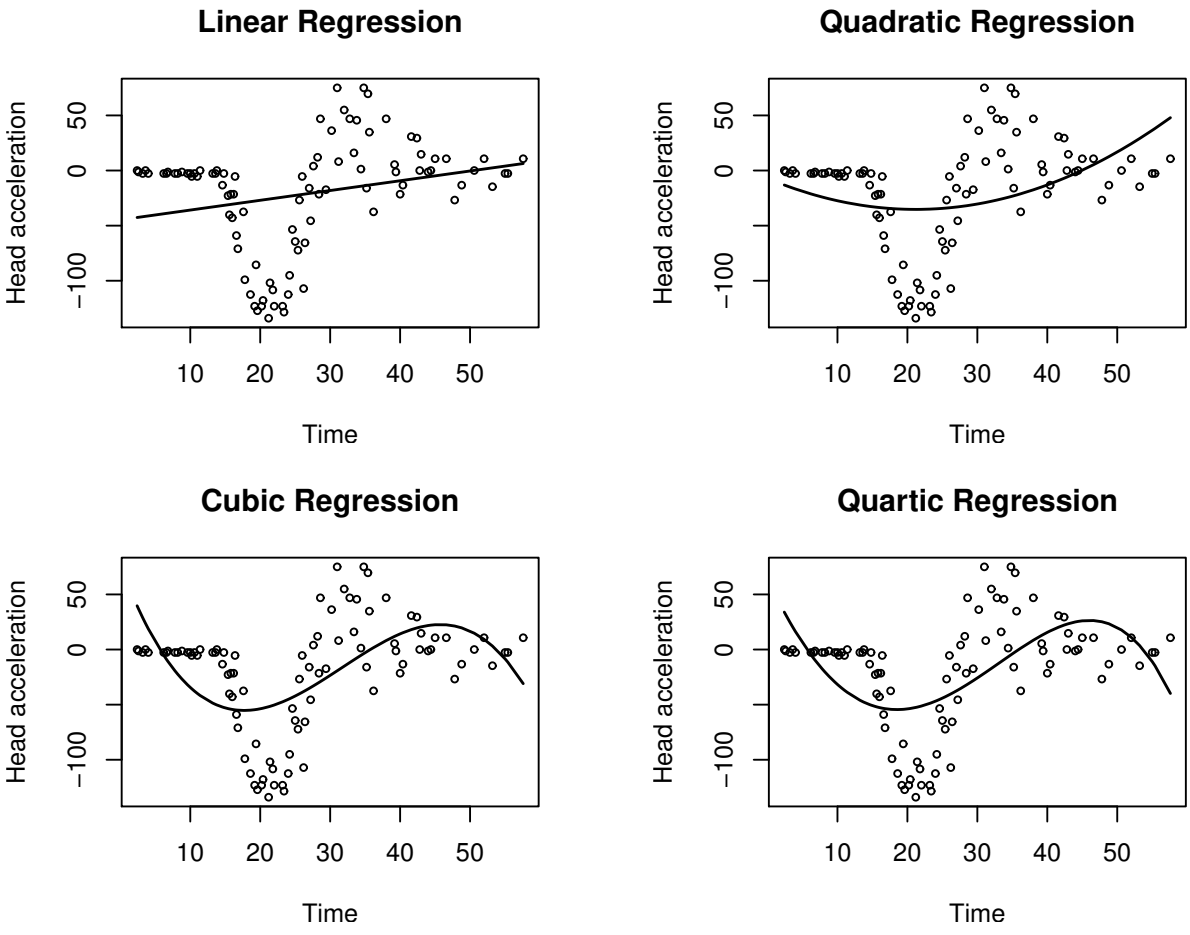


illustrative estimated results are visually reported in Figure 2.4.

It can be seen visually that, compared with linear regression, the quadratic, cubic or quartic fit may reduce the modelling bias to some extent, but leads to an estimator with larger variance. In addition, the polynomial models also suffer from the drawback that the remote individual observations can impact largely on the curve.

There are various related methods for fixing the problems arising from polynomial modellings such as splines approaches and orthogonal series modelling. However, in this chapter and the entire thesis, we shall focus our attention on local polynomial modelling.

Figure 2.4: Example of global polynomial models



Various local modeling regression estimators were studied in diverse statistical context. For example, one may consider using a weighted average of the response variables,

$$\hat{m}_h(x) = \frac{\sum_{i=1}^{n} K_h(X_i - x) Y_i}{\sum_{i=1}^{n} K_h(X_i - x)}, \tag{2.9}$$

proposed by Nadaraya (1964) and Watson (1964). $K_h(\cdot) = K(\cdot/h)/h$ is a probability density function with a kernel function $K(\cdot)$ and the bandwidth h, which is the smooth parameter.

There is another example, the Gasser-Muller estimator which originally proposed by Gasser and Muller (1984). This estimator is defined as follows

$$\hat{m}_h(x) = \sum_{i=1}^{n} \int_{s_i-1}^{s_i} K_h(u-x) Y_i du, \quad \text{with } s_i = (X_i - X_{i+1})/2, \tag{2.10}$$

where $X_0 = -\infty$ and $X_{n+1} = +\infty$.

Both (2.9) and (2.10) can be regarded as a local constant approximations for $m(\cdot)$. Indeed, by considering an arbitrary local least squares regression

$$\hat{\theta} = \arg\min_\theta \sum_{i=1}^{n} (Y_i - \theta)^2 w_i = \sum_{i=1}^{n} w_i Y_i / \sum_{i=1}^{n} w_i. \tag{2.11}$$

It is easy to see that (2.9) and (2.10) are special cases with $w_i = K_h(X_i - x)$ and $w_i = \int_{s_i-1}^{s_i} K_h(u-x)du$ respectively. However, local linear regression outperforms local constant regression in most cases. As explained by Fan (1992), this is because when going from a local constant estimation to a local linear estimation, the variance does not increase while the bias decreases. Therefore, the generalisation performance of local linear smoothing is better than the local constant smoothing. Additionally, Fan and Gijbels (1996) state that local linear regression adapts well to random and fixed designs, as well as highly clustered and nearly uniform designs.

To provide more insights into local linear regression, we introduce an independently and identically distributed bivari-

ate samples $\{(X_1, Y_1), \cdots, (X_n, Y_n)\}$ form a population (X, Y). Assume that the data is generated from the model

$$Y = m(X) + \sigma(X)\epsilon, \tag{2.12}$$

where $\mathbb{E}(\epsilon) = 0, \mathrm{Var}(\epsilon) = 1$, and ϵ is independent of X. We would like to estimate the unknown regression function $m(x_0) = \mathbb{E}(Y|X = x_0)$ and its derivatives $\dot{m}(x_0), \ddot{m}(x_0), \cdots, m^{(p)}(x_0)$. Assume that the $(p+1)$-th derivative of $m(\cdot)$ exists at the point x_0.

Consider a Taylor expansion for the unknown function $m(x)$ for x in a neighbourhood of x_0

$$\begin{aligned} m(x) \approx & m(x_0) + \dot{m}(x_0)(x - x_0) + \frac{\ddot{m}(x_0)}{2!}(x - x_0)^2 \\ & + \cdots + \frac{m^{(p)}(x_0)}{p!}(x - x_0)^p. \end{aligned} \tag{2.13}$$

We can consider $m(x_0), \dot{m}(x_0), \cdots, m^{(p)}(x_0)$ as unknown parameters to be estimated. From this point of view, we use the notation:

$$\frac{m^{(j)}(x_0)}{j!} = \beta_j, \quad \text{for } j = 0, 1, \cdots, p,$$

which leads us to rewrite (2.13) as

$$m(x) \approx \beta_0 + \beta_1(x - x_0) + \beta_2(x - x_0)^2 + \cdots + \beta_p(x - x_0)^p. \tag{2.14}$$

To obtain the estimators of unknown parameters, which are denoted by $\hat{\beta}_0, \hat{\beta}_1, \cdots, \hat{\beta}_p$, we solve a minimisation problem of a locally weighted least squares regression

$$\sum_{i=1}^{n}\left\{Y_i-\sum_{j=0}^{p}\beta_j(X_i-x_0)^j\right\}^2 K_h(X_i-x_0), \tag{2.15}$$

with respect to β_j, $j=0,\ldots,p$, where h is the bandwidth, and $K_h(\cdot)=K(\cdot/h)/h$ is a kernel function (a symmetric probability density function) allocating weights to every observation. Based on the estimates $\hat{\beta}_j$, we can obtain the estimator of function $m(x)$ and its derivatives $m^{(v)}(x_0)$ by $\hat{m}_v(x_0)=v!\hat{\beta}_v$ for each $v=0,\cdots,p$. Following the notations used in Fan and Gijbels (1996), the locally weighted least squares problem (2.15) can be rewritten in the matrix notation as

$$\min_{\boldsymbol{\beta}}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top\mathbf{W}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}),$$

where

$$\mathbf{X}=\begin{pmatrix}1 & (X_1-x_0) & \cdots & (X_1-x_0)^p\\ \vdots & \vdots & \ddots & \vdots\\ 1 & (X_n-x_0) & \cdots & (X_n-x_0)^p\end{pmatrix},$$

$$\mathbf{y}=(Y_1,\cdots,Y_n)^\top,$$
$$\boldsymbol{\beta}=(\beta_0,\cdots,\beta_p)^\top,$$

and

$$\mathbf{W}=\mathrm{diag}\{K_h(X_1-x_0),\ldots,K_h(X_n-x_0)\}.$$

It follows from least squares theory that

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{W}\mathbf{X})^{-1}\mathbf{X}^\top \mathbf{W}\mathbf{y}. \tag{2.16}$$

By solving the local polynomial estimators via (2.16), we are going to discuss how to evaluate the performance of the estimators. As the mean squared error (MSE) represents the sum of conditional bias and conditional variance, it can be a qualified metric to measure the goodness of estimates. Additionally determined by both the conditional bias and conditional variance of an estimator, mean integrated squared error (MISE) are always introduced as a metric to examine the fitting of estimators for unknown curves. Therefore, it is necessary to gain more insights on bias and variance. The conditional expectation and variance of $\hat{\boldsymbol{\beta}}$ is given by

$$\begin{aligned}\mathbb{E}(\hat{\boldsymbol{\beta}}|\mathbb{X}) &= (\mathbf{X}^\top \mathbf{W}\mathbf{X})^{-1}\mathbf{X}^\top \mathbf{W}m \\ &= \boldsymbol{\beta} + (\mathbf{X}^\top \mathbf{W}\mathbf{X})^{-1}\mathbf{X}^\top \mathbf{W}r\end{aligned}$$

And

$$\mathrm{Var}(\hat{\boldsymbol{\beta}}|\mathbb{X}) = (\mathbf{X}^\top \mathbf{W}\mathbf{X})^{-1}(\mathbf{X}^\top \Sigma \mathbf{X})(\mathbf{X}^\top \mathbf{W}\mathbf{X})^{-1}$$

where $m = \{m(\mathbf{X}_1), \cdots, m(\mathbf{X}_n)\}^\top$, $\boldsymbol{\beta} = \{m(x_0), \cdots, m^{(p)}(x_0)/p!\}^\top$, $\mathbf{r} = \mathbf{m} - \mathbf{X}\boldsymbol{\beta}$, the vector of residuals of the local polynomial approximation, and $\Sigma = \mathrm{diag}\{K_h^2(X_1 - x_0)\sigma^2(X_1), \ldots, K_h^2(X_n - x_0)\sigma^2(X_n)\}$. Nevertheless, we note that these equations cannot be directly used because of the unknown quantities $\mathbf{r}$ and σ. Therefore a first order asymptotic expansion of the bias and variance

of $\hat{m}_v(x_0) = v!\hat{\boldsymbol{\beta}}_v$ serves as an approximation and is given in the theorem below. The theorem is directly quoted from Fan and Gijbels (1996) but was originally proven by Ruppert and Wand (1994). We use the following notation:

$$
\begin{aligned}
&u_j = \int u^j K(u)du, \quad v_j = \int u^j K^2(u)du, \\
&S = (u_{j+l})_{0\leq j,l\leq p}, \quad \tilde{S} = (u_{j+l+1})_0 \leq j,l \leq p, \\
&S^* = (u_{j+l+1})_0 \leq j,l \leq p, \quad c_p = (\mu_{p+1},\cdots,\mu_{2p+1})^\top, \\
&\tilde{c}_p = (\mu_{p+2},\cdots,\mu_{2p+2})^\top, \quad e_{v+1} = (0,\cdots,0,1,0,\cdots,0)^\top,
\end{aligned}
$$

where e_{v+1} has a 1 on the $(v+1)$ -th position. We also use $Op(1)$ to represent a random quantity that is tending to zero in probability.

Theorem 1. *Assume that $f(x_0) > 0$ and that $f(\cdot), m^{(p+1)}(\cdot)$ and $\sigma^2(\cdot)$ are continuous in a neighborhood of x_0. Further assume that $h \to 0$ and $nh \to \infty$. Then the asymptotic conditional variance of $\hat{m}_v(x_0)$ is given by*

$$
\begin{aligned}
Var(\hat{m}_v(x_0)|\mathbb{X}) = e_{v+1}^\top S^{-1} S^* S^{-1} e_{v+1} \frac{v!^2 \sigma^2(x_0)}{f(x_0) n h^{1+2v}} \\
+ o_p\left(\frac{1}{nh^{1+2v}}\right).
\end{aligned} \tag{2.17}
$$

The asymptotic conditional bias for $p - v$ odd is given by

$$
\begin{aligned}
Bias(\hat{m}_v(x_0)|\mathbb{X}) = e_{v+1}^\top S^{-1} c_p \frac{v!}{(p+1)!} m^{(p+1)}(x_0) h^{p+1-v} \\
+ o_p(h^{p+1-v}).
\end{aligned} \tag{2.18}
$$

Further, for $p-v$ even the asymptotic conditional bias is

$$\begin{aligned} Bias(\hat{m}_v(x_0)|\mathbb{X}) = e_{v+1}^{\top} S^{-1} \tilde{c}_p \frac{v!}{(p+1)!} \{ & m^{(p+2)}(x_0) \\ & + (p+2) m^{(p+1)}(x_0) \frac{f'(x_0)}{f(x_0)} \} h^{p+2-v} \\ & + op(h^{p+2-v}) \end{aligned} \tag{2.19}$$

provided that $f'(\cdot)$ and $m^{(p+2)}(\cdot)$ are continuous in a neighborhood of x_0 and $nh^3 \to \infty$.

From the above theorem, we notice that there is a theoretical difference between odd order fits and even order fits with respect to the asymptotic bias. In fact, Ruppert and Wand (1994) proved that odd order fits are always more desirable over even order fits.

2.4 Varying coefficient models

The varying coefficient model is proven to be a very important generalisation of the linear model whose coefficients are allowed to be functions with respect to some random variable. Equipped with good interpretability, this model is quite useful in exploring the dynamic pattern in many scientific areas, such as economics, finance, politics, epidemiology, medical science, ecology and so on. In past decades, the varying coefficient models have experienced deep and exciting devel-

opments on methodological, theoretical and applied sides. In this section, we give a concise review of the major methodology of the varying coefficient model.

The varying coefficient models were originally introduced by Cleveland, Grosse and Shyu (1991) to extend the applications of local regression techniques from a one dimensional to multi-dimensional setting. The varying coefficient models assume the form of multi-variate regression function as

$$m(Z,\mathbf{X}) = \mathbf{X}^\top \mathbf{g}(Z), \tag{2.20}$$

for unknown functional coefficient $\mathbf{g}(Z) = (g_1(Z),\ldots,g_p(Z))^\top$ and a given scalar Z, where $m(Z,\mathbf{X}) = E(y|Z,\mathbf{X})$ is the regression function.

There are some different approaches to estimate the vector of functional coefficients $\mathbf{g}(\cdot)$ in model (2.20). For example, smoothing spline, see Hastie and Tibshirani (1993) and polynomial spline, proposed by Huang *et al.*(2002, 2004). Because the varying coefficient models are locally linear models, the kernel smoothing has been shown as one of the best methods, see Wu *et al.*(1998) and Fan and Zhang (1999). In the following, we are going to outline the kernel smoothing on varying coefficient model.

Assume that we have the function

$$Y = \sum_{k=1}^{d} g_k(Z)X_k + \epsilon, \tag{2.21}$$

for the univariate index variables Z, covariates $X_1, \cdots, X_d$ and response variable Y with

$$\mathbb{E}(\epsilon|Z, X_1, \cdots, X_d) = 0, \quad \text{Var}(\epsilon|Z, X_1, \cdots, X_d) = \sigma^2(Z).$$

And we note that it is possible for us to consider an intercept by setting $X_1 \equiv 1$. Now, we can directly fit them by the kernel regression locally around the index Z.

Suppose that we have a sample $(z_i, x_{i1}, \ldots, x_{id}, y_i)$, $i = 1, \ldots, n$ from $(Z, X_1, \ldots, X_d, Y)$ in model (2.21), then following the local linear smoothing in Fan and Zhang (1999), for each given z, we locally approximate the function by Taylor expansion that gives

$$g_k(Z_i) \approx a_k + b_k(Z_i - z),$$

for Z_i in a neighbourhood of z. This leads to the local estimation procedure with the smoothing parameter (bandwidth) h as follows

$$\sum_{i=1}^{n} \left\{ y_i - \sum_{k=1}^{d} [a_k + b_k(Z_i - z)] x_{ik} \right\}^2 K_h(Z_i - z), \tag{2.22}$$

The locally weighted least squares (2.22) can be rewritten as

$$\min_{\boldsymbol{\theta}} (\mathbf{y} - \mathbb{X}\boldsymbol{\theta})^\top W (\mathbf{y} - \mathbb{X}\boldsymbol{\theta})$$

where $\mathbf{y} = (y_1, \ldots, y_n)^\top$ and

$$\boldsymbol{\theta} = (a_1, b_1, \cdots, a_d, b_d)^\top,$$
$$W = \text{diag}\{K_h(Z_1 - z), \cdots, K_h(Z_n - z)\},$$

$$\mathbb{X} = \begin{pmatrix} x_{11} & x_{11}(Z_1 - z) & \cdots & x_{1d} & x_{1p}(Z_1 - z) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{n1} & x_{n1}(Z_n - z) & \cdots & x_{nd} & x_{np}(Z_n - z) \end{pmatrix}.$$

The solution is given by the least squares theory that

$$\hat{\boldsymbol{\theta}} = (\mathbb{X}^\top W \mathbb{X})^{-1} \mathbb{X}^\top W \mathbf{y}.$$

The estimator $\hat{\boldsymbol{\theta}}(\cdot)$ is a local linear estimator of $\boldsymbol{\theta}(\cdot)$. The estimate of coefficient function $g_k(z)$ is

$$\hat{g}_k(z) = \boldsymbol{e}_{2k-1,2d}^\top (\mathbb{X}^\top W \mathbb{X})^{-1} \mathbb{X}^\top W \mathbf{y} \tag{2.23}$$

where $\boldsymbol{e}_{2k-1,2d}$ is the unit vector of with length $2d$ and the $2k-1$ component being 1.

In classic varying coefficient models, the index variable Z is known. In order to alleviate the "curse of dimensionality", we introduce the single index model (Hardle and Stoker, 1990) to incorporate with the varying coefficient models. The single index models can be expressed by the following basic form

$$Y = g(X^\top \boldsymbol{\beta}, \epsilon), \tag{2.24}$$

where X is a d dimensional covariate, Y is the response variable, q is an integer smaller than the dimension d and ϵ is the random error. We remark that in the model (2.24), the known index Z is replaced by the linear combination of variables and index parameters $\boldsymbol{\beta}$. By assuming the index

is unknown, Fan *et al.*(2003) explored the adaptive varying coefficient model.

Specifically, suppose that Y is a random variable and $\mathbf{X}$ is a $d \times 1$ random vector. The adaptive varying coefficient linear model in Fan *et al.*(2003) is defined to be the model structure as follows

$$m(\mathbf{x}) = \sum_{k=0}^{d} f_k(\boldsymbol{\beta}^\top \mathbf{x}) x_k, \tag{2.25}$$

where $\mathbf{x} = (x_1 \cdots x_d)^\top, x_0 = 1$, $\boldsymbol{\beta} \in \mathbb{R}^d$ is the vector of unknown index parameters and the varying coefficients $f_0(\cdot), \cdots, f_d(\cdot)$ are unknown functions. We get the estimators of coefficient functions $f_k(\cdot)$ and index parameters $\boldsymbol{\beta}$ by minimising $\mathbb{E}\{G(\mathbf{X}) - g(\mathbf{X})\}^2$. We note that once $\boldsymbol{\beta}$ has been properly fitted, the model (2.25) actually becomes a typical varying coefficient model (2.21) which can be smoothed via the foregoing kernel smoothing. In the case that $\boldsymbol{\beta}$ is unknown, Fan *et al.*(2003) proposed a hybrid backfitting algorithm to estimate the model, which is implemented by an alternating iteration between fitting the index through a one-step scheme and fitting functional coefficients through local linear regression.

CHAPTER 3

ESTIMATION OF SINGLE-INDEX VECTOR AUTOREGRESSIVE MODEL

In this chapter, we first specify the single-index vector autoregressive models (SIVARM) and then explore an iterative method to simultaneously work out the estimators of the index parameters and the local linear estimators of the functional coefficients. By considering the computational cost, we thereby develop an upgraded algorithm to estimate the model more efficiently.

Developing these algorithms has two purposes. Firstly, it establishes efficient penalty-free approaches to fit the SIVARM, which serves as a useful stepping stone for the following model selection and estimation of SIVARM in Chapter 4. Secondly, we will employ these penalty-free iterative procedures to es-

timate the true sub-model directly to obtain the oracle estimates, which will be used as a benchmark to evaluate the estimation accuracy of our proposed penalised approach.

Additionally, a simulation study is used in this chapter to demonstrate the performance of these estimators.

3.1 Model description

Let $\mathbf{y}_t$, $t = 1, \cdots, n$, be a d-dimensional stationary time series, and $\mathbf{X}_t$, $t = 1, \cdots, n$, be identically independent distributed $\mathbf{X}_t$ is a $d \times q$ matrix. The single-index vector autoregressive model is defined as

$$\mathbf{y}_t = \sum_{j=1}^{p} \mathbf{A}_j(\mathbf{y}_{t-\ell}^{\top}\boldsymbol{\beta})\mathbf{y}_{t-j} + \mathbf{X}_t\boldsymbol{\alpha} + \boldsymbol{\epsilon}_t, \tag{3.1}$$

where $\mathbf{A}_j(\cdot)$, $j = 1, \ldots, p$, is the $d \times d$ matrix of varying coefficients, and the first column of $\mathbf{A}_\ell(\cdot)$, $1 \le \ell \le p$, is $\mathbf{0}_d$, $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_d)^{\top} \in \mathbb{R}^d$ and satisfies

$$\|\boldsymbol{\beta}\| = 1, \quad \beta_1 > 0, \tag{3.2}$$

$\alpha \in \mathbb{R}^q$ is the constant coefficients of $\mathbf{X}_t$ and the random noise $\boldsymbol{\epsilon}_t$, $t = 1, \cdots, n$, are identically independent distributed with

$$E(\boldsymbol{\epsilon}_t|\mathscr{F}_t) = \mathbf{0}_d, \qquad \operatorname{cov}(\boldsymbol{\epsilon}_t|\mathscr{F}_t) = \sigma^2\mathbf{I}_d,$$

where $\mathscr{F}_t = \{(Y_{l-1}, \mathbf{X}_l) : \ l \le t\}$. We will estimate the unknown parameters $\boldsymbol{\beta}$, $\mathbf{A}_j(\cdot)$, $\boldsymbol{\alpha}$ and σ^2.

We remark that, for the identifiability purposes, in exactly the same way in the Section 1.3, the first column of $\mathbf{A}_\ell(\cdot)$ is set to be $\mathbf{0}_d$.

3.2 Methodology

In this section, we present an estimation procedure based on the kernel smoothing for the SIVARM. In order to illustrate the estimation procedure, we firstly simplify the model by ignoring the item $\mathbf{X}_t\boldsymbol{\alpha}$. Thus, the model is written as

$$\mathbf{y}_t = \sum_{j=1}^{p} \mathbf{A}_j(\mathbf{y}_{t-\ell}^\top \boldsymbol{\beta})\mathbf{y}_{t-j} + \boldsymbol{\epsilon}_t. \tag{3.3}$$

It is worth noting that once an estimate $\hat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$ is given, model (3.3) becomes a varying coefficient vector auto-regressive model with known index $\mathbf{y}_{t-\ell}^\top \hat{\boldsymbol{\beta}}$. In this situation, estimation of the matrix of coefficient function, $\mathbf{A}_j(\cdot)$, can be achieved by using local linear regression, which is a classical methodology associated with fitting typical varying coefficient models . Approximate $\mathbf{A}_j(\mathbf{y}_{t-\ell}^\top \hat{\boldsymbol{\beta}}), j = 0, \cdots, q$, locally by a Taylor expansion

$$\mathbf{A}_j(\mathbf{y}_{t-\ell}^\top \hat{\boldsymbol{\beta}}) \approx \mathbf{A}_j(z) + \dot{\mathbf{A}}_j(z)(\mathbf{y}_{t-\ell}^\top \hat{\boldsymbol{\beta}} - z),$$

for $\mathbf{y}_{t-\ell}^\top \hat{\boldsymbol{\beta}}$ in a neighbourhood of a given grid point z. By minimising

$$\sum_{t=p+1}^{n} \left\| \mathbf{y}_t - \sum_{j=1}^{p} [\mathbf{A}_j(z) + \dot{\mathbf{A}}_j(z)(\mathbf{y}_{t-\ell}^\top \hat{\boldsymbol{\beta}} - z)]\mathbf{y}_{t-j} \right\|^2 K_h(\mathbf{y}_{t-\ell}^\top \hat{\boldsymbol{\beta}} - z)$$

with respect to $\mathbf{A}_j(z)$ and $\dot{\mathbf{A}}_j(z)$ for $j = 0, \cdots, p$, it follows from the least squares theory that

$$\hat{\boldsymbol{\theta}} = (\mathbb{X}^\top \mathbf{W} \mathbb{X})^{-1} \mathbb{X}^\top \mathbf{W} \mathbf{Y} \tag{3.4}$$

where $\hat{\boldsymbol{\theta}} = (\hat{\mathbf{A}}_1(z), \hat{\dot{\mathbf{A}}}_1(z), \cdots, \hat{\mathbf{A}}_p(z), \hat{\dot{\mathbf{A}}}_p(z)) \in \mathbb{R}^{d \times 2pd}$, $W = \text{diag}\{K_h(\mathbf{y}_{p+1-\ell}^\top \hat{\boldsymbol{\beta}} - z), \cdots, K_h(\mathbf{y}_{n-\ell}^\top \hat{\boldsymbol{\beta}} - z)\}$ is a $(n-p) \times (n-p)$ diagonal matrix with $K_h(\cdot) = K(\cdot/h)/h$ where $K(\cdot)$ is a kernel function and h is a smoothing paramter,

$$\mathbb{X} = \begin{pmatrix} \mathbf{y}_p^\top & (\mathbf{y}_{p+1-\ell}^\top \hat{\boldsymbol{\beta}} - z)\mathbf{y}_p^\top & \cdots & \mathbf{y}_1^\top & (\mathbf{y}_{p+1-\ell}^\top \hat{\boldsymbol{\beta}} - z)\mathbf{y}_1^\top \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{y}_{n-1}^\top & (\mathbf{y}_{n-\ell}^\top \hat{\boldsymbol{\beta}} - z)\mathbf{y}_{n-1}^\top & \cdots & \mathbf{y}_{n-p} & (\mathbf{y}_{n-\ell}^\top \hat{\boldsymbol{\beta}} - z)\mathbf{y}_{n-p}^\top \end{pmatrix},$$

is a $(n-p) \times 2pd$ matrix with $\mathbf{y}_i = (y_1, \cdots, y_d)^\top \in \mathbb{R}^d, i = 1, \cdots, n$ and $\mathbf{Y} = (\mathbf{y}_{p+1}, \cdots, \mathbf{y}_n)^\top \in \mathbb{R}^{(n-p) \times d}$.

With this in mind, the primary target now is to estimate the index parameter $\boldsymbol{\beta}$. Because of the complex relationship between the response and covariates in the autoregressive model, it is hard to work out an estimator of $\boldsymbol{\beta}$ directly. Hence we need to develop an iterative computational algorithm.

3.2.1 Iterative procedure for estimating the single-index vector autoregressive models

Let us go back to our SIVARM. For any given j, $j = 1, \cdots, p$, and k, $k = 1, \cdots, n-\ell$, by the Taylor's expansion,

$$\mathbf{A}_j(\mathbf{y}_i^\top \boldsymbol{\beta}) \approx \mathbf{A}_j(\mathbf{y}_k^\top \boldsymbol{\beta}) + \dot{\mathbf{A}}_j(\mathbf{y}_k^\top \boldsymbol{\beta}) \left[(\mathbf{y}_i - \mathbf{y}_k)^\top \boldsymbol{\beta}\right],$$

when $\mathbf{y}_i^\top \boldsymbol{\beta}$ is in a small neighbourhood of $\mathbf{y}_k^\top \boldsymbol{\beta}$, where $\dot{\mathbf{A}}_j(\cdot)$ is the derivative of $\mathbf{A}_j(\cdot)$. Therefore we can approximate model (3.1) by

$$\mathbf{y}_t \approx \sum_{j=1}^{p} \{\mathbf{A}_j(\mathbf{y}_k^\top \boldsymbol{\beta}) + \dot{\mathbf{A}}_j(\mathbf{y}_k^\top \boldsymbol{\beta})(\mathbf{y}_{t-\ell}^\top \boldsymbol{\beta} - \mathbf{y}_k^\top \boldsymbol{\beta})\}\mathbf{y}_{t-j} + \mathbf{X}_t \boldsymbol{\alpha} + \boldsymbol{\epsilon}_t. \quad (3.5)$$

For concision purposes, we use the notation

$$\begin{aligned}
&\mathbf{B}_{j,k} = \dot{\mathbf{A}}_{j,k} \in \mathbb{R}^{d\times d},\\
&\boldsymbol{\theta}_k = (\mathbf{A}_{1,k}, \mathbf{B}_{1,k}, \cdots, \mathbf{A}_{p,k}, \mathbf{B}_{p,k}) \in \mathbb{R}^{d\times 2pd},\\
&\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_{n-l}) \in \mathbb{R}^{d\times 2pd(n-l)},
\end{aligned}$$

where $\mathbf{A}_j(\mathbf{y}_k^\top \boldsymbol{\beta})$ and $\dot{\mathbf{A}}_j(\mathbf{y}_k^\top \boldsymbol{\beta})$ are denoted by $\mathbf{A}_{j,k}$ and $\dot{\mathbf{A}}_{j,k}$, respectively.

By using approximation (3.5) together with the idea of least squares, we can form the following local discrepancy loss

function:

$$L(\boldsymbol{\beta},\boldsymbol{\alpha},\boldsymbol{\Theta}) = \sum_{k=1}^{n-l}\sum_{t=p+1}^{n}\Bigg\|\mathbf{y}_t - \sum_{j=1}^{p}\Big\{\mathbf{A}_{j,k} + \mathbf{B}_{j,k}\left[(\mathbf{y}_{t-\ell}-\mathbf{y}_k)^\top\boldsymbol{\beta}\right]\Big\}\mathbf{y}_{t-j} - \mathbf{X}_t\boldsymbol{\alpha}\Bigg\|^2 K_h((\mathbf{y}_{t-\ell}-\mathbf{y}_k)^\top\boldsymbol{\beta}), \tag{3.6}$$

where $K_h(\cdot) = K(\cdot/h)/h$ is a probability density function with a kernel function $K(\cdot)$ and the bandwidth h. Then, the estimator $\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\Theta}}$, and $\hat{\boldsymbol{\alpha}}$ can be obtained by solving

$$(\hat{\boldsymbol{\beta}},\hat{\boldsymbol{\alpha}},\hat{\boldsymbol{\Theta}}) = \underset{\boldsymbol{\beta},\boldsymbol{\alpha},\boldsymbol{\Theta}}{\arg\min}\, L(\boldsymbol{\beta},\boldsymbol{\alpha},\boldsymbol{\Theta}), \tag{3.7}$$

subject to the constraints $\|\boldsymbol{\beta}\| = 1$ and $\beta_1 > 0$. A global minimum of the local discrepancy function cannot be found analytically and thereby an iterative procedure is proposed for implementation. We note that at each step, there exists a closed form solution. The proposed iterative algorithm can be broken down as follows:

Step 1. Choose initial value $\boldsymbol{\beta}^0$. Before the iterative procedure, a consistent initial value of $\boldsymbol{\beta}$ should be specified, which is denoted by $\boldsymbol{\beta}^0$. The initial value $\boldsymbol{\beta}^0$ also satisfies the constraints that $\beta_1^0 > 0$ and $\|\boldsymbol{\beta}^0\| = 1$, where β_1^0 is the first component of $\boldsymbol{\beta}^0$.

Step 2. Estimate $\boldsymbol{\Theta}, \boldsymbol{\alpha}$ with $\boldsymbol{\beta}$ taking the initial estimated

value. Set $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}^0$, we now estimate $\boldsymbol{\Theta}, \boldsymbol{\alpha}$ by solving

$$(\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\alpha}}) = \arg\min_{\boldsymbol{\Theta}, \boldsymbol{\alpha}} L(\boldsymbol{\Theta}, \boldsymbol{\alpha} | \boldsymbol{\beta}).$$

For each k, the estimators of $\boldsymbol{\theta}_k, \boldsymbol{\alpha}$ can be obtained by minimising the following locally weighted function

$$L(\boldsymbol{\Theta}, \boldsymbol{\alpha} | \boldsymbol{\beta}) = \sum_{t=p+1}^{n} \Big\| \mathbf{y}_t - \sum_{j=1}^{p} \Big\{ \mathbf{A}_{j,k} + \mathbf{B}_{j,k} \left[(\mathbf{y}_{t-\ell} - \mathbf{y}_k)^\top \hat{\boldsymbol{\beta}} \right] \Big\} \mathbf{y}_{t-j} - \mathbf{X}_t \boldsymbol{\alpha} \Big\|^2 K_h((\mathbf{y}_{t-\ell} - \mathbf{y}_k)^\top \hat{\boldsymbol{\beta}}). \tag{3.8}$$

Then, we write (3.8) in a matrix notation

$$L(\boldsymbol{\Theta}, \boldsymbol{\alpha} | \boldsymbol{\beta}) = \sum_{t=p+1}^{n} \Big\| \mathbf{y}_t - \mathbf{Q}_{t,k}^\top \boldsymbol{\theta}_k - \mathbf{X}_t \boldsymbol{\alpha} \Big\|^2 K_h((\mathbf{y}_{t-\ell} - \mathbf{y}_k)^\top \hat{\boldsymbol{\beta}}), \tag{3.9}$$

with respect to $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_{n-l})$ and $\boldsymbol{\alpha}$, where $\mathbf{Q}_{t,k} = (\mathbf{y}_{t-1}^\top, (\mathbf{y}_{t-\ell}^\top \hat{\boldsymbol{\beta}} - z_k)\mathbf{y}_{t-1}^\top, \cdots, \mathbf{y}_{t-p}^\top, (\mathbf{y}_{t-\ell}^\top \hat{\boldsymbol{\beta}} - z_k)\mathbf{y}_{t-p}^\top)$ with $z_k = \mathbf{y}_k^\top \hat{\boldsymbol{\beta}}$.

To fulfil the condition of identifiability that the first column of $\mathbf{A}_\ell(\cdot)$ is $\mathbf{0}_d$, instead of estimating the first column of $\mathbf{A}_\ell(\cdot)$ in the estimation procedure, we let the elements of the first column of $\mathbf{A}_\ell(\cdot)$ equal 0 directly, which leads to a reduction of dimension of $\boldsymbol{\theta}_k$ from $d \times 2pd$ to $d \times (2pd - 2)$. Correspondingly, both of the first element of $\mathbf{y}_{t-\ell}$ and $(\mathbf{y}_{t-\ell}^\top \hat{\boldsymbol{\beta}} - z_k)\mathbf{y}_{t-\ell}$ of $\mathbf{Q}_{t,k}$ need to be eliminated in the estimation procedure, therefore, the dimension of $\mathbf{Q}_{t,k}$ reduces from $1 \times 2pd$ to $1 \times (2pd - 2)$.

To build a classical weighted least squares, we would like to transform (3.6) into the function as follows:

$$L(\gamma_k | \boldsymbol{\beta}) = \big\| \mathbf{y}_t - \mathbb{X}_{t,k} \boldsymbol{\gamma}_k \big\|^2 K_h((\mathbf{y}_{t-\ell} - \mathbf{y}_k)^\top \hat{\boldsymbol{\beta}}), \tag{3.10}$$

with respect to $\boldsymbol{\gamma}_k = [\boldsymbol{\theta}_k, \boldsymbol{\alpha}]$, where $\mathbb{X}_{t,k} = [\mathbf{Q}_{t,k}, \mathbf{X}_t]$, $\boldsymbol{\Gamma} = (\boldsymbol{\gamma}_1, \cdots, \boldsymbol{\gamma}_{n-l})$.

However, the augmented matrix $\boldsymbol{\gamma}_k = [\boldsymbol{\theta}_k, \boldsymbol{\alpha}]$ cannot be integrated by $\boldsymbol{\theta}_k \in \mathbb{R}^{d\times(2pd-2)}, \boldsymbol{\alpha} \in \mathbb{R}^{q\times 1}$ directly. Analogically, a befitting $\mathbb{X}_{t,k}$ cannot be directly merged by $\mathbf{Q}_{t,k} \in \mathbb{R}^{1\times(2pd-2)}$ and $\mathbf{X}_t \in \mathbb{R}^{d\times q}$. Therefore, we need to do some matrix transformations to obtain the proper augmented matrices $\boldsymbol{\gamma}_k$ and $\mathbb{X}_{t,k}$.

Let $a_{j\imath\jmath k}$ and $b_{j\imath\jmath k}$ be the $(\imath, \jmath)$ -th element of $\mathbf{A}_{j,k}$ and $\mathbf{B}_{j,k}$ respectively. We transfer matrix $\boldsymbol{\theta}_k = (\mathbf{A}_{1,k}, \mathbf{B}_{1,k}, \cdots, \mathbf{A}_{p,k}, \mathbf{B}_{p,k})$ to a vector

$$\begin{aligned}\boldsymbol{\theta}_k^{\dagger} \;=\; & (a_{111k}, \cdots, a_{1ddk}, b_{111k}, \cdots, b_{1ddk}, \cdots, a_{l12k}, \\ & \cdots, a_{lddk}, b_{l12k}, \cdots, b_{lddk}, \cdots, a_{p11k}, \cdots, a_{pddk}, \\ & b_{p11k}, \cdots, b_{pddk})^{\top} \in \mathbb{R}^{(2pdd-2d)\times 1}.\end{aligned}$$

Hence $\boldsymbol{\gamma}_k = [\boldsymbol{\theta}_k^{\dagger}, \boldsymbol{\alpha}]^{\top}$ is constructed to be a $(2pd^2 - 2d + q) \times 1$ vector.

The same method is used in $\mathbb{X}_{t,k} = [\mathbf{Q}_{t,k}, \mathbf{X}_t]$. Because the $\mathbf{X}_t$ is a $d \times q$ matrix, we design a matrix based on $\mathbf{Q}_{t,k}$ which has the same number of rows as $\mathbf{X}_t$:

$$\mathbf{Q}_{t,k}^{\dagger} = \begin{pmatrix} \mathbf{Q}_{t,k} & 0 & \cdots & 0 \\ 0 & \mathbf{Q}_{t,k} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & 0 & 0 & \mathbf{Q}_{t,k} \end{pmatrix},$$

where $\mathbf{Q}_{t,k}^{\dagger} \in \mathbb{R}^{d\times(2pdd-2d)}$. With this new $\mathbf{Q}_{t,k}^{\dagger}$, we have an augmented matrix $\mathbb{X}_{t,k} \in \mathbb{R}^{d\times(2pdd-2d+q)}$. Through the least squares theory, the solution is given by

$$\hat{\boldsymbol{\gamma}}_k = (\sum_{t=p+1}^{n} \mathbb{X}_{t,k}^{\top} K_h(\cdot)\mathbb{X}_{t,k})^{-1}(\sum_{t=p+1}^{n} \mathbb{X}_{t,k}^{\top} K_h(\cdot)\mathbf{y}_t), \tag{3.11}$$

and hence we obtain $\hat{\boldsymbol{\Gamma}} = (\hat{\boldsymbol{\gamma}}_1, \cdots, \hat{\boldsymbol{\gamma}}_{n-l}) \in \mathbb{R}^{(2pdd-2d+q)\times(n-l)}$.

Step 3. Estimate $\boldsymbol{\beta}$ given $\boldsymbol{\Gamma}$. Using the estimators $\hat{\boldsymbol{\Gamma}}$ from Step 2, we would like to find the estimator of $\boldsymbol{\beta}$:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} L(\boldsymbol{\beta}|\hat{\boldsymbol{\Gamma}})$$

which is equivalent to

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{k=1}^{n-l}\sum_{t=p+1}^{n} \Big\| \mathbf{y}_t - \sum_{j=1}^{p} \Big\{ \hat{\mathbf{A}}_{j,k} + \hat{\mathbf{B}}_{j,k}\left[(\mathbf{y}_{t-\ell} - \mathbf{y}_k)^{\top}\boldsymbol{\beta}\right] \Big\} \mathbf{y}_{t-j} - \mathbf{X}_t\hat{\boldsymbol{\alpha}} \Big\|^2 K_h((\mathbf{y}_{t-\ell} - \mathbf{y}_k)^{\top}\hat{\boldsymbol{\beta}}), \tag{3.12}$$

We note that only $\boldsymbol{\beta}$ in the least squares part of the loss function is the parameter to be estimated, the $\hat{\boldsymbol{\beta}}$ appearing in the kernel function is the estimator of $\boldsymbol{\beta}$ we used in Step 2. For the sake of distinguishing the two $\boldsymbol{\beta}$s, we use $\boldsymbol{\beta}_{new}$ and $\boldsymbol{\beta}_{old}$ to rewrite the approximation:

$$\hat{\boldsymbol{\beta}}_{new} = \arg\min_{\boldsymbol{\beta}_{new}} \sum_{k=1}^{n-l}\sum_{t=p+1}^{n} \Big\| \mathbf{y}_t - \sum_{j=1}^{p} \Big\{ \hat{\mathbf{A}}_{j,k} + \hat{\mathbf{B}}_{j,k}\left[(\mathbf{y}_{t-\ell} - \mathbf{y}_k)^{\top}\boldsymbol{\beta}_{new}\right] \Big\} \times \mathbf{y}_{t-j} - \mathbf{X}_t\hat{\boldsymbol{\alpha}} \Big\|^2 K_h((\mathbf{y}_{t-\ell} - \mathbf{y}_k)^{\top}\hat{\boldsymbol{\beta}}_{old}), \tag{3.13}$$

To obtain the estimator $\boldsymbol{\beta}_{new}$, we shall minimise the following function:

$$L(\boldsymbol{\beta}|\hat{\boldsymbol{\Gamma}}) = \sum_{k=1}^{n-l}\sum_{t=p+1}^{n}\Big\{\mathbf{y}_t - \sum_{j=1}^{p}\hat{\mathbf{A}}_{j,k}\mathbf{y}_{t-j} + \sum_{j=1}^{p}\hat{\mathbf{B}}_{j,k}\mathbf{y}_{t-j}(\mathbf{y}_{t-\ell}-\mathbf{y}_k)^\top \times \boldsymbol{\beta}_{new} - \mathbf{X}_t\hat{\boldsymbol{\alpha}}\Big\}^2 K_h((\mathbf{y}_{t-\ell}-\mathbf{y}_k)^\top\hat{\boldsymbol{\beta}}_{old}). \tag{3.14}$$

To obtain the closed form solution, we rewrite the minimisation problem (3.14) in matrix notation:

$$L(\boldsymbol{\beta}|\hat{\boldsymbol{\Gamma}}) = \arg\min_{\boldsymbol{\beta}_{new}} \left\|\mathbf{c}_{t,k} - \mathbf{M}_{t,k}\boldsymbol{\beta}_{new}\right\|^2 K_h((\mathbf{y}_{t-\ell}-\mathbf{y}_k)^\top\hat{\boldsymbol{\beta}}_{old}), \tag{3.15}$$

where:

$$\mathbf{c}_{t,k} = \mathbf{y}_t - \sum_{j=1}^{p}\hat{\mathbf{A}}_{j,k}\mathbf{y}_{t-j} - \mathbf{X}_t\hat{\boldsymbol{\alpha}} \in \mathbb{R}^{d\times 1},$$

$$\mathbf{M}_{t,k} = \sum_{j=1}^{p}\hat{\mathbf{B}}_{j,k}\mathbf{y}_{t-j}(\mathbf{y}_{t-\ell}-\mathbf{y}_k)^\top \in \mathbb{R}^{d\times d}.$$

Following the least squares theory, the solution can be obtained by :

$$\hat{\boldsymbol{\beta}}_{new} = \Big(\sum_{k=1}^{n-l}\sum_{t=p+1}^{n}\mathbf{M}_{t,k}^\top K_h((\mathbf{y}_{t-\ell}-\mathbf{y}_k)^\top\hat{\boldsymbol{\beta}}_{old})\mathbf{M}_{t,k}\Big)^{-1} \times \Big(\sum_{k=1}^{n-l}\sum_{t=p+1}^{n}\mathbf{M}_{t,k}^\top K_h((\mathbf{y}_{t-\ell}-\mathbf{y}_k)^\top\hat{\boldsymbol{\beta}}_{old})\mathbf{c}_{t,k}\Big). \tag{3.16}$$

Then, to satisfy the identifiability conditions that $\|\boldsymbol{\beta}\| = 1$ and $\beta_1 > 0$, we define $\hat{\boldsymbol{\beta}}_{new} = \hat{\boldsymbol{\beta}}_{new}/\|\hat{\boldsymbol{\beta}}_{new}\|$ if the first component of

$\hat{\boldsymbol{\beta}}_{new}$ is positive, otherwise, $\hat{\boldsymbol{\beta}}_{new} = -\hat{\boldsymbol{\beta}}_{new}/\|\hat{\boldsymbol{\beta}}_{new}\|$. We update the initial value $\boldsymbol{\beta}^0$ in Step 2 to $\boldsymbol{\beta}_{new}$, and iterate between Step 2 and Step 3 until convergence.

3.2.2 Computationally efficient estimation method

In Section 3.2.1, we proposed an iterative estimation method to effectively fit the SIVARM. However, we notice that with the increase of the dimension d, the computational cost of the previous algorithm increases rapidly, and hence care must be taken from a computational point of view to reduce this cost.

By recalling the previous estimation approach, we note that for the sake of applying local weighted least square to simultaneously estimate varying coefficients $\boldsymbol{\theta}_k$ and the parameter $\boldsymbol{\alpha}$ given initial value of $\boldsymbol{\beta}$, the vector $\mathbf{Q}_{t,k} \in \mathbb{R}^{1\times(2pd-2)}$ is transformed to a matrix with largely increased dimensions:

$$\mathbf{Q}_{t,k}^{\dagger} = \begin{pmatrix} \mathbf{Q}_{t,k} & 0 & \cdots & 0 \\ 0 & \mathbf{Q}_{t,k} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \mathbf{Q}_{t,k} \end{pmatrix} \in \mathbb{R}^{d\times(2pdd-2d)}.$$

Obviously, it leads to a huge computational cost.

Therefore, in this section, we will take the challenge to explore a more efficient estimation approach. Additionally, To

verify the efficiency of this method, a comparison between the two estimation methods will be provided in Section 4.2.2.

Let $\mathbf{c}_{m,j}(\cdot), m = 1, \cdots, d$ be the m -th row of the matrix $\mathbf{A}_j(\cdot)$, $\mathbf{y}_t = (y_{t,1}, \cdots, y_{t,d})^\top$, $\mathbf{X}_t = (\mathbf{x}_{t,1}, \cdots, \mathbf{x}_{t,d})^\top$, with $\mathbf{x}_{t,m} \in \mathbb{R}^{1\times q}, m = 1, \cdots, d$ and $\boldsymbol{\epsilon}_t = (\epsilon_{t,1}, \cdots, \epsilon_{t,d})$, $t = 1, \cdots, n$, be i.i.d. with

$$E(\boldsymbol{\epsilon}_t|\mathscr{F}_t) = \mathbf{0}_d \qquad \text{cov}(\boldsymbol{\epsilon}_t|\mathscr{F}_t) = \sigma^2\mathbf{I}_d,$$

where $\mathscr{F}_t = \{(Y_{l-1},\ \mathbf{X}_l) :\ l \le t\}$. We rewrite the model (3.1) by

$$\begin{pmatrix} y_{t,1} \\ y_{t,2} \\ \vdots \\ y_{t,d} \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^p \mathbf{c}_{1,j}(\mathbf{y}_{t-\ell}^\top\boldsymbol{\beta})\mathbf{y}_{t-j} \\ \sum_{j=1}^p \mathbf{c}_{2,j}(\mathbf{y}_{t-\ell}^\top\boldsymbol{\beta})\mathbf{y}_{t-j} \\ \vdots \\ \sum_{j=1}^p \mathbf{c}_{d,j}(\mathbf{y}_{t-\ell}^\top\boldsymbol{\beta})\mathbf{y}_{t-j} \end{pmatrix} + \begin{pmatrix} \mathbf{x}_{t,1}\boldsymbol{\alpha} \\ \mathbf{x}_{t,2}\boldsymbol{\alpha} \\ \vdots \\ \mathbf{x}_{t,d}\boldsymbol{\alpha} \end{pmatrix} + \begin{pmatrix} \epsilon_{t,1} \\ \epsilon_{t,2} \\ \vdots \\ \epsilon_{t,d} \end{pmatrix}. \tag{3.17}$$

To avoid applying the complicated matrix transformation described in Section 3.2.1, we approximate the parameter $\boldsymbol{\alpha}$ by $\boldsymbol{\alpha} \approx 1/d \sum_{m=1}^d \alpha_m$, $m = 1, \ldots, d$, and thereby the $d \times 1$ vector $\mathbf{X}_t\boldsymbol{\alpha}$ can been substituted by the approximation $(\mathbf{x}_{t,1}\alpha_1, \ldots, \mathbf{x}_{t,d}\alpha_d)^\top \in \mathbb{R}^{d\times 1}$. Accordingly, the model (3.17) can be approximated as

$$\begin{pmatrix} y_{t,1} \\ y_{t,2} \\ \vdots \\ y_{t,d} \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^p \mathbf{c}_{1,j}(\mathbf{y}_{t-\ell}^\top\boldsymbol{\beta})\mathbf{y}_{t-j} \\ \sum_{j=1}^p \mathbf{c}_{2,j}(\mathbf{y}_{t-\ell}^\top\boldsymbol{\beta})\mathbf{y}_{t-j} \\ \vdots \\ \sum_{j=1}^p \mathbf{c}_{d,j}(\mathbf{y}_{t-\ell}^\top\boldsymbol{\beta})\mathbf{y}_{t-j} \end{pmatrix} + \begin{pmatrix} \mathbf{x}_{t,1}\alpha_1 \\ \mathbf{x}_{t,2}\alpha_2 \\ \vdots \\ \mathbf{x}_{t,d}\alpha_d \end{pmatrix} + \begin{pmatrix} \epsilon_{t,1} \\ \epsilon_{t,2} \\ \vdots \\ \epsilon_{t,d} \end{pmatrix}. \tag{3.18}$$

Hence, we can obtain the expression of $y_{t,m}$, the m -th compo-

nent of $\mathbf{y}_t$, as

$$y_{t,m} = \sum_{j=1}^{p} \boldsymbol{c}_{m,j}(\mathbf{y}_{t-\ell}^{\top}\boldsymbol{\beta})\mathbf{y}_{t-j} + \mathbf{x}_{t,m}\alpha_m + \epsilon_{t,m}. \tag{3.19}$$

For any given j, $j = 1, \cdots, p$, and k, $k = 1, \cdots, n-\ell$, by applying Taylor's expansion,

$$\boldsymbol{c}_{m,j}(\mathbf{y}_i^{\top}\boldsymbol{\beta}) \approx \boldsymbol{c}_{m,j}(\mathbf{y}_k^{\top}\boldsymbol{\beta}) + \dot{\boldsymbol{c}}_{m,j}(\mathbf{y}_k^{\top}\boldsymbol{\beta})\left[(\mathbf{y}_i - \mathbf{y}_k)^{\top}\boldsymbol{\beta}\right],$$

when $\mathbf{y}_i^{\top}\boldsymbol{\beta}$ is in a small neighbourhood of $\mathbf{y}_k^{\top}\boldsymbol{\beta}$, where $\dot{\boldsymbol{c}}_{m,j}(\cdot)$ is the derivative of $\boldsymbol{c}_{m,j}(\cdot)$. Therefore we can approximate model (3.19) by

$$\begin{aligned} y_{t,m} \approx \sum_{j=1}^{p} \left\{\boldsymbol{c}_{m,j}(\mathbf{y}_k^{\top}\boldsymbol{\beta}) + \dot{\boldsymbol{c}}_{m,j}(\mathbf{y}_k^{\top}\boldsymbol{\beta})(\mathbf{y}_{t-\ell}^{\top}\boldsymbol{\beta} - \mathbf{y}_k^{\top}\boldsymbol{\beta})\right\}\mathbf{y}_{t-j} + \\ \mathbf{x}_{t,m}\alpha_m + \epsilon_{t,m}, \end{aligned} \tag{3.20}$$

For brevity purposes, we use the notation

$$\begin{aligned} &\boldsymbol{d}_{m,j,k} = \dot{\boldsymbol{c}}_{m,j,k}, \\ &\boldsymbol{\theta}_{m,k} = (\boldsymbol{c}_{m,1,k}, \boldsymbol{d}_{m,1,k}, \cdots, \boldsymbol{c}_{m,p,k}, \boldsymbol{c}_{m,p,k}), \\ &\boldsymbol{\Theta} = (\sum_{m=1}^{d} \boldsymbol{\theta}_{m,1}, \cdots, \sum_{m=1}^{d} \boldsymbol{\theta}_{m,n-l}), \end{aligned}$$

where we denote $\boldsymbol{c}_{m,j}(\mathbf{y}_k^{\top}\boldsymbol{\beta})$ and $\dot{\boldsymbol{c}}_{m,j}(\mathbf{y}_k^{\top}\boldsymbol{\beta})$ by $\boldsymbol{c}_{m,j,k}$ and $\dot{\boldsymbol{d}}_{m,j,k}$ respectively.

Using approximation (3.20) together with the idea of least squares, we can form the following local discrepancy loss

function:

$$L(\boldsymbol{\beta},\boldsymbol{\alpha},\boldsymbol{\Theta}) = \sum_{m=1}^{d}\sum_{k=1}^{n-l}\sum_{t=p+1}^{n}\Big\{y_{t,m} - \sum_{j=1}^{p}\left[\boldsymbol{c}_{m,j,k} + \boldsymbol{d}_{m,j,k}(\mathbf{y}_{t-\ell}^{\top}\boldsymbol{\beta} - \mathbf{y}_{k}^{\top}\boldsymbol{\beta})\right]\mathbf{y}_{t-j} - \mathbf{x}_{t,m}\alpha_m\Big\}^2 K_h((\mathbf{y}_{t-\ell} - \mathbf{y}_k)^{\top}\boldsymbol{\beta}), \tag{3.21}$$

where $K_h(\cdot) = K(\cdot/h)/h$, $K(\cdot)$ is a kernel function, h is a bandwidth.

The estimator $\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\Theta}}$, and $\hat{\boldsymbol{\alpha}}$ can be obtained by solving

$$(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\Theta}}) = \underset{\boldsymbol{\beta},\boldsymbol{\alpha},\boldsymbol{\Theta}}{\arg\min}\, L(\boldsymbol{\beta},\boldsymbol{\alpha},\boldsymbol{\Theta}) \tag{3.22}$$

subject to the constraints $\|\boldsymbol{\beta}\| = 1$ and $\beta_1 > 0$. As with the previous estimation method, the global minimum of the local discrepancy function cannot be found analytically and therefore an iterative procedure is proposed for implementation purposes, which is broken down as follows:

Step 1. Choose initial value $\boldsymbol{\beta}^0$. Before the iterative procedure, an initial value of $\boldsymbol{\beta}$ should be specified, which is denoted by $\boldsymbol{\beta}^0$. The initial value $\boldsymbol{\beta}^0$ needs to satisfy the constraints $\boldsymbol{\beta}_1^0 > 0$ and $\|\boldsymbol{\beta}^0\| = 1$, where $\boldsymbol{\beta}_1^0$ is the first component of $\boldsymbol{\beta}^0$.

Step 2. Estimate $\boldsymbol{\Theta}, \boldsymbol{\alpha}$ with $\boldsymbol{\beta}$ taking the initial estimated value. Set $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}^0$, we now estimate $\boldsymbol{\Theta}, \boldsymbol{\alpha}$ by solving

$$(\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\alpha}}) = \underset{\boldsymbol{\Theta},\boldsymbol{\alpha}}{\arg\min}\, L(\hat{\boldsymbol{\beta}}, \boldsymbol{\Theta}, \boldsymbol{\alpha})$$

For each k and m, we choose the estimator of $\boldsymbol{\theta}_{m,k}, \alpha_m$ by minimising

$$\sum_{t=p+1}^{n} \Big\{ y_{t,m} - \sum_{j=1}^{p} \left[\boldsymbol{c}_{m,j,k} + \boldsymbol{d}_{m,j,k} (\mathbf{y}_{t-\ell}^{\top} \hat{\boldsymbol{\beta}} - \mathbf{y}_{k}^{\top} \hat{\boldsymbol{\beta}}) \right] \mathbf{y}_{t-j} - \mathbf{x}_{t,m} \alpha_m \Big\}^2 \times K_h((\mathbf{y}_{t-\ell} - \mathbf{y}_k)^{\top} \hat{\boldsymbol{\beta}}). \tag{3.23}$$

Rewriting the minimisation problem yields

$$\sum_{t=p+1}^{n} \left\| y_{t,m} - \mathbf{Q}_{t,k} \boldsymbol{\theta}_{m,k}^{\top} - \mathbf{x}_{t,m} \alpha_m \right\|^2 K_h((\mathbf{y}_{t-\ell} - \mathbf{y}_k)^{\top} \hat{\boldsymbol{\beta}}), \tag{3.24}$$

where $\mathbf{Q}_{t,k} = (\mathbf{y}_{t-1}^{\top}, (\mathbf{y}_{t-\ell}^{\top} \hat{\boldsymbol{\beta}} - z_k) \mathbf{y}_{t-1}^{\top}, \cdots, \mathbf{y}_{t-p}^{\top}, (\mathbf{y}_{t-\ell}^{\top} \hat{\boldsymbol{\beta}} - z_k) \mathbf{y}_{t-p}^{\top})$, in which $z_k = \mathbf{y}_k^{\top} \hat{\boldsymbol{\beta}}$.

Analogically, the parameters in the model should satisfy the constraints that $\|\boldsymbol{\beta}\| = 1, \beta_1 > 0$ and the first column of $\mathbf{A}_{\ell}(\cdot)$ is $\mathbf{0}_d$. In exactly the same way, we set the elements of the first column of $\mathbf{A}_{\ell}(\cdot)$ equal to 0, which leads to a reduction of dimension of $\boldsymbol{\theta}_{m,k}$ from $(2pd \times 1)$ to $(2pd - 2) \times 1$. Correspondingly, both of the first element of $\mathbf{y}_{t-\ell}$ and $(\mathbf{y}_{t-\ell}^{\top} \hat{\boldsymbol{\beta}} - z_k) \mathbf{y}_{t-\ell}$ of $\mathbf{Q}_{t,k}$ need to be eliminated in the estimation procedure, therefore, the dimension of $\mathbf{Q}_{t,k}$ reduces from $1 \times 2pd$ to $1 \times (2pd - 2)$.

To reduce more computational cost, we notice that we can rewrite the single summation in the equation (3.24) in the form of stacked vectors and matrices. This can be achieved by defining the $(n-p) \times 1$ vector $\mathbf{y}_m$, the $(n-p) \times (2pd-2)$ matrix $\mathbb{Q}_k$, the $(n-p) \times q$ matrix $\mathbf{X}_m$ and the $(n-p) \times (n-p)$ diagonal

matrix $\mathbf{W}_k$ as follows:

$$\mathbf{y}_m = (y_{1,m}, \cdots, y_{n-p,m})^\top, \quad \mathbb{Q}_k = (\mathbf{Q}_{1,k}^\top, \cdots, \mathbf{Q}_{n-p,k}^\top)^\top,$$

$$\mathbf{X}_m = (\mathbf{x}_{1,m}^\top, \cdots, \mathbf{x}_{n-p,m}^\top)^\top,$$

$$\mathbf{W}_k = \operatorname{diag}\Big\{K_h((\mathbf{y}_{1-\ell} - \mathbf{y}_k)^\top \hat{\boldsymbol{\beta}}), \cdots, K_h((\mathbf{y}_{n-p-\ell} - \mathbf{y}_k)^\top \hat{\boldsymbol{\beta}})\Big\},$$

With these, equation (3.24) can be written in matrix notation as:

$$(\mathbf{y}_m - \mathbb{Q}_k \boldsymbol{\theta}_{m,k}^\top - \mathbf{X}_m \alpha_m)^\top \mathbf{W}_k (\mathbf{y}_m - \mathbb{Q}_k \boldsymbol{\theta}_{m,k}^\top - \mathbf{X}_m \alpha_m), \tag{3.25}$$

To obtain the closed form solution, we rewrite the equation (3.25) by:

$$\hat{\boldsymbol{\gamma}}_{m,k} = \underset{\boldsymbol{\gamma}_{m,k}}{\arg\min} \left\| \mathbf{y}_m - \mathbb{X}_{m,k} \boldsymbol{\gamma}_{m,k} \right\|^2 \mathbf{W}_k, \tag{3.26}$$

where $\boldsymbol{\gamma}_{m,k} = [\boldsymbol{\theta}_{m,k}^\top, \alpha_m]^\top$ is a $(2pd - 2 + q) \times 1$ vector, $\mathbb{X}_{m,k} = [\mathbb{Q}_k, \mathbf{X}_m] \in \mathbb{R}^{(n-p)\times(2pd-2+q)}$, $\boldsymbol{\Gamma} = (\sum_{m=1}^d \boldsymbol{\gamma}_{m,1}, \cdots, \sum_{m=1}^d \boldsymbol{\gamma}_{m,n-l})$.

It follows the least square theory that

$$\hat{\boldsymbol{\gamma}}_{m,k} = (\mathbb{X}_{m,k}^\top \mathbf{W}_k \mathbb{X}_{m,k})^{-1} (\mathbb{X}_{m,k}^\top \mathbf{W}_k \mathbf{y}_m), \tag{3.27}$$

and hence we obtain $\hat{\boldsymbol{\Gamma}} = (\sum_{m=1}^d \hat{\boldsymbol{\gamma}}_{m,1}, \cdots, \sum_{m=1}^d \hat{\boldsymbol{\gamma}}_{m,n-l})$.

Step 3. Estimate $\boldsymbol{\beta}$ given $\boldsymbol{\Gamma}$. Using the estimators $\hat{\boldsymbol{\Gamma}}$ from Step 1, we would like to find the estimator of $\boldsymbol{\beta}$:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\arg\min}\, L(\boldsymbol{\beta}, \hat{\boldsymbol{\Gamma}})$$

which is equivalent to minimising

$$\sum_{k=1}^{n-l}\sum_{t=p+1}^{n}\left\|\mathbf{y}_t-\sum_{j=1}^{p}\left\{\hat{\mathbf{A}}_{j,k}+\hat{\mathbf{B}}_{j,k}\left[(\mathbf{y}_{t-\ell}-\mathbf{y}_k)^\top\boldsymbol{\beta}\right]\right\}\mathbf{y}_{t-j}-\mathbf{X}_t\hat{\boldsymbol{\alpha}}\right\|^2 \times K_h((\mathbf{y}_{t-\ell}-\mathbf{y}_k)^\top\hat{\boldsymbol{\beta}}), \tag{3.28}$$

This step is similar to step 3 of the previous algorithm, therefore, we can rewrite the equation (3.28) in matrix notation as

$$\left\|\mathbf{c}_{t,k}-\mathbf{M}_{t,k}\boldsymbol{\beta}_{new}\right\|^2 K_h((\mathbf{y}_{t-\ell}-\mathbf{y}_k)^\top\hat{\boldsymbol{\beta}}_{old}), \tag{3.29}$$

where $\mathbf{c}_{t,k}=\mathbf{y}_t-\sum_{j=1}^{p}\hat{\mathbf{A}}_{j,k}\mathbf{y}_{t-j}-\mathbf{X}_t\hat{\boldsymbol{\alpha}}$ is a $d\times 1$ vector, $\mathbf{M}_{t,k}=\sum_{j=1}^{p}\hat{\mathbf{B}}_{j,k}\mathbf{y}_{t-j}(\mathbf{y}_{t-\ell}-\mathbf{y}_k)^\top\in\mathbb{R}^{d\times d}$, $\boldsymbol{\beta}_{old}$ is the estimator of $\boldsymbol{\beta}$ we used in Step 1 and $\boldsymbol{\beta}_{new}$ is the parameter we need to estimate here. Then we have the solution by :

$$\hat{\boldsymbol{\beta}}_{new}=\left(\sum_{k=1}^{n-l}\sum_{t=p+1}^{n}\mathbf{M}_{t,k}^\top K_h((\mathbf{y}_{t-\ell}-\mathbf{y}_k)^\top\hat{\boldsymbol{\beta}}_{old})\mathbf{M}_{t,k}\right)^{-1} \times\left(\sum_{k=1}^{n-l}\sum_{t=p+1}^{n}\mathbf{M}_{t,k}^\top K_h((\mathbf{y}_{t-\ell}-\mathbf{y}_k)^\top\hat{\boldsymbol{\beta}}_{old})\mathbf{c}_{t,k}\right). \tag{3.30}$$

At this point, in order to satisfy the identifiability conditions $\|\boldsymbol{\beta}\|=1$ and $\beta_1>0$, we define $\hat{\boldsymbol{\beta}}_{new}=-\hat{\boldsymbol{\beta}}_{new}/\|\hat{\boldsymbol{\beta}}_{new}\|$ if the first component of $\hat{\boldsymbol{\beta}}_{new}$ is negative, otherwise, let $\hat{\boldsymbol{\beta}}_{new}=-\hat{\boldsymbol{\beta}}_{new}/\|\hat{\boldsymbol{\beta}}_{new}\|$.

We then update the initial value $\boldsymbol{\beta}^0$ in Step 2 to $\boldsymbol{\beta}_{new}$, and continue Step 2 and Step 3 until $L(\boldsymbol{\beta},\boldsymbol{\alpha},\boldsymbol{\Theta})$ differs insignificantly.

3.3 Simulation study

In this section, we use a simulated example to illustrate the performance of our methodology, evaluate the accuracy of the proposed estimation approaches and compare the performance of two estimators.

We now consider a data-generating model with $q = 3, p = 2$ and sample size $n = 800$, which is defined by

$$\mathbf{y}_t = \mathbf{A}_1(z_{t-l})\mathbf{y}_{t-1} + \mathbf{A}_2(z_{t-l})\mathbf{y}_{t-2} + \mathbf{X}_t\boldsymbol{\alpha} + \boldsymbol{\epsilon}_t, \tag{3.31}$$

where $z_{t-l} = \mathbf{y}_{t-l}^{\top}\boldsymbol{\beta}$, $\mathbf{y}_t, t = 1, \cdots, 800$, is a $d-$ dimensional stationary time series with 2 lags, $\mathbf{X}_t$ is a $d \times 3$ matrix generated from d-dimensional Gaussian distribution, the $\boldsymbol{\epsilon}_t$ are independently generated from the normal distribution with $E(\boldsymbol{\epsilon}_t|\mathscr{F}_t) = \mathbf{0}_d$ and $\text{cov}(\boldsymbol{\epsilon}_t|\mathscr{F}_t) = \sigma^2\mathbf{I}_d$, where $\mathscr{F}_t = \{(Y_{l-1}, \mathbf{X}_l) : l \leq t\}$. In our example, we set:

$$l = 1, \quad \sigma = \frac{1}{3}, \quad d = 3, \quad \boldsymbol{\alpha} = (1,2,2), \quad \boldsymbol{\beta} = (0.6, 0.8, 0)^{\top},$$

$$\mathbf{A}_1(z_{t-l}) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0.9\exp(-z_{t-l}^2) & 0 \\ 0 & 0 & 0.25(\sin(\pi z_{t-l}) - 0.75) \end{pmatrix},$$

$$\mathbf{A}_2(z_{t-l}) = \begin{pmatrix} 0.33(\cos(\pi z_{t-l}) + 0.5) & 0 & 0 \\ 0.8\exp(-z_{t-l}^2) & 0.75\exp(-z_{t-l}^2) & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

where $\boldsymbol{\beta}$ is a $d \times 1$ vector and both $\mathbf{A}_1(z_{t-l})$ and $\mathbf{A}_2(z_{t-l})$ are $d \times d$ matrices. The first two predictors should be generated independently from a normal distribution, and then we generate $100 + n$ observations followed by the process (3.31). In order to improve the accuracy of estimation, we will discard the first 100 predictors and let $\mathbf{y}_t, t = 1, \cdots, n$ be the remaining predictors we have. For each case, we conduct the simulation over a total of 1000 replications. Meanwhile, in our simulation, we also employ the estimators from the model with known $\boldsymbol{\beta}$ as the benchmark.

Throughout this section, the kernel function we applied is Epanechnikov kernel $K(t) = 0.75(1 - t^2)_+$ and the bandwidth we set is 0.25 of the whole range, which is $0.25 \times (\max\{\mathbf{y}_1^\top \hat{\boldsymbol{\beta}}, \cdots, \mathbf{y}_t^\top \hat{\boldsymbol{\beta}}\} - \min\{\mathbf{y}_1^\top \hat{\boldsymbol{\beta}}, \cdots, \mathbf{y}_t^\top \hat{\boldsymbol{\beta}}\})$.

With the purpose of evaluating the performance of $\hat{\boldsymbol{\beta}}$ and $\hat{\mathbf{A}}_j, j = 1, \cdots, p$, the squared error metrics are used as follows:

$$\Delta(\hat{\mathbf{A}}) = \frac{\sum_{k=1}^{n-l} \sum_{j=1}^{p} \sum_{\imath=1}^{d} \sum_{\jmath=1}^{d} (\hat{a}_{j\imath\jmath k} - a_{j\imath\jmath k})^2}{\sum_{k=1}^{n-l} \sum_{j=1}^{p} \sum_{\imath=1}^{d} \sum_{\jmath=1}^{d} a_{j\imath\jmath k}^2}, \quad \Delta(\hat{\boldsymbol{\beta}}) = \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|^2, \tag{3.32}$$

The expectation and standard deviation of $\Delta(\hat{\mathbf{A}}_j)$ and $\Delta(\hat{\boldsymbol{\beta}})$ can be approximated by averaging over the 1000 replications using

$$\mathbb{E}(\Delta(\hat{\mathbf{A}})) \approx \frac{1}{1000} \sum_{L=1}^{1000} \Delta_L(\hat{\mathbf{A}}), \quad \mathbb{E}(\Delta(\hat{\boldsymbol{\beta}})) \approx \frac{1}{1000} \sum_{L=1}^{1000} \Delta_L(\hat{\boldsymbol{\beta}}),$$

$$SD(\Delta(\hat{\mathbf{A}})) \approx (\frac{1}{1000}\sum_{L=1}^{1000}\left\{\Delta_L(\hat{\mathbf{A}}) - \mathbb{E}(\Delta(\hat{\mathbf{A}}))\right\}^2)^{1/2},$$

$$SD(\Delta(\hat{\boldsymbol{\beta}})) \approx (\frac{1}{1000}\sum_{L=1}^{1000}\left\{\Delta_L(\hat{\boldsymbol{\beta}}) - \mathbb{E}(\Delta(\hat{\boldsymbol{\beta}}))\right\}^2)^{1/2},$$

where $\Delta_L(\cdot)$ denotes the square error metric of the Lth simulated dataset. One crucial thing to note is that $\mathbb{E}(\Delta(\hat{\boldsymbol{\beta}}))$ is the well known metric mean square error (MSE) in terms of the index parameters, and we can also call the metric $\mathbb{E}(\Delta(\hat{\mathbf{A}}))$ relative mean integrated squared error (RMISE) of the estimators of the unknown varying auto-regression coefficients.

The simulated results of the estimation accuracy are reported in Table 3.1.

Table 3.1: Comparison of estimates

	$\mathbb{E}(\Delta(\hat{\boldsymbol{\beta}}))$	$SD(\Delta(\hat{\boldsymbol{\beta}}))$	$\mathbb{E}(\Delta(\hat{\mathbf{A}}))$	$SD(\Delta(\hat{\mathbf{A}}))$
Method I	0.002	0.028	0.663	0.231
Method II	0.002	0.010	0.631	0.201
Method I (true $\boldsymbol{\beta}$)	0.000	0.000	0.689	0.242
Method II (true $\boldsymbol{\beta}$)	0.000	0.000	0.661	0.270

"Method I (True $\boldsymbol{\beta}$) " presents the one step estimation of $\mathbf{A}$ with the given true $\boldsymbol{\beta}$ via method I, " Method II (True $\boldsymbol{\beta}$)" presents the one step estimation of $\mathbf{A}$ with the given true $\boldsymbol{\beta}$ via method II.

As it can be seen from Table 3.1, Both of these two methods can yield reasonably good estimators with small corresponding estimated errors. We would also remark that the performance of the corresponding estimators in the model

with unknown $\boldsymbol{\beta}$ is similar to the estimates from the model with known $\boldsymbol{\beta}$, which greatly corroborates the effectiveness and accuracy of the proposed approach.

CHAPTER 4

ALGORITHM

In the sparse single-index vector autoregressive models (SSI-VARM), when the dimension of the covariates is fixed, the resulting nonparametric estimators can be obtained by local linear smoothing as we discussed in Chapter 3. However, if the model is of a higher dimension and allows sparsity, direct use of nonparametric regression may overestimate the model's complexity and lead to an over-fitted result. To sort out this issue, we will introduce a locally weighted group selection method by adding the SCAD penalty to the iterative approaches in Chapter 3. The penalised least squares can help select the importance of the model and shrink the insignificance to 0. Then, in our implementation, a unified solver for SCAD -type penalised least squares will be employed to consistently select the model, and hence to work out the resulting

estimators.

In this chapter, we shall focus on the model selection in SSIVARM, which is also the main subject of our thesis. Concretely, our model selection includes: (i) variable selection, which is equivalent to searching the null coefficients; (ii) specification of the constant coefficients, namely, detection of the coefficients with zero derivatives; (iii) identification of the index, which is realised by identifying zero-elements of the vector of index parameters $\boldsymbol{\beta}$.

In Section 4.1, we will concisely recall the SSIVARM. In Section 4.2, we will elaborate the methodology of the computational algorithm for selecting and estimating the SSIVARM simultaneously.

4.1 Model specification

Let $\mathbf{y}_t$, $t = 1, \cdots, n$, be a d-dimensional stationary time series, and $\mathbf{X}_t$, $t = 1, \cdots, n$, be i.i.d.. $\mathbf{X}_t$ is a $d \times q$ matrix. The sparse single-index vector autoregressive model is defined as

$$\mathbf{y}_t = \sum_{j=1}^{p} \mathbf{A}_j(\mathbf{y}_{t-\ell}^\top \boldsymbol{\beta})\mathbf{y}_{t-j} + \mathbf{X}_t \boldsymbol{\alpha} + \boldsymbol{\epsilon}_t, \tag{4.1}$$

where $1 \le \ell \le p$, the first column of $\mathbf{A}_\ell(\cdot)$ is $\mathbf{0}_d$, $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_d)^\top$ and satisfies

$$\|\boldsymbol{\beta}\| = 1, \quad \beta_1 > 0, \tag{4.2}$$

$\boldsymbol{\epsilon}_t$, $t = 1, \cdots, n$, are i.i.d. with

$$E(\boldsymbol{\epsilon}_t|\mathscr{F}_t) = \mathbf{0}_d \quad \text{cov}(\boldsymbol{\epsilon}_t|\mathscr{F}_t) = \sigma^2\mathbf{I}_d,$$

where $\mathscr{F}_t = \{(Y_{l-1}, \mathbf{X}_l) : l \leq t\}$. We also assume that $\boldsymbol{\beta}$ and $\mathbf{A}_j(\cdot)$, $j = 1, \cdots, p$, are sparse. The unknown parameters $\boldsymbol{\beta}$, $\mathbf{A}_j(\cdot)$, $\boldsymbol{\alpha}$ and σ^2 shall be estimated.

In this thesis, we focus on selecting the sparse model in finite dimension, but some of proposed approaches can also be introduced to divergent or high dimensionality, which will be left to discuss in the future works.

4.2 Methodology

In a similar way to Section 3.2, an iterative algorithm will be applied: firstly we need to obtain the estimators of the traditional varying coefficient model with known index $\mathbf{y}_t^\top\boldsymbol{\beta}$ by choosing an initial value $\hat{\boldsymbol{\beta}}$, the second step is to estimate $\boldsymbol{\beta}$ with the gotten estimators of $\boldsymbol{\Theta}$ and $\boldsymbol{\alpha}$, finally we continue these two steps until $L(\boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\Theta})$ differs insignificantly. However, apart from the estimation, the model selection work will be added after every estimation because the model is sparse and overparameterized. After the model estimation and selection, the most ideal estimators for us are that the significant variables are kept at the same value and the insignificant ones will be shrunk to 0. Hence, the objective of this section

is to introduce the model selection procedure of each method after the estimation.

Another important topic in this section is to determine a penalty function to consistently select the model and to find a proper algorithm to solve penalised least squares. We have introduced three penalty functions in Chapter 2, hard thresholding penalty, LASSO penalty and SCAD penalty. Hard thresholding penalty cannot meet the property Continuity and LASSO is biased, therefore, the penalty function SCAD which result in an estimator with three properties : unbiasedness, sparsity and continuity is the final choice for us in this section. This continuous differentiable penalty function is defined by

$$p'_\lambda(\beta) = \lambda\left\{I(\beta \le \lambda) + \frac{(a\lambda - \beta)_+}{(a-1)\lambda} I(\beta > \lambda)\right\} \quad \text{for some } a > 2 \text{ and } \beta > 0, \tag{4.3}$$

Hence we can have the penalised least square problem like

$$L(\boldsymbol{\beta}|\boldsymbol{\alpha}, \boldsymbol{\Theta}) + n\sum_{j=1}^{d} p_\lambda(|\beta_j|). \tag{4.4}$$

We meet two challenges here, one is the SCAD function is singular at the origin and they do not have continuous second derivatives, the other challenge is the computational cost is too heavy in the traditional optimising methods. However, this problem can be solved by the local quadratic approximation

(LQA) algorithm introduced by Fan and Li (2001). In this algorithm, we set the initial $\boldsymbol{\beta}_0$ which is close to the minimise of (4.4). If $\beta_{0,j}$ is very close to 0, then let $\hat{\beta}_j = 0$. Otherwise they can be locally approximated by a quadratic function as

$$[p_\lambda(|\beta_j|)]' = p'_\lambda(|\beta_j|)\text{sgn}(\beta_j) \approx \{p'_\lambda(|\beta_{0,j}|)/|\beta_{0,j}|\}\beta_j,$$

when $\beta_j \neq 0$. It means,

$$p_\lambda(|\beta_j|) \approx p_\lambda(|\beta_{0,j}|) + \frac{1}{2}\{p'_\lambda(|\beta_{0,j}|)/|\beta_{0,j}|\}(\beta_j^2 - \beta_{0,j}^2). \quad \text{for} \beta_j \approx \beta_{0,j} \qquad (4.5)$$

With this method, the penalised least squares problem becomes a convex quadratic optimisation problem which can be solved by obtaining the closed form solution. However, the shortage of this approximation is that the value 0 is an absorbing state, in the other words, once a coefficient is set to 0, it remains 0 in subsequent iterations.

It is worth noting that, in Section 4.2.1 and Section 4.2.2, we shall follow the idea of local quadratic approximation, proposed by Fan and Li (2001) to solve the weighted penalised least squares. As preliminary estimates are required to initiate this iterative algorithm, we will employ similar methods in Section 3.2.1 and Section 3.2.2 respectively to obtain the preliminary estimates from ordinary least squares, and thereby, according to the algorithm of local quadratic approximation,

the preliminary estimates will be used to get the penalised estimators.

4.2.1 Matrix transformation method

In this section, we will introduce the estimation procedure for our model (4.1) including model selection, based on the estimator given in Section 3.2.1 and penalised least squares. This estimation approach is named as the Matrix transformation method because we use some matrix transformations in obtaining the closed form solution of the least square problem in the estimation procedure.

From section 3.2.1, we get the local discrepancy loss function:

$$L(\boldsymbol{\beta},\boldsymbol{\alpha},\boldsymbol{\Theta}) = \sum_{k=1}^{n-l}\sum_{t=p+1}^{n} \Big\| \mathbf{y}_t - \sum_{j=1}^{p} \left\{ \mathbf{A}_{j,k} + \mathbf{B}_{j,k} \left[(\mathbf{y}_{t-\ell} - \mathbf{y}_k)^\top \boldsymbol{\beta} \right] \right\} \mathbf{y}_{t-j} - \mathbf{X}_t \boldsymbol{\alpha} \Big\|^2 K_h((\mathbf{y}_{t-\ell} - \mathbf{y}_k)^\top \boldsymbol{\beta}), \tag{4.6}$$

Because $\mathbf{A}_j(\cdot), j = 1, \cdots, p$, are functional, to deal with the sparsity in $\mathbf{A}_j(\cdot)$, the penalty has to be imposed on the function values of $\mathbf{A}_j(\cdot)$ at all $\mathbf{y}_k^\top, k = 1, \cdots, n-l$. So the form of the

penalised least squares is constructed as

$$\begin{aligned}\mathscr{L}(\boldsymbol{\beta},\boldsymbol{\alpha},\boldsymbol{\Theta}) =& L(\boldsymbol{\beta},\boldsymbol{\alpha},\boldsymbol{\Theta}) + \sum_{l=1}^{d} P_l(|\beta_l|) \\ &+ \sum_{j=1}^{p}\sum_{\imath=1}^{d}\sum_{\jmath=1}^{d}\left\{P_{j\imath\jmath}(\|\mathbf{a}_{j\imath\jmath}\|) + P_{j\imath\jmath}(\|\mathbf{b}_{j\imath\jmath}\|)\right\} \qquad (4.7)\end{aligned}$$

where

$$P_l(\cdot) = \mathscr{P}_{\lambda_l}(\cdot), \quad P_{j\imath\jmath}(\cdot) = \mathscr{P}_{\lambda_{j\imath\jmath}}(\cdot),$$

$$\mathbf{a}_{j\imath\jmath} = (a_{j\imath\jmath 1}, \cdots, a_{j\imath\jmath(n-\ell)})^\top, \quad \mathbf{b}_{j\imath\jmath} = (b_{j\imath\jmath 1}, \cdots, b_{j\imath\jmath(n-\ell)})^\top$$

λ_l and $\lambda_{j\imath\jmath}$ are tuning parameters. $\mathscr{P}_\lambda(\cdot)$ is a penalty function with tuning parameter λ. In this section, the penalty function is taken to be the SCAD function proposed by Fan and Li (2001). $a_{j\imath\jmath k}$ and $b_{j\imath\jmath k}$ are the $(\imath,\ \jmath)$th element of $\mathbf{A}_{j,k}$ and $\mathbf{B}_{j,k}$, respectively.

The estimator $\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\Theta}}$, and $\hat{\boldsymbol{\alpha}}$ can be obtained by solving

$$(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\Theta}}) = \underset{\boldsymbol{\beta},\boldsymbol{\alpha},\boldsymbol{\Theta}}{\arg\min}\, \mathscr{L}(\boldsymbol{\beta},\boldsymbol{\alpha},\boldsymbol{\Theta}) \qquad (4.8)$$

subject to the constraints $\|\boldsymbol{\beta}\| = 1$ and $\beta_1 > 0$. Following the idea of quadratic approximation, see Fan and Li (2001), by properly defining the threshold δ, we let $\|\hat{\mathbf{a}}_{j\imath\jmath}\|$ be 0 when $\|\hat{\mathbf{a}}_{j\imath\jmath}\| < \delta$. Once $\|\hat{\mathbf{a}}_{j\imath\jmath}\| = 0$, we use 0 to estimate the $(\imath,\ \jmath)$th element of $\mathbf{A}_j(\cdot)$. When $\|\hat{\mathbf{a}}_{j\imath\jmath}\| \neq 0$, we use $\hat{\mathbf{a}}_{j\imath\jmath}$ to estimate $(a_{j\imath\jmath}(\mathbf{y}_1^\top\boldsymbol{\beta}), \cdots, a_{j\imath\jmath}(\mathbf{y}_{n-\ell}^\top\boldsymbol{\beta}))^\top$. As in the last chapter, the global minimum of the local discrepancy function cannot be found

analytically and therefore an iterative procedure is proposed for implementation purposes as follows:

Step 1. Choose initial value $\boldsymbol{\beta}^0$. Before the iterative procedure, an initial value of $\boldsymbol{\beta}$ should be specified, which is denoted by $\boldsymbol{\beta}^0$. The initial value $\boldsymbol{\beta}^0$ needs to satisfy the constraints $\boldsymbol{\beta}_1^0 > 0$ and $\|\boldsymbol{\beta}^0\| = 1$, where $\boldsymbol{\beta}_1^0$ is the first component of $\boldsymbol{\beta}^0$.

Step 2. Estimate $\boldsymbol{\Theta}, \boldsymbol{\alpha}$ and select $\boldsymbol{\Theta}$ with $\boldsymbol{\beta}$ taking the initial estimated value. Set $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}^0$, we now estimate $\boldsymbol{\Theta}, \boldsymbol{\alpha}$ by solving

$$\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\Theta}, \boldsymbol{\alpha}}{\arg\min}\ \mathscr{L}(, \boldsymbol{\Theta}, \boldsymbol{\alpha} | \hat{\boldsymbol{\beta}})$$

in the other words, we need to minimise the form of penalised least squares

$$L(\boldsymbol{\alpha}, \boldsymbol{\Theta} | \hat{\boldsymbol{\beta}}) + \sum_{j=1}^{p} \sum_{\imath=1}^{d} \sum_{\jmath=1}^{d} \left\{ P_{j\imath\jmath}(\|\mathbf{a}_{j\imath\jmath}\|) + P_{j\imath\jmath}(\|\mathbf{b}_{j\imath\jmath}\|) \right\} \tag{4.9}$$

In order to deal with the minimisation problem (4.9), a unified algorithm of local quadratic approximation proposed by Fan and Li (2001) is applied here. For the penalised least square problem, it is straightforward to find that we can locally approximate the first term in (4.9) by a quadratic function. As for the second term, SCAD penalty functions, they can also be locally approximated by a quadratic function, which will be discussed as follows. Therefore, the expression (4.9) will

be formed into a quadratic function where we can obtain the minimiser as an explicit solution.

For this approximation of the SCAD penalty function, as mentioned previously, we have to set the initial value, which is close to the minimiser of (4.9). In this section, we will use the minimiser of $L(\hat{\boldsymbol{\beta}}, \boldsymbol{\Theta}, \boldsymbol{\alpha})$ as the initial value of $\mathbf{a}_{j\imath\jmath}$ and $\mathbf{b}_{j\imath\jmath}$. Let $\hat{\boldsymbol{\Theta}}^{\dagger} = (\hat{\mathbf{a}}_{111}, \cdots, \hat{\mathbf{a}}_{1dd}, \hat{\mathbf{b}}_{111}, \cdots, \hat{\mathbf{b}}_{1dd}, \cdots, \hat{\mathbf{a}}_{p11}, \cdots, \hat{\mathbf{a}}_{pdd}, \hat{\mathbf{b}}_{p11} \cdots, \hat{\mathbf{b}}_{pdd})$ be the current "minimiser" of $L(\hat{\boldsymbol{\beta}}, \boldsymbol{\Theta}, \boldsymbol{\alpha})$, the penalty functions $P_{j\imath\jmath}(\|\mathbf{a}_{j\imath\jmath}\|)$ and $P_{j\imath\jmath}(\|\mathbf{b}_{j\imath\jmath}\|)$ will be applied to the approximation:

$$P_{j\imath\jmath}(\|\mathbf{a}_{j\imath\jmath}\|) \approx P_{j\imath\jmath}(\|\hat{\mathbf{a}}_{j\imath\jmath}\|) + \frac{P'_{j\imath\jmath}(\|\hat{\mathbf{a}}_{j\imath\jmath}\|)}{2\|\hat{\mathbf{a}}_{j\imath\jmath}\|}(\mathbf{a}_{j\imath\jmath}^{\top}\mathbf{a}_{j\imath\jmath} - \hat{\mathbf{a}}_{j\imath\jmath}^{\top}\hat{\mathbf{a}}_{j\imath\jmath}),$$

$$P_{j\imath\jmath}(\|\mathbf{b}_{j\imath\jmath}\|) \approx P_{j\imath\jmath}(\|\hat{\mathbf{b}}_{j\imath\jmath}\|) + \frac{P'_{j\imath\jmath}(\|\hat{\mathbf{b}}_{j\imath\jmath}\|)}{2\|\hat{\mathbf{b}}_{j\imath\jmath}\|}(\mathbf{b}_{j\imath\jmath}^{\top}\mathbf{b}_{j\imath\jmath} - \hat{\mathbf{b}}_{j\imath\jmath}^{\top}\hat{\mathbf{b}}_{j\imath\jmath}),$$

Then the penalised least square problem (4.9) can be locally approximated (except for a constant term) by:

$$L(\boldsymbol{\alpha}, \boldsymbol{\Theta}|\hat{\boldsymbol{\beta}}) + \sum_{j=1}^{p}\sum_{\imath=1}^{d}\sum_{\jmath=1}^{d}\left\{\frac{P'_{j\imath\jmath}(\|\hat{\mathbf{a}}_{j\imath\jmath}\|)}{2\|\hat{\mathbf{a}}_{j\imath\jmath}\|}\mathbf{a}_{j\imath\jmath}^{\top}\mathbf{a}_{j\imath\jmath} + \frac{P'_{j\imath\jmath}(\|\hat{\mathbf{b}}_{j\imath\jmath}\|)}{2\|\hat{\mathbf{b}}_{j\imath\jmath}\|}\mathbf{b}_{j\imath\jmath}^{\top}\mathbf{b}_{j\imath\jmath}\right\} \tag{4.10}$$

which is equivalent to

$$L(\boldsymbol{\alpha}, \boldsymbol{\Theta}|\hat{\boldsymbol{\beta}}) + \sum_{j=1}^{p}\sum_{\imath=1}^{d}\sum_{\jmath=1}^{d}\left\{\sum_{k=1}^{n-l}\frac{P'_{j\imath\jmath}(\|\hat{\mathbf{a}}_{j\imath\jmath}\|)}{2\|\hat{\mathbf{a}}_{j\imath\jmath}\|}\alpha_{j\imath\jmath k}^{2} + \sum_{k=1}^{n-l}\frac{P'_{j\imath\jmath}(\|\hat{\mathbf{b}}_{j\imath\jmath}\|)}{2\|\hat{\mathbf{b}}_{j\imath\jmath}\|}b_{j\imath\jmath k}^{2}\right\} \tag{4.11}$$

Therefore, for the sake of solving this minimisation problem (4.11), we need to obtain $\hat{\boldsymbol{\Theta}}^{\dagger}$, the minimiser of $L(\hat{\boldsymbol{\beta}}, \boldsymbol{\Theta}, \boldsymbol{\alpha})$ firstly.

For brevity purposes, we use the notations

$$\mathbf{B}_{j,k} = \dot{\mathbf{A}}_{j,k}$$

$$\boldsymbol{\theta}_k = (\mathbf{A}_{1,k}, \mathbf{B}_{1,k}, \cdots, \mathbf{A}_{p,k}, \mathbf{B}_{p,k}),$$

$$\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_{n-l}),$$

where we denote $\mathbf{A}_j(\mathbf{y}_k^{\top}\boldsymbol{\beta})$ and $\dot{\mathbf{A}}_j(\mathbf{y}_k^{\top}\boldsymbol{\beta})$ by $\mathbf{A}_{j,k}$ and $\dot{\mathbf{A}}_{j,k}$ respectively.

For each k, we choose the estimator of $\boldsymbol{\theta}_k, \boldsymbol{\alpha}$ by minimising

$$\sum_{t=p+1}^{n} \left\| \mathbf{y}_t - \sum_{j=1}^{p} \left\{ \mathbf{A}_{j,k} + \mathbf{B}_{j,k} \left[(\mathbf{y}_{t-\ell} - \mathbf{y}_k)^{\top} \hat{\boldsymbol{\beta}} \right] \right\} \mathbf{y}_{t-j} - \mathbf{X}_t \boldsymbol{\alpha} \right\|^2 \times K_h((\mathbf{y}_{t-\ell} - \mathbf{y}_k)^{\top} \hat{\boldsymbol{\beta}}), \tag{4.12}$$

We can write it (4.12) in matrix notation as

$$(\hat{\boldsymbol{\theta}}_k, \hat{\boldsymbol{\alpha}}) = \underset{\boldsymbol{\theta}_k, \boldsymbol{\alpha}}{\arg\min} \sum_{t=p+1}^{n} \left\| \mathbf{y}_t - \mathbf{Q}_{t,k}^{\top} \boldsymbol{\theta}_k - \mathbf{X}_t \boldsymbol{\alpha} \right\|^2 K_h((\mathbf{y}_{t-\ell} - \mathbf{y}_k)^{\top} \hat{\boldsymbol{\beta}}), \tag{4.13}$$

where $\mathbf{Q}_{t,k} = (\mathbf{y}_{t-1}^{\top}, (\mathbf{y}_{t-\ell}^{\top}\hat{\boldsymbol{\beta}} - z_k)\mathbf{y}_{t-1}^{\top}, \cdots, \mathbf{y}_{t-p}^{\top}, (\mathbf{y}_{t-\ell}^{\top}\hat{\boldsymbol{\beta}} - z_k)\mathbf{y}_{t-p}^{\top})$, in which $z_k = \mathbf{y}_k^{\top}\hat{\boldsymbol{\beta}}$; $\mathbf{y}_i = (y_1, \cdots, y_d)^{\top}, i = 1, \cdots, n$.

One thing needs to be emphasized is that we have to compose the identifiability conditions so that the model must satisfy the conditions $\|\boldsymbol{\beta}\| = 1, \beta_1 > 0$ and the first column of $\mathbf{A}_{\ell}(\cdot)$ is $\mathbf{0}_d$,

which can be practically realised that instead of estimating the first column of $\mathbf{A}_\ell(\cdot)$ in the estimation procedure, we let the elements of the first column of $\mathbf{A}_\ell(\cdot)$ equal 0 directly, which will lead to a reduction of dimension of $\boldsymbol{\theta}_k$ from $d \times 2pd$ to $d \times (2pd-2)$. Correspondingly, both of the first element of $\mathbf{y}_{t-\ell}$ and $(\mathbf{y}_{t-\ell}^\top \hat{\boldsymbol{\beta}} - z_k)\mathbf{y}_{t-\ell}$ of $\mathbf{Q}_{t,k}$ need to be eliminated in the estimation procedure, therefore, the dimension of $\mathbf{Q}_{t,k}$ reduces from $1 \times 2pd$ to $1 \times (2pd-2)$.

In order to solve the weighted least squares problem to calculate the explicit solution , we need to rewrite the equation (4.13) as:

$$\hat{\boldsymbol{\gamma}}_k = \arg\min_{\boldsymbol{\gamma}_k} \sum_{t=p+1}^{n} \left\| \mathbf{y}_t - \mathbb{X}_{t,k}\boldsymbol{\gamma}_k \right\|^2 K_h((\mathbf{y}_{t-\ell} - \mathbf{y}_k)^\top \hat{\boldsymbol{\beta}}), \qquad (4.14)$$

where $\boldsymbol{\gamma}_k = [\boldsymbol{\theta}_k, \boldsymbol{\alpha}]$, $\mathbb{X}_{t,k} = [\mathbf{Q}_{t,k}, \mathbf{X}_t]$, $\boldsymbol{\Gamma} = (\boldsymbol{\gamma}_1, \cdots, \boldsymbol{\gamma}_{n-l})$. However, the augmented matrix $\boldsymbol{\gamma}_k = [\boldsymbol{\theta}_k, \boldsymbol{\alpha}]$ cannot be constructed by $\boldsymbol{\theta}_k \in \mathbb{R}^{d\times(2pd-2)}, \boldsymbol{\alpha} \in \mathbb{R}^{q\times 1}$ directly, similarly, $\mathbb{X}_{t,k}$ meets the same problem that $\mathbf{Q}_{t,k} \in \mathbb{R}^{1\times(2pd-2)}$ and $\mathbf{X}_t \in \mathbb{R}^{d\times q}$ cannot construct $\mathbb{X}_{t,k}$ directly. Therefore, we need to do some matrix transformations to obtain the proper augmented matrices $\boldsymbol{\gamma}_k$ and $\mathbb{X}_{t,k}$.

In our observation, the matrix $\boldsymbol{\theta}_k = (\mathbf{A}_{1,k}, \mathbf{B}_{1,k}, \cdots, \mathbf{A}_{p,k}, \mathbf{B}_{p,k})$ is transferred to a vector $\boldsymbol{\theta}_k^\dagger = (a_{111k}, \cdots, a_{1ddk}, b_{111k}, \cdots, b_{1ddk}, \cdots, a_{l12k}, \cdots, a_{pddk}, b_{l12k}, \cdots, b_{pddk}, \cdots, a_{p11k}, \cdots, a_{pddk}, b_{p11k}, \cdots,$

$b_{pddk})^\top$, where $\boldsymbol{\theta}_k^\dagger \in \mathbb{R}^{(2pdd-2d)\times 1}$. Hence $\boldsymbol{\gamma}_k = [\boldsymbol{\theta}_k^\dagger, \boldsymbol{\alpha}]^\top$ is constructed to a $(2pdd-2d+q)\times 1$ matrix.

The same method is used in $\mathbb{X}_{t,k} = [\mathbf{Q}_{t,k}, \mathbf{X}_t]$. Because the $\mathbf{X}_t$ is an $d \times q$ matrix, we design a matrix based on $\mathbf{Q}_{t,k}$ which has the same number of rows as $\mathbf{X}_t$:

$$\mathbf{Q}_{t,k}^\dagger = \begin{pmatrix} \mathbf{Q}_{t,k} & 0 & \cdots & 0 \\ 0 & \mathbf{Q}_{t,k} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \mathbf{Q}_{t,k} \end{pmatrix},$$

where $\mathbf{Q}_{t,k}^\dagger \in \mathbb{R}^{d\times(2pdd-2d)}$. With this new $\mathbf{Q}_{t,k}^\dagger$, we have an augmented matrix $\mathbb{X}_{t,k} \in \mathbb{R}^{d\times(2pdd-2d+q)}$. Through the least squares theory, the solution is given by

$$\hat{\boldsymbol{\gamma}}_k = (\sum_{t=p+1}^{n} \mathbb{X}_{t,k}^\top K_h(\mathbf{y}_{t-\ell} - \mathbf{y}_k)\mathbb{X}_{t,k})^{-1}(\sum_{t=p+1}^{n} \mathbb{X}_{t,k}^\top K_h(\mathbf{y}_{t-\ell} - \mathbf{y}_k)\mathbf{y}_t) \tag{4.15}$$

and hence we obtain $\hat{\boldsymbol{\Gamma}} = (\hat{\boldsymbol{\gamma}}_1, \cdots, \hat{\boldsymbol{\gamma}}_{n-l})$.

After working out $\hat{\boldsymbol{\Theta}}^\dagger$ and $\hat{\boldsymbol{\alpha}}$, let us go back to the penalised least square problem, the expression (4.11) can be written in matrix notation as

$$\sum_{k=1}^{n-l}\sum_{t=p+1}^{n} \left\| \mathbf{y}_t - \mathbf{X}_t\boldsymbol{\alpha} - \mathbf{Q}_{t,k}^\dagger\boldsymbol{\theta}_k^\dagger \right\|^2 K_h(\mathbf{y}_{t-\ell} - \mathbf{y}_k) + \frac{1}{2}\sum_{k=1}^{n-l} \boldsymbol{\theta}_k^\dagger \Sigma_{\boldsymbol{\Theta}} (\boldsymbol{\theta}_k^\dagger)^\top, \tag{4.16}$$

where $\Sigma_{\boldsymbol{\theta}_k} = \text{diag}\Big\{P'_{111}(\|\hat{\mathbf{a}}_{111}\|)/\|\hat{\mathbf{a}}_{111}\|, \cdots, P'_{1dd}(\|\hat{\mathbf{a}}_{1dd}\|)/\|\hat{\mathbf{a}}_{1dd}\|, P'_{111}(\|\hat{\mathbf{b}}_{111}\|)/\|\hat{\mathbf{b}}_{111}\|, \cdots, P'_{1dd}(\|\hat{\mathbf{b}}_{1dd}\|)/\|\hat{\mathbf{b}}_{1dd}\|, \cdots, P'_{p11}(\|\hat{\mathbf{a}}_{p11}\|)/\|\hat{\mathbf{a}}_{p11}\|,$

$\cdots, P'_{pdd}(\|\hat{\mathbf{a}}_{pdd}\|)/\|\hat{\mathbf{a}}_{pdd}\|, P'_{p11}(\|\hat{\mathbf{b}}_{p11}\|)/\|\hat{\mathbf{b}}_{p11}\|, \cdots, P'_{pdd}(\|\hat{\mathbf{b}}_{pdd}\|)/\|\hat{\mathbf{b}}_{pdd}\|\}$.

The continuous differentiable penalty function is defined previously in (4.3), and we can obtain the solution by penalised least square approach:

$$\check{\boldsymbol{\theta}}_k^{\dagger} = \Big(\sum_{t=p+1}^{n} (\mathbf{Q}_{t,k}^{\dagger})^{\top} K_h(\mathbf{y}_{t-\ell} - \mathbf{y}_k) \mathbf{Q}_{t,k}^{\dagger} + \frac{n-p}{2} \Sigma_{\boldsymbol{\theta}_k} \Big)^{-1} \times \Big(\sum_{t=p+1}^{n} (\mathbf{Q}_{t,k}^{\dagger})^{\top} K_h(\mathbf{y}_{t-\ell} - \mathbf{y}_k)(\mathbf{y}_t - \mathbf{X}_t \boldsymbol{\alpha}) \Big). \tag{4.17}$$

Hence, we have $\check{\boldsymbol{\Theta}}^{\dagger} = (\check{\boldsymbol{\theta}}_1^{\dagger}, \cdots, \check{\boldsymbol{\theta}}_{n-l}^{\dagger})$.

Step 3. Estimate and select $\boldsymbol{\beta}$ given $\boldsymbol{\Theta}$,$\boldsymbol{\alpha}$. Using the estimators $\check{\boldsymbol{\Theta}}^{\dagger}$ and $\hat{\boldsymbol{\alpha}}$ from Step two, we would like to find the estimator of $\boldsymbol{\beta}$:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \mathscr{L}(\boldsymbol{\beta} | \check{\boldsymbol{\Theta}}^{\dagger}, \hat{\boldsymbol{\alpha}}),$$

in other words, we need to minimise the form of penalised least squares

$$L(\boldsymbol{\beta} | \check{\boldsymbol{\Theta}}^{\dagger}, \hat{\boldsymbol{\alpha}}) + \sum_{l=1}^{d} P_l(|\beta_l|). \tag{4.18}$$

In order to deal with the penalised least square problem (4.18), the same algorithm as the last step will be applied here. We will use the local quadratic approximation to reduce the minimisation problem to a quadratic minimisation problem.

Similarly, the minimise of $L(\boldsymbol{\beta}, \check{\boldsymbol{\Theta}}^{\dagger}, \hat{\boldsymbol{\alpha}})$ will be set as the initial value of $\boldsymbol{\beta}$.

Let $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \cdots, \hat{\beta}_d)^{\top}$ be the current 'minimiser' of $L(\boldsymbol{\beta}, \check{\boldsymbol{\Theta}}^{\dagger}, \hat{\boldsymbol{\alpha}})$, apply the approximation

$$P_l(|\beta_l|) \approx P_l(|\hat{\beta}_l|) + \frac{P_l'(|\hat{\beta}_l|)}{2|\hat{\beta}_l|}(\beta_l^2 - \hat{\beta}_l^2),$$

to the $P_l(|\beta_l|)$ in (4.18), then the penalised least squares problem (4.18) can be locally approximated (except for a constant term) by:

$$L(\boldsymbol{\beta}|\check{\boldsymbol{\Theta}}^{\dagger}, \hat{\boldsymbol{\alpha}}) + \sum_{l=1}^{d} \frac{P_l'(|\hat{\beta}_l|)}{2|\hat{\beta}_l|}\beta_l^2 \tag{4.19}$$

Therefore, for the sake of solving this minimisation problem (4.19), we need to obtain $\hat{\boldsymbol{\beta}}$, the minimise of $L(\boldsymbol{\beta}, \check{\boldsymbol{\Theta}}^{\dagger}, \hat{\boldsymbol{\alpha}})$ firstly,

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} L(\boldsymbol{\beta}, \check{\boldsymbol{\Theta}}^{\dagger}, \hat{\boldsymbol{\alpha}})$$

which is equivalent to minimising

$$\sum_{k=1}^{n-l} \sum_{t=p+1}^{n} \left\| \mathbf{y}_t - \sum_{j=1}^{p} \left\{ \hat{\mathbf{A}}_{j,k} + \hat{\mathbf{B}}_{j,k} \left[(\mathbf{y}_{t-\ell} - \mathbf{y}_k)^{\top} \boldsymbol{\beta} \right] \right\} \mathbf{y}_{t-j} - \mathbf{X}_t \hat{\boldsymbol{\alpha}} \right\|^2 \times K_h((\mathbf{y}_{t-\ell} - \mathbf{y}_k)^{\top} \hat{\boldsymbol{\beta}}), \tag{4.20}$$

Two things need to be emphasised. The first one is $\hat{\mathbf{A}}_{j,k}, \hat{\mathbf{B}}_{j,k}$ and $\hat{\boldsymbol{\alpha}}$ come from $\hat{\boldsymbol{\Gamma}}$ because of the matrix transformation in the last step. Secondly, only $\boldsymbol{\beta}$ in the least squares part of the loss function is the parameter we need to estimate, the $\hat{\boldsymbol{\beta}}$

appears in the kernel function is the estimator of $\boldsymbol{\beta}$ we used in Step 1. For the sake of distinguishing the two $\boldsymbol{\beta}$s, we use $\boldsymbol{\beta}_{new}$ and $\boldsymbol{\beta}_{old}$ to rewrite the approximation:

$$\sum_{k=1}^{n-l}\sum_{t=p+1}^{n}\left\|\mathbf{y}_t-\sum_{j=1}^{p}\left\{\hat{\mathbf{A}}_{j,k}+\hat{\mathbf{B}}_{j,k}\left[(\mathbf{y}_{t-\ell}-\mathbf{y}_k)^\top\boldsymbol{\beta}_{new}\right]\right\}\mathbf{y}_{t-j}-\mathbf{X}_t\hat{\boldsymbol{\alpha}}\right\|^2 \times K_h((\mathbf{y}_{t-\ell}-\mathbf{y}_k)^\top\hat{\boldsymbol{\beta}}_{old}), \tag{4.21}$$

Let us expand the expression (4.21) as below:

$$\sum_{k=1}^{n-l}\sum_{t=p+1}^{n}\left\|\mathbf{y}_t-\sum_{j=1}^{p}\hat{\mathbf{A}}_{j,k}\mathbf{y}_{t-j}+\sum_{j=1}^{p}\hat{\mathbf{B}}_{j,k}\mathbf{y}_{t-j}(\mathbf{y}_{t-\ell}-\mathbf{y}_k)^\top\boldsymbol{\beta}_{new}-\mathbf{X}_t\hat{\boldsymbol{\alpha}}\right\|^2 \times K_h((\mathbf{y}_{t-\ell}-\mathbf{y}_k)^\top\hat{\boldsymbol{\beta}}_{old}), \tag{4.22}$$

In order to obtain the closed form solution, it is straightforward to rewrite the problem in matrix notation:

$$\hat{\boldsymbol{\beta}}_{new}=\underset{\boldsymbol{\beta}_{new}}{\arg\min}\sum_{k=1}^{n-l}\sum_{t=p+1}^{n}\left\|\mathbf{c}_{t,k}-\mathbf{M}_{t,k}\boldsymbol{\beta}_{new}\right\|^2 K_h((\mathbf{y}_{t-\ell}-\mathbf{y}_k)^\top\hat{\boldsymbol{\beta}}_{old}), \tag{4.23}$$

where:

$$\mathbf{c}_{t,k}=\mathbf{y}_t-\sum_{j=1}^{p}\hat{\mathbf{A}}_{j,k}\mathbf{y}_{t-j}-\mathbf{X}_t\hat{\boldsymbol{\alpha}},\quad \mathbf{M}_{t,k}=\sum_{j=1}^{p}\hat{\mathbf{B}}_{j,k}\mathbf{y}_{t-j}(\mathbf{y}_{t-\ell}-\mathbf{y}_k)^\top.$$

in which $\mathbf{c}_{t,k}$ is a $d\times 1$ matrix and the dimension of $\mathbf{M}_{t,k}$ is

$d \times d$. Then we have the solution by :

$$\begin{aligned}\hat{\boldsymbol{\beta}}_{new} &= \Big(\sum_{k=1}^{n-l}\sum_{t=p+1}^{n} \mathbf{M}_{t,k}^{\top} K_h((\mathbf{y}_{t-\ell}-\mathbf{y}_k)^{\top}\hat{\boldsymbol{\beta}}_{old})\mathbf{M}_{t,k}\Big)^{-1} \\ &\quad \times\Big(\sum_{k=1}^{n-l}\sum_{t=p+1}^{n} \mathbf{M}_{t,k}^{\top} K_h((\mathbf{y}_{t-\ell}-\mathbf{y}_k)^{\top}\hat{\boldsymbol{\beta}}_{old})\mathbf{c}_{t,k}\Big) \quad (4.24)\end{aligned}$$

After working out $\hat{\boldsymbol{\beta}}_{new}$, let us go back to the penalised least square problem, the expression (4.19) can be written in matrix notation as

$$\sum_{k=1}^{n-l}\sum_{t=p+1}^{n} \left\|\mathbf{c}_{t,k}-\mathbf{M}_{t,k}\boldsymbol{\beta}_{new}\right\|^2 K_h((\mathbf{y}_{t-\ell}-\mathbf{y}_k)^{\top}\hat{\boldsymbol{\beta}}_{old}) + \frac{1}{2}\boldsymbol{\beta}_{new}\Sigma_{\boldsymbol{\beta}}\boldsymbol{\beta}_{new} \quad (4.25)$$

where $\Sigma_{\boldsymbol{\beta}} = \text{diag}\Big\{P_1'(|\hat{\beta}_{new,1}|)/|\hat{\beta}_{new,1}|, \cdots, P_d'(|\hat{\beta}_{new,d}|)/|\hat{\beta}_{new,d}|\Big\}$.

The continuous differentiable penalty function is defined previously (4.3), we can obtain the solution by penalised least square approach:

$$\begin{aligned}\check{\boldsymbol{\beta}}_{new} &= \Big(\sum_{k=1}^{n-l}\sum_{t=p+1}^{n} \mathbf{M}_{t,k}^{\top} K_h((\mathbf{y}_{t-\ell}-\mathbf{y}_k)^{\top}\hat{\boldsymbol{\beta}}_{old})\mathbf{M}_{t,k} + \frac{n-p}{2}\Sigma_{\boldsymbol{\beta}}\Big)^{-1} \\ &\quad \times\Big(\sum_{k=1}^{n-l}\sum_{t=p+1}^{n} \mathbf{M}_{t,k}^{\top} K_h((\mathbf{y}_{t-\ell}-\mathbf{y}_k)^{\top}\hat{\boldsymbol{\beta}}_{old})\mathbf{c}_{t,k}\Big) \quad (4.26)\end{aligned}$$

At this point, in order to satisfy the identifiability conditions $\|\boldsymbol{\beta}\| = 1$ and $\beta_1 > 0$, we define $\check{\boldsymbol{\beta}}_{new} = -\check{\boldsymbol{\beta}}_{new}/\|\check{\boldsymbol{\beta}}_{new}\|$ if the first component of $\check{\boldsymbol{\beta}}_{new}$ is negative, otherwise, let $\check{\boldsymbol{\beta}}_{new} = \check{\boldsymbol{\beta}}_{new}/\|\check{\boldsymbol{\beta}}_{new}\|$.

4.2.2 Computationally Efficient Estimation Method

As discussed in chapter 3, when we work on the model in relative high dimension with the matrix transformation method, we will meet a challenge that the estimation procedure cannot be completed in an acceptable time. Therefore, we will propose a practically feasible approach named the computationally efficient estimation method to alleviate the computational burden with the model in a higher dimensional situation.

Let $\mathbf{c}_{m,j}(\cdot), m = 1, \cdots, d$ be the mth row of the matrix $\mathbf{A}_j(\cdot)$, $\mathbf{y}_t = (y_{t,1}, \cdots, y_{t,d})^\top$, $\mathbf{X}_t = (\mathbf{x}_{t,1}, \cdots, \mathbf{x}_{t,d})^\top$, where $\mathbf{x}_{t,m} \in \mathbb{R}^{1\times q}, m = 1, \cdots, d$. $\boldsymbol{\epsilon}_t = (\epsilon_{t,1}, \cdots, \epsilon_{t,d})$, where $\boldsymbol{\epsilon}_t$, $t = 1, \cdots, n$, are i.i.d. with

$$E(\boldsymbol{\epsilon}_t|\mathscr{F}_t) = \mathbf{0}_d, \quad \text{cov}(\boldsymbol{\epsilon}_t|\mathscr{F}_t) = \sigma^2\mathbf{I}_d,$$

where $\mathscr{F}_t = \{(Y_{l-1}, \mathbf{X}_l) : l \le t\}$.

In a similar way to Section 3.2.2, we approximate the following model

$$\begin{pmatrix} y_{t,1} \\ y_{t,2} \\ \vdots \\ y_{t,d} \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^p \mathbf{c}_{1,j}(\mathbf{y}_{t-\ell}^\top\boldsymbol{\beta})\mathbf{y}_{t-j} \\ \sum_{j=1}^p \mathbf{c}_{2,j}(\mathbf{y}_{t-\ell}^\top\boldsymbol{\beta})\mathbf{y}_{t-j} \\ \vdots \\ \sum_{j=1}^p \mathbf{c}_{d,j}(\mathbf{y}_{t-\ell}^\top\boldsymbol{\beta})\mathbf{y}_{t-j} \end{pmatrix} + \begin{pmatrix} \mathbf{x}_{t,1}\boldsymbol{\alpha} \\ \mathbf{x}_{t,2}\boldsymbol{\alpha} \\ \vdots \\ \mathbf{x}_{t,d}\boldsymbol{\alpha} \end{pmatrix} + \begin{pmatrix} \epsilon_{t,1} \\ \epsilon_{t,2} \\ \vdots \\ \epsilon_{t,d} \end{pmatrix}, \qquad (4.27)$$

by

$$\begin{pmatrix} y_{t,1} \\ y_{t,2} \\ \vdots \\ y_{t,d} \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^{p} \mathbf{c}_{1,j}(\mathbf{y}_{t-\ell}^{\top}\boldsymbol{\beta})\mathbf{y}_{t-j} \\ \sum_{j=1}^{p} \mathbf{c}_{2,j}(\mathbf{y}_{t-\ell}^{\top}\boldsymbol{\beta})\mathbf{y}_{t-j} \\ \vdots \\ \sum_{j=1}^{p} \mathbf{c}_{d,j}(\mathbf{y}_{t-\ell}^{\top}\boldsymbol{\beta})\mathbf{y}_{t-j} \end{pmatrix} + \begin{pmatrix} \mathbf{x}_{t,1}\alpha_1 \\ \mathbf{x}_{t,2}\alpha_2 \\ \vdots \\ \mathbf{x}_{t,d}\alpha_d \end{pmatrix} + \begin{pmatrix} \epsilon_{t,1} \\ \epsilon_{t,2} \\ \vdots \\ \epsilon_{t,d} \end{pmatrix}. \quad (4.28)$$

Hence, we obtain the expression of $y_{t,m}, m = 1, \cdots, d$ as

$$y_{t,m} = \sum_{j=1}^{p} \boldsymbol{c}_{m,j}(\mathbf{y}_{t-\ell}^{\top}\boldsymbol{\beta})\mathbf{y}_{t-j} + \mathbf{x}_{t,m}\alpha_m + \epsilon_{t,m} \quad (4.29)$$

For any given j, $j = 1, \cdots, p$, and k, $k = 1, \cdots, n-\ell$, by applying Taylor's expansion,

$$\boldsymbol{c}_{m,j}(\mathbf{y}_i^{\top}\boldsymbol{\beta}) \approx \boldsymbol{c}_{m,j}(\mathbf{y}_k^{\top}\boldsymbol{\beta}) + \dot{\boldsymbol{c}}_{m,j}(\mathbf{y}_k^{\top}\boldsymbol{\beta})\left[(\mathbf{y}_i - \mathbf{y}_k)^{\top}\boldsymbol{\beta}\right],$$

when $\mathbf{y}_i^{\top}\boldsymbol{\beta}$ is in a small neighbourhood of $\mathbf{y}_k^{\top}\boldsymbol{\beta}$, where $\dot{\boldsymbol{c}}_{m,j}(\cdot)$ is the derivative of $\boldsymbol{c}_{m,j}(\cdot)$. Therefore we can approximate model (4.29) by

$$y_{t,m} \approx \sum_{j=1}^{p} \left\{ \boldsymbol{c}_{m,j}(\mathbf{y}_k^{\top}\boldsymbol{\beta}) + \dot{\boldsymbol{c}}_{m,j}(\mathbf{y}_k^{\top}\boldsymbol{\beta})(\mathbf{y}_{t-\ell}^{\top}\boldsymbol{\beta} - \mathbf{y}_k^{\top}\boldsymbol{\beta}) \right\} \mathbf{y}_{t-j} + \mathbf{x}_{t,m}\alpha_m + \epsilon_{t,m}.$$

For brevity purposes, we let $a_{j\imath\jmath k}$ be the $(\imath, \jmath)$th element of $\mathbf{A}_{j,k}$ and use the notation

$$\mathbf{a}_{j\imath\jmath} = (a_{j\imath\jmath 1}, \cdots, a_{j\imath\jmath(n-\ell)})^{\top},$$

$$\boldsymbol{d}_{m,j,k} = \dot{\boldsymbol{c}}_{m,j,k},$$

$$\boldsymbol{\theta}_{m,k} = (\boldsymbol{c}_{m,1,k}, \boldsymbol{d}_{m,1,k}, \cdots, \boldsymbol{c}_{m,p,k}, \boldsymbol{c}_{m,p,k}),$$

$$\boldsymbol{\Theta} = (\sum_{m=1}^{d} \boldsymbol{\theta}_{m,1}, \cdots, \sum_{m=1}^{d} \boldsymbol{\theta}_{m,n-l}),$$

where we denote $\boldsymbol{c}_{m,j}(\mathbf{y}_k^\top \boldsymbol{\beta})$ and $\dot{\boldsymbol{c}}_{m,j}(\mathbf{y}_k^\top \boldsymbol{\beta})$ by $\boldsymbol{c}_{m,j,k}$ and $\dot{\boldsymbol{d}}_{m,j,k}$ respectively.

Using approximation (4.2.2) together with the idea of least squares, we can form the following local discrepancy loss function:

$$L(\boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\Theta}) = \sum_{m=1}^{d} \sum_{k=1}^{n-l} \sum_{t=p+1}^{n} \Big\{ y_{t,m} - \sum_{j=1}^{p} \big[\boldsymbol{c}_{m,j,k} + \boldsymbol{d}_{m,j,k} (\mathbf{y}_{t-\ell}^\top \boldsymbol{\beta} - \mathbf{y}_k^\top \boldsymbol{\beta}) \big] \mathbf{y}_{t-j} - \mathbf{x}_{t,m} \alpha_m \Big\}^2 K_h((\mathbf{y}_{t-\ell} - \mathbf{y}_k)^\top \boldsymbol{\beta}), \tag{4.30}$$

where $K_h(\cdot) = K(\cdot/h)/h$, $K(\cdot)$ is a kernel function, h is a bandwidth.

Because $\boldsymbol{c}_{m,j}(\cdot), j = 1, \cdots, p, m = 1, \cdots, d$ are functional, to deal with the sparsity in $\boldsymbol{c}_{m,j}(\cdot)$, the penalty has to be imposed on the function values of $\boldsymbol{c}_{m,j}(\cdot)$, at all $\mathbf{y}_k^\top, k = 1, \cdots, n-l$. Therefore, the form of the penalised least squares is constructed as

$$\begin{aligned} \mathscr{L}(\boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\Theta}) =& L(\boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\Theta}) + \sum_{l=1}^{d} P_l(|\beta_l|) \\ &+ \sum_{j=1}^{p} \sum_{m=1}^{d} \Big\{ P_{jm}(\|\boldsymbol{C}_{m,j}\|) + P_{jm}(\|\boldsymbol{D}_{m,j}\|) \Big\} \end{aligned} \tag{4.31}$$

where

$$P_l(\cdot) = \mathscr{P}_{\lambda_l}(\cdot), \quad P_{jm}(\cdot) = \mathscr{P}_{\lambda_{jm}}(\cdot),$$

$$\boldsymbol{C}_{m,j} = (\boldsymbol{c}_{m,j,1}, \cdots, \boldsymbol{c}_{m,j,n-l})^\top, \quad \boldsymbol{D}_{m,j} = (\boldsymbol{d}_{m,j,1}, \cdots, \boldsymbol{c}_{m,j,n-l})^\top.$$

The estimators $\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\Theta}}$, and $\hat{\boldsymbol{\alpha}}$ can be obtained by solving

$$(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\Theta}}) = \underset{\boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\Theta}}{\arg\min} \mathscr{L}(\boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\Theta}) \tag{4.32}$$

subject to the constraints $\|\boldsymbol{\beta}\| = 1$ and $\beta_1 > 0$. Following the idea of quadratic approximation, see Fan and Li (2001), by properly defining the threshold δ, we let $\|\hat{\mathbf{a}}_{j\imath\jmath}\|$ be 0 when $\|\hat{\mathbf{a}}_{j\imath\jmath}\| < \delta$. Once $\|\hat{\mathbf{a}}_{j\imath\jmath}\| = 0$, we use 0 to estimate the $(\imath, \jmath)$th element of $\mathbf{A}_j(\cdot)$. When $\|\hat{\mathbf{a}}_{j\imath\jmath}\| \neq 0$, we use $\hat{\mathbf{a}}_{j\imath\jmath}$ to estimate $(a_{j\imath\jmath}(\mathbf{y}_1^\top \boldsymbol{\beta}), \cdots, a_{j\imath\jmath}(\mathbf{y}_{n-\ell}^\top \boldsymbol{\beta}))^\top$. As with the previous estimation method, the global minimum of the local discrepancy function cannot be found analytically and therefore an iterative procedure is proposed for implementation purposes, which is written as follows:

Step 1. Choose initial value $\boldsymbol{\beta}^0$. Before the iterative procedure, an initial value of $\boldsymbol{\beta}$ should be specified, which is denoted by $\boldsymbol{\beta}^0$. The initial value $\boldsymbol{\beta}^0$ needs to satisfy the constraints $\boldsymbol{\beta}_1^0 > 0$ and $\|\boldsymbol{\beta}^0\| = 1$, where $\boldsymbol{\beta}_1^0$ is the first component of $\boldsymbol{\beta}^0$.

Step 2. Select and Estimate $\boldsymbol{\Theta}$,and fit$\boldsymbol{\alpha}$ giving the known $\boldsymbol{\beta}$. Set $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}^0$, we now estimate $\boldsymbol{\Theta}, \boldsymbol{\alpha}$ by solving

$$(\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\alpha}}) = \underset{\boldsymbol{\Theta}, \boldsymbol{\alpha}}{\arg\min}\ \mathscr{L}(\boldsymbol{\Theta}, \boldsymbol{\alpha} | \hat{\boldsymbol{\beta}})$$

in other words, we need to minimise the form of penalised least squares

$$\mathscr{L}(\boldsymbol{\Theta}, \boldsymbol{\alpha}|\hat{\boldsymbol{\beta}}) = L(\boldsymbol{\Theta}, \boldsymbol{\alpha}|\hat{\boldsymbol{\beta}}) + \sum_{l=1}^{d} P_l(|\beta_l|) + \sum_{j=1}^{p}\sum_{m=1}^{d} \left\{ P_{jm}(\|\boldsymbol{C}_{m,j}\|) + P_{jm}(\|\boldsymbol{D}_{m,j}\|) \right\} \quad (4.33)$$

In order to deal with the minimisation problem (4.33), a unified algorithm of local quadratic approximation proposed by Fan and Li (2001) is applied here. For the penalised least squares problem, it is straightforward to find that we can locally approximate the first term in (4.33) by a quadratic function. As for the second term, SCAD penalty functions, they can also be locally approximated by a quadratic function, which will be discussed as follows. Therefore, the expression (4.33) will be formed into a quadratic function where we can obtain the minimiser as an explicit solution.

For this approximation of the SCAD penalty function, as mentioned previously, we have to set the initial value, which is close to the minimiser of (4.31). In this section, we will use the minimiser of $L(\hat{\boldsymbol{\beta}}, \boldsymbol{\Theta}, \boldsymbol{\alpha})$ as the initial value of $\boldsymbol{C}_{m,j}$ and $\boldsymbol{D}_{m,j}$. Let $\hat{\boldsymbol{\Theta}} = (\hat{\boldsymbol{C}}_{1,1}, \cdots, \hat{\boldsymbol{C}}_{1,d}, \hat{\boldsymbol{D}}_{1,1}, \cdots, \hat{\boldsymbol{D}}_{1,d}, \cdots, \hat{\boldsymbol{C}}_{p,1}, \cdots, \hat{\boldsymbol{C}}_{p,d}, \hat{\boldsymbol{D}}_{p,1}, \cdots, \hat{\boldsymbol{D}}_{p,d})$ be the current "minimiser" of $L(\hat{\boldsymbol{\beta}}, \boldsymbol{\Theta}, \boldsymbol{\alpha})$, the penalty functions $P_{jm}(\|\boldsymbol{C}_{m,j}\|)$ and $P_{jm}(\|\boldsymbol{D}_{m,j}\|)$ will be applied to the

approximation:

$$P_{jm}(\|\boldsymbol{C}_{m,j}\|) \approx P_{jm}(\|\hat{\boldsymbol{C}}_{m,j}\|) + \frac{P'_{jm}(\|\hat{\boldsymbol{C}}_{m,j}\|)}{2\|\hat{\boldsymbol{C}}_{m,j}\|}(\boldsymbol{C}_{m,j}^{\top}\boldsymbol{C}_{m,j} - \hat{\boldsymbol{C}}_{m,j}^{\top}\hat{\boldsymbol{C}}_{m,j}),$$

$$P_{jm}(\|\boldsymbol{D}_{m,j}\|) \approx P_{jm}(\|\hat{\boldsymbol{D}}_{m,j}\|) + \frac{P'_{jm}(\|\hat{\boldsymbol{D}}_{m,j}\|)}{2\|\hat{\boldsymbol{D}}_{m,j}\|}(\boldsymbol{D}_{m,j}^{\top}\boldsymbol{D}_{m,j} - \hat{\boldsymbol{D}}_{m,j}^{\top}\hat{\boldsymbol{D}}_{m,j}),$$

Then the penalised least square problem (4.33) can be locally approximated (except for a constant term) by:

$$L(\boldsymbol{\Theta}, \boldsymbol{\alpha}|\hat{\boldsymbol{\beta}}) = \sum_{j=1}^{p}\sum_{m=1}^{d}\left\{\frac{P'_{jm}(\|\hat{\boldsymbol{C}}_{m,j}\|)}{2\|\hat{\boldsymbol{C}}_{m,j}\|}\boldsymbol{C}_{m,j}^{\top}\boldsymbol{C}_{m,j} + \frac{P'_{jm}(\|\hat{\boldsymbol{D}}_{m,j}\|)}{2\|\hat{\boldsymbol{D}}_{m,j}\|}\boldsymbol{D}_{m,j}^{\top}\boldsymbol{D}_{m,j}\right\} \tag{4.34}$$

which is equivalent to

$$\begin{aligned} L(\boldsymbol{\Theta}, \boldsymbol{\alpha}|\hat{\boldsymbol{\beta}}) \;=\; & \sum_{k=1}^{n-l}\sum_{j=1}^{p}\sum_{m=1}^{d}\left\{\frac{P'_{jm}(\|\hat{\boldsymbol{C}}_{m,j}\|)}{2\|\hat{\boldsymbol{C}}_{m,j}\|}\|\boldsymbol{c}_{m,j,k}\|^2 + \frac{P'_{jm}(\|\hat{\boldsymbol{D}}_{m,j}\|)}{2\|\hat{\boldsymbol{D}}_{m,j}\|}\right. \\ & \left.\times\|\boldsymbol{d}_{m,j,k}\|^2\right\} \end{aligned} \tag{4.35}$$

Therefore, for the sake of solving this minimisation problem (4.35), we need to obtain $\hat{\boldsymbol{\Theta}}$, the minimise of $L(\hat{\boldsymbol{\beta}}, \boldsymbol{\Theta}, \boldsymbol{\alpha})$ firstly. For each k and m, we choose the estimator of $\boldsymbol{\theta}_{m,k}, \alpha_m$ by minimising

$$\begin{aligned} & \sum_{t=p+1}^{n}\left\{y_{t,m} - \sum_{j=1}^{p}\left[\boldsymbol{c}_{m,j,k} + \boldsymbol{d}_{m,j,k}(\mathbf{y}_{t-\ell}^{\top}\hat{\boldsymbol{\beta}} - \mathbf{y}_k^{\top}\hat{\boldsymbol{\beta}})\right]\mathbf{y}_{t-j}\right. \\ & \left. - \mathbf{x}_{t,m}\alpha_m\right\}^2 K_h((\mathbf{y}_{t-\ell} - \mathbf{y}_k)^{\top}\hat{\boldsymbol{\beta}}), \end{aligned} \tag{4.36}$$

Rewriting the minimisation problem yields

$$\sum_{t=p+1}^{n} \left\| y_{t,m} - \mathbf{Q}_{t,k}\boldsymbol{\theta}_{m,k}^{\top} - \mathbf{x}_{t,m}\alpha_m \right\|^2 K_h((\mathbf{y}_{t-\ell} - \mathbf{y}_k)^{\top}\hat{\boldsymbol{\beta}}), \quad (4.37)$$

where $\mathbf{Q}_{t,k} = (\mathbf{y}_{t-1}^{\top}, (\mathbf{y}_{t-\ell}^{\top}\hat{\boldsymbol{\beta}} - z_k)\mathbf{y}_{t-1}^{\top}, \cdots, \mathbf{y}_{t-p}^{\top}, (\mathbf{y}_{t-\ell}^{\top}\hat{\boldsymbol{\beta}} - z_k)\mathbf{y}_{t-p}^{\top})$ with $z_k = \mathbf{y}_k^{\top}\hat{\boldsymbol{\beta}}$.

For the identifiability purpose, in exactly the same way as in Section 3.1, the model is set to satisfy the conditions $\|\boldsymbol{\beta}\| = 1, \beta_1 > 0$ and the first column of $\mathbf{A}_{\ell}(\cdot)$ is $\mathbf{0}_d$, which can be practically realised instead of estimating the first column of $\mathbf{A}_{\ell}(\cdot)$ in the estimation procedure, we let the elements of the first column of $\mathbf{A}_{\ell}(\cdot)$ equal 0 directly. It leads to a reduction of dimension of $\boldsymbol{\theta}_{m,k}$ from $(2pd \times 1)$ to $(2pd-2) \times 1$. Correspondingly, both of the first element of $\mathbf{y}_{t-\ell}$ and $(\mathbf{y}_{t-\ell}^{\top}\hat{\boldsymbol{\beta}} - z_k)\mathbf{y}_{t-\ell}$ of $\mathbf{Q}_{t,k}$ need to be eliminated in the estimation procedure, therefore, the dimension of $\mathbf{Q}_{t,k}$ reduces from $1 \times 2pd$ to $1 \times (2pd-2)$.

In order to reduce more computation cost, we find we can rewrite the single summation in the equation (4.37) in the form of stacked vectors and matrices. This can be achieved by defining the $(n-p) \times 1$ vector $\mathbf{y}_m$, the $(n-p) \times (2pd-2)$ matrix $\mathbb{Q}_k$, the $(n-p) \times q$ matrix $\mathbf{X}_m$ and the $(n-p) \times (n-p)$ diagonal matrix $\mathbf{W}_k$ as follows:

$$\mathbf{y}_m = (y_{1,m}, \cdots, y_{n-p,m})^{\top}, \quad \mathbb{Q}_k = (\mathbf{Q}_{1,k}^{\top}, \cdots, \mathbf{Q}_{n-p,k}^{\top})^{\top},$$

$$\mathbf{X}_m = (\mathbf{x}_{1,m}^{\top}, \cdots, \mathbf{x}_{n-p,m}^{\top})^{\top},$$

$$\mathbf{W}_k = \text{diag}\Big\{K_h((\mathbf{y}_{1-\ell} - \mathbf{y}_k)^\top \hat{\boldsymbol{\beta}}), \cdots, K_h((\mathbf{y}_{n-p-\ell} - \mathbf{y}_k)^\top \hat{\boldsymbol{\beta}})\Big\},$$

With these, equation (4.37) can be written in matrix notation:

$$(\hat{\boldsymbol{\theta}}_{m,k}, \hat{\alpha}_m) = \underset{\boldsymbol{\theta}_{m,k}, \alpha_m}{\arg\min} \left\| \mathbf{y}_m - \mathbb{Q}_k \boldsymbol{\theta}_{m,k}^\top - \mathbf{X}_m \alpha_m \right\|^2 \mathbf{W}_k, \qquad (4.38)$$

In order to obtain the closed form solution, we have to rewrite the equation (4.38) as:

$$\hat{\boldsymbol{\gamma}}_{m,k} = \underset{\boldsymbol{\gamma}_{m,k}}{\arg\min} \left\| \mathbf{y}_m - \mathbb{X}_{m,k} \boldsymbol{\gamma}_{m,k} \right\|^2 \mathbf{W}_k, \qquad (4.39)$$

where $\boldsymbol{\gamma}_{m,k} = [\boldsymbol{\theta}_{m,k}^\top, \alpha_m]^\top$ is a $(2pd - 2 + q) \times 1$ vector, $\mathbb{X}_{m,k} = [\mathbb{Q}_k, \mathbf{X}_m] \in \mathbb{R}^{(n-p)\times(2pd-2+q)}$, $\boldsymbol{\Gamma} = (\sum_{m=1}^d \boldsymbol{\gamma}_{m,1}, \cdots, \sum_{m=1}^d \boldsymbol{\gamma}_{m,n-l})$. and the solution is given by

$$\hat{\boldsymbol{\gamma}}_{m,k} = (\mathbb{X}_{m,k}^\top \mathbf{W}_k \mathbb{X}_{m,k})^{-1} (\mathbb{X}_{m,k}^\top \mathbf{W}_k \mathbf{y}_m), \qquad (4.40)$$

and hence we obtain $\hat{\boldsymbol{\Gamma}} = (\sum_{m=1}^d \hat{\boldsymbol{\gamma}}_{m,1}, \cdots, \sum_{m=1}^d \hat{\boldsymbol{\gamma}}_{m,n-l})$.

After working out $\hat{\boldsymbol{\Theta}}$ and $\hat{\boldsymbol{\alpha}}$, let us go back to the penalised least square problem, the expression (4.41) can be written in the matrix notation as

$$\sum_{m=1}^{d} \sum_{k=1}^{n-l} \left\| \mathbf{y}_m - \mathbb{Q}_k \boldsymbol{\theta}_{m,k}^\top - \mathbf{X}_m \alpha_m \right\|^2 \mathbf{W}_k + \frac{1}{2} \sum_{m=1}^{d} \sum_{k=1}^{n-l} \boldsymbol{\theta}_k \Sigma_{\boldsymbol{\Theta}} \boldsymbol{\theta}_k^\top, \quad (4.41)$$

where $\Sigma_{\boldsymbol{\Theta}} = \text{diag}\Big\{P'_{m,1}(\|\hat{\boldsymbol{C}}_{m,1}\|)/\|\hat{\boldsymbol{C}}_{m,1}\|, P'_{m,1}(\|\hat{\boldsymbol{D}}_{m,1}\|)/\|\hat{\boldsymbol{D}}_{m,1}\|, \cdots, P'_{m,p}(\|\hat{\boldsymbol{C}}_{m,p}\|)/\|\hat{\boldsymbol{C}}_{m,p}\|, P'_{m,p}(\|\hat{\boldsymbol{D}}_{m,p}\|)/\|\hat{\boldsymbol{D}}_{m,p}\|\Big\}$.

The continuous differentiable penalty function is defined previously in (4.3), we can obtain the solution by the penalised least square approach:

$$\check{\boldsymbol{\theta}}_{m,k} = \left(\mathbb{Q}_k^\top \mathbf{W}_k \mathbb{Q}_k + \frac{n-p}{2}\Sigma_{\boldsymbol{\theta}_k}\right)^{-1}\left(\mathbb{Q}_k^\top \mathbf{W}_k(\mathbf{y}_m - \mathbf{X}_m \alpha_m)\right) \quad (4.42)$$

Hence, we have $\check{\boldsymbol{\Theta}} = (\sum_{m=1}^d \check{\boldsymbol{\theta}}_{m,1}, \cdots, \sum_{m=1}^d \check{\boldsymbol{\theta}}_{m,n-l})$.

Step 3. Estimate and select $\boldsymbol{\beta}$ given $\boldsymbol{\Theta}, \boldsymbol{\alpha}$. Using the estimators $\check{\boldsymbol{\Theta}}$ and $\hat{\boldsymbol{\alpha}}$ from Step two, we would like to find the estimator of $\boldsymbol{\beta}$:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \mathscr{L}(\boldsymbol{\beta}|\check{\boldsymbol{\Theta}}, \hat{\boldsymbol{\alpha}})$$

in the other words, we need to minimise the form of penalised least squares

$$L(\boldsymbol{\beta}|\check{\boldsymbol{\Theta}}, \hat{\boldsymbol{\alpha}}) + \sum_{l=1}^{d} P_l(|\beta_l|) \quad (4.43)$$

In order to deal with the penalised least square problem (4.43), the same algorithm as the last step will be applied here. We will use the local quadratic approximation to reduce the minimisation problem to a quadratic minimisation problem. Similarly, the minimise of $L(\boldsymbol{\beta}, |\check{\boldsymbol{\Theta}}, \hat{\boldsymbol{\alpha}})$ will be set as the initial value of $\boldsymbol{\beta}$.

Let $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \cdots, \hat{\beta}_d)^\top$ be the current minimiser of $L(\boldsymbol{\beta}, |\check{\boldsymbol{\Theta}}, \hat{\boldsymbol{\alpha}})$, apply the approximation

$$P_l(|\beta_l|) \approx P_l(|\hat{\beta}_l|) + \frac{P_l'(|\hat{\beta}_l|)}{2|\hat{\beta}_l|}(\beta_l^2 - \hat{\beta}_l^2),$$

to the $P_l(|\beta_l|)$ in (4.43), then the penalised least square problem (4.43) can be locally approximated (except for a constant term) by:

$$L(\boldsymbol{\beta}, |\check{\boldsymbol{\Theta}}, \hat{\boldsymbol{\alpha}}) + \sum_{l=1}^{d} \frac{P_l'(|\hat{\beta}_l|)}{2|\hat{\beta}_l|} \beta_l^2 \tag{4.44}$$

Therefore, for the sake of solving this minimisation problem (4.44), we need to obtain $\hat{\boldsymbol{\beta}}$, the minimise of $L(\boldsymbol{\beta}, \check{\boldsymbol{\Theta}}, \hat{\boldsymbol{\alpha}})$ firstly,

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} (\boldsymbol{\beta}, \check{\boldsymbol{\Theta}}, \hat{\boldsymbol{\alpha}})$$

which is equivalent to minimise

$$\sum_{k=1}^{n-l} \sum_{t=p+1}^{n} \left\| \mathbf{y}_t - \sum_{j=1}^{p} \left\{ \hat{\mathbf{A}}_{j,k} + \hat{\mathbf{B}}_{j,k} \left[(\mathbf{y}_{t-\ell} - \mathbf{y}_k)^\top \boldsymbol{\beta} \right] \right\} \mathbf{y}_{t-j} - \mathbf{X}_t \hat{\boldsymbol{\alpha}} \right\|^2 \times K_h((\mathbf{y}_{t-\ell} - \mathbf{y}_k)^\top \hat{\boldsymbol{\beta}}), \tag{4.45}$$

This step is nearly the same as the step 3 of the purposed algorithm previous presented, therefore, we can rewrite the equation (4.45) in matrix notation as

$$\hat{\boldsymbol{\beta}}_{new} = \arg\min_{\boldsymbol{\beta}_{new}} \sum_{k=1}^{n-l} \sum_{t=p+1}^{n} \left\| \mathbf{c}_{t,k} - \mathbf{M}_{t,k} \boldsymbol{\beta}_{new} \right\|^2 K_h((\mathbf{y}_{t-\ell} - \mathbf{y}_k)^\top \hat{\boldsymbol{\beta}}_{old}), \tag{4.46}$$

where $\mathbf{c}_{t,k} = \mathbf{y}_t - \sum_{j=1}^{p} \hat{\mathbf{A}}_{j,k} \mathbf{y}_{t-j} - \mathbf{X}_t \hat{\boldsymbol{\alpha}}$ is a $d \times 1$ vector, $\mathbf{M}_{t,k} = \sum_{j=1}^{p} \hat{\mathbf{B}}_{j,k} \mathbf{y}_{t-j} (\mathbf{y}_{t-\ell} - \mathbf{y}_k)^\top \in \mathbb{R}^{d \times d}$, $\boldsymbol{\beta}_{old}$ is the estimator of $\boldsymbol{\beta}$ we used in Step 1 and $\boldsymbol{\beta}_{new}$ is the parameter we need to estimate

here. Then we have the solution by :

$$\begin{aligned}\hat{\boldsymbol{\beta}}_{new} &= \Big(\sum_{k=1}^{n-l}\sum_{t=p+1}^{n}\mathbf{M}_{t,k}^{\top}K_h((\mathbf{y}_{t-\ell}-\mathbf{y}_k)^{\top}\hat{\boldsymbol{\beta}}_{old})\mathbf{M}_{t,k}\Big)^{-1} \\ &\quad\times\Big(\sum_{k=1}^{n-l}\sum_{t=p+1}^{n}\mathbf{M}_{t,k}^{\top}K_h((\mathbf{y}_{t-\ell}-\mathbf{y}_k)^{\top}\hat{\boldsymbol{\beta}}_{old})\mathbf{c}_{t,k}\Big) \quad (4.47)\end{aligned}$$

After working out $\hat{\boldsymbol{\beta}}_{new}$, let us go back to the penalised least square problem, the expression (4.44) can be written in matrix notation as

$$\sum_{k=1}^{n-l}\sum_{t=p+1}^{n}\left\|\mathbf{c}_{t,k}-\mathbf{M}_{t,k}\boldsymbol{\beta}_{new}\right\|^2 K_h((\mathbf{y}_{t-\ell}-\mathbf{y}_k)^{\top}\hat{\boldsymbol{\beta}}_{old}))+\frac{1}{2}\boldsymbol{\beta}_{new}\Sigma_{\boldsymbol{\beta}}\boldsymbol{\beta}_{new} \quad (4.48)$$

where $\Sigma_{\boldsymbol{\beta}}=\mathrm{diag}\Big\{P_1'(|\hat{\beta}_{new,1}|)/|\hat{\beta}_{new,1}|,\cdots,P_d'(|\hat{\beta}_{new,d}|)/|\hat{\beta}_{new,d}|\Big\}$.

The continuous differentiable penalty function is defined previously (4.3), we can obtain the solution by the penalised least square approach:

$$\begin{aligned}\check{\boldsymbol{\beta}}_{new} &= \Big(\sum_{k=1}^{n-l}\sum_{t=p+1}^{n}\mathbf{M}_{t,k}^{\top}K_h((\mathbf{y}_{t-\ell}-\mathbf{y}_k)^{\top}\hat{\boldsymbol{\beta}}_{old})\mathbf{M}_{t,k}+\frac{n-p}{2}\Sigma_{\boldsymbol{\beta}}\Big)^{-1} \\ &\quad\times\Big(\sum_{k=1}^{n-l}\sum_{t=p+1}^{n}\mathbf{M}_{t,k}^{\top}K_h((\mathbf{y}_{t-\ell}-\mathbf{y}_k)^{\top}\hat{\boldsymbol{\beta}}_{old})\mathbf{c}_{t,k}\Big) \quad (4.49)\end{aligned}$$

At this point, in order to satisfy the identifiability conditions $\|\boldsymbol{\beta}\|=1$ and $\beta_1>0$, we define $\check{\boldsymbol{\beta}}_{new}=-\check{\boldsymbol{\beta}}_{new}/\|\check{\boldsymbol{\beta}}_{new}\|$ if the first component of $\check{\boldsymbol{\beta}}_{new}$ is negative, otherwise, let $\check{\boldsymbol{\beta}}_{new}=\check{\boldsymbol{\beta}}_{new}/\|\check{\boldsymbol{\beta}}_{new}\|$.

Furthermore, it can be seen that in the foregoing computational algorithms, we shrink the irrelevant components of the underlying model to zero only after the iterative procedure is completed. This implementation leads to a "double check" mechanism which works as follows: if after an iteration a coefficient or an index parameter is shrunken to be insignificant, it still has an opportunity to be reselected into the model in the following iteration. Thanks to this mechanism, our algorithm can overcome the main drawback in typical local quadratic approximation, which is that once a coefficient is lessened to zero, it will remain at zero. Meanwhile, since we do not eliminate the insignificant components in each iteration, the algorithm is not very sensitive to the choice of initial values, namely, the choice of the initial estimate $\boldsymbol{\beta}^0$ of $\boldsymbol{\beta}$.

4.3 Simulation study

In this section, we are going to build on the same simulation example in Section 4.3 by exploring how the model selection work affects the performance of the estimation in the two methods. Apart from this, we will simulate the example in a higher dimension to compare the running time and performance of two estimators.

We now consider a data - generating model with $q = 3, p =$

2 and sample size $n = 800$, which is defined by

$$\mathbf{y}_t = \mathbf{A}_1(z_{t-l})\mathbf{y}_{t-1} + \mathbf{A}_2(z_{t-l})\mathbf{y}_{t-2} + \mathbf{X}_t\boldsymbol{\alpha} + \boldsymbol{\epsilon}_t, \tag{4.50}$$

where $z_{t-l} = \mathbf{y}_{t-l}^\top \boldsymbol{\beta}$, $\mathbf{y}_t, t = 1, \cdots, n$ is a $d-$ dimensional stationary time series with 2 lags, $\mathbf{X}_t, t = 1, \cdots, n$ is a $d \times 3$ matrix generated from $d-$ dimensional Gaussian distribution, the $\boldsymbol{\epsilon}_t$ are independently generated from the normal distribution with $E(\boldsymbol{\epsilon}_t|\mathscr{F}_t) = \mathbf{0}_d$ and $\text{cov}(\boldsymbol{\epsilon}_t|\mathscr{F}_t) = \sigma^2\mathbf{I}_d$, where $\mathscr{F}_t = \{(Y_{l-1}, \mathbf{X}_l) : \ l \le t\}$. In this example, we set:

$$l = 1, \quad \sigma = \frac{1}{3}, \quad d = 3, \quad \boldsymbol{\alpha} = (1,2,2)^\top, \quad \boldsymbol{\beta} = (0.6, 0.8, 0)^\top,$$

$$\mathbf{A}_1(z_{t-l}) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0.9\exp(-z_{t-l}^2) & 0 \\ 0 & 0 & 0.25(\sin(\pi z_{t-l}) - 0.75) \end{pmatrix},$$

$$\mathbf{A}_2(z_{t-l}) = \begin{pmatrix} 0.33(\cos(\pi z_{t-l}) + 0.5) & 0 & 0 \\ 0.8\exp(-z_{t-l}^2) & 0.75\exp(-z_{t-l}^2) & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

where $\boldsymbol{\beta}$ is a $d \times 1$ vector and both $\mathbf{A}_1(z_{t-l})$ and $\mathbf{A}_2(z_{t-l})$ are $d \times d$ matrices. The first two predictors should be generated independently from a normal distribution, and then we generate $100 + n$ observations followed by the process (4.50). In order to improve the accuracy of estimation, we will discard the first 100 predictors and let $\mathbf{y}_t, t = 1, \cdots, n$ be the remaining predictors we have. For each case, we conduct the simulation over a total of 1000 replications.

Throughout this section, the kernel function we applied is Epanechnikov kernel $K(t) = 0.75(1-t^2)_+$ and the bandwidth we set is 0.25 of the whole range, which is $0.25 \times (\max\{\mathbf{y}_1^\top \hat{\boldsymbol{\beta}}, \cdots, \mathbf{y}_t^\top \hat{\boldsymbol{\beta}}\} - \min\{\mathbf{y}_1^\top \hat{\boldsymbol{\beta}}, \cdots, \mathbf{y}_t^\top \hat{\boldsymbol{\beta}}\})$. As for the tuning parameters, from $\Sigma_{\boldsymbol{\Theta}}$ and $\Sigma_{\boldsymbol{\beta}}$ in Section 5.2.1 and Section 5.2.2 and the given continuous differentiable SCAD penalty function:

$$p'_\lambda(\beta) = \lambda\left\{I(\beta \le \lambda) + \frac{(a\lambda - \beta)_+}{(a-1)\lambda} I(\beta > \lambda)\right\}.$$

We find that every continuous differentiable penalty function in $\Sigma_{\boldsymbol{\Theta}}$ and $\Sigma_{\boldsymbol{\beta}}$ will have a unique tuning parameter λ. However, in this situation, the closed form solution of the penalised least square problem will be difficult to implement. Therefore, for practical purpose, we simplify the tuning parameters as λ_A for $\Sigma_{\boldsymbol{\Theta}}$ and λ_β for $\Sigma_{\boldsymbol{\beta}}$. Thus, the original high dimensional problem of tuning parameter selection has now become a bivariate problem about $\{\lambda_A, \lambda_\beta\} \in \mathbb{R}^2$. According to the idea of choosing the tuning parameter in Fan and Li (2002), here we use $\lambda = \sqrt{2\log(d)}$ to get $\lambda_A = 2.33$ and $\lambda_\beta = 1.48$.

With the purpose of evaluating the accuracy of $\hat{\boldsymbol{\beta}}$ and $\hat{\mathbf{A}}_j, j = 1, \cdots, p$, the squared error metrics are used as follows:

$$\Delta(\hat{\mathbf{A}}) = \frac{\sum_{k=1}^{n-l}\sum_{j=1}^{p}\sum_{\iota=1}^{d}\sum_{\jmath=1}^{d}(\hat{a}_{j\iota\jmath k} - a_{j\iota\jmath k})^2}{\sum_{k=1}^{n-l}\sum_{j=1}^{p}\sum_{\iota=1}^{d}\sum_{\jmath=1}^{d} a_{j\iota\jmath k}^2}, \quad \Delta(\hat{\boldsymbol{\beta}}) = \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|^2, \tag{4.51}$$

The expectation and standard deviation of $\Delta(\hat{\mathbf{A}}_j)$ and $\Delta(\hat{\boldsymbol{\beta}})$ can be approximated by averaging over the 1000 replications using

$$\mathbb{E}(\Delta(\hat{\mathbf{A}})) \approx \frac{1}{1000}\sum_{L=1}^{1000}\Delta_L(\hat{\mathbf{A}}), \quad \mathbb{E}(\Delta(\hat{\boldsymbol{\beta}})) \approx \frac{1}{1000}\sum_{L=1}^{1000}\Delta_L(\hat{\boldsymbol{\beta}}),$$

$$SD(\Delta(\hat{\mathbf{A}})) \approx (\frac{1}{1000}\sum_{L=1}^{1000}\left\{\Delta_L(\hat{\mathbf{A}}) - \mathbb{E}(\Delta(\hat{\mathbf{A}}))\right\}^2)^{1/2},$$

$$SD(\Delta(\hat{\boldsymbol{\beta}})) \approx (\frac{1}{1000}\sum_{L=1}^{1000}\left\{\Delta_L(\hat{\boldsymbol{\beta}}) - \mathbb{E}(\Delta(\hat{\boldsymbol{\beta}}))\right\}^2)^{1/2},$$

where $\Delta_L(\cdot)$ denotes the square error metric of the Lth simulated dataset. One crucial thing to note is that $\mathbb{E}(\Delta(\hat{\boldsymbol{\beta}}))$ is the well known metric mean square error (MSE) in terms of the index parameters, and we can also call the metric $\mathbb{E}(\Delta(\hat{\mathbf{A}}))$ relative mean integrated squared error (RMISE) of the estimators of the unknown varying auto-regression coefficients.

In order to measure the performance of model selection, we report the ratio of the correct model, under-fitted model, over-fitted model and other models. Whenever the estimated model identifies the true submodel precisely, we classify it as a “correct model”. Whenever the resulting model discards at least one significant covariates but does not include any irrelevant covariates, we classify it as a “under-fitted model”. Whenever the estimated model includes at least one insignificant covariates but does not eliminate any relevant covariates,

it is classified as an “over-fitted model”. The “other models” means that the estimated submodel not only includes the irrelevant candidate covariates but also ignores relevant covariates. In particular, these three ratios are calculated from 1000 replications:

$$\text{rate of "correct model"} = \frac{\text{number of "correct model"}}{1000}$$

$$\text{rate of "under-fitted model"} = \frac{\text{number of "under-fitted" model}}{1000}$$

$$\text{rate of "over-fitted model"} = \frac{\text{number of "over-fitted" model}}{1000}$$

$$\text{rate of "other models"} = \frac{\text{number of "other models"}}{1000}$$

The simulated results of the estimation accuracy are reported in Table 4.1. Compared with the estimation performance of the penalised-free approach, whose simulated results are reported in Table 3.1, the proposed penalised estimators perform better. Furthermore, we also report the simulated results of model selection in Table 4.2. From the results in Table 4.2, we can see that both methods perform reasonably well on model selection in modest dimensionality.

Table 4.1: Comparison of estimates

	$\mathbb{E}(\Delta(\hat{\boldsymbol{\beta}}))$	$SD(\Delta(\hat{\boldsymbol{\beta}}))$	$\mathbb{E}(\Delta(\hat{\mathbf{A}}))$	$SD(\Delta(\hat{\mathbf{A}}))$
Method I	0.002	0.032	0.570	0.033
Method II	0.003	0.009	0.587	0.175
Method I (true $\boldsymbol{\beta}$)	0.000	0.000	0.579	0.092
Method II (true $\boldsymbol{\beta}$)	0.000	0.000	0.579	0.076

NOTE: *"Method* I *(True* $\boldsymbol{\beta}$*)" presents the one step estimation of* $\mathbf{A}$ *with the given true* $\boldsymbol{\beta}$ *via method* I*, "Method* II *(True* $\boldsymbol{\beta}$*)" presents the one step estimation of* $\mathbf{A}$ *with the given true* $\boldsymbol{\beta}$ *via method* II*.*

Table 4.2: Comparison of Model selection

	Correct	Overfitting	Underfitting	Others
I	0.847	0.138	0.004	0.001
I (true $\boldsymbol{\beta}$)	0.894	0.103	0.003	0.000
II	0.842	0.155	0.003	0.000
II (true $\boldsymbol{\beta}$)	0.864	0.134	0.002	0.000

NOTE: *'* I *' stands for the estimation of* $\mathbf{A}$ *via method* I*, while '* I *(True* $\boldsymbol{\beta}$*) ' presents the one step estimation of* $\mathbf{A}$ *with the given true* $\boldsymbol{\beta}$ *via method* I*; '*II *' presents the estimation of* $\mathbf{A}$ *via method* II*, and '*II *(True* $\boldsymbol{\beta}$*) ' presents the one step estimation of* $\mathbf{A}$ *with the given true* $\boldsymbol{\beta}$ *via method* II*.*

4.4 Comparison of the computational cost of the proposed two approaches

In this section, we will use a simulation study to explore the computational cost of each proposed shrinkage approach. By the numerical results, we shall demonstrate the efficiency of the second iterative procedure.

We first consider to make d flexible in the example (4.50) to illustrate how the increase of dimension of the model affects the running time and thereby to prove the second proposed method is more computational efficient. We have the same data-generating model with $q = 3, p = 2$ and sample size $n = 600$, which is defined by

$$\mathbf{y}_t = \mathbf{A}_1(z_{t-l})\mathbf{y}_{t-1} + \mathbf{A}_2(z_{t-l})\mathbf{y}_{t-2} + \mathbf{X}_t\boldsymbol{\alpha} + \boldsymbol{\epsilon}_t, \tag{4.52}$$

Keep the other parameters unchanged, we set:

$$l = 1, \quad \sigma = \frac{1}{3}, \quad \boldsymbol{\alpha} = (1,2,2)^\top, \quad \boldsymbol{\beta} = (0.6, 0.8, 0, \cdots, 0)^\top,$$

$$\mathbf{A}_1(z_{t-l}) = \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0.9\exp(-z_{t-l}^2) & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0.25(\sin(\pi z_{t-l}) - 0.75) & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix},$$

$$\mathbf{A}_2(z_{t-l}) = \begin{pmatrix} 0.33(\cos(\pi z_{t-l}) + 0.5) & 0 & 0 & 0 & \cdots & 0 \\ 0.8\exp(-z_{t-l}^2) & 0.75\exp(-z_{t-l}^2) & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix},$$

where $\boldsymbol{\beta}$ is a $d \times 1$ vector and both $\mathbf{A}_1(z_{t-l})$ and $\mathbf{A}_2(z_{t-l})$ are $d \times d$ matrices.

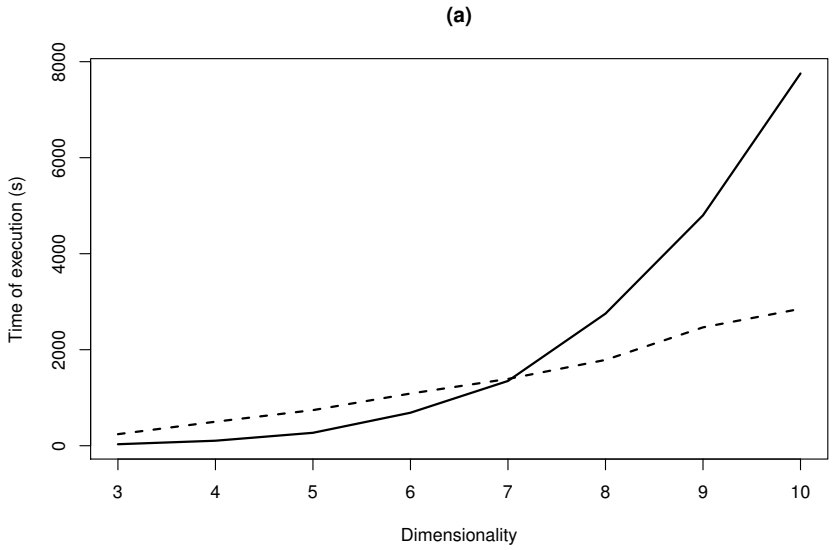
The implementation of this simulation is broken down as follows. We employ the regression example (4.52) in exactly the same way but change the dimensions from $d = 3$ to $d = 20$. Thus, there are as many as 18 models with different dimensionality that will be taken into consideration. Then, we record the time cost of both two iterative shrinkage methods on these 18 models. The simulations are conducted in 200 replications each with 600 samples. We will report the median of the time cost from all the replications.

The simulated results are virtually reported in the Figure 4.1 and Figure 4.2. The first figure is the simulated results on the models in modest dimensions and the second one summarises the simulation performance changing to a higher performance.

As illustrated in Figure 4.1, the second iterative approach may cost more time in low dimensionality, but the marginal computational cost is much less than the first iterative procedure. From the dimension $d = 7$, the cost of the second

estimation method is cheaper than the other one.

Figure 4.1: The computational cost of the two proposed approaches on the modest dimensional models



NOTE: *The solid line stands for the time cost of the first iterative procedure; the dashed line refers to the time cost of the second iterative approach.*

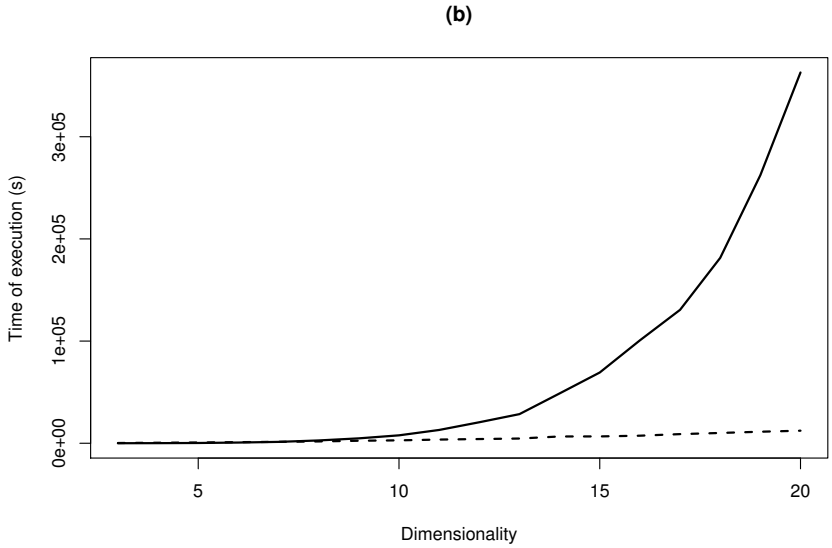
To extend our idea to a higher dimensionality, we summarise our simulated results on models with the dimension from $d = 3$ to $d = 20$ in Figure 4.2. The Figure 4.2 corroborates the findings in Figure 4.1 very well. Meanwhile, we intuitively notice that the time required of the first iterative approach approximately grows at rate $O(e^d)$ with the increasing of dimensionality and the growth is about rate $O(\sqrt{d})$ for the second iterative approach. Precisely, we also provide the accurate simulated time cost in Table 4.3.

Table 4.3: Median time cost of two iterative approaches

d	The 1st approach	The 2nd approach
3	31	241
4	104	500
5	269	741
6	687	1087
7	1349	1391
8	2752	1787
9	4800	2466
10	7754	2852
11	13026	3612
12	20558	4126
13	28514	4634
14	48784	6639
15	69255	6700
16	100720	7408
17	130569	8951
18	181440	10083
19	262400	11296
20	362880	12299

NOTE: *The column labelled with "The 1st approach" refers to the median time cost of the first iterative approach; the column labelled with "The 1st approach" represents the median time cost of the 2nd approach.*

Figure 4.2: The time cost of the two proposed approaches on models with the dimension from $d = 3$ to $d = 20$



NOTE: *The solid line stands for the time cost of the first iterative procedure; the dashed line refers to the time cost of the second iterative approach.*

From all the aforementioned numerical evidence, we conclude that the computational burden of the second iterative approach is less than the first one, and hence, we shall use the second iterative approach to select and estimate the SSIVAR model in real implementation.

CHAPTER 5

SELECTION OF HYPER-PARAMETERS

As selecting the optimal bandwidth is an essential topic in local polynomial regression and the selection of tuning parameters largely determine whether the SCAD -type penalised approach can consistently select the true model, we shall discuss the choice of these hyper-parameters in this chapter, and thus develop some efficient criteria to select them. We will explore the selection of bandwidth in Section 5.1 and address how to choose tuning parameter in Section 5.2. Meanwhile, we give simulation studies to demonstrate the performance of the corresponding metrics.

5.1 Bandwidth selection

We have set a bandwidth in the simulation study of previous chapters, the bandwidth h we use is 0.25 of the range of estimated indices, which is $25\% \times (\max\{\mathbf{y}_1^\top \hat{\boldsymbol{\beta}}, \cdots, \mathbf{y}_t^\top \hat{\boldsymbol{\beta}}\} - \min\{\mathbf{y}_1^\top \hat{\boldsymbol{\beta}}, \cdots, \mathbf{y}_t^\top \hat{\boldsymbol{\beta}}\})$. In this chapter, we will test different bandwidths and we find it is quite hard to visualise whether a particular value of h is "large" or "small". Here we prefer to use the percentage H which is the percentage of the whole range, to notate the size of bandwidth.

It states in Fan and Gijbels (1996) that a theoretical optimal bandwidth is obtained by minimising the conditional Mean Square Error (MSE) given $\mathbb{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_n)$ or the conditional weighted Mean Integrated Square Error (MISE) given $\mathbb{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_n)$. Accordingly, the criteria used for assessing the performance of the resulting estimates are their MSE and Relative MISE.

Although MSE-criterion or RMISE-criterion can be applied in the simulation study, in the real dataset, the true parameters are unknown and either of them is unable to be used. Thus, the cross-validation was considered as a possible alternative to select the bandwidth. Wu *et al.*(1998) proposed to use this statistic to choose the bandwidth.

However, it has been well studied in the literature that the cross-validation cannot consistently identify the optimal

bandwidth, whose choice always leads to overtting results, see Yang (2005) and Shao (1997). The other shortage of CV is that the computational burden of a grid-search approach based on cross-validation is very heavy. From our experience, in the high dimensional situation, the parallel computing should be applied to speed up the computation of left-one-out cross-validation.

Consequently, we are going to explore a data-driven method to evaluate the performance of the estimation with a sequence of bandwidth parameters from 0 to 100%. If the resulting estimates of our proposed approach are not very sensitive to the choice of the bandwidth as long as H is chosen to be within a reasonable range, and thus we can practically choose a befitting bandwidth in that range.

Let $(\mathbf{X}_i, Y_i)$, $i = 1, \ldots, n$ denote the observations, it states in Fan and Gijbels(1996) that a theoretical optimal bandwidth is obtained by minimising the conditional Mean Square Error (MSE) given $\mathbb{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_n)$ or the conditional weighted Mean Integrated Square Error (MISE) given $\mathbb{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_n)$. Accordingly, the criteria used for assessing the performance of the resulting estimates are their MSE and Relative MISE.

Specifically, we employ median MSE in all the replications to measure the goodness of the estimated index parameter $\hat{\boldsymbol{\beta}}$, which is defined as follows:

$$\text{MSE}_{\boldsymbol{\beta}} = \sum_{m=1}^{d} (\hat{\beta}_m^l - \beta_m)^2, \tag{5.1}$$

where $\hat{\beta}_m^l$, $m = 1, \ldots, d$ is either the unpenalised estimator or the penalised estimator from the l-th, which gives a median MSE from all the MSEs from L replications, β_m is the true index parameter; and we evaluate the goodness of estimators of coefficients in terms of the median relative MISE (RMISE), which can be approximated by

$$\text{RMISE}_{A(\cdot)} \approx \frac{\sum_{k=1}^{n-l} \sum_{j=1}^{p} \sum_{\imath=1}^{d} \sum_{\jmath=1}^{d} (\hat{a}_{j\imath\jmath}^l(z_k) - a_{j\imath\jmath}(z_k))^2}{\sum_{k=1}^{n-l} \sum_{j=1}^{p} \sum_{\imath=1}^{d} \sum_{\jmath=1}^{d} a_{j\imath\jmath}(z_k)^2}, \tag{5.2}$$

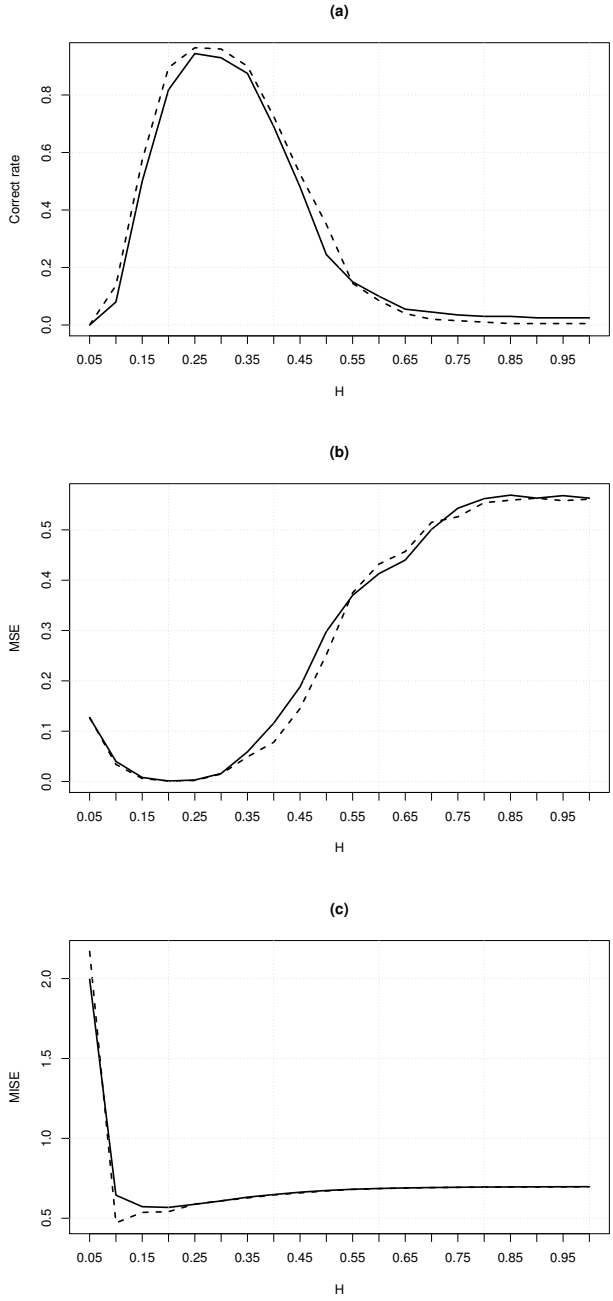
where $\hat{a}_{j\imath\jmath}^l(z_k)$, is either the unpenalised estimator or the penalised estimator of the function at the $(\imath, \jmath)$ entry of the matrix A_j, in the l-th replication, which leads to a median RMISE from all the RMISEs from L replications, and $z_k = \mathbf{y}_k^\top \hat{\boldsymbol{\beta}}$, $k = 1, \ldots, n-l$. Analogically, we will report the median RMISE from the simulation replications.

As the choice of bandwidth will impact both the model fitting and model selection, we also introduce the rate of "correct model" described in Section 4.3 to measure the performance of model selection.

Now, we execute a simulation on the regression example (4.52) with dimension $d = 3$ to explore the relationship between different bandwidth and the accuracy of estimation

and selection. The results are intuitively reported in Figure 5.1 and concretely summarised in Table 5.1 and Table 5.2.

Figure 5.1: Sensitivity of the choice of bandwidth H on the model with dimensions $d = 3$



NOTE: *Simulation results: (a) sensitivity of model selection to estimates to H; (b) sensitivity of MSE of index parameters to H; (c) sensitivity of RMISE of functional coefficients to H. In all cases: solid line, estimate on underlying model with noise ε_i; dashed line, estimate on underlying model without noise.*

Table 5.1: The sensitivity of model selection to the bandwidth

Bandwidth (H)	Correct Rate	
	With ϵ	Without ϵ
0.05	0.000	0.000
0.10	0.080	0.138
0.15	0.500	0.573
0.20	0.818	0.895
0.25	0.944	0.964
0.30	0.929	0.960
0.35	0.875	0.900
0.40	0.690	0.725
0.45	0.480	0.525
0.50	0.245	0.350
0.55	0.150	0.145
0.60	0.100	0.085
0.65	0.055	0.040
0.70	0.045	0.020
0.75	0.035	0.015
0.80	0.030	0.010
0.85	0.030	0.005
0.90	0.025	0.005
0.95	0.025	0.005
1.00	0.025	0.005

Table 5.2: The sensitivity of model estimation to the bandwidth

Bandwidth	MSE		RMISE	
(H)	With ϵ	Without ϵ	With ϵ	Without ϵ
0.05	0.127	0.127	1.997	2.170
0.10	0.040	0.034	0.644	0.471
0.15	0.008	0.006	0.572	0.536
0.20	0.001	0.001	0.568	0.541
0.25	0.003	0.002	0.587	0.588
0.30	0.016	0.015	0.608	0.609
0.35	0.059	0.049	0.632	0.627
0.40	0.116	0.078	0.648	0.646
0.45	0.188	0.145	0.663	0.658
0.50	0.298	0.252	0.674	0.671
0.55	0.370	0.375	0.681	0.681
0.60	0.413	0.432	0.686	0.685
0.65	0.440	0.457	0.690	0.689
0.70	0.501	0.515	0.692	0.692
0.75	0.543	0.526	0.694	0.693
0.80	0.562	0.554	0.695	0.695
0.85	0.569	0.559	0.696	0.695
0.90	0.563	0.563	0.697	0.696
0.95	0.568	0.558	0.697	0.696
1.00	0.563	0.561	0.697	0.697

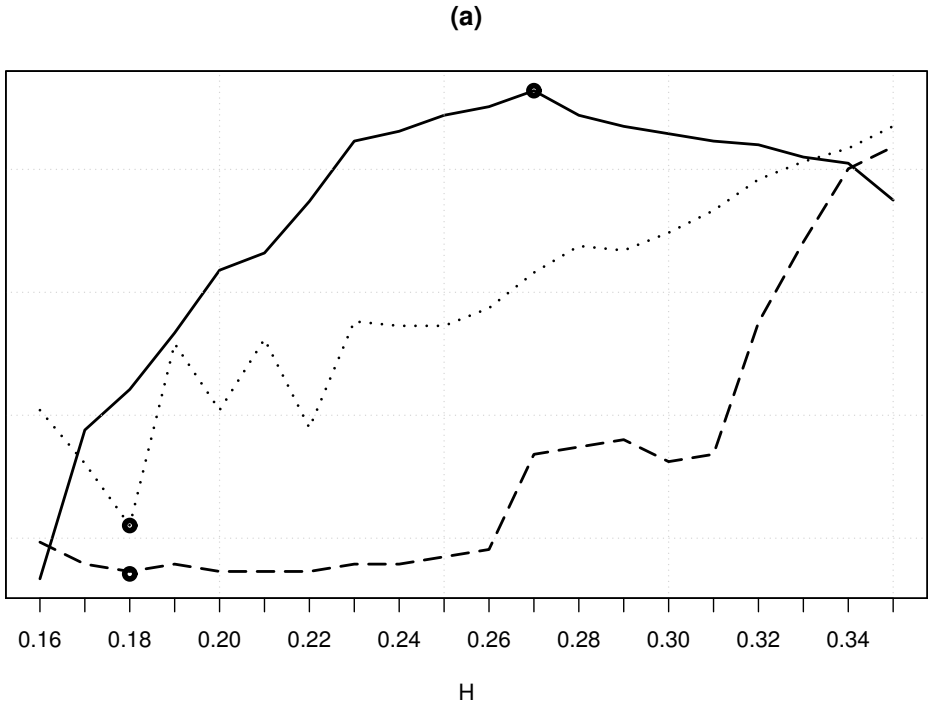
The finding from the simulated results is twofold. Firstly, there indeed exists the optimal bandwidth for the penalised

estimates of both index parameters and varying coefficients, which is inside the range $(0.1, 0.3)$. Secondly, as illustrated, both the performance of model selection and estimation are not very sensitive to the choice of the bandwidth once H is chosen to be within the range $(0.1, 0.3)$, hence an empirical reasonable bandwidth can be selected with this range. Concretely, we would like to choose the optimal bandwidth within that range. Hence, we conduct another simulation to show the sensitivity of each metric to the bandwidth in a lessened range which is from 0.16 to 0.35. The simulated results are summarised in Table 5.3 and visually reported in Figure 5.2.

Table 5.3: The sensitivity of model selection and estimation to the bandwidth

Bandwidth	Correct Rate	MSE	RMISE
0.16	0.567	0.005	0.568
0.17	0.688	0.002	0.556
0.18	0.721	0.001	0.542
0.19	0.767	0.002	0.583
0.20	0.818	0.001	0.568
0.21	0.832	0.001	0.584
0.22	0.874	0.001	0.564
0.23	0.923	0.002	0.588
0.24	0.931	0.002	0.587
0.25	0.944	0.003	0.587
0.26	0.951	0.004	0.591
0.27	0.964	0.017	0.599
0.28	0.944	0.018	0.605
0.29	0.935	0.019	0.604
0.30	0.929	0.016	0.608
0.31	0.923	0.017	0.613
0.32	0.920	0.035	0.620
0.33	0.910	0.046	0.624
0.34	0.905	0.056	0.627
0.35	0.875	0.059	0.632

Figure 5.2: The sensitivity of model selection and estimation to the bandwidth within the range $(0.16, 0.35)$



NOTE: *Solid line: sensitivity of model selection to H; dashed line: sensitivity of MSE of index parameters to H; dotted line: sensitivity of RMISE of functional coefficients to H.*

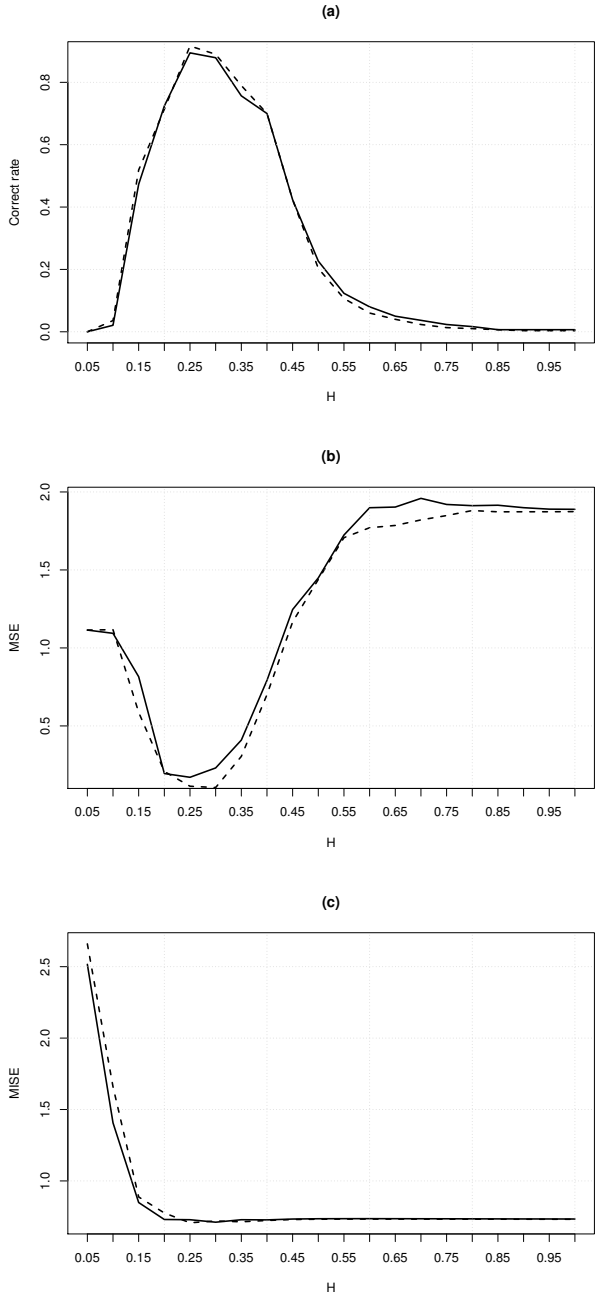
From Table 5.3 and Figure 5.2, we can find the optimal bandwidth from each metric. As the primary target of the proposed approach is to identify the true sub-model, we shall firstly follow the metric of “Correct Rate" to find the optimal bandwidth, which is 0.27. Then, if the model has been appropriately selected, we will get the optimal bandwidth regarding MSE or RMISE.

Furthermore, we also execute a simulation on the under-

lying model with dimension $d = 10$ by using the penalised approach to demonstrate the sensitivity of estimation accuracy to the bandwidth in higher dimensional situation. The tuning parameters used here are well selected based on the GIC, which we will concretely explain in Section 5.2.

Similarly, we conduct a simulation with sample size $n = 1000$ in a total of 300 replications. The results are reported in Figure 5.3. From these three figures, we remark that in the relatively high-dimensional model, both the optimal choice of bandwidth and the sensitivity to the choice of this hyper-parameter are fairly similar to the situation in the modest-dimensional model. The optimal bandwidth for penalised estimates of index parameters and the optimal bandwidth for penalised estimates of functional coefficients exists in range $(0.15, 0.35)$. Additionally, we provide the details of the simulated results in Table 5.4 and Table 5.5.

Figure 5.3: Sensitivity of the choice of bandwidth H on the model with dimensions $d = 10$



NOTE: *Simulation results: (a) sensitivity of model selection to estimates to H; (b) sensitivity of MSE of index parameters to H; (c) sensitivity of RMISE of functional coefficients to H. In all cases: solid line, estimate on underlying model with noise ε_i; dashed line, estimate on underlying model without noise.*

Table 5.4: The sensitivity of model estimation to the bandwidth on the model with dimensions $d = 10$

Bandwidth	Correct Rate	
(H)	With ϵ	Without ϵ
0.05	0.000	0.000
0.10	0.021	0.036
0.15	0.473	0.519
0.20	0.724	0.714
0.25	0.895	0.917
0.30	0.879	0.890
0.35	0.757	0.790
0.40	0.700	0.700
0.45	0.423	0.423
0.50	0.227	0.203
0.55	0.123	0.107
0.60	0.080	0.060
0.65	0.050	0.040
0.70	0.037	0.023
0.75	0.023	0.013
0.80	0.017	0.010
0.85	0.007	0.007
0.90	0.007	0.003
0.95	0.007	0.003
1.00	0.007	0.003

Table 5.5: The sensitivity of model estimation to the bandwidth on the model with dimensions $d = 10$

Bandwidth	MSE		RMISE	
(H)	With ϵ	Without ϵ	With ϵ	Without ϵ
0.05	0.372	0.371	2.447	2.586
0.10	0.365	0.372	1.368	1.613
0.15	0.272	0.196	0.825	0.863
0.20	0.065	0.069	0.710	0.754
0.25	0.057	0.038	0.708	0.687
0.30	0.077	0.035	0.691	0.700
0.35	0.136	0.102	0.708	0.694
0.40	0.264	0.233	0.707	0.703
0.45	0.415	0.389	0.713	0.710
0.50	0.483	0.480	0.715	0.711
0.55	0.575	0.569	0.715	0.712
0.60	0.633	0.590	0.715	0.713
0.65	0.634	0.595	0.715	0.712
0.70	0.653	0.607	0.715	0.712
0.75	0.640	0.616	0.715	0.712
0.80	0.637	0.627	0.714	0.712
0.85	0.638	0.624	0.714	0.712
0.90	0.633	0.624	0.713	0.711
0.95	0.630	0.624	0.713	0.711
1.00	0.630	0.625	0.713	0.711

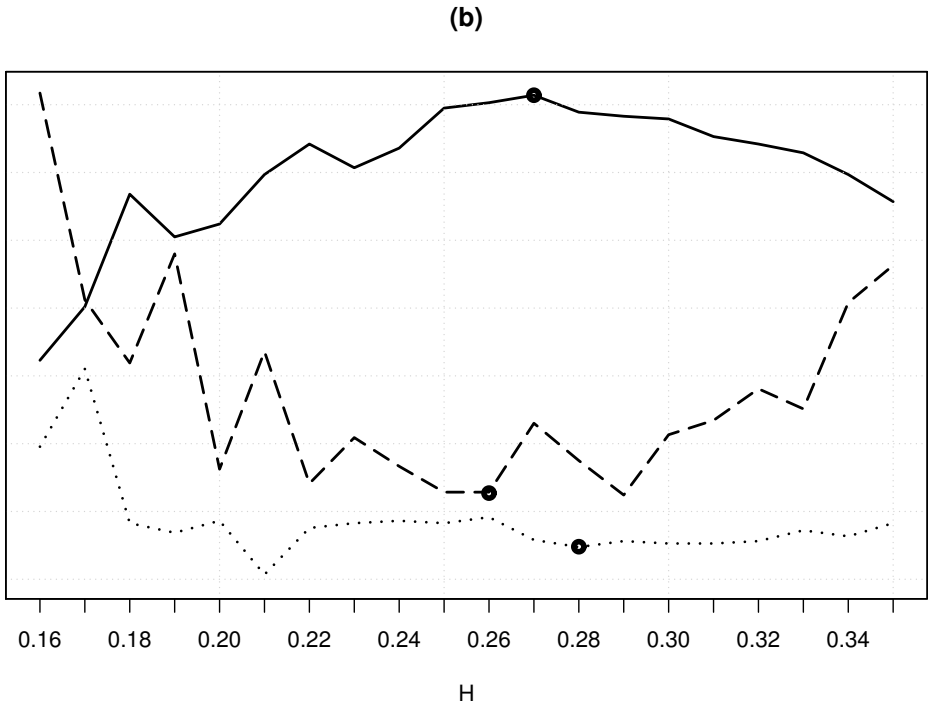
The aforementioned simulation studies provide the empirical evidence that the optimal bandwidth H can always be

found within a reasonable range. Hence, we conduct another simulation to describe the sensitivity of model selection and estimation to the bandwidth in the reasonable range which is chosen from 0.16 to 0.35. The simulated results are reported in Table 5.6 and Figure 5.4, respectively.

Table 5.6: The sensitivity of model selection and estimation to the bandwidth in the 10 dimensional model

Bandwidth	Correct Rate	MSE	RMISE
0.16	0.523	0.196	0.772
0.17	0.602	0.124	0.837
0.18	0.768	0.102	0.708
0.19	0.705	0.140	0.700
0.20	0.724	0.065	0.710
0.21	0.797	0.106	0.665
0.22	0.842	0.060	0.704
0.23	0.807	0.076	0.708
0.24	0.836	0.066	0.710
0.25	0.895	0.057	0.708
0.26	0.903	0.057	0.713
0.27	0.914	0.081	0.694
0.28	0.889	0.068	0.688
0.29	0.883	0.056	0.693
0.30	0.879	0.077	0.691
0.31	0.853	0.082	0.691
0.32	0.842	0.093	0.693
0.33	0.829	0.086	0.702
0.34	0.797	0.123	0.697
0.35	0.757	0.136	0.708

Figure 5.4: The sensitivity of model selection and estimation to the bandwidth within the range $(0.16, 0.35)$ in the 10 dimensional model



NOTE: *Solid line: sensitivity of model selection to H; dashed line: sensitivity of MSE of index parameters to H; dotted line: sensitivity of RMISE of functional coefficients to H.*

Furthermore, we remark that our simulated results are in line with the idea of Li, Ke and Zhang (2015) to empirically choose the bandwidth as $H = 0.6(d/n)^{0.2}$. Therefore, we also recommend to follow their idea to select the bandwidth in real implementation.

5.2 Selection of tuning parameters

In the thesis, we use the generalised information criterion (GIC) to determine the tuning parameters, which is proposed in Fang and Tang (2013). The tuning parameter vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \cdots, \lambda_{pd^2+d}, \tilde{\lambda}_1, \cdots, \tilde{\lambda}_d)$ is of dimension $pd^2 + d$, but to simultaneously choose a total of $pd^2 + d$ tuning parameters is very challenging. Therefore, for brevity purposes, we consider a 2-dimensional problem about $\boldsymbol{\lambda} = (\lambda_A, \lambda_\beta) \in \mathbb{R}^2$.

As in the true model, the non-zero coefficients may consist of functional coefficients and the constant coefficients, it is necessary to firstly calculate that each unknown functional parameter amounts to how many unknown constant parameters. We follow the idea of Cheng, Zhang and Chen (2009), which suggests that when sample size n is sufficiently large, an unknown functional parameter equals to approximate $1.028571h^{-1}$ constant parameters when Epanechnikov kernel is used. Hence, we select the optimal tuning parameters $\boldsymbol{\lambda} = (\lambda_A, \lambda_\beta) \in \mathbb{R}^2$ by minimising

$$\mathrm{GIC}_{\boldsymbol{\lambda}} = \log(\sigma_{\boldsymbol{\lambda}}) + \frac{a_n}{n} \times (\mathrm{DF}_{\boldsymbol{\lambda}} + 1.028571h^{-1}\mathrm{DF}_{\tilde{\boldsymbol{\lambda}}}), \tag{5.3}$$

where $a_n = \log\{\log(n)\}\log(1.028571h^{-1}d^2 + d)$, $\mathrm{DF}_{\boldsymbol{\lambda}}$ is the number of significant constant parameters, $\mathrm{DF}_{\tilde{\boldsymbol{\lambda}}}$ is the number of the significant functional parameters and σ_λ is the residual sum of squares which is defined as follows,

$$\sigma_\lambda = \frac{1}{(n-p)} \sum_{t=p+1}^{n} \left\{ \mathbf{y}_t - \sum_{j=1}^{p} \hat{A}_{j(\lambda_A)}(\mathbf{y}_{t-l}\hat{\boldsymbol{\beta}}_{\lambda_\beta})\mathbf{y}_{t-j} - \mathbf{X}_t\hat{\alpha} \right\}^2.$$

Then, we denote the optimal tuning parameter $\boldsymbol{\lambda} = (\lambda_A, \lambda_\beta) \in \mathbb{R}^2$ by $\hat{\boldsymbol{\lambda}}$, which is determined by

$$\hat{\boldsymbol{\lambda}} = \arg\min_{\boldsymbol{\lambda}} \mathrm{GIC}_{\boldsymbol{\lambda}}.$$

Since in real application, it is challenging to simultaneously work out these two tuning parameters λ_A and λ_β by 5.3, we construct an iterative algorithm to get the optimal tuning parameters.

1. Choose an initial value of λ_β^0, the tuning parameter λ_A is selected according to

   $$\mathrm{GIC}_{\lambda_A} = \log(\sigma_{\lambda^0}) + \frac{\bar{a}_n}{n} \times (DF_{\hat{A}} + 1.028571h^{-1}DF_{\tilde{A}}),$$

   where $\bar{a}_n = \log\{\log(n)\}\log(1.028571h^{-1}d^2)$, $DF_{\hat{A}}$ is the number of significant covariates with constant coefficients, $DF_{\tilde{A}}$ is the number of significant covariates with varying coefficients and σ_{λ^0} is

   $$\sigma_{\lambda^0} = \frac{1}{(n-p)} \sum_{t=p+1}^{n} \left\{ \mathbf{y}_t - \sum_{j=1}^{p} \hat{A}_{j(\hat{\lambda}_A)}(\mathbf{y}_{t-l}\hat{\boldsymbol{\beta}}_{\lambda_\beta^0})\mathbf{y}_{t-j} - \mathbf{X}_t\hat{\alpha} \right\}^2.$$

   The selected tuning parameter $\hat{\lambda}_A$ is obtained as

   $$\hat{\lambda}_A = \arg\min_{\lambda_A} \mathrm{GIC}_{\lambda_A}.$$

2. By updating $\hat{\lambda}_A$, the selected tuning parameter $\hat{\lambda}_\beta$ can be selected by minimising

$$\mathrm{GIC}_{\lambda_\beta} = \log(\sigma_{\lambda^1}) + n^{-1}\log\{\log(n)\}\log d \times DF_{\hat{\beta}},$$

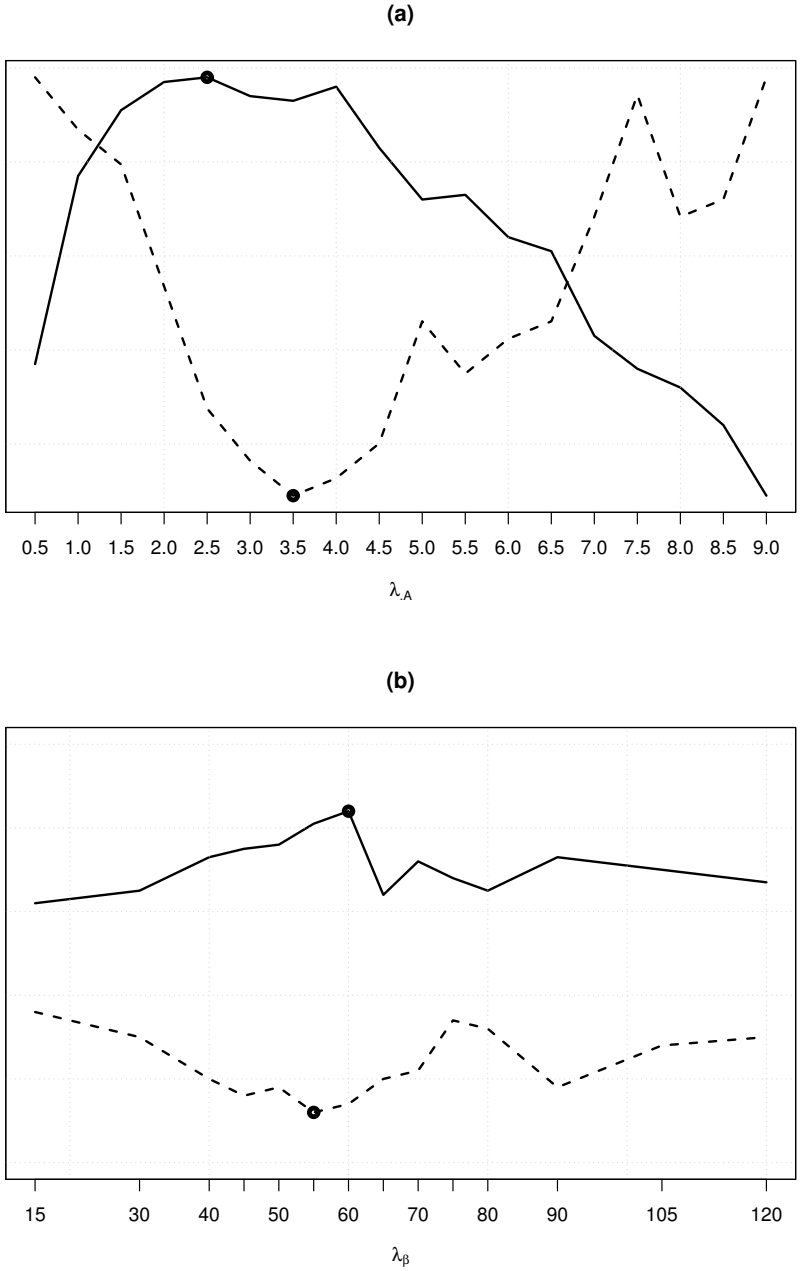
where $DF_{\hat{\beta}}$ is the number of relevant index parameters identified by $\hat{\boldsymbol{\beta}}$ and $\sigma_{\lambda^1}$is

$$\sigma_{\lambda^1} = \frac{1}{(n-p)}\sum_{t=p+1}^{n}\left\{\mathbf{y}_t - \sum_{j=1}^{p}\hat{A}_{j(\hat{\lambda}_A)}(\mathbf{y}_{t-l}\hat{\boldsymbol{\beta}}_{\hat{\lambda}_\beta})\mathbf{y}_{t-j} - \mathbf{X}_t\hat{\alpha}\right\}^2.$$

Replace λ_β^0 in Step 1 by $\hat{\lambda}_\beta$ and keep repeating the two steps above until convergence. Then, we take the converged $\hat{\lambda}_A$ and $\hat{\lambda}_\beta$ as the selected optimal tuning parameters.

Now, we employ a simulation study to illustrate the goodness of GIC and show that the sensitivity to the choice of λ_A and λ_β are different. The performance of GIC is assessed in terms of "Correct Rate". We first fix $\lambda_\beta = 70$ and bandwidth $H = 0.27$ in a model with dimension $d = 3$ to examine the performance of GIC_{λ_A}, and then we conduct another simulation by fixing $\lambda_A = 2.5$ to test the performance of $\mathrm{GIC}_{\lambda_\beta}$. We intuitively present the simulated results in Figure 5.5 and concretely reported the results in Table 5.7 - Table 5.10.

Figure 5.5: GIC with respect to different tuning parameters



NOTE: *The solid curve depicts the Correct Rate and the dashed curve indicates the GIC with respect to different tuning parameters λ_A in (a) and the GIC with respect to a sequence of λ_β in (b), respectively.*

Table 5.7: The performance of GIC_{λ_A} from the simulation on the model with $d = 3$

λ_A	Correct	Overfitting	Underfitting	Others
0.5	0.897	0.103	0.000	0.000
1.0	0.937	0.063	0.000	0.000
1.5	0.951	0.049	0.000	0.000
2.0	0.957	0.043	0.000	0.000
2.5	0.958	0.039	0.003	0.000
3.0	0.954	0.020	0.026	0.000
3.5	0.953	0.019	0.028	0.000
4.0	0.956	0.012	0.032	0.000
4.5	0.943	0.012	0.045	0.000
5.0	0.932	0.016	0.052	0.000
5.5	0.933	0.014	0.053	0.000
6.0	0.924	0.012	0.064	0.000
6.5	0.921	0.012	0.067	0.000
7.0	0.903	0.011	0.086	0.000
7.5	0.896	0.010	0.094	0.000
8.0	0.892	0.010	0.098	0.000
8.5	0.884	0.008	0.107	0.001
9.0	0.869	0.007	0.124	0.000

NOTE: *The column labeled by "Correct", "Under-fitted", "Over-fitted" and "Others" refers to the ratio of correct models, ratio of under-fitted models, ratio of over-fitted models and ratio of other models, respectively.*

Table 5.8: The performance of GIC_{λ_A} and BIC_{λ_A} on the model with $d = 3$

λ_A	Correct Rate	GIC_{λ_A}	BIC_{λ_A}
0.5	0.897	0.160	0.156
1.0	0.937	0.157	0.152
1.5	0.951	0.155	0.151
2.0	0.957	0.148	0.142
2.5	0.958	0.141	0.136
3.0	0.954	0.138	0.134
3.5	0.953	0.136	0.131
4.0	0.956	0.137	0.133
4.5	0.943	0.139	0.134
5.0	0.932	0.146	0.141
5.5	0.933	0.143	0.138
6.0	0.924	0.145	0.140
6.5	0.921	0.146	0.141
7.0	0.903	0.152	0.147
7.5	0.896	0.159	0.154
8.0	0.892	0.152	0.148
8.5	0.884	0.153	0.148
9.0	0.869	0.160	0.155

Table 5.9: The performance of $\text{GIC}_{\lambda_\beta}$ from the simulation on the model with $d = 3$

λ_β	Correct	Overfitting	Underfitting	Others
15	0.942	0.058	0.000	0.000
30	0.945	0.055	0.000	0.000
40	0.953	0.047	0.000	0.000
45	0.955	0.045	0.000	0.000
50	0.956	0.044	0.000	0.000
55	0.961	0.038	0.001	0.000
60	0.964	0.034	0.002	0.000
65	0.944	0.054	0.002	0.000
70	0.958	0.039	0.003	0.000
75	0.948	0.051	0.001	0.000
80	0.945	0.052	0.003	0.000
90	0.953	0.045	0.002	0.000
105	0.950	0.047	0.003	0.000
120	0.947	0.048	0.005	0.000

NOTE: *The column labeled by "Correct", "Under-fitted", "Over-fitted" and "Others" refers to the ratio of correct models, ratio of under-fitted models, ratio of over-fitted models and ratio of other models, respectively.*

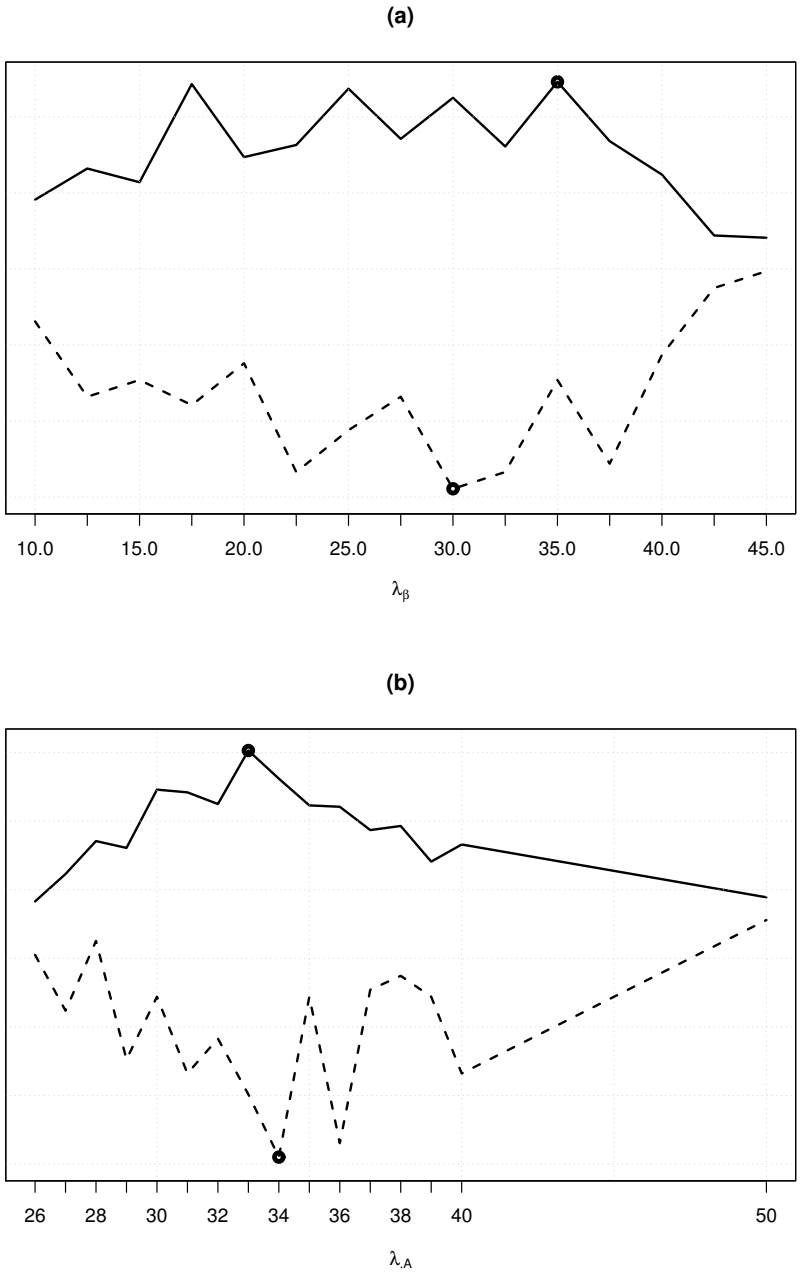
Table 5.10: The performance of $\mathrm{GIC}_{\lambda_\beta}$ and $\mathrm{BIC}_{\lambda_\beta}$ on the model with $d = 3$

	Correct Rate	$\mathrm{GIC}_{\lambda_\beta}$	$\mathrm{BIC}_{\lambda_\beta}$
15	0.942	0.148	0.144
30	0.945	0.145	0.140
40	0.953	0.140	0.135
45	0.955	0.138	0.134
50	0.956	0.139	0.134
55	0.961	0.136	0.132
60	0.964	0.137	0.133
65	0.944	0.140	0.135
70	0.958	0.141	0.136
75	0.948	0.147	0.141
80	0.945	0.146	0.141
90	0.953	0.139	0.137
105	0.950	0.144	0.139
120	0.947	0.145	0.140

Furthermore, we would like to examine the performance of this metric in higher dimensionality and thereby extend the dimension of the model to $d = 10$. By fixing $\lambda_A = 30$ and bandwidth $H = 0.27$, we conduct a simulation to describe the performance of $\mathrm{GIC}_{\lambda_\beta}$. Analogically, the detailed simulation is executed with sample size $n = 1000$ in a total of 300 replications. The simulated results are reported in Table 5.11 and Table 5.12.

Then, to illustrate the accuracy of GIC_{λ_A}, we fix $\lambda_\beta = 35$ and bandwidth $H = 0.27$, and then conduct a simulation, whose results are reported in Table 5.13 and Table 5.14. The simulated results are visually depicted in Plot (a) and Plot (b) respectively in Figure 5.6.

Figure 5.6: GIC with respect to different tuning parameters on the model with dimension $d = 10$



NOTE: *The solid curve depicts the Correct Rate and the dashed curve indicates the GIC concerning different tuning parameters λ_β in (a) and GIC with respect to a sequence of λ_A in (b), respectively.*

Table 5.11: The performance of $\mathrm{GIC}_{\lambda_\beta}$ on the model with $d = 10$

λ_β	Correct	Overfitting	Underfitting	Others
10.0	0.691	0.309	0.000	0.000
12.5	0.732	0.268	0.000	0.000
15.0	0.714	0.286	0.000	0.000
17.5	0.843	0.157	0.000	0.000
20.0	0.747	0.221	0.032	0.000
22.5	0.763	0.196	0.041	0.000
25.0	0.837	0.163	0.000	0.000
27.5	0.771	0.247	0.043	0.000
30.0	0.825	0.128	0.047	0.000
32.5	0.761	0.195	0.044	0.000
35.0	0.846	0.099	0.055	0.000
37.5	0.768	0.160	0.072	0.000
40.0	0.724	0.223	0.053	0.000
42.5	0.644	0.253	0.103	0.000
45.0	0.641	0.252	0.097	0.000

NOTE: *The column labeled by "Correct", "Under-fitted", "Over-fitted" and "Others" refers to the ratio of correct models, ratio of under-fitted models, ratio of over-fitted models and ratio of other models, respectively.*

Table 5.12: The performance of $\mathrm{GIC}_{\lambda_\beta}$ and $\mathrm{BIC}_{\lambda_\beta}$ on the model with $d = 10$

	Correct Rate	$\mathrm{GIC}_{\lambda_\beta}$	$\mathrm{BIC}_{\lambda_\beta}$
10.0	0.691	0.912	0.698
12.5	0.732	0.824	0.667
15.0	0.714	0.848	0.723
17.5	0.843	0.814	0.657
20.0	0.747	0.861	0.679
22.5	0.763	0.736	0.579
25.0	0.837	0.781	0.626
27.5	0.771	0.821	0.683
30.0	0.825	0.713	0.542
32.5	0.761	0.736	0.625
35.0	0.846	0.842	0.654
37.5	0.768	0.745	0.571
40.0	0.724	0.874	0.665
42.5	0.644	0.951	0.756
45.0	0.641	0.974	0.793

Table 5.13: The performance of GIC_{λ_A} on the model with $d = 10$

λ_A	Correct	Overfitting	Underfitting	Others
26	0.683	0.317	0.000	0.000
27	0.723	0.263	0.014	0.000
28	0.771	0.214	0.015	0.000
29	0.761	0.168	0.071	0.000
30	0.846	0.099	0.055	0.000
31	0.842	0.142	0.016	0.000
32	0.825	0.121	0.054	0.000
33	0.903	0.081	0.016	0.000
34	0.862	0.043	0.095	0.000
35	0.823	0.063	0.114	0.000
36	0.821	0.057	0.122	0.000
37	0.787	0.067	0.146	0.000
38	0.793	0.063	0.144	0.000
39	0.741	0.082	0.177	0.000
40	0.766	0.031	0.203	0.000
50	0.689	0.042	0.269	0.000

NOTE: *The column labeled by "Correct", "Under-fitted", "Over-fitted" and "Others" refers to the ratio of correct models, ratio of under-fitted models, ratio of over-fitted models and ratio of other models, respectively.*

Table 5.14: The performance of GIC_{λ_A} and BIC_{λ_A} on the model with $d = 10$

λ_A	Correct Rate	GIC_{λ_A}	GIC_{λ_A}
26	0.683	0.901	0.689
27	0.723	0.825	0.654
28	0.771	0.928	0.743
29	0.761	0.754	0.588
30	0.846	0.841	0.654
31	0.842	0.733	0.607
32	0.825	0.783	0.621
33	0.903	0.706	0.518
34	0.862	0.612	0.474
35	0.823	0.842	0.656
36	0.821	0.635	0.489
37	0.787	0.856	0.663
38	0.793	0.872	0.715
39	0.741	0.843	0.729
40	0.766	0.736	0.572
50	0.689	0.951	0.732

From the numeric results given above, it can be seen that GIC is able to precisely and consistently identify the optimal tuning parameters. Moreover, from Figure 5.5 and Figure 5.6, we notice that this criterion can even describe the accurate pattern of the change of corresponding correct rates to a sequence of tuning parameters. All the simulated results corroborate that the GIC works quite well in choosing the tuning

parameters of the index and varying coefficients from both modest dimensional and relative high dimensional models.

CHAPTER 6

SIMULATION STUDY

In previous chapters, we use the model (4.52) for all the numerical analysis, to avoid over specified, thus, it is reasonable to conduct simulations on another example. Therefore, in this section, we use a new simulation example to verify the effective and accuracy of the proposed iterative algorithm. For the choice of hyper-parameters, we shall follow the idea in Chapter 5.1 that the bandwidth is selected in terms of $H = 0.6(d/n)^{0.2}$ and the tuning parameters are determined by the iterative GIC criterion we described in Section 5.2.

We now consider the following example of SSIVARM,

$$\mathbf{y}_t = \mathbf{A}_1(z_{t-l})\mathbf{y}_{t-1} + \mathbf{A}_2(z_{t-l})\mathbf{y}_{t-2} + \mathbf{X}_t\boldsymbol{\alpha} + \boldsymbol{\epsilon}_t, \tag{6.1}$$

where $z_{t-l} = \mathbf{y}_{t-l}^{\top}\boldsymbol{\beta}$, $\mathbf{y}_t, t = 1, \cdots, n$ is a $d-$ dimensional stationary time series with 2 lags, $\mathbf{X}_t, t = 1, \cdots, n$ is a $d \times 3$ matrix gen-

erated from $d-$ dimensional Gaussian distribution, the $\boldsymbol{\epsilon}_t$ are independently generated from the normal distribution with $E(\boldsymbol{\epsilon}_t|\mathscr{F}_t) = \mathbf{0}_d$ and $\text{cov}(\boldsymbol{\epsilon}_t|\mathscr{F}_t) = \sigma^2\mathbf{I}_d$, with $\mathscr{F}_t = \{(Y_{l-1}, \mathbf{X}_l) : l \leq t\}$. We set l=1, $\sigma = \frac{1}{3}$, $\boldsymbol{\alpha} = (1,2,2)^\top$, $\boldsymbol{\beta} = (0.6, 0.8, 0, \cdots, 0)^\top$,

$$\mathbf{A}_1(z_{t-l}) = \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0.25(\cos(\pi z_{t-l}) + 0.5) & \exp(-z_{t-l}^2) & 0 & \cdots & 0 \\ 0 & 0 & 0.8\exp(-z_{t-l}^2) & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix},$$

$$\mathbf{A}_1(z_{t-l}) = \begin{pmatrix} 0.75 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0.5\sin(\pi z_{t-l}) + 0.3 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}.$$

In a similar way to the simulation studies in Chapter 5, we consider the models in the dimension of $d = 3$ and $d = 10$, respectively. The first two predictors should be generated independently from a normal distribution, and then we generate $100 + n$ observations followed by the process (6.1). Hence, we will discard the first 100 predictors and let $\mathbf{y}_t, t = 1, \cdots, n$ be the remaining predictors we have. For the lower dimensional model, we conduct the simulation with sample size $n = 800$,

over a total of 1000 replications. Since the SSIVARM (6.1) has a very fancy structure, in order to have a reasonably good simulated results in higher dimensionality $d = 10$, we will increase the sample size to $n = 1000$ to give more information to fit the model.

Analogically, we will measure the accuracy of the proposed approach on model selection and estimation. To evaluate the performance of model selection, we report the rate of correct model, Under-fitting model, Over-fitting model and other models. In one simulation, once the resulting model simultaneously detects the true model and identifies the modelling structure correctly, we classify it as a "correct model"; when the estimated model deletes at least one important variables but does not include any irrelevant variables, we classify it as an "under-fitting model"; when the estimated model includes at least one insignificant variables but does not miss any relevant variables, it is regarded as an "over-fitting model". The "other models" represent that the estimated model not only includes the insignificant variables but also ignores significant variables.

To evaluate the estimation accuracy of the proposed estimate, in this section, we use median MSE_{β_m} with respect to each non-zero index parameter and $\mathrm{RMISE}_{a_{jıJ}}$ with respect to each non-zero coefficient parameter. Concretely, the MSE_{β_m} is calculated in component-wise manner, which can be defined

as

$$\text{MSE}_{\beta_m} = (\hat{\beta}_m^l - \beta_m)^2, \tag{6.2}$$

where $\hat{\beta}_m^{(l)}, m = 1, 2, \ldots, d, l = 1, 2, \ldots, L$, is the estimator of the $m-$th index parameter from the $l-$th iteration, which leads to a median MSE_{β_m} from all the MSEs_{β_m} from L replications; the median $\text{RMISE}_{a_{j\iota J}}$ for the estimates of each relevant coefficient, which is approximated as follows

$$\text{RMISE}_{a_{j\iota J}} \approx [\frac{\sum_{k=1}^{n-l}(\hat{a}_{j\iota J}^l(z_k) - a_{j\iota J}(z_k))^2}{\sum_{k=1}^{n-l}(z_k)^2}], \tag{6.3}$$

where $\hat{a}_{j\iota J}^l(z_k)$, is the estimator of the function at the ι entry of the matrix A_j, in the l-th replication, which leads to a median $\text{RMISE}_{a_{j\iota J}}$ from all the $\text{RMISE}_{a_{j\iota J}}$ from L replications, and $z_k = \mathbf{y}_k^\top \hat{\boldsymbol{\beta}}, k = 1, \ldots, n-l$. The MSE of the unknown parameter α will also be taken into consideration. Additionally, introducing a benchmark is necessary for evaluating the accuracy of the estimation. We also use the "oracle estimators" as the benchmark and hence, we calculate the $\text{RMISE}_{a_{j\iota J}}$ and MSE_{β_m} of oracle estimators as well. We report the simulated results of model selection and estimation in Table 6.1.

We can see from Table 6.1 that, in all cases, the percentage of the correctly selected models is no less than 86%, which can verify the accuracy of the proposed method on selecting the

Table 6.1: The simulation results of model selection and estimation

	Oracle	d=3	d=10
Correct Rate	1.000	0.936	0.862
Over-fitting Rate	0.000	0.010	0.037
Under-fitting Rate	0.000	0.019	0.049
Rate of Other Models	0.000	0.035	0.052
MSE_{β_1}	0.0027	0.0087	0.0288
MSE_{β_2}	0.0018	0.0218	0.0511
MSE_{α}	0.0021	0.0027	0.0025
$\text{RMISE}_{a_{122}}$	0.6986	0.7149	0.7569
$\text{RMISE}_{a_{123}}$	0.5847	0.6489	0.7341
$\text{RMISE}_{a_{133}}$	0.5416	0.6700	0.7615
$\text{RMSE}_{c_{211}}$	0.0002	0.0053	0.0336
$\text{RMISE}_{a_{222}}$	0.6075	0.6160	0.6318

NOTE: *The columns labeled by* $d = 3$ *and* $d = 10$ *represent the estimators from the proposed method in the 3-dimensional models and 10-dimensional models, respectively. the rows labeled by "oracle" depict the oracle estimators. The coefficient* c_{211} *is assume to be a constant and hence it is measured by* $MSE_{c_{211}}$*is.*

true model. Additionally, the fact that the ratio of correctly fitted models increases with the decrease of dimension also makes sense. Moreover, all the values of MSE and RMISE are reasonably small, and gradually become smaller with the decrease of the dimension of models. Additionally, the oracle

estimates are always more accurate. Therefore, we conclude that our proposed method can simultaneously select the true model correctly and estimate the model precisely.

CHAPTER 7

REAL DATA ANALYSIS

We next apply the proposed sparse single-index vector autoregressive model to analyse and forecast the change of house price in ten major metropolitan areas in the United States. The data is chosen from the SP CoreLogic Case-Shiller Home Price NSA Index, which measures the average change in the value of the residential real estate in a specific city given a constant level of quality. For instance, the change of house price of Los Angles are in terms of the corresponding SP CoreLogic Case-Shiller Los Angeles Home Price NSA Index. Meanwhile, the choice of the ten metropolitan areas are determined by the cities listed in a Composite Home Price Index called " S&P CoreLogic Case-Shiller 10-City Composite Home Price NSA Index". This data set was collected monthly across thirty years from January 1987 to January 2017, which leads to the

sample size $n = 361$ and all the variables in this dataset are seasonally adjusted. These ten metropolitan areas are:

1. Los Angeles-Long Beach-Santa Ana, CA Metropolitan Statistical Area (coded by y_{t1}),

2. Las Vegas-Paradise, NV Metropolitan Statistical Area (coded by y_{t2}),

3. San Diego-Carlsbad-San Marcos, CA Metropolitan Statistical Area (coded by y_{t3}),

4. San Francisco-Oakland-Fremont, CA Metropolitan Statistical Area (coded by y_{t4}),

5. Denver-Aurora, CO Metropolitan Statistical Area (coded by y_{t5}),

6. New York City Area (coded by y_{t6}),

7. Miami-Fort Lauderdale-Pompano Beach, FL Metropolitan Statistical Area (coded by y_{t7}),

8. Washington-Arlington-Alexandria, DC-VA-MD-WV Metropolitan Statistical Area (coded by y_{t8}),

9. Boston-Cambridge-Quincy, MA-NH Metropolitan Statistical Area (coded by y_{t9}),

10. Chicago-Naperville-Joliet, IL Metropolitan Division (coded by y_{t10}).

Before the modelling, we make a transformation of these variables that is taking the second difference of logarithms in order to construct models on stationary time series, which reduces the sample size to $n = 359$. Furthermore, we standardise the data such that they have the sample mean 0 and sample covariance matrix I_{10}.

In this section, the focus is to forecast the home price growth of these ten areas at the state level, that is, in the prediction of the home price change in a target area, the effect from all the other areas will be taken into consideration. Moreover, we are also interested in explicitly revealing which area contribute significant effects on the forecasting of a target area, and whether their impacts vary over the change of a national home price index (coded by $z_{t-l} = \mathbf{y}_{t-l}^{\top}\boldsymbol{\beta}, \mathbf{y_{t-l}} = (y_{t-l,1}, y_{t-l,2}, \ldots, y_{t-l,10})^{\top}$), which is a linear combination of the index parameter (coded by $\boldsymbol{\beta}$) and all the collected predictors. Our study leads to a relative high-dimensional problem and standard regression techniques often fail to estimate it. Typically, before building a time-series model on sequential observations, we shall first specify the number of lags in the model, which are commonly determined by AIC or BIC. In our real data analysis, considering the trade-

off between computational cost and accuracy of prediction, we plan to simplify this problem by assuming the number of lags to be 1. Hence, to realise the object, we employ our proposed forecast model

$$\mathbf{y}_t = \sum_{j=1}^{p} \mathbf{A}_j(Z_{t-l})\mathbf{y}_{t-j} + \epsilon_{\mathbf{t}},$$

where we set $l = 1$. As our aim is to identify the significant predictors to a target area and only a handful of lags will be necessary for prediction, the regression result should be sparse. We apply our proposed method to obtain the penalised estimators of varying coefficients $\mathbf{A}_j(\cdot)$ and the index parameters $\boldsymbol{\beta}$, respectively. The resulting penalised estimators suggest that the important index parameters are $\boldsymbol{\beta} = (\beta_1, \beta_3, \beta_4, \beta_5, \beta_8)^\top$, which includes the metropolitan area of Los Angeles, San Diego, San Francisco, Denver and Washington. Meanwhile, by our proposed method, we have selected the true model as follows,

$$\begin{pmatrix} y_{t,2} \\ y_{t,3} \\ y_{t,5} \\ y_{t,6} \\ y_{t,7} \\ y_{t,8} \end{pmatrix} = \begin{pmatrix} A_{(1)22}y_{t-1,2} \\ A_{(1)33}y_{t-1,3} + A_{(1)39}y_{t-1,9} \\ A_{(1)54}y_{t-1,4} \\ 0 \\ A_{(1)77}y_{t-1,7} \\ A_{(1)84}y_{t-1,4} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ A_{(2)61}y_{t-2,1} \\ A_{(2)7,10}y_{t-2,10} \\ A_{(2)8,10}y_{t-2,10} \end{pmatrix} + \begin{pmatrix} \epsilon_{t,2} \\ \epsilon_{t,3} \\ \epsilon_{t,5} \\ \epsilon_{t,6} \\ \epsilon_{t,7} \\ \epsilon_{t,8} \end{pmatrix}.$$

It can be seen that the autoregressions of metropolitan areas of Las Vegas, San Diego, Denver, New York City, Miami and Washington are captured by the sparse single-index VAR model and concretely, our submodel also detects the cross-dependence of these mojor areas across the country. Concretely, it indicates that the home price in metropolitan area of San Diego has statistical correlation with Boston; home price in metropolitan area of Denver has statistical correlation with San Francisco; home price in metropolitan area of New York City has statistical correlation with Los Angeles; home price in metropolitan area of Miami has correlation with Chicago and home price in metropolitan area of Washington has correlation with Chicago and San Francisco.

Then, we introduce a similar estimation procedure without penalised approaches to estimate the specified model. By applying the proposed estimation procedure, we obtain the estimators of index parameters, which are

$$\begin{aligned} \boldsymbol{\beta} &= (\beta_1, \beta_3, \beta_4, \beta_5, \beta_8)^\top \\ &= (0.3820, 0.3869, 0.1764, 0.8077, 0.1371)^\top. \end{aligned}$$

From an intuitive evaluation, we find that the estimated curves are not smooth enough, thus, we employ a second-step local linear regression as a modifiction to generate better estimated cures. We provide in Figure 7.1 and Figure 7.2 these estimated curves from our purposed method and a second-step

local linear regression.

Figure 7.1: Estimated curves of varying coefficients in $\mathbf{A}_1(\cdot)$

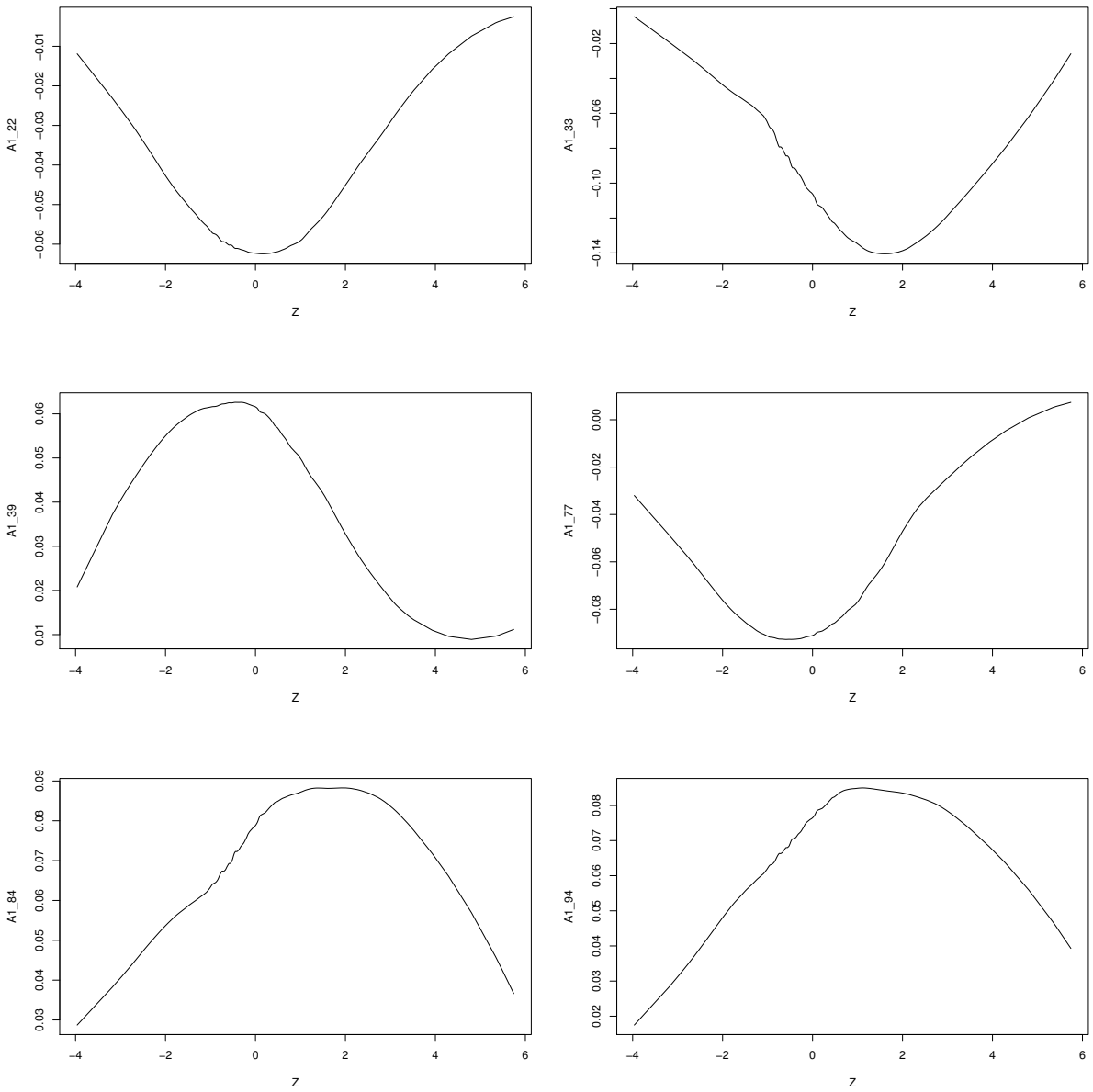
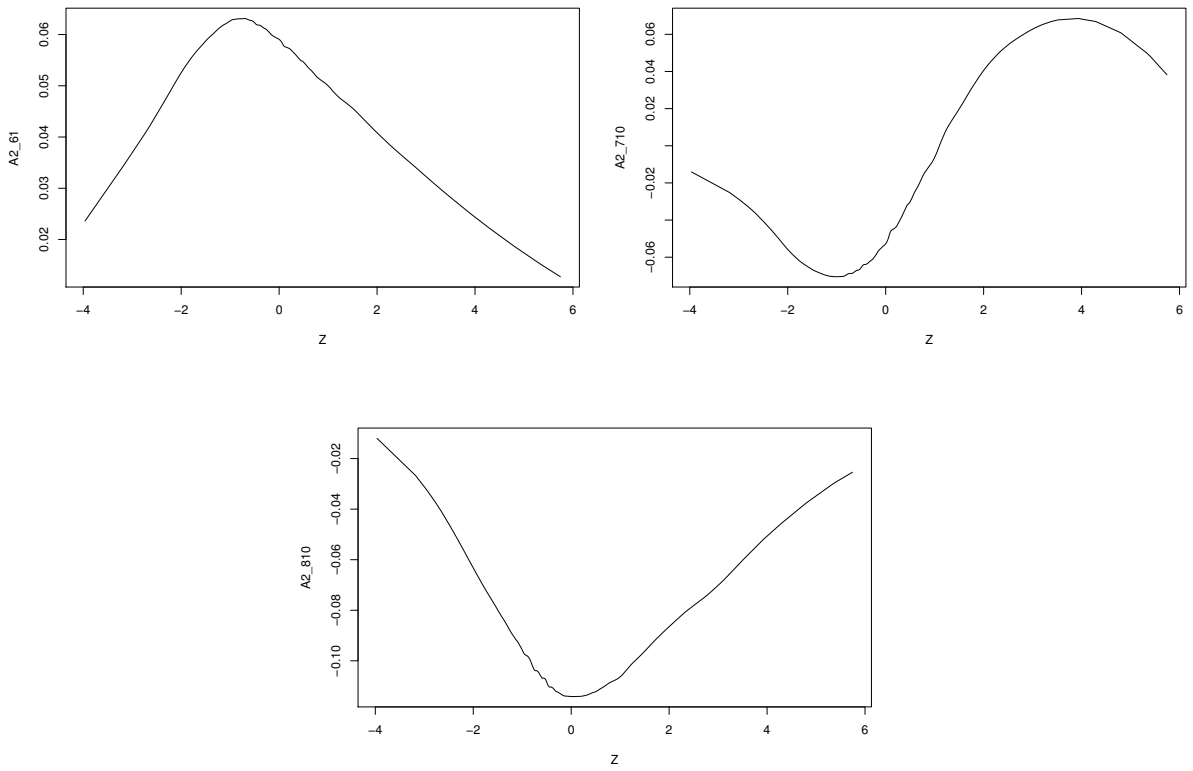


Figure 7.2: Estimated curves of varying coefficients in $\mathbf{A}_2(\cdot)$



As we can see from Figure 7.1 and Figure 7.2 , the estimated coefficients are unlikely to be null or other constants, and they all vary over the range of national home price index.

Moreover, we would like to further analyse our estimates by examining whether each of the resulting residuals $\{\epsilon_{t,2}, \epsilon_{t,3}, \epsilon_{t,5}, \epsilon_{t,6}, \epsilon_{t,7}, \epsilon_{t,8}\}$, $t = 1, \ldots, n$, is white noise. Then, we plot the estimated residuals ϵ_2 in figure 7.3.

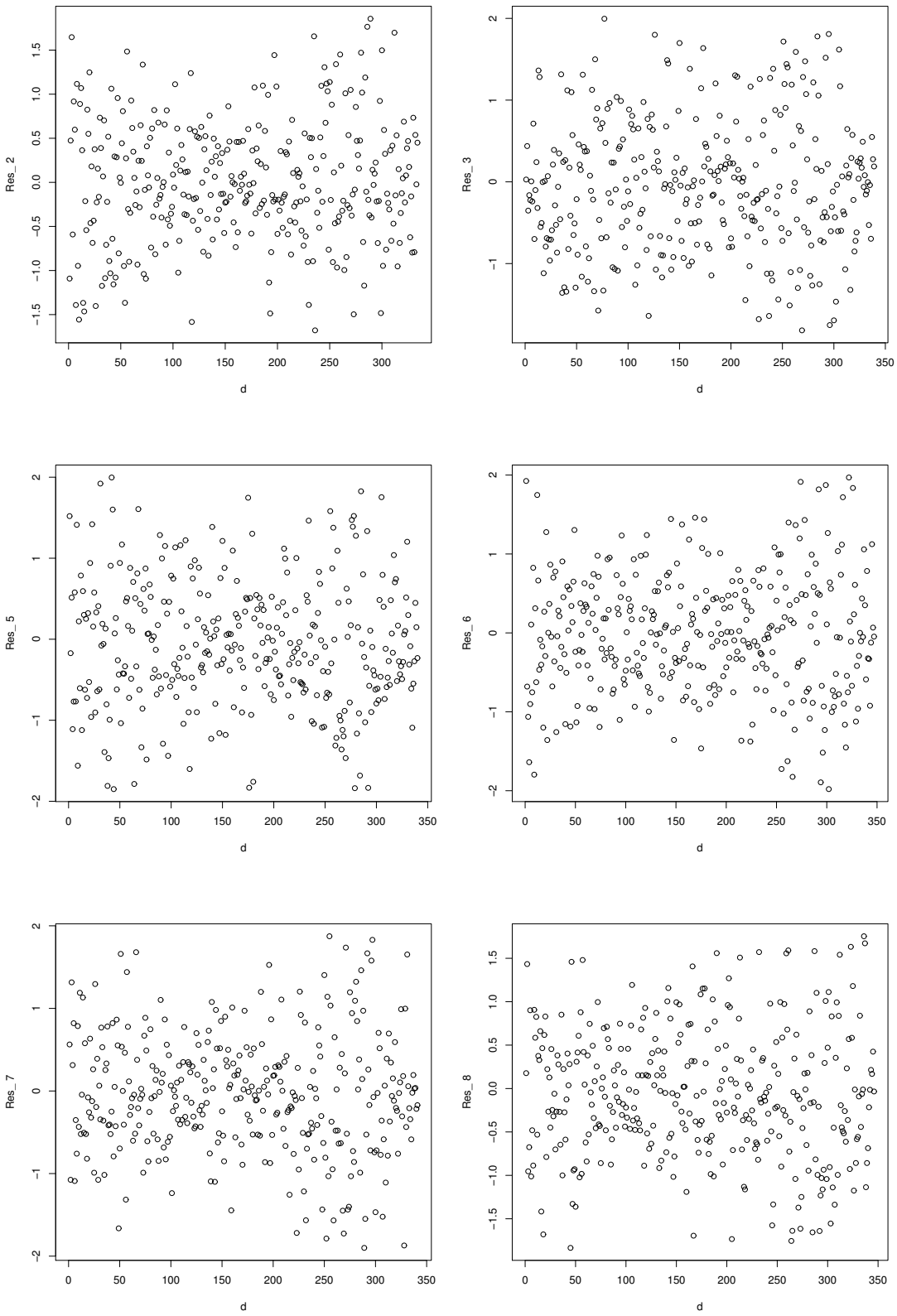
As we can see in Figure 7.3, there is no obvious tendency, which also corroborates the purposed selection and estimation methods very well.

In addition, we would like to evaluate the forecasting performance of the selected model and have a further comparison with the ARIMA model. For either model, we use the first 349 observations as the training data to estimate the conditional expectation of the 350th observation. Then we repeat this one-step forward prediction by adding one more observation into the training set at a time. Finally, we end with using the first 358th observations to forecast the 359th observation. We use the mean squared prediction error (MSPE) as a measure of overall performance which is defined as follows:

$$\text{MSPE} = \frac{1}{\tau \hat{d}} \sum_{s=1}^{\tau} \|\hat{\mathbf{y}}_{T+s} - \mathbf{y}_{T+s}\|^2,$$

where $\tau = 10$, $\hat{d} = 6$ and $T = 349$. The MSPE of the selected model is 0.1925 while the MSPE of the ARIMA model is 0.2645. Hence, the selected model from our proposed method performs better in terms of prediction error than the ARIMA model.

Figure 7.3: Residuals



CHAPTER 8

CONCLUSION AND FUTURE WORK

We proposed in this thesis a novel method for model selection and nonparametric estimation in sparse single-index varying coefficient VAR model via local linear smoothing and penalised least squares. Based on the idea of generalised information criterion, a specified metric to determine the involved tuning parameters (regularisation parameters) has been established. With the properly selected regularisation parameters, we have shown that the proposed estimates perform as good as the oracle estimates by solid numerical evidence. Moreover, as a side-product from the exploration of the proposed algorithm, we also developed an efficient method to fit the single-index varying coefficient VAR model without sparsity, which can be applied to calculate the oracle estimators. Additionally, the accuracy of the estimation from the proposed

algorithm is backed up by intensive simulation studies.

After concluding the thesis, we would like to discuss some interesting extensions for future study. First, the algorithm used for minimising penalised least squares is the local quadratic approximation, a more recent minimisation approach like the local linear approximation, which has proven to be a better solver, can be employed in the future work. Secondly, the proposed model selection and estimation of a semiparametric model inevitably includes two hyper-parameters: the smoothing parameter and the regularisation parameter. In our current work, for the purpose of simplicity, we select those two class of parameters separately. Such a simplification makes our method easier to implement and more computational efficiency, though possibly not optimal. How to jointly tune both hyper-parameters is another interesting topic open for the future work.

BIBLIOGRAPHY

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. *Proc. 2nd Int. Symp. Information Theory*. Budapest: Akademiai Kiado.

Antoniadis, A. and Fan, J. (2001) Regularization of Wavelet Approximations. *Journal of the American Statistical Association*, **96**, 939-967

Bickel, P. J., Ritov, Y. and Tsybakov, A. (2009). Simultaneous analysis of Lasso and Dantzig selector. *The Annals of Statistics*, **37**, 1705-1732.

Boyd, S. and Vandenberghe, L (2004). Convex optimization. Cambridge University Press.

Breiman, L. (1995). Better subset regression using the nonnegative garrote. *Technometrics*, **37**, 373-384.

Breiman, L. (1996). Heuristics of instability and stabilization in model selection. *The Annals of Statistics*, **24**, 2350-2383.

Cai, Z., Fan, J. and Yao, Q. (2000). Functional-coefficient regression models for nonlinear time series. *Journal of the American Statistical Association*, **95**, 941-956.

Cai, Z., Fan, J. and Li, R. (2000). Efficient estimation and inferences for varying-coefficient models. *Journal of the American Statistical Association*, **95**, 888-902.

Carroll, R. J., Ruppert, D. and Welsh, A. H. (1998). Local estimating equations. *Journal of the American Statistical Association*, **93**, 214-227.

Cheng, M.-Y., Zhang, W. and Chen, L.-H. (2009). Statistical estimation in generalized multiparameter likelihood models. *Journal of the American Statistical Association*, **104**, 1179–1191.

Cheng, M.-Y., Honda, T., Li, J., and Peng, H. (2014). Nonparametric independence screening and structure identification for ultra high dimensional longitudinal data. *The Annals of Statistics*, **42**, 1819–1849.

Chen, R. and Tsay, R. S. (1993). Nonlinear Additive ARX Models. *Journal of the American Statistical Association*, **88**, 955-967.

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, **32**, 407-499.

Fan, J. and Gijbels, I. (1996). Local Polynomial Modeling and Its Applications, London: Chapman & Hall.

Fan, J. and Zhang, W. (1999). Statistical estimation in varying coefficient models. *The Annals of Statistics*, **27**, 1491-1518.

Fan, J. and Zhang, W. (2000). Simultaneous confidence bands and hypothesis testing in varying-coefficient models. *Scandinavian Journal of Statistics* , **27**, 715-731.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, **96**, 1348-1360.

Fan, J., Zhang, C. and Zhang, J. (2001). Generalized Likelihood Ratio Statistics and Wilks Phenomenon. *The Annals of Statistics*, **29**, 153-193.

Fan, J., Yao, Q. and Cai, Z. (2003). Adaptive varying-coefficient linear models. *Journal of Royal Statistical Society, Series B*, **65**, 57-80.

Fan, J. and Huang, t. (2003). Profile likelihood inferences on semiparametric varying-coefficient partially linear models. *Bernoulli*, **11**, 1031-1057.

Fan, J. and Li, R. (2004). New estimation and model selection procedures for semiparametric modeling in longitudinal data analysis. *Journal of the American Statistical Association*, **99**, 710-723.

Fan, J. and Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space (with discussion). *Journal of the Royal Statistical Society, Series B*, **70**, 849-911

Fan, J. and Zhang, W. (2008). Statistical methods with varying coefficient models. *Statistics and Its Interface*, **1**, 179–195.

Fan, J., Feng, Y. and Wu, Y. (2009). Network exploration via the adaptive LASSO and SCAD penalties. *The Annals of Applied Statistics*, **3**, 521-541.

Fan, J., Lv, J. and Qi, L., (2011). Sparse high dimensional models in economics. *Annual Review of Economics*, **3**, 291-317.

Fan, Y. and Tang, C. Y. (2013). Tuning parameter selection in high dimensional penalized likelihood. *Journal of Royal Statistical Society, Series B*, **75**, 531-552.

Fan, Y., Kong, Y., Li, D. and Zheng, Z. (2015). Innovated interaction screening for high-dimensional nonlinear classification. *The Annals of Statistics*, **43**, 1243-1272.

Fan, Y. and Lv, J. (2016). Innovated scalable efficient estimation in ultra-large Gaussian graphical models. *The Annals of Statistics*, **44**, 2098-2126.

Fang, X., Li, J., Wong, W. K., and Fu, B. (2014). Detecting the violation of variance homogeneity in mixed models. *Statistical Methods in Medical Research*, DOI: 10.1177/09622802145

Frank, E. and Friedman, J. H. (1993). A Statistical View of Some Chemometrics Regression Tools. *Technometrics*, **35**, 109-135.

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

Hastie, T. J. and Tibshirani, R. J. (1993). Varying-coefficient models. *Journal of Royal Statistical Society, Series B*, **55**, 757-796.

Hoover, D. R., Rice, J. A., Wu, C. O. and Yang, L.-P. (1998). Nonparametric smoothing estimates of time-varying coefficient models with longitudinal data. *Biometrika*, **85**, 809–822.

Huang, J., and Xie, H. (2007). Asymptotic oracle properties of SCAD-penalized least squares estimators. *Lecture Notes-Monograph Series*, 149–166.

Hunter, D. R. and Li, R. (2005). Variable selection using MM algorithms. *The Annals of Statistics*, **33**, 1617-1642.

Kauermann, G. and Tutz, G. (1999). On model diagnostics using varying coefficient models. *Biometrika*, **86**, 119–128.

Kong, E. and Xia, Y. (2006). Variable selection for the single-index model. *Biometrika*, **94**, 217-229.

Lavergne, P. (1998). A Cauchy-Schwarz inequality for expectation of matrices.

Li, R. and Liang, H. (2008). Variable Selection in Semiparametric Regression Modeling. *The Annals of Statistics*, **36**, 261–286.

Liu, J., Li, R. and Wu, R. (2014). Feature selection for varying coefficient models with ultrahigh dimensional covariates.

Journal of the American Statistical Association, **109**, 266–274.

Li, D., Ke, Y. and Zhang, W. (2015). Model selection and structure specification in ultra-high dimensional generalised semi-varying coefficient models. *The Annals of Statistics*, **43**, 2676-2705.

Lin, Y. and Zhang, H. H. (2003). Component selection and smoothing spline analysis of variance models. *The Annals of Statistics*, **34**, 2272-2297.

Nishii, R. (1984). Asymptotic properties of criteria for selection of variables in multiple regression. *The Annals of Statistics*, **12**, 758–765.

Nolan, D. and Pollard, D. (1987). U-processes: Rates of convergence. *The Annals of Statistics*, **15**, 780-799.

Pakes, A. and Pollard, D. (1989). Simulation and the Asymptotics of Optimization Estimators. *Econometrica* **57**, 1027-1057.

Pollard, D. (1984). Convergence of Stochastic Processes. Springer.

Rapach, D. E. and Strauss, J. (2007). Forecasting real housing price growth in the Eighth District states. *Regional Economic Development*, , 11, 33-42.

Ruppert, D. and Wand, M. P. (1994). Multivariate Locally Weighted Least Squares Regression. *The Annals of Statistics*, , 22, 1346-1370.

Shao, J. (1997). An asymptotic theory for linear model selection. *Statistica Sinica*, **7**, 221–264.

Song, R., Yi, F. and Zuo, H. (2012). On varying-coefficient independence screening for high-dimensional varying-coefficient models. *Statistica Sinica*, **24**, 1735–1752.

Stefanski, L. A., Wu, Y., and White, K. (2014). Variable selection in nonparametric classification via measurement error model selection likelihoods. *Journal of the American Statistical Association*, **109**, 574-589.

Tsay, R. S. (2014). Multivariate Time Series Analysis: With R and Financial Applications, John Wiley Sons, Inc., Hoboken, New Jersey.

Wang, L., Peng, B., and Li, R. (2015). A high-dimensional nonparametric multivariate test for mean vector. *Journal of the American Statistical Association*, **110**, 1658-1669.

Wu, C. O., Chiang, C. -T. and Hoover, D. R. (1998). Asymptotic Confidence Regions for Kernel Smoothing of a Varying-Coefficient Model with Longitudinal Data. *Journal of the American Statistical Association*, **93**, 1388-1402.

Xia, Y. and Li, W. K. (1999). On single-index coefficient regression models. *Journal of the American Statistical Association*, **94**, 1275-1285.

Xia, Y. (2006). Asymptotic distributions for two estimators of the single-index model. *Econometric Theory*, **22**, 1112-1137.

Yang, Y. (2005). Can the strengths of aic and bic be shared? A conflict between model identification and regression estimation. *Biometrika*, **92**, 937–950.

Yuan, M and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of Royal Statistical Society, Series B*, **68**, 49–67.

Zhang, C-H. (2007). Penalized linear unbiased selection. *Technical Report*, 2007-003

Zhang, X., Wu, Y., Wang, L., and Li, R. (2016). Variable Selection for Support Vector Machines in Moderately High Dimensions. *Journal of the Royal Statistical Society, Series B*, **78**, 53-76.

Zhang, W. and Lee, s.-Y. (2000). Variable Bandwidth Selection in Varying-Coefficient Models. *Journal of Multivariate Analysis*, **74**, 116-134.

Zhang, Y., Li, R. and Tsai, C-L. (2010). Regularization Parameter Selections via Generalized Information Criterion. *Journal of the American Statistical Association*, **105**, 312–323.

Zou, H. (2006). The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association*, **101**, 1418-1429.

Zou, H. and Li, R. (2008). One-step sparse estimates in nonconcave penalized likelihood models. *The Annals of Statistics*, **36**, 1509-1533.