Transition Models for Longitudinal Data Analysis with Sparse Hierarchical Penalization

Peter Park, Department of Mathematics and Statistics

McGill University, Montreal

Jan, 2021

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of

Master of Statistics

Abstract

Longitudinal data analysis employ repeated measurements of individuals over a period of time. It has been used by statisticians to study the change in outcome measurement associated with the change in exposure conditions. In some scenario, we would want to study which factors are associated in these changes. Variable selection, a machine learning technique that has been used widely for selecting the subset of relevant features, is widely used for this purpose. Furthermore, such technique has been previously extended to exhibit certain properties such as maintaining a predefined structure within the features that are selected. In this work we present the usage of variable selection on longitudinal data and provide the details regarding its application on both simulated and real dataset.

Abrégé

L'analyse des données longitudinales utilise des mesures répétées d'individus sur une période donnée. Il a été utilisé par les statisticiens pour étudier le changement de mesure des résultats associé au changement des conditions d'exposition. Dans certains scénarios, nous voudrions étudier les facteurs associés à ces changements. La sélection de variables, une technique d'apprentissage automatique qui a été largement utilisée pour sélectionner le sous-ensemble de fonctionnalités pertinentes, est largement utilisée à cette fin. En outre, une telle technique a été précédemment étendue pour présenter certaines propriétés telles que le maintien d'une structure prédéfinie dans les caractéristiques qui sont sélectionnées. Dans ce travail, nous présentons l'utilisation de la sélection de variables sur des données longitudinales et fournissons les détails concernant son application sur des ensembles de données simulés et réels.

Acknowledgements

I would first like to thank my parents for ongoing unconditional support in everything I have pursued in my life. Without them, I would not be who I am today. I also would like to thank my older brothers John and Jake who encouraged me to further my academic pursuit.

I direct my gratitude and appreciation to the staff members and professors of Department of Mathematics and Statitics, who have taught me invaluable knowledge and academic integrity.

I would also like to thank my friends at McGill who I have spent great time interacting and learning from them.

Lastly, I would like to thank my girlfriend Wendy Zhang. She has been an inspirational and a supportive individual who had a significant influence in my life. Much of what I have accomplished during the current chapter of my life were only possible due to her support. Because of this, I cannot be more grateful.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The world's population is aging, with the total of 962 million population aged 60 years or over in 2017 and this number is expected to double again by 2050. Dementia is one of common diseases and major causes of disability and dependency among older people worldwide. According to World Health Organization, around 50 million people are suffering from dementia and nearly 10 million new cases every year in the worldwide. It is estimated that over 130 million people will live with dementia by 2050. Studies have proved that depression is a common comorbidity in dementia. Patients with dementia, especially comorbid with depression, use more than 70% of health services and 50% of care organization than their age matched controls [Gutterman et al., 1999]. With the rising tide of dementia, the economic and social impact of dementia is likely to increase. It is critical to understand the health service use of dementia patients with psychiatric comorbidity may help to establish a framework for considering change in the current system of care.

Despite that many epidemiological studies, largely cross-sectional or short-term longitudinal studies, have suggested that the comorbid psychiatric diagnosis and dementia predict more health care utilizations including medical and psychiatric inpatient days of care and outpatient visits [Kunik et al., 2003]. There has been no long-term longitudinal study conducted to assess the degree to which psychiatric comorbidity in patients

with dementia is associated with health care utilization. Furthermore, there is even less research conducted to portrait the potential population characteristics associated with increased utilizations of health care services. It is critical to understand the trend of long-term health care utilizations for the most vulnerable populations suffering depression and dementia.

Longitudinal data analysis has been widely used in medical and quantitative psychology research. In the longitudinal data, observations are usually collected at multiple follow-up times. For example, in clinical trial settings, the data often collected after treatments or exposures are administered for an experimental design to study whether outcome (response) changes according to exposure of some event. There are two groups of subjects, one is given placebo while the other is given active drug, these two groups can be compared over multiple time points to observe whether there is a noticeable difference between the outcome of the control and the treated group. Most common techniques for analyzing longitudinal data include random effect models, marginal models, and transition models.

Denote $Y_{i,t}$ as the random variable corresponding to response for individual i at timepoint t. Let $\mathbf{x}_{i,t} = (x_{i,t,1}, \ldots, x_{i,t,p})^\top$ be the corresponding p-dimensional vector of covariates and $\boldsymbol{\beta}^* = (\beta_1^*, \ldots, \beta_p^*)^\top$ be the true coefficients. Marginal models [Liang and Zeger, 1986] assumes that the marginal expectation of response $Y_{i,t}$ is a function of $\mathbf{x}_{i,t}$

$$g(E(Y_{i,t}|\mathbf{x}_{i,t})) = \mathbf{x}_{i,t}^\top \boldsymbol{\beta}^*,$$

where g is a link function. The marginal variance of $Y_{i,t}$ is assumed to be dependent on the marginal mean through $\text{Var}(Y_{i,t}) = \theta v(\mu_{i,t})$, where $v(\mu_{i,t})$ is known as a variance function and θ is known as a scale parameter. Given the effects of covariates on the average probability of response, this model is often used for "population-averaged" interpretation.

Random effect models [Stiratelli et al., 1984] differ from marginal model in a sense that there is an additional unobservable random effect term U_i associated with each indi-

vidual. The margin models consider population-level effects whereas the random effect models also considers "subject-specific" effects. For example, let $Y_{i,t}$ be the values of response variable for individual i at time point t, random effect models make the following assumptions

$$g(E(Y_{i,t}|U_i)) = \mathbf{x}_{i,t}^\top \boldsymbol{\beta}^* + U_i.$$

In a situation where the past q measurements of the response $Y_{i,t-1}, \cdots, Y_{i,t-q}$ influence present observation $Y_{i,t}$, a transition model [Zeng and Cook, 2007] can be used. In transition models, the conditional distribution of each variable $Y_{i,t}$ is an explicit function of past response $Y_{i,t-1}, \cdots, Y_{i,t-q}$ and covariates $\mathbf{x}_{i,t}$. For example, a transitional model with a linear function of the covariates and autoregressive terms with Gaussian errrors becomes a Markov model

$$Y_{i,t} = \mathbf{x}_{i,t}^\top \boldsymbol{\beta}^* + \sum_{r=1}^{q} \phi_r^* (Y_{i,t-r} - \mathbf{x}_{i,t-r}^\top \boldsymbol{\beta}^*) + \epsilon_{i,t}, \tag{1.1}$$

where $\epsilon_{i,t} \sim \mathcal{N}(0, \sigma^2)$.

Variable selection is a very important issue for the above longitudinal regression modeling as it can select the most relevant covariates and provide a more parsimonious model. This could enhances the prediction accuracy and grants more interpretability of the resulting models. In addition to the traditional variable selection techniques such as the forward and backward stepwise selection, the sparse penalization such as Lasso (least absolute shrinkage and selection operator) [Tibshirani, 1996] has been a popular approach for simultaneous variable selection and estimation. Lasso was initially formulated for linear regression but is extended to other statistical models such as generalized linear models (GLM) [Cardot and Sarda, 2005,She, 2012], generalized estimating equations (GEE) [Wang et al., 2012,Johnson et al., 2008], and longitudinal data analysis [Groll and Tutz, 2014]. Assume that we observe n observations of the p-dimentional covariates $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$, the design matrix can be written as $\mathbf{X} = (\mathbf{x}_1^\top, \ldots, \mathbf{x}_n^\top)^\top$. The observed response variable

can be denoted by $\mathbf{y} = (y_1, \ldots, y_n)^\top$. Lasso estimates $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\top$ by solving the following problem

$$\min_{\boldsymbol{\beta}} \frac{1}{n} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1.$$

where $\lambda \geq 0$ controls the amount regularization and $\|\cdot\|_1$ is a ℓ_1-norm. Unlike ridge regression that uses ℓ_2-norm for penalization, solution of Lasso has sparsity. But a problem of Lasso is that it tends to over-select the true variables and lacks of selection consistency [Zhao and Yu, 2006]. To overcome such issue, [Zou, 2006] proposed the adaptive Lasso, which retains estimation consistency and oracle property in variable selection. In the adaptive Lasso, different penalty weights $w_j \geq 0$ are applied to the coefficients β_j's

$$\min_{\boldsymbol{\beta}} \frac{1}{n} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{j=1}^{p} w_j |\beta_j|,$$

where the weights $w_j = \frac{1}{|\hat{\beta}_j|^\gamma}$, for $\gamma > 0$, and $\hat{\beta}$ is a $\sqrt{n}$-consistent estimator such as the OLS, ridge or Lasso estimator.

When there is strong correlation between the covariates, Lasso tends to select only one covariate from a set of highly correlated covariates, [Zou and Hastie, 2005] proposed elastic net to deal with such drawback of Lasso. In the elastic net, an additional ℓ_2-norm penalty is added

$$\min_{\boldsymbol{\beta}} \frac{1}{n} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1 \|\boldsymbol{\beta}\|_1 + \lambda_2 \|\boldsymbol{\beta}\|_2^2.$$

In many applications with categorical variables in the data, the categorical variables are usually encoded as groups of dummy variables. In this case, it does not make sense to select these dummy variables belonging to the same categorical variable separately, i.e. a same group of dummy variables should selected/excluded simultaneously in the model selection procedure. [Yuan and Lin, 2006] introduced the group lasso which penalizes and selects predefined groups of covariates together. Occasionally, sparsity in both of groups and within each group may be preferred. For example, when identifying particularly im-

portant genes in pathway of gene expression pathway, we would want sparsity amongst genes, and its corresponding pathways.

One of the limitations of the group lasso is its restriction on the non-overlapping group structure, which limits its applicability in many scenarios. There have been many recent works to generalize the group Lasso formulation to allow overlaps among groups. The overlapping group Lasso has been particularly useful for selecting groups of features that are expected to have a certain contiguous patterns [Rapaport et al., 2008]. [Simon et al., 2013] introduced the sparse group lasso by adding an additional ℓ_1-norm penalty to the model to allow within group sparsity in addition to group-wise sparsity. Another important work for the overlapping group Lasso is by [Obozinski et al., 2011], which introduced latent group Lasso norms for structured sparsity with supports that are unions of predefined overlapping group of variables.

The overlapping group Lasso has also been extended to two-dimensional grid setting. For example, such penalization was applied in topographic dictionary learning [Jenatton et al., 2011], wavelet-based de-noising and for face recognition with corruption by occlusions [Mairal et al., 2011, Mairal et al., 2010, Jenatton et al., 2011]. The overlapping group lasso has also been used for variable selection according to a tree-like hierarchical structure, with applications in topic modeling, image restoration, probabilistic graphical models, mining of fMRI data [Jenatton et al., 2012], and natural language processing [Martins et al., 2011]. For example, [Kim and Xing, 2010] proposed a tree-guided group Lasso to estimate a sparse multi-response regression model. The model was applied in expression quantitative trait locus (eQTL) mapping in which the goal is to discover genetic variations that influence gene-expression levels. Since gene expression is multi-level process, the tree structure in the tree-guided group Lasso aims to capture such effect of gene expression path starting from genome to phenotype. The resulting optimization of the hierarchical group Lasso, however, is much more challenging to solve due to the complex group structures. [Jenatton et al., 2011] proposed an efficient solution for solving the proximal

operator of the hierarchical group Lasso through an algorithm based on the primal-dual formulation.

This overlapping group Lasso penalization has also been used in time series setting, [Nicholson et al., 2017] have proposed a general approach for hierarchical variable selection in multivariate autoregressive models. In their work, they introduce the concept of lag order to describe the variables associated with time. Higher lag coefficient corresponds coefficients that are associated with older response variable. Their methodology can be described as following multivariate autoregresssion

$$\mathbf{y}_t = \nu + \Phi^{(1)}\mathbf{y}_{t-1} + \cdots + \Phi^{(p)}\mathbf{y}_{t-p} + \mathbf{u}_t, \text{ for } t = 1, \cdots, T,$$

which conditions on initial values $\{\mathbf{y}_{-(p-1)}, \cdots, \mathbf{y}_0\}$. Here ν is the intercept vector and $\{\mathbf{\Phi}^{(\ell)}\}_{\ell=1}^{p}$ are the lag coefficients with $\mathbf{u}_t$ being a white noise vector. In their work, they define three different Lag structures among which the componentwise lag structure is closely resemble that of our approach.

In this article, we propose a double Lasso penalized transitional models to eliminate the redundant parts in the regression coefficients $\boldsymbol{\beta}$ and the autoregressive coefficients $\boldsymbol{\phi}$, which can in turn improve the prediction and estimation accuracy. In addition, for some applications, in order to obtain a more parsimonious model, we hope to maintain a hierarchical sparse structure of $\boldsymbol{\phi}$ in the model selection procedure, i.e. when an ancestor variable is zero, the model selection procedure can guarantee all of its descendents to be zero also, thus maintaining the desired the *strong heredity* property. For such purpose, we will introduce a hierarchical group Lasso penalization [Jenatton et al., 2010] into the transitional models. The resulting models can be efficiently solved by combining an iterative updating scheme on $\boldsymbol{\beta}$ and $\boldsymbol{\phi}$ with a computationally efficient proximal gradient algorithm. The proximal operation can also be efficiently conducted using a one-time update based on the primal-dual formulation.

The rest of the article is organized as follows. In Section 2, we introduce the methodology of our models, including the transition models using double Lasso penalization and hierarchical penalization. In Section 3, we provide the algorithms for solving the aforementioned models, and the implementation details of these algorithms are provided in Section 4. We also conduct simulation studies in Section 5 and a real data analysis in Section 6.

Chapter 2

METHODOLOGY

In this section, we introduce two different model selection techniques for the transition models using double Lasso penalization and hierarchical penalization, respectively.

2.1 Transition models using double Lasso penalization

We first give a brief description of the transition model. Given longitudinal dataset, assume that we observe n individuals $i = 1, \cdots, n$ who are measured on the time points $t = 1, \cdots, T_i$. Denote $y_{i,t}$ as the response variable for subject i measured on the time point t. Consider the REGression with AutoRegressive errors (REGAR) [Wang et al., 2007] model

$$y_{i,t} = \mathbf{x}_{i,t}^{\top}\boldsymbol{\beta}^* + e_{i,t} \qquad t = 1, \ldots, T_i,$$

where $\mathbf{x}_{i,t} = (x_{i,t,1}, \ldots, x_{i,t,p})^{\top}$ is the p-dimensional regression covariate and $\boldsymbol{\beta}^* = (\beta_1^*, \ldots, \beta_p^*)^{\top}$ is the associated true regression coefficient. Assume further that variable $e_{i,t}$ follows the autoregressive process with order q:

$$e_{i,t} = \phi_1^* e_{i,t-1} + \phi_2^* e_{i,t-2} + \ldots + \phi_q^* e_{i,t-q} + \varepsilon_{i,t},$$

where $\boldsymbol{\phi}^* = (\phi_1^*, \ldots, \phi_q^*)^\top$ are the true autoregressive coefficients and $\varepsilon_{i,t}$ are independent and identically distributed random variables from $\mathcal{N}(0, \sigma^2)$. The transition model can be expressed as a function of both the covariates $\mathbf{x}_{i,t}$ and of the past responses $y_{i,t-1}, \ldots, y_{i,t-q}$. We assume that the past q response variables affect the present response variable as in the following model:

$$y_{i,t} = \mathbf{x}_{i,t}^\top \boldsymbol{\beta}^* + \sum_{r=1}^{q} \phi_r^* (y_{i,t-r} - \mathbf{x}_{i,t-r}^\top \boldsymbol{\beta}^*) + \varepsilon_{i,t}. \tag{2.1}$$

In above equation we can see that present observation $y_{i,t}$ is a linear function of $\mathbf{x}_{i,t}$ and of $y_{i,t-r} - \mathbf{x}_{i,t-r}^\top \boldsymbol{\beta}^*$, for $r = 1, \ldots, q$. Suppose now that the number of previous time points q to be considered in the model is fixed.

Given a dataset $\{(\mathbf{x}_{i,t}, y_{i,t}) \in \mathbb{R}^{p+1}\colon i = 1, \cdots, n,\ t = 1, \cdots, T_i\}$, where T_i's are used for indicating maximum number of time points observed for the subject i, one can estimate $(\boldsymbol{\beta}^*, \boldsymbol{\phi}^*)$ in (2.1) by maximizing the following conditional likelihood function with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\phi}$:

$$\left(\frac{1}{2\pi\sigma^2}\right)^{n \times \sum T_i} \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^{n} \sum_{t=q+1}^{T_i} \left\{ y_{i,t} - \mathbf{x}_{i,t}^\top \boldsymbol{\beta} - \sum_{r=1}^{q} \phi_r (y_{i,t-r} - \mathbf{x}_{i,t-r}^\top \boldsymbol{\beta}) \right\}^2 \right].$$

which is equivalent to the minimization of the negative log-likelihood:

$$\ell(\boldsymbol{\beta}, \boldsymbol{\phi}) \equiv \frac{1}{2n} \sum_{i=1}^{n} \sum_{t=q+1}^{T_i} \left[y_{i,t} - \mathbf{x}_{i,t}^\top \boldsymbol{\beta} - \sum_{r=1}^{q} \phi_r (y_{i,t-r} - \mathbf{x}_{i,t-r}^\top \boldsymbol{\beta}) \right]^2.$$

In many applications, some elements of $\boldsymbol{\beta}^*$ and $\boldsymbol{\phi}^*$ might be exactly zero. Those zero elements of the coefficients correspond to the noise variables in $\boldsymbol{\beta}^*$ and $\boldsymbol{\phi}^*$ that have no effect to the outcomes of the model. In order to obtain a more parsimonious models without those noise variables, a model selection procedure becomes necessary. [Wang et al., 2007] introduced a Lasso penalized REGAR model to eliminate those redundant parts in $\boldsymbol{\beta}^*$ and $\boldsymbol{\phi}^*$ and estimate their values, which in turn can also help improve the prediction and estimation accuracy. In their method, they estimate the model coefficients

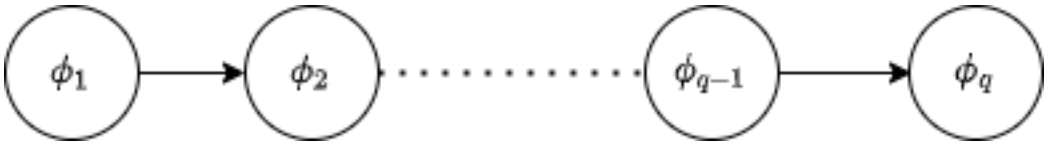

Figure 2.1: Chain Structure of $\boldsymbol{\phi}$

by minimizing the following objective function

$$(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\phi}}) = \arg\min_{\boldsymbol{\beta},\boldsymbol{\phi}} \ell(\boldsymbol{\beta}, \boldsymbol{\phi}) + \lambda_1 \sum_{r=1}^{q} w_r^{\phi} |\phi_r| + \lambda_2 \sum_{j=1}^{p} w_j^{\beta} |\beta_j|, \tag{2.2}$$

which applies a separate weighted ℓ_1 norm on each coefficient vector $\boldsymbol{\beta}$ and $\boldsymbol{\phi}$ with $\lambda_1, \lambda_2 \geq 0$ as the tuning parameters controlling the amount of penalization. By adding two separate Lasso penalty on $\boldsymbol{\beta}$ and $\boldsymbol{\phi}$, some estimated coefficients will be exactly zero, thus the corresponding covariates of $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\phi}}$ can be removed. From now on we will refer to the model in (2.2) as the double Lasso penalized transition model.

2.2 Transition models with hierarchical penalization

One drawback of the Lasso penalization in (2.2) is that the elements of $\boldsymbol{\phi}$ are selected individually and thus the impact of the autoregressive effects in $\boldsymbol{\phi}$ on their sparsity is ignored. Specifically, the temporal structure of $\boldsymbol{\phi}$ can be represented in a way as shown in Figure 1, in which node ϕ_1 comes before nodes $\phi_2, \ldots, \phi_q$ thus can be represented as the ancestor node and node $\phi_2, \ldots, \phi_q$ can be regarded as the descendants of ϕ_1. Similarly, node ϕ_2 can be viewed as the ancestor node of $\phi_3, \ldots, \phi_q$, and so on.

For some applications, in order to obtain a more parsimonious model, we hope to maintain a hierarchical sparse structure of $\boldsymbol{\phi}$ in the model selection procedure. Specifically, the model selection procedure should satisfy the so called *strong heredity* property:

$$\widehat{\phi}_j = 0 \qquad \Longrightarrow \qquad \text{descendents of } \widehat{\phi}_j \text{ are all zero, i.e. } (\widehat{\phi}_{j+1}, \widehat{\phi}_{j+2}, \ldots, \widehat{\phi}_q) = \mathbf{0}.$$

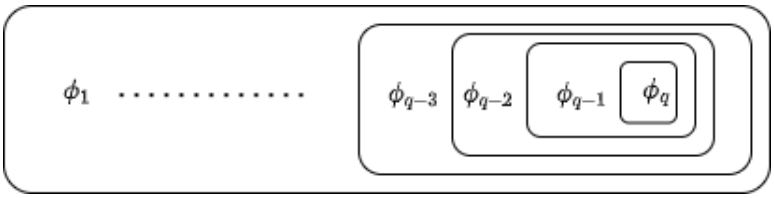

Figure 2.2: Hierarchical structure

A model selection procedure with such a property can ensure that if an autoregressive effect ϕ_j is removed from the model, then all the effects ϕ_i that come after ϕ_j (i.e. for $i > j$) in the temporal sequence will also be zero. To impose such hierarchical structure in the model, we introduce a hierarchical group Lasso penalty [Jenatton et al., 2010], which is a generalization of the group Lasso penalty [Yuan and Lin, 2006]. To impose the strong heredity, we group each variable with all of its descendent variables using a ℓ_2 norm. Specifically, the penalty has the following form

$$P(\boldsymbol{\phi}) = \sum_j \|\phi_j, \text{descendents of } \phi_j\|_2.$$

Specifically, in our model, the descendent of ϕ_j are $\phi_{j+1}, \phi_{j+2}, \ldots, \phi_q$. We can see such grouping structure in Figure 2.2 By the singularity of the ℓ_2 norm, when $\widehat{\phi}_j$ is zero, this penalty can enforce all of its descendents $\widehat{\phi}_{j+1}, \widehat{\phi}_{j+2}, \ldots, \widehat{\phi}_q$ to be zero also, thus maintaining the required strong heredity. Denote $\boldsymbol{\phi}_{r:q} = (\phi_r, \phi_{r+1}, \cdots, \phi_q)^\top$, we propose a hierarchical penalized transition model

$$(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\phi}}) = \underset{(\boldsymbol{\beta},\boldsymbol{\phi})}{\arg\min}\, f(\boldsymbol{\beta}, \boldsymbol{\phi}) \equiv \underset{(\boldsymbol{\beta},\boldsymbol{\phi})}{\arg\min}\, \ell(\boldsymbol{\beta}, \boldsymbol{\phi}) + P_{hier}(\boldsymbol{\beta}, \boldsymbol{\phi}),$$

where

$$P_{hier}(\boldsymbol{\beta}, \boldsymbol{\phi}) = \lambda_1 \sum_{r=1}^{q} w_r^{\phi} \|\boldsymbol{\phi}_{r:q}\|_2 + \lambda_2 \sum_{j=1}^{p} w_j^{\beta} |\beta_j|, \tag{2.3}$$

where $w_j^{\beta} \geq 0$ and $w_r^{\phi} \geq 0$ are the weights for penalty terms of $\boldsymbol{\beta}$ and $\boldsymbol{\phi}$ respectively. In (2.3), the first term of $P(\boldsymbol{\beta}, \boldsymbol{\phi})$ penalizes the autoregressive term $\boldsymbol{\phi}$ with hierarchical

property and the second term penalizes the regression coefficients β with the regular Lasso.

Chapter 3

Algorithm

In this section, we will discuss the numerical algorithms for solving the transition model with the double Lasso penalization and the hierarchical penalization. The implementation details such as the hyper-parameter tuning are also discussed.

3.1 Computation of the double lasso penalized model

The optimization of the double Lasso penalized transition model in (2.2) involves minimizing an objective function with non-smooth penalty function using ℓ_1-norn. The computation of such non-smooth optimization problem can be efficiently carried out using coordinate descent [Friedman et al., 2010a], the accelerated proximal gradient descent or the algorithms that explores the piecewise linearity of the solution path such as LARS [Efron et al., 2004]. Since in our problem there are two coefficient vectors $\boldsymbol{\beta}$ and $\boldsymbol{\phi}$ to optimize, we will combine an iterative scheme on $\boldsymbol{\beta}$ and $\boldsymbol{\phi}$ with the computationally efficient coordinate descent algorithm. Similarly to [Wang et al., 2007], in the iterative algorithm we fix one of the coefficient vector while optimizing the objective function with respect to the other alternatively till convergence. Specifically, for a given dataset $\{(\mathbf{x}_{i,t}, y_{i,t}) \in \mathbb{R}^{p+1}\colon\ i = 1, \cdots, n,\ t = 1, \cdots, T_i\}$, the double Lasso penalized transition model solves the following objective function

$$f_{\text{Lasso}}(\boldsymbol{\beta}, \boldsymbol{\phi}) = \frac{1}{2n} \sum_{i=1}^{n} \sum_{t=q+1}^{T_i} \left[y_{i,t} - \mathbf{x}_{i,t}^{\top} \boldsymbol{\beta} - \sum_{r=1}^{q} \phi_r (y_{i,t-r} - \mathbf{x}_{i,t-r}^{\top} \boldsymbol{\beta}) \right]^2 + \lambda_1 \sum_{r=1}^{q} w_r^{\phi} |\phi_r| + \lambda_2 \sum_{j=1}^{p} w_j^{\beta} |\beta_j|. \tag{3.1}$$

The iterative optimization scheme is conducted in the following way: first we fix $\boldsymbol{\beta}$ at its current value $\widetilde{\boldsymbol{\beta}}$ and update $\boldsymbol{\phi}$ by minimizing

$$\widetilde{\boldsymbol{\phi}}^{(\text{new})} \leftarrow \arg\min_{\boldsymbol{\phi}} f_{\text{Lasso}}(\boldsymbol{\phi} | \widetilde{\boldsymbol{\beta}}). \tag{3.2}$$

Next we fix $\boldsymbol{\phi} = \widetilde{\boldsymbol{\phi}}$ and update $\boldsymbol{\beta}$ by minimizing

$$\widetilde{\boldsymbol{\beta}}^{(\text{new})} \leftarrow \arg\min_{\boldsymbol{\beta}} f_{\text{Lasso}}(\boldsymbol{\beta} \mid \widetilde{\boldsymbol{\phi}}). \tag{3.3}$$

For mathematically convenience, those two sub-problems can be represented in matrix format. Define

$$z_{i,t} = y_{i,t} - \sum_{r=1}^{q} \widetilde{\phi}_r y_{i,t-r}, \qquad z'_{i,t} = y_{i,t} - \mathbf{x}_{i,t}^{\top} \widetilde{\boldsymbol{\beta}},$$

$$\mathbf{w}_{i,t} = \mathbf{x}_{i,t} - \sum_{r=1}^{q} \widetilde{\phi}_r \mathbf{x}_{i,t-r}, \qquad \mathbf{w'}_{i,t} = [y_{i,t-1} - \mathbf{x}_{i,t-1}^{\top} \widetilde{\boldsymbol{\beta}}, \cdots, y_{i,t-q} - \mathbf{x}_{i,t-q}^{\top} \widetilde{\boldsymbol{\beta}}]^{\top},$$

for $z_{i,t}, z'_{i,t} \in \mathbb{R}$, $\mathbf{w}_{i,t} \in \mathbb{R}^p, \mathbf{w'}_{i,t} \in \mathbb{R}^q$. Note that we consider terms where $i \in \{1, \cdots, n\}$ and $t \in \{q+1, \cdots, T_i\}$. Then (3.2) and (3.3) can be expressed as:

$$f_{\text{Lasso}}(\boldsymbol{\phi} | \widetilde{\boldsymbol{\beta}}) = \frac{1}{2n} \sum_{i=1}^{n} \sum_{t=q+1}^{T_i} \left[z'_{i,t} - \mathbf{w'}_{i,t}^{\top} \boldsymbol{\phi} \right]^2 + \lambda_1 \sum_{r=1}^{q} w_r^{\phi} |\phi_r|, \tag{3.4}$$

and

$$f_{\text{Lasso}}(\boldsymbol{\beta} | \widetilde{\boldsymbol{\phi}}) \frac{1}{2n} \sum_{i=1}^{n} \sum_{t=q+1}^{T_i} \left[z_{i,t} - \mathbf{w}_{i,t}^{\top} \boldsymbol{\beta} \right]^2 + \lambda_2 \sum_{j=1}^{p} w_j^{\beta} |\beta_j|, \tag{3.5}$$

which can be further simplified as

$$\widetilde{\boldsymbol{\phi}}^{(\text{new})}_{\text{Lasso}} \leftarrow \underset{\boldsymbol{\phi}\in\mathbb{R}^q}{\arg\min}\, \frac{1}{2n}\|\mathbf{z}' - \mathbf{W}'\boldsymbol{\phi}\|_2^2 + \lambda_1 \sum_{r=1}^{q} w_r^{\phi}|\phi_r|, \tag{3.6}$$

and

$$\widetilde{\boldsymbol{\beta}}^{(\text{new})}_{\text{Lasso}} \leftarrow \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\, \frac{1}{2n}\|\mathbf{z} - \mathbf{W}\boldsymbol{\beta}\|_2^2 + \lambda_2 \sum_{j=1}^{p} w_j^{\beta}|\beta_j|, \tag{3.7}$$

where

$$\begin{aligned}
\mathbf{z} &= (z_{1,q+1}, z_{1,q+2}, \cdots, z_{1,T_1}, z_{2,q+1}, \cdots, z_{2,T_2}, \cdots, z_{n,q+1}, \cdots, z_{n,T_n})^{\top},\\
\mathbf{z}' &= (z'_{1,q+1}, z'_{1,q+2}, \cdots, z'_{1,T_1}, z'_{2,q+1}, \cdots, z'_{2,T_2}, \cdots, z'_{n,q+1}, \cdots, z'_{n,T_n})^{\top},\\
\mathbf{W} &= [\mathbf{w}_{1,q+1}, \mathbf{w}_{1,q+2}, \cdots, \mathbf{w}_{1,T_1}, \mathbf{w}_{2,q+1}, \cdots, \mathbf{w}_{2,T_2}, \cdots, \mathbf{w}_{n,q+1}, \cdots, \mathbf{w}_{n,T_n}]^{\top},\\
\mathbf{W}' &= [\mathbf{w}'_{1,q+1}, \mathbf{w}'_{1,q+2}, \cdots, \mathbf{w}'_{1,T_1}, \mathbf{w}'_{2,q+1}, \cdots, \mathbf{w}'_{2,T_2}, \cdots, \mathbf{w}'_{n,q+1}, \cdots, \mathbf{w}'_{n,T_n}]^{\top}.
\end{aligned}$$

From the formula (3.6) and (3.7), we have found that the optimization problems (3.2) and (3.3) have been converted into two separate Lasso penalized least squares problems, which then can be efficiently solved using the existing solver via the coordinate descent such as `glmnet` proposed by [Friedman et al., 2010a]. The details of the algorithm is provided in Algorithm (1).

3.2 Computation of the hierarchical penalized model

In this section, we discuss the computation of the hierarchical penalized transition models. We consider of solving the following objective function

$$f_{\text{Hier}}(\boldsymbol{\beta}, \boldsymbol{\phi}) = \frac{1}{2n}\sum_{i=1}^{n}\sum_{t=q+1}^{T_i}\left[y_{i,t} - \mathbf{x}_{i,t}^{\top}\boldsymbol{\beta} - \sum_{r=1}^{q}\phi_r(y_{i,t-r} - \mathbf{x}_{i,t-r}^{\top}\boldsymbol{\beta})\right]^2 + \lambda_1\sum_{r=1}^{q} w_r^{\phi}\|\boldsymbol{\phi}_{r:q}\|_2 + \lambda_2\sum_{j=1}^{p} w_j^{\beta}|\beta_j|. \tag{3.8}$$

Algorithm 1: Double LASSO implementation of penalization of Transition model

Input : $\mathbf{y} = (y_{1,1}, \cdots, y_{n,T_n})^\top$, $\mathbf{X} = (\mathbf{x}_{1,1}, \cdots, \mathbf{x}_{n,T_n})^\top$; the tuning parameters λ_1, λ_2.
Output : $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\phi}})$

- Initialize $(\widetilde{\boldsymbol{\beta}}, \widetilde{\boldsymbol{\phi}}) \leftarrow (\mathbf{0}, \mathbf{0})$.
- Repeat (a) and (b) until convergence of $\boldsymbol{\beta}$ and $\boldsymbol{\phi}$:

  (a) Fix $\boldsymbol{\beta} = \widetilde{\boldsymbol{\beta}}$ and update $\boldsymbol{\phi}$ by minimizing

  $$\widetilde{\boldsymbol{\phi}}_{\text{Lasso}}^{(\text{new})} \leftarrow \arg\min_{\boldsymbol{\phi}} f(\boldsymbol{\phi}|\widetilde{\boldsymbol{\beta}}),$$

  with respect to $\boldsymbol{\phi}$, where

  $$f_{\text{Lasso}}(\boldsymbol{\phi}|\widetilde{\boldsymbol{\beta}}) = \frac{1}{2n}\sum_{i=1}^{n}\sum_{t=q+1}^{T_i}\left[y_{i,t} - \mathbf{x}_{i,t}^\top\widetilde{\boldsymbol{\beta}} - \sum_{r=1}^{q}\phi_r(y_{i,t-r} - \mathbf{x}_{i,t-r}^\top\widetilde{\boldsymbol{\beta}})\right]^2 + \lambda_1\sum_{r=1}^{q} w_r^{\phi}|\phi_r|.$$

  (b) Fix $\boldsymbol{\phi} = \widetilde{\boldsymbol{\phi}}$ and update $\boldsymbol{\beta}$ by minimizing:

  $$\widetilde{\boldsymbol{\beta}}_{\text{Lasso}}^{(\text{new})} \leftarrow \arg\min_{\boldsymbol{\beta}} f(\boldsymbol{\phi}|\widetilde{\boldsymbol{\beta}}),$$

  with respect to $\boldsymbol{\beta}$, where

  $$f_{\text{Lasso}}(\boldsymbol{\beta}|\widetilde{\boldsymbol{\phi}}) = \frac{1}{2n}\sum_{i=1}^{n}\sum_{t=q+1}^{T_i}\left[y_{i,t} - \sum_{r=1}^{q}\widetilde{\phi}_r y_{i,t-r} + \sum_{r=1}^{q}(\widetilde{\phi}_r\mathbf{x}_{i,t-r}^\top - \mathbf{x}_{i,t}^\top)\boldsymbol{\beta}\right]^2 + \lambda_2\sum_{j=1}^{p} w_j^{\beta}|\beta_j|.$$

- Return $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\phi}}) = (\widetilde{\boldsymbol{\beta}}_{\text{Lasso}}^{(\text{new})}, \widetilde{\boldsymbol{\phi}}_{\text{Lasso}}^{(\text{new})})$.

Similar to the double Lasso penalization, we find $\boldsymbol{\beta}$ and $\boldsymbol{\phi}$ with iterative algorithm. That is, we first fix $\boldsymbol{\beta} = \widetilde{\boldsymbol{\beta}}$ in $f(\boldsymbol{\beta}, \boldsymbol{\phi})$ and minimize

$$\widetilde{\boldsymbol{\phi}}^{(\text{new})} \leftarrow \arg\min_{\boldsymbol{\phi}} f_{\text{Hier}}(\boldsymbol{\phi}|\widetilde{\boldsymbol{\beta}}) \tag{3.9}$$

for $\boldsymbol{\phi}$, then we fix $\boldsymbol{\phi} = \widetilde{\boldsymbol{\phi}}$ in $f(\boldsymbol{\beta}, \boldsymbol{\phi})$ and minimize

$$\widetilde{\boldsymbol{\beta}}^{(\text{new})} \leftarrow \arg\min_{\boldsymbol{\beta}} f_{\text{Hier}}(\boldsymbol{\beta} \mid \widetilde{\boldsymbol{\phi}}) \tag{3.10}$$

for $\boldsymbol{\beta}$. These steps are alternatively repeated until convergence of both $\boldsymbol{\beta}$ and $\boldsymbol{\phi}$. Obviously the only difference in the algorithm is how we penalize in the sub-problem 3.9. The detail of this algorithm is outlined in Algorithm 2.

Again, similarly to the double Lasso case, we can also rewrite (3.9) and (3.10) in matrix forms as

$$\widetilde{\boldsymbol{\phi}}_{\text{Hier}}^{(\text{new})} \leftarrow \arg\min_{\boldsymbol{\phi}\in\mathbb{R}^q} \frac{1}{2n}\|\mathbf{z}' - \mathbf{W}'\boldsymbol{\phi}\|_2^2 + \lambda_1 \sum_{r=1}^{q} w_r^{\phi}\|\boldsymbol{\phi}_{r:q}\|_2, \tag{3.17}$$

and

$$\widetilde{\boldsymbol{\beta}}_{\text{Hier}}^{(\text{new})} \leftarrow \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \frac{1}{2n}\|\mathbf{z} - \mathbf{W}\boldsymbol{\beta}\|_2^2 + \lambda_2 \sum_{j=1}^{p} w_j^{\beta}|\beta_j|. \tag{3.18}$$

In the next section, we will discuss the computation of the sub-problem (3.17), which is a hierarchical penalized least squares.

3.3 Computation of the hierarchical penalized least squares

3.3.1 The proximal gradient

We use the proximal gradient algorithm [Beck and Teboulle, 2009] to solve the sub-problem (3.17). The algorithm iteratively performs a gradient descent update within a proximal operator, which is computationally efficient. It has the convergence guarantee for

Algorithm 2: Iterative scheme for the hierarchical penalized transition model.

Input : $\mathbf{y} = (y_{1,1}, \cdots, y_{n,T_n})^\top$, $\mathbf{X} = (\mathbf{x}_{1,1}, \cdots, \mathbf{x}_{n,T_n})^\top$; the tuning parameters λ_1, λ_2.
Output : $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\phi}})$

- Initialize $(\widetilde{\boldsymbol{\beta}}, \widetilde{\boldsymbol{\phi}}) \leftarrow (\mathbf{0}, \mathbf{0})$.
- Repeat the following step (a) and (b) until convergence of $\boldsymbol{\beta}$ and $\boldsymbol{\phi}$:

  (a) Fix $\boldsymbol{\beta} = \widetilde{\boldsymbol{\beta}}$ and update $\boldsymbol{\phi}$ by minimizing

$$\widetilde{\boldsymbol{\phi}}_{\text{Hier}}^{(\text{new})} \leftarrow \arg\min_{\boldsymbol{\phi}} f(\boldsymbol{\phi}|\widetilde{\boldsymbol{\beta}}), \tag{3.11}$$

  with respect to $\boldsymbol{\phi}$, where

$$f_{\text{Hier}}(\boldsymbol{\phi}|\widetilde{\boldsymbol{\beta}}) = \frac{1}{2n}\sum_{i=1}^{n}\sum_{t=q+1}^{T_i}\left[y_{i,t} - \mathbf{x}_{i,t}^\top\widetilde{\boldsymbol{\beta}} - \sum_{r=1}^{q}\phi_r(y_{i,t-r} - \mathbf{x}_{i,t-r}^\top\widetilde{\boldsymbol{\beta}})\right]^2 \tag{3.12}$$

$$+ \lambda_1\sum_{r=1}^{q} w_r^{\phi}\|\boldsymbol{\phi}_{r:q}\|_2. \tag{3.13}$$

  (b) Fix $\boldsymbol{\phi} = \widetilde{\boldsymbol{\phi}}$ and update $\boldsymbol{\beta}$ by minimizing:

$$\widetilde{\boldsymbol{\beta}}_{\text{Hier}}^{(\text{new})} \leftarrow \arg\min_{\boldsymbol{\beta}} f(\boldsymbol{\phi}|\widetilde{\boldsymbol{\beta}}), \tag{3.14}$$

  with respect to $\boldsymbol{\beta}$, where

$$f_{\text{Hier}}(\boldsymbol{\beta}|\widetilde{\boldsymbol{\phi}}) = \frac{1}{2n}\sum_{i=1}^{n}\sum_{t=q+1}^{T_i}\left[y_{i,t} - \sum_{r=1}^{q}\widetilde{\phi}_r y_{i,t-r} + (\sum_{r=1}^{q}(\mathbf{x}_{i,t}^\top - \widetilde{\phi}_r\mathbf{x}_{i,t-r}^\top))\boldsymbol{\beta}\right]^2 \tag{3.15}$$

$$+ \lambda_2\sum_{j=1}^{p} w_j^{\beta}|\beta_j|. \tag{3.16}$$

- Return $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\phi}}) = (\widetilde{\boldsymbol{\beta}}_{\text{Hier}}^{(\text{new})}, \widetilde{\boldsymbol{\phi}}_{\text{Hier}}^{(\text{new})})$.

the non-smooth convex optimization problems and has linear convergence rate under strong convexity case. Specifically, we denote the value of $\boldsymbol{\phi}$ at k-th iteration as $\boldsymbol{\phi}^{(k)}$. Let $\ell(\boldsymbol{\phi}) = \frac{1}{2n}\|\mathbf{z}' - \mathbf{W}'\boldsymbol{\phi}\|_2^2$ and its gradient is denoted by $\nabla_{\boldsymbol{\phi}}\ell(\boldsymbol{\phi}) = -\frac{1}{n}\mathbf{W}'^{\top}(\mathbf{z}' - \mathbf{W}'\boldsymbol{\phi})$. Denote by $P(\boldsymbol{\phi}) = \lambda\sum_{r=1}^{q} w_r\|\boldsymbol{\phi}_{r:q}\|_2$ the hierarchical penalty. The proximal gradient algorithm iteratively update $\boldsymbol{\phi}^{(k)}$ with

$$\begin{aligned}\boldsymbol{\phi}^{(k)} &\leftarrow \underset{\boldsymbol{\phi}\in\mathbb{R}^q}{\arg\min}\,\frac{1}{2t}\|\boldsymbol{\phi} - (\boldsymbol{\phi}^{(k-1)} - t\nabla_{\boldsymbol{\phi}}\ell(\boldsymbol{\phi}^{(k-1)}))\|_2^2 + P(\boldsymbol{\phi}) \\ &= \text{prox}_{tP}(\boldsymbol{\phi}^{(k-1)} - t\nabla_{\boldsymbol{\phi}}\ell(\boldsymbol{\phi}^{(k-1)})),\end{aligned} \tag{3.19}$$

where $t > 0$ is the step size. The proximal operator used above is the minimizer of the penalized problem

$$\text{prox}_{tP}(\mathbf{u}) = \underset{\boldsymbol{\phi}\in\mathbb{R}^q}{\arg\min}\,\frac{1}{2}\|\mathbf{u} - \boldsymbol{\phi}\|_2^2 + \lambda_1\sum_{r=1}^{q} w_r^{\phi}\|\boldsymbol{\phi}_{r:q}\|_2. \tag{3.20}$$

Therefore in iteration k, we update $\boldsymbol{\phi}^{(k)}$ by computing the proximal operator (3.20) with

Algorithm 3: Proximal methods for solving for $\boldsymbol{\phi}$

Input : $\mathbf{z}' \in \mathbb{R}^{\sum_i T_i - nq}$, $\mathbf{W}' \in \mathbb{R}^{(\sum T_i - nq)\times q}$
Output : $\widetilde{\boldsymbol{\phi}}^{(\text{new})}$
Initialize $\boldsymbol{\phi}^{(0)}$, let $k = 1$
repeat
  Use the backtracking line search to find step size t (call Algorithm 4)
  $\mathbf{u}^{(k-1)} \leftarrow \boldsymbol{\phi}^{(k-1)} - t\nabla_{\boldsymbol{\phi}}\ell(\boldsymbol{\phi}^{(k-1)})$
  $\boldsymbol{\phi}^{(k)} \leftarrow \arg\min_{\boldsymbol{\phi}\in\mathbb{R}^q} \text{prox}_{tP}(\mathbf{u}^{(k-1)})$ (call Algorithm 5)
  $k \leftarrow k + 1$
until *convergence of* $\boldsymbol{\phi}$;
$\widetilde{\boldsymbol{\phi}}^{(\text{new})} \leftarrow \boldsymbol{\phi}^{(k)}$

$\mathbf{u} = \boldsymbol{\phi}^{(k-1)} - t\nabla_{\boldsymbol{\phi}}\ell(\boldsymbol{\phi}^{(k-1)}) = \boldsymbol{\phi}^{(k-1)} + \frac{t}{n}\mathbf{W}'^{\top}(\mathbf{z}' - \mathbf{W}'\boldsymbol{\phi}^{(k-1)})$.

We use the backtracking line search to select step size t in each iteration. After the initialization of $t = t_{\text{init}} > 0$, we shrink t by a factor β using $t = \beta t$ until the following

condition becomes satisfied:

$$\ell(\mathring{\boldsymbol{\phi}}) > \ell(\boldsymbol{\phi}^{(k-1)}) - \nabla_{\boldsymbol{\phi}}\ell(\boldsymbol{\phi}^{(k-1)})^\top[\boldsymbol{\phi}^{(k-1)} - \mathring{\boldsymbol{\phi}}] + \frac{1}{2t}\|\boldsymbol{\phi}^{(k-1)} - \mathring{\boldsymbol{\phi}}\|_2^2,$$

where $\mathring{\boldsymbol{\phi}} = \text{prox}_{tP}(\boldsymbol{\phi}^{(k-1)} - t\nabla_{\boldsymbol{\phi}}\ell(\boldsymbol{\phi}))$ and $\ell(\boldsymbol{\phi}) = \frac{1}{2n}\|\mathbf{z}' - \mathbf{W}'\boldsymbol{\phi}\|_2^2$. The details of the line search is provided in Algorithm (4).

Algorithm 4: Backtracking Line Search Implementation for solving for $\boldsymbol{\phi}$

Input : $\boldsymbol{\phi}^{(k-1)}$ **and** $\nabla_{\boldsymbol{\phi}}\ell(\boldsymbol{\phi}^{(k-1)}))$
Output : step size t
Initialize step size $= t_{\text{init}} > 0$, choose shrinking factor $0 < \beta < 1$
$\mathring{\boldsymbol{\phi}}$ =prox$_{tP}((\boldsymbol{\phi}^{(k-1)} - t\nabla_{\boldsymbol{\phi}}\ell(\boldsymbol{\phi}^{(k-1)}))$ (call Algorithm 5)
while $\ell(\mathring{\boldsymbol{\phi}}) < \ell(\boldsymbol{\phi}^{(k-1)}) - \nabla_{\boldsymbol{\phi}}\ell(\boldsymbol{\phi}^{(k-1)})^\top[\boldsymbol{\phi}^{(k-1)} - \mathring{\boldsymbol{\phi}}] + \frac{1}{2t}\|\boldsymbol{\phi}^{(k-1)} - \mathring{\boldsymbol{\phi}}\|_2^2$ **do**
  $t = \beta t$
  $\mathring{\boldsymbol{\phi}}$ =prox$_{tP}((\boldsymbol{\phi}^{(k-1)} - t\nabla_{\boldsymbol{\phi}}\ell(\boldsymbol{\phi}^{(k-1)}))$ (call Algorithm 5)
end
return t

3.3.2 The proximal operator for the hierarchical penalization

For the computation of the proximal operator for the hierarchical penalization in (3.20), we adopt the algorithm proposed by [Jenatton et al., 2010], which provides an efficient way to solve the proximal operators for the overlapping group lasso with general tree structures. The hierarchical penalization used in our model (3.17) has a chain structure as shown in Figure 2.2, which can be viewed as a special case of the tree structures. Specifically, we optimize the dual form of (3.20), which can be written as

$$\begin{aligned} \max_{\xi \in \mathbb{R}^{q\times q}} \; & -\frac{1}{2}\big[\|\mathbf{u} - \sum_{r=1}^{q}\boldsymbol{\xi}_{r:q}\|_2^2 - \|\mathbf{u}\|_2^2\big] \\ \text{s.t.} \quad & \forall r \in \{1, \cdots, q\},\ \|\boldsymbol{\xi}_{r:q}\|_2 \le \lambda_1 w_r \end{aligned} \quad . \tag{3.21}$$

Here, $\boldsymbol{\xi}_{r:q}$ represents the concatenation of a zero matrix and the last r columns of dual variable $\boldsymbol{\xi} \in \mathbb{R}^{q\times q}$,

$$\boldsymbol{\xi}_{r:q} = (0, \cdots, \xi_r, \xi_{r+1}, \cdots, \xi_q)^\top.$$

Denote by $\Pi^*_{\lambda_1 w_r}$ the projection operator of the vector $\mathbf{v}$ to an ℓ_2-norm ball with radius $\lambda_1 w_r$

$$\Pi^*_{\lambda_1 w_r}(\mathbf{v}) = \begin{cases} \mathbf{v} & \text{if } \|\mathbf{v}\|_2 \leq \lambda_1 w_{r,} \\ \frac{\mathbf{v}}{\|\mathbf{v}\|_2}\lambda_1 w_r & \text{otherwise.} \end{cases}.$$

After initialization of $\mathbf{v} = \mathbf{u}$ and $\boldsymbol{\xi} = 0$, for $r = q, q-1, \ldots, 2, 1$, we iteratively update $\boldsymbol{\xi}_{r:q}$ by

$$\boldsymbol{\xi}_{r:q} \leftarrow \Pi^*_{\lambda_1 w_r}(\mathbf{v}_{r:q})$$

where $\mathbf{v} = \mathbf{u} - \sum_{j\neq r} \boldsymbol{\xi}_{j:q}$. In the end of the iteration, we report the solution of (3.20) as

$$\text{prox}_{tP}(\mathbf{u}) = \mathbf{v} = \mathbf{u} - \sum_{r=1}^{q} \boldsymbol{\xi}_{r:q}$$

Algorithm 5: Proximal methods for solving for ϕ

Input : $\mathbf{u} \in \mathbb{R}^q$
Output : $(\mathbf{v}, \boldsymbol{\xi})$ (primal-dual solutions)
Initialize $\mathbf{v} = \mathbf{u},\ \boldsymbol{\xi} = 0_{(q\times q)}$
for $r \in \mathcal{G} = \{q, q-1, \cdots, 1\}$ **do**
  $\mathbf{v} \leftarrow \mathbf{u} - \sum_{j\neq r} \boldsymbol{\xi}_{j:q}$
  $\boldsymbol{\xi}_{r:q} \leftarrow \Pi^*_{\lambda_1 w_g}(\mathbf{v}_{r:q})$
end
$\mathbf{v} \leftarrow \mathbf{u} - \sum_{r=1}^{q} \boldsymbol{\xi}_{r:q}$

Chapter 4

Implementation details

In this section we will describe some details of the implementation of the algorithms for solving the double Lasso and the hierarchical penalized transition models.

4.1 The double Lasso penalization

For solving the double Lasso penalized problem in (3.6) and (3.7), we use the existing `R` package `glmnet` [Friedman et al., 2010b], which adopts an efficient coordinate descent algorithm for solving least squares loss

$$\min_{\boldsymbol{\beta}} \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda[(1-\alpha)\|\boldsymbol{\beta}\|_2^2 + \alpha\|\boldsymbol{\beta}\|_1].$$

For the argument of `glmnet`, we exclude the intercept term using `intercept=FALSE`, disable the standardization by `standardization=FALSE` and we set the parameter `alpha=1` to enforce a Lasso penalty. Suppose we have the observations $x_i \in \mathbb{R}^p$ and the response $y_i \in \mathbb{R}$ for $i = 1, \ldots, n$, when we optimize $\boldsymbol{\beta}$, we set $\mathbf{X} = \mathbf{W}$ and $\mathbf{y} = \mathbf{z}$; When we optimize $\boldsymbol{\phi}$, we set $\mathbf{X} = \mathbf{W}'$ and $\mathbf{y} = \mathbf{z}'$ with the same settings on `glmnet` .

4.2 The hierarchical penalization

The solution to the proximal operator in (3.20) for the hierarchical penalization is obtained using `SPAMS` package [Mairal, 2014]. After finding the correct step size t with 4, we use `SPAMS.proximalGraph` function in `SPAMS` to solve the optimization problem (3.20), specifically it solves the following problem

$$\min_{\mathbf{v}\in\mathbb{R}^p} \frac{1}{2}\|\mathbf{u}-\mathbf{v}\|_2 + \lambda \sum_{g\in\mathcal{G}} \eta^g \|\mathbf{v}_g\|_*, \tag{4.1}$$

where $\mathbf{v} = (v_1, \ldots, v_p)^\top$ is a p-dimensional vector and g is defined to be the index of the group of the variables to be penalized together. The function `proximalGraph` has a parameter `graph` that the user need to specify in order to define a given structured regularization. There are three attributes associated with the parameter `graph`:

- `graph.eta_g`,
- `graph.groups`
- `graph.groups_var`

The first attribute `graph.eta_g` is used to determine the weight of penalization assigned to each group g. In our case, we set `graph.eta_g`=$(1, \ldots, 1)^\top$. The second and the third attributes `graph.groups` and `graph.group_var` controls the structure of the penalization. The attribute `graph.groups` sets the inclusion relationships between the groups. Suppose that there are k groups $\mathcal{G} = \{g_1, \cdots, g_k\}$ in the structured penalization. The value of this attribute should be a $k \times k$ matrix and the (i, j) entry of this matrix is one if and only if the group g_i is included in group g_j (for $i \neq j$). The diagonal entries of this matrix are all zero. The third attribute `graph.group_var` controls the inclusion relationships between groups and variables. It takes a $p \times k$ matrix, where the (i, j) entry of this matrix is one if and only if the variable v_i in (4.1) is included in group g_j but not in any descendants group of g_j. For example, if the group structure $\mathcal{G} = \{g_1, g_2, g_3, g_4, g_5\}$ is

defined in the following way

$$
\begin{aligned}
g_1 &= \{1, 2, 3, 4\} \\
g_2 &= \{4, 5, 6, 7\} \\
g_3 &= \{7, 8, 9, 10\} \\
g_4 &= \{1, 2, 3, 4, 5, 6\} \\
g_5 &= \{7, 8, 9\},
\end{aligned}
$$

the entries for `graph.groups` and `graph.group_var` will be

$$
\text{graph.groups} = \begin{pmatrix}
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0
\end{pmatrix}, \quad \text{graph.groups\_var} = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0
\end{pmatrix}.
$$

For argument `graph.groups`, the entry at (1,4) of `graph.groups` is non-zero because g_1 is different from g_4 and g_1 is included in g_4. Similarly, the entry at (5,3) of `graph.groups` is also non-zero because g_3 is different from g_5 and g_5 is included in g_3. All other entries of `graph.groups` are zeroes because it does not satisfy the aforementioned condition.

Now looking at `graph.groups_var`, the entry at (1,1) is non-zero because variable 1 is included in g_1 and is not included in any children of g_1 (since g_1 does not have a descendent group), whereas the entry (1,4) is zero because even that variable 1 is included

in g_4 but it also included in g_1 (which is a descendant group of g_4), thus doesn't satisfy the condition. Likewise, for the hierarchical penalization case, we can construct the two graph structure parameters for our chain-like structure as,

$$\text{graph.group} = \begin{pmatrix} 0 & 1 & \cdots & 1 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \cdots & \cdots & 0 \end{pmatrix}, \text{ graph.groups\_var} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 1 \end{pmatrix},$$

for regularization we select `regul='graph'` which will apply group lasso to the coefficients. We set λ_1 as the given regularization parameter and exclude intercept term using `intercept=FALSE`, and we allow negative numbers in the output of the proximal computation by setting `pos=FALSE` (setting this argument to `TRUE` will truncation the negative entries of the results to zero). As mentioned previously, we use this at each iteration k of backtracking-line search by setting $\mathbf{u} = \boldsymbol{\phi}^{(k-1)} - t\nabla_{\boldsymbol{\phi}}\ell(\boldsymbol{\phi}^{(k-1)})$ in (3.20). The output of this function will give the solution to 3.20.

4.3 Model selection

To find the appropriate values of the tuning parameters λ_1 and λ_2 associated in the models for time-dependent data, we use the cross-validation procedure proposed by [Song and Bickel, 2011] and [Bańbura et al., 2010]. We write $(\widehat{\boldsymbol{\beta}}^{(\lambda_1,\lambda_2)}, \widehat{\boldsymbol{\phi}}^{(\lambda_1,\lambda_2)})$ as the estimated coefficients for a given pair (λ_1, λ_2), the point estimates of the *one-step-ahead* forecasts are denoted by

$$\hat{y}_{i,t+1}^{(\lambda_1,\lambda_2)} = \mathbf{x}_{i,t}^{\top}\widehat{\boldsymbol{\beta}}^{(\lambda_1,\lambda_2)} + \sum_{r=1}^{q} \hat{\phi}_r^{(\lambda_1,\lambda_2)}(y_{i,t-r} - \mathbf{x}_{i,t-r}^{\top}\boldsymbol{\beta}^{(\lambda_1,\lambda_2)}),$$

which is obtained using only the information up to time t. Denote by T^0 and T^1 the beginning and the end of the evaluation period. For each evaluation time period $t = T^0, \ldots, T^1 - 1$, we compute the one-step-ahead forecasts $\hat{y}_{i,t+1}^{(\lambda_1,\lambda_2)}$ using only the information

up to time t. The out-of-sample forecast error is measured using mean squared forecast error (MSFE)

$$\text{MSFE}^{(\lambda_1,\lambda_2)} \equiv \frac{1}{n \times (T^1 - T^0 - 1)} \sum_{i=1}^{n} \sum_{t=T^0}^{T^1-1} (y_{i,t+1} - \hat{y}_{i,t+1}^{(\lambda_1,\lambda_2)})^2. \tag{4.2}$$

We select the optimal pair $(\hat{\lambda}_1, \hat{\lambda}_2)$ according to

$$(\hat{\lambda}_1, \hat{\lambda}_2) = \underset{(\lambda_1,\lambda_2)}{\arg\min} \text{MSFE}^{(\lambda_1,\lambda_2)}.$$

Chapter 5

Simulation

The simulation results presented in this section will compare and evaluate the double Lasso penalized models 2.2 and the hierarchical penalized models 3.8 in terms of their prediction accuracy and variable selection accuracy, and how those performances vary with different sample sizes and the numbers of longitudinal measurements. Throughout this section, the tuning parameters are selected by the cross validation procedure describe in Section 4.3 and we use grid search on the tuning parameters λ_1 and λ_2 from the grid points $\{0, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$. For convergence criterion, we choose $\epsilon = 10^{-3}$ and we stop the iterative process when $\|\boldsymbol{\phi}^{(k+1)} - \boldsymbol{\phi}^{(k)}\|_2 < \epsilon$ and $\|\boldsymbol{\beta}^{(k+1)} - \boldsymbol{\beta}^{(k)}\|_2 < \epsilon$. We select $(\hat{\lambda}_1, \hat{\lambda}_2)$ which minimizes MSFE defined in (4.2). The simulation data is generated from the REGAR model adopted from [Wang et al., 2007] under the following setting:

$$y_{i,t} = 3.0x_{i,t,1} + 1.5x_{i,t,2} + 2.0x_{i,t,5} + x_{i,} + e_{i,t}$$

$$e_{i,t} = 0.5e_{i,t-1} - 0.70e_{i,t-2} + \sigma\epsilon_t,$$

for $\epsilon_t \sim \mathcal{N}(0,1)$, $t = 1, \cdots, T_i$ and $i = 1, \cdots, n$. Our true regression and autocorrelation coefficients are set at $\boldsymbol{\beta}^* = (3.0, 1.5, 0, 0, 2, 0, 0, 1)^\top$ and $\boldsymbol{\phi}^* = (0.5, -0.7, 0, 0, 0)^\top$. Also, we generate $\mathbf{x}_{i,t} = (x_{i,t,1}, \cdots, x_{i,t,p})^\top$ independently from multivariate normal distribution

where the pairwise correlation between x_{i,t,j_1} and x_{i,t,j_2} is $\rho^{|j_1-j_2|}$. Then we consider cases where $T_i = T$ for all $i \in \{1, \cdots, n\}$ with $T \in \{50, 100\}$ and $n \in \{10, 100\}$. For the standard deviation of the noise variable we have $\sigma = 0.5$ and for correlation coefficients we have $\rho \in \{0.25, 0.75\}$. The two values of correlation coefficients are used to show high and low linear correlation between the covariates. For each simulation setting, we repeat the experiment for 100 time, and the averaged value of precision and recall of the variable selection results for both $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\phi}}$ are reported. Here precision is defined as the fraction of selected variables that are true variables, and recall (also known as sensitivity) is defined as the fraction of the true variables that are selected by variable selection. Then we report the precision and recall of variable selection as well as the estimation error for $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\phi}}$. We can see from the reported Table 5.1 and 5.2 that when ρ is smaller, both models tend to perform better in terms of variable selection error and estimation error. Also, when we have higher n and T, the estimation and variables selection results are better.

Estimator	$\widehat{\boldsymbol{\beta}}$			$\widehat{\boldsymbol{\phi}}$		
	Pre. (%)	Rec. (%)	$\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2$	Pre. (%)	Rec. (%)	$\|\widehat{\boldsymbol{\phi}} - \boldsymbol{\phi}^*\|_2$
$n = 10, T = 50$						
Double Lasso	56.9	100.0	0.544	42.0	100.0	0.118
	(0.6)	(0.0)	(0.001)	(0.5)	(0.0)	(0.005)
Hierarchical	57.9	100.0	0.083	30.8	80.0	0.882
	(0.7)	(0.0)	(0.002)	(1.6)	(4.0)	(0.005)
$n = 100, T = 50$						
Double Lasso	84.5	100.0	0.021	66.7	100.0	0.476
	(1.3)	(0.0)	(0.001)	(0.0)	(0.0)	(0.002)
Hierarchical	78.7	100.0	0.025	40.0	100.0	0.866
	(1.2)	(0.0)	(0.001)	(0.0)	(0.0)	(0.002)
$n = 10, T = 100$						
Double Lasso	100.0	100.0	0.113	57.1	100.0	0.135
	(0.0)	(0.0)	(0.001)	(0.0)	(0.0)	(0.005)
Hierarchical	100.0	100.0	0.109	40.0	100.0	0.119
	(0.0)	(0.0)	(0.001)	(0.0)	(0.0)	(0.005)
$n = 100, T = 100$						
Double Lasso	50.0	100.0	0.015	66.6	100.0	0.460
	(0.0)	(0.0)	(0.035)	(0.0)	(0.0)	(0.001)
Hierarchical	53.1	100.0	0.012	40.0	100.0	0.029
	(0.6)	(0.0)	(0.031)	(0.0)	(0.0)	(0.001)

Table 5.1: Simulation Result for $\rho = 0.25$

Estimator	$\widehat{\boldsymbol{\beta}}$			$\widehat{\boldsymbol{\phi}}$		
	Pre. (%)	Rec. (%)	$\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2$	Pre. (%)	Rec. (%)	$\|\widehat{\boldsymbol{\phi}} - \boldsymbol{\phi}^*\|_2$
$n = 10, T = 50$						
Double Lasso	88.7	100.0	0.109	40.1	100.0	0.137
	(0.0)	(0.0)	(0.003)	(0.0)	(0.0)	(0.051)
Hierarchical	89.0	100.0	0.111	40.0	100.0	0.137
	(0.0)	(0.0)	(0.003)	(0.0)	(0.0)	(0.005)
$n = 100, T = 50$						
Double Lasso	50.0	100.0	0.034	40.0	100.0	0.049
	(0.0)	(0.0)	(0.001)	(0.0)	(0.0)	(0.002)
Hierarchical	50.0	100.0	0.034	40.0	100.0	0.050
	(0.0)	(0.0)	(0.001)	(0.0)	(0.0)	(0.002)
$n = 10, T = 100$						
Double Lasso	52.3	100.0	0.075	69.3	100.0	0.469
	(0.0)	(0.0)	(0.002)	(0.0)	(0.0)	(0.006)
Hierarchical	50.0	100.0	0.109	37.9	94.5	0.871
	(0.0)	(0.0)	(0.003)	(0.0)	(0.0)	(0.004)
$n = 100, T = 100$						
Double Lasso	50.0	100.0	0.022	48.6	100.0	0.029
	(0.0)	(0.0)	(0.001)	(0.0)	(0.0)	(0.001)
Hierarchical	50.0	100.0	0.023	40.0	100.0	0.254
	(0.0)	(0.0)	(0.001)	(0.0)	(0.0)	(0.002)

Table 5.2: Simulation Result for $\rho = 0.75$

Chapter 6

Real Data Analysis

To identify which variable selection techniques are appropriate for analyzing longitudinal regression modeling, we use a provincial dataset from Saskatchewan to examine which of the variable selection approach outperforms than others in terms of having a better prediction accuracy and a more interpretability of the resulting models. We aim to predict the frequency of hospital visits among those with incident dementia and depression and identify factors that are associated with the frequency of hospital visits. Data analyzed here is from the Saskatchewan health care utilization datafiles. The data files cover health services (for instance, hospital discharge and physician services) delivered to all Saskatchewan residents, which represent almost 99% of the provincial population. Because the comorbidity of dementia and depression is highly prevalent and an in-depth understanding the health service use of patients suffering from the comorbidity could not only help to better allocate the limited health services resources to meet the need of population with the comorbidity but also highlights the clinical treatment and management efforts towards to the subpopulation with more disadvantaged characteristics. The current study included all of the Saskatchewan residents who, in 2000, were eligible for health coverage, and were newly diagnosed with dementia and depression during the year of 2000 and 2006. A total of 328 patients who were diagnosed with incident of dementia and depression. The incidence density of dementia in the provincial popula-

tion of Saskatchewan in year 2000 was 0.01 per 1000 person for individuals between age 18-64 years and 3.13 per 10000 person for age 75 to 84. To demonstrate the variable selection approaches, we used 17 variables in the analysis. The description of the variables of the dataset corresponding to the variable name is included in 6.1. All the features of this dataset except *age2000* and *incidentage* are categorical, as such we one-hot encode the dataset to create dummy variables before applying our models to them. Specifically, for covariate *SEX*, we use binary variables *SEX_1* for male patients and *SEX_2* for female patients; for covariate *MARSTATI*, the *MARSTATI_1* is used for entries corresponding to whether or not an individual is single , for covariate *MARSTATI_2*, the *MARSTATI_2* is used for entries corresponding to whether or not an individual is married or in common-law, and the *MARSTATI_3* is used for entries corresponding to whether or not individual is Separated, divorced or widowed; for covariate *RESINDX*, the *RESINDX_1* is used for entries corresponding to whether or not the individual is living in urban area, and the *RESINDX_2* is used for entries corresponding to whether or not the individual is living in a rural area; for covariate *INCSECI*, the *INCSECI_0* is used for entries corresponding to not receiving any social welfare at the beginning of the study period, *INCSECI_1* is used for entries corresponding to receiving the Saskatchewan Assistance plan at the beginning of the study period, *INCSECI_2* is used for entries corresponding to receiving the Family-based income security benefits at the beginning of the study period, *INCSECI_3* is used for entries corresponding to receiving the Senior-based income security beginning of study period; for covariate *INCSECE*, the *INCSECE_0* is used for entries corresponding to not receiving any social welfare at the end of the study period, *INCSECE_1* is used for entries corresponding to receiving the Saskatchewan Assistance plan at the end of the study period, *INCSECE_3* is used for entries corresponding to receiving the Senior-based income security benefits at the end of the study period. After such reparameterization, the total number of covariates in the model is 17.

Variable	Description
age2000	Age at 2000
incidentyear	Year of incident of dementia
incidentage	Age at incident of dementia
SEX	Subject sex (1: "Male", 2: "Female")
MARSTATI	Marital status (1: "single", 2: "married/common-law", 3: "Separated, divorced, or widowed")
RESINDX	Residence Status (1: "urban", 2: "rural")
INCSECI	Income security benefits status at index (0: "None" , 1: "Saskatchewan Assistance plan", 2: "Family-based income security benefits", 3 "Senior- based income security benefits").
INCSECE	Income security benefits status at exit (0: "None" , 1: "Saskatchewan Assistance plan", 2: "Family-based income security benefits", 3 "Senior- based income security benefits").

Table 6.1: A description of the covariates in the psychological dataset

For this dataset, the yearly measurements were gathered from year of 1999 to year of 2006. Hence the total number of time points is 8. While training the models to this dataset first 7 time points were used for training and the last 2 time points were used for cross validation and reporting. We also note that the value of input does not change year to year, as all the variables in 6.1 for each individual are same across the time period. For example, *age2000* corresponding to subject's age at 2000, and contribution to the outcome remains constant between 1999 to 2006. Since the value of the input does not change, the values $X_{i,t}$ that it corresponds to in the model remains unchanged while the response variable $Y_{i,t}$ varies.

We model the response $Y_{i,t}$, which is the frequency of individual i's visit to the hospital for psychiatric treatment measured at time point t, as an explicit function of past response $Y_{i,t-1}, \cdots, Y_{i,t-q}$ and covariates $\mathbf{x}_{i,t}$. Here $t = 1$ corresponds to year 1999 and $t = 9$ corresponds to year 2006. The description of the response variable is provided in Table 6.2.

We apply three different kinds models to this dataset: (i) an unpenalized transition model (ii) a double Lasso penalized transition model and (iii) a hierarchical penalized transition model. In order to tune the parameters for model (ii) and (iii), we used the cross validation method discussed in Section 4.3. Specifically, we train model (ii) and (iii)

Variable	Description
NumSPs1999	Number of service for psychiatric diagnoses in 1999
NumSPs2000	Number of service for psychiatric diagnoses in 2000
NumSPs2001	Number of service for psychiatric diagnoses in 2001
NumSPs2002	Number of service for psychiatric diagnoses in 2002
NumSPs2003	Number of service for psychiatric diagnoses in 2003
NumSPs2004	Number of service for psychiatric diagnoses in 2004
NumSPs2005	Number of service for psychiatric diagnoses in 2005
NumSPs2006	Number of service for psychiatric diagnoses in 2006

Table 6.2: Repeated measurements of the response variable in the psychological dataset

	Value
$\text{MSFE}^{(\hat{\lambda}_1^{\text{hier}}, \hat{\lambda}_2^{\text{hier}})}$	12513
$\text{MSFE}^{(\hat{\lambda}_1^{\text{dlasso}}, \hat{\lambda}_2^{\text{dlasso}})}$	12497
$\text{MSFE}^{(10^{-6}, 10^{-6})}$	4817155

Table 6.3: MSFE values for model (i), (ii) and (iii) on Psychological dataset

using the first seven time points and validate the fitted model using the last two time points, i.e. we set $T_0 = 7$ and $T_1 = 9$ in (4.2) to obtain the MSFEs in the model selection procedure. The optimal values $(\hat{\lambda}_1^{\text{dlasso}}, \hat{\lambda}_2^{\text{dlasso}})$ corresponding to the smallest MSFEs are chosen for double Lasso penalized models and $(\hat{\lambda}_1^{\text{hier}}, \hat{\lambda}_2^{\text{hier}})$ are chosen for the hierarchical penalized models, respectively.

In order to compare the performance of model (i), (ii), and (iii), we compare their MSFE values, which are presented in Table 6.3. Both double lasso and hierarchical penalization model have similar MSFE. They have noticeably lower values than model (iii). Although we cannot conclude which one of these two penalization model are better than the other, they perform better than the unpenalized model.

Additionally, we study the importance of each variable in $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\phi}}$ by using the solution paths. The solution path can show which of the variables become non-zero as the amount of sparse penalization decreases. The plot demonstrates the order in which the variables are added to the model are shown in Figure 6.1, and the corresponding results are reported in Table. We plot the solution path with varying λ_1 and λ_2 for both the hi-

erarchical penalized model and the double Lasso penalized model. For example, for plot a), we show the solution path of $\widehat{\boldsymbol{\beta}}$ for the double lasso case. We fix λ_2 at $\hat{\lambda}_2^{\text{dlasso}}$ selected by cross validation and fit the double lasso for a sequence of 30 different λ_1 values spanning across $\{10^{-6}, 10^{-5.8}, \cdots, 10^{-1}\}$. Then we plot the fitted values $\widehat{\boldsymbol{\beta}}$ at each λ_1. Each line in the the Figure 4(a) corresponds to the value of $\hat{\beta}_i$, the i-th element of $\widehat{\boldsymbol{\beta}}$. On the y-axis we have the values of $\hat{\beta}_i$ at each λ_1, and on the x-axis, we have the values of corresponding λ_1 on the logarithmic scale. For Figure 4(b), we plot the solution path of $\widehat{\boldsymbol{\phi}}$ for the double lasso. Here, we fix λ_1 on $\hat{\lambda}_1^{\text{dlasso}}$ and then fit the double lasso while varying λ_2. Then we plot the values of $\widehat{\boldsymbol{\phi}}$ at each λ_2. Again, each line corresponds to the value of $\hat{\phi}_i$, i-th coefficient in $\widehat{\boldsymbol{\phi}}$. For Figure 4(c) and (d), we carry out this exact same procedure except that we use the hierarchical penalization instead of the double Lasso.

The results shown in Figure 4 demonstrate which variables in the models are deemed more important. The variables that become non-zero at a smaller penalization value are less relevant to the prediction of the response variable. In the double Lasso model, we see that $\hat{\phi}_1$ becomes non-zero at highest value of λ_1, followed by $\hat{\phi}_2$ at second highest and so on. In hierarchical Lasso model, all the $\hat{\phi}_j's$ becomes non-zero at the same λ value. These results are displayed in Table 6.4 where we see the order variables in which they enter the model is also shown. Variables including *age2000, incidentyear*, and *incidentage* enter the model with higher λ values than the other variables indicating that they are the most important variables for determining the health services usage across the given time period.

Table 6.6 shows the fitted values of both double lasso and hierachical penalization $(\hat{\lambda}_1^{\text{dlasso}}, \hat{\lambda}_2^{\text{dlasso}})$ and $(\hat{\lambda}_1^{\text{hier}}, \hat{\lambda}_2^{\text{hier}})$, respectively. These values are more complicated to interpret as the response variable depends on both $\boldsymbol{\phi}$ and $\boldsymbol{\beta}$ as well as the presence of dummy variables that are present in the model. Let $\boldsymbol{\beta}^*$ and $\boldsymbol{\phi}^*$ be the fitted values from the model and $\boldsymbol{\beta}^*_{-j} = (\beta_1^*, \cdots \beta_{j-1}^*, \beta_{j+1}^*, \cdots, \beta_p^*) \in \mathbb{R}^{p-1}$. We also denote $x_{i,t,j}$ as the binary dummy variable mentioned earlier in this section. From 1.1, that the expected value of the response variable can be calculated as

$$\begin{aligned}\mathbb{E}[Y_{i,t}|x_{i,t,j}] &= \mathbf{x}_{i,t}^{\top}\boldsymbol{\beta}^{*} + \sum_{r=1}^{q}\phi_r^{*}(Y_{i,t-r} - \mathbf{x}_{i,t-r}^{\top}\boldsymbol{\beta}^{*}),\\ &= x_{i,t,j}\beta_j^{*} + \mathbf{x}_{i,t-j}^{\top}\boldsymbol{\beta}_{-j}^{*} + \sum_{r=1}^{q}\phi_r^{*}(Y_{i,t-r} - x_{i,t-r,j}\beta_j^{*} - \mathbf{x}_{i,t,-j}^{\top}\boldsymbol{\beta}_{-j}^{*}).\end{aligned}$$

Now, given that dummy variables take the value 0 or 1, the expected value of response variable when $x_{i,t,j}$ is 0 and 1 are

$$\mathbb{E}[Y_{i,t}|x_{i,t,j}=0] = \mathbf{x}_{i,t-j}^{\top}\boldsymbol{\beta}_{-j}^{*} + \sum_{r=1}^{q}\phi_r^{*}(Y_{i,t-r} - \mathbf{x}_{i,t,-j}^{\top}\boldsymbol{\beta}_{-j}^{*}),$$

and

$$\mathbb{E}[Y_{i,t}|x_{i,t,j}=1] = \beta_j^{*} + \mathbf{x}_{i,t-j}^{\top}\boldsymbol{\beta}_{-j}^{*} + \sum_{r=1}^{q}\phi_r^{*}(Y_{i,t-r} - \beta_j^{*} - \mathbf{x}_{i,t,-j}^{\top}\boldsymbol{\beta}_{-j}^{*})$$

respectively. Hence the difference between these two expected value is

$$\begin{aligned}\mathbb{E}[Y_{i,t}|x_{i,t,j}=1] - \mathbb{E}[Y_{i,t}|x_{i,t,j}=0] &= \beta_j^{*} + \sum_{r=1}^{q}\phi_r^{*}(-\beta_j^{*})\\ &= \beta_j^{*}(1 - \sum_{r=1}^{q}\phi_r^{*}).\end{aligned}$$

This means that when dummy variable $x_{i,t,j}$ is 1, then the expected value $Y_{i,t}$ increases by $\beta_j^{*}(1 - \sum_{r=1}^{q}\phi_r^{*})$. Now to put this in the context of the data, we consider an individual with following properties:

1. Being a male
2. Being single
3. Living in urban areas

The results from the hierarchical penalization model indicate:

- The expected value of frequency of hospital visit changes by $-0.162 \times [1 - (0.735 - 0.201 + 0.220 - 0.166 - 0.027)] = -0.071$ due to this individual being a male.

- The expected value of frequency of hospital visit changes by $0.419 \times [1 - (0.735 - 0.201 + 0.220 - 0.166 - 0.027)] = 0.184$ due to this individual being single.

- The expected value of frequency of hospital visit changes by $0.048 \times [1 - (0.735 - 0.201 + 0.220 - 0.166 - 0.027)] = 0.021$ due to this individual living in urban areas.

For continous variables *age2000, incidentyear*, and *incidentage*, we can interpret the result as folllowing:

- One unit increase of *age2000,* changes the expected value of frequency of hospital visit by $-0.499 \times [1 - (0.735 - 0.201 + 0.220 - 0.166 - 0.027)] = -0.219$

- One unit increase of *incidentyear,* changes the expected value of frequency of hospital visit by $0.003 \times [1 - (0.735 - 0.201 + 0.220 - 0.166 - 0.027)] = 0.001$

- One unit increase of *incidentage,* changes the expected value of frequency of hospital visit by $0.465 \times [1 - (0.735 - 0.201 + 0.220 - 0.166 - 0.027)] = 0.204$

Similar calculation for the double Lasso model shows that:

- The expected value of frequency of hospital visit changes by $0.877 \times [1 - (0.614 - 0.096 + 0.007 - 0.029 - 0.007)] = 0.448$ due to this individual being a male.

- The expected value of frequency of hospital visit changes by $0.782 \times [1 - (0.614 - 0.096 + 0.007 - 0.029 - 0.007)] = 0.400$ due to this individual being single.

- No changes are expected to the value of frequency of hospital visit due to this individual living in urban areas.

Note that the result of the Double Lasso penalization model is more stable and efficient than the hierarchical Lasso penalization model while training on the datasets. Therefore, we recommend to use the results from Double Lasso penalization model. To summa-

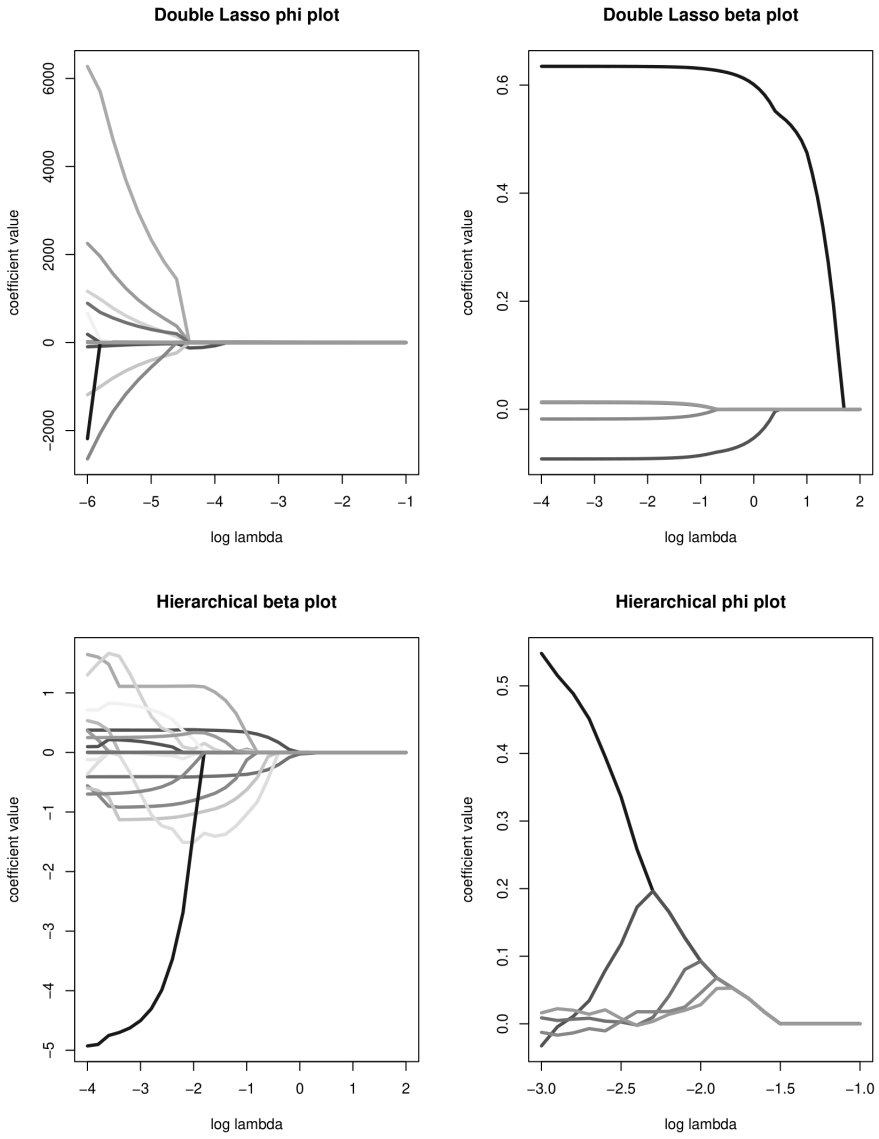

Figure 6.1: Solution paths for (a) $\widehat{\beta}$ for the double Lasso case; (b) $\widehat{\phi}$ for the double Lasso case;

Variable	Double Lasso	Hierarchical Lasso
age2000	3	2
incidentyear	1	1
incidentage	2	3
SEX	4	6
MARSTATI	6	5
RESINDX	8	4
INCSECI	7	8
INCSECE	5	7

Table 6.4: Order of entrance of variables in $\hat{\boldsymbol{\beta}}$ into the model

Variable	Double Lasso	Hierarchical Lasso
ϕ_1	1	(1)
ϕ_2	2	(1)
ϕ_3	3	(1)
ϕ_4	4	(1)

Table 6.5: Order of entrance of variables in $\hat{\boldsymbol{\phi}}$ into the model

Variable	Hierarchical Lasso	Double Lasso
age2000	-0.499	-0.137
incidentyear	0.003	0.702
incidentage	0.465	-0.017
SEX_1	-0.162	0.877
SEX_2	0.071	18.936
MARSTATI_1	0.419	0.782
MARSTATI_2	0	0
MARSTATI_3	-0.828	1.184
RESINDX_1	0.048	0
RESINDX_2	-1.386	0
INCSECI_0	0	0
INCSECI_1	0	0.436
INCSECI_2	0	-1.489
INCSECI_3	0	0.126
INCSECE_0	0	0
INCSECE_1	0	0
INCSECE_3	0	0.311

Table 6.6: Value of fitted $\boldsymbol{\beta}$

Variable	Hierarchical Lasso	Double Lasso
ϕ_1	0.735	0.614
ϕ_2	-0.201	-0.096
ϕ_3	0.220	0.007
ϕ_4	-0.166	-0.029
ϕ_5	-0.027	-0.007

Table 6.7: Value of fitted $\boldsymbol{\phi}$

rize, the results from the plot (6.1) and the tables (6.4), and (6.6) allow us to interpret the model. Table (6.4) shows the significance of the variables to the model in which the order of entrance indicating the order of significance of variables to the model. Here, variables that enter the model at lower order are higher in significance compared to the models that enter the model at a higher order. Since the expected value of the response variable $Y_{i,t}$ depends on both $\boldsymbol{\beta}$ and $\boldsymbol{\phi}$, we could interpet that the change of the expected value of frequency of hospital visit of an individual due to these values. Analyzing the values in Table (6.6) for the Double Lasso, we can find that being female increases the expected value of frequency of hospital visit more compared to male. For marital status, "Separated, divorced, or widowed" increases the expected value more than being "Single". For residential status, there are no significant difference between living in urban areas and living in rural ares. For income security benefits status at index, receiving in "Saskatchewan Assistance plan" has a higher increase followed by "Senior- based income security benefits", and then "Family-based income security benefits". For income security benefits status at exit, being "Senior- based income security benefits" increases expected value of response variable. Similar analysis can be made with the hierarchical Lasso but the results are less reliable than Double Lasso implementation.

Chapter 7

Conclusion

Longitudinal data analysis is frequently used to model changes in the outcome of interest overtime. In the present study, we provided an overview of usage of traditional machine learning methods such as penalization techniques in longitudinal data analysis and compare the penalization methods that induce sparsity in the coefficients while constructing a hierarchy between the variables. There are many penalization methods that induces sparsity. One of the most widely used model for variable selection is Lasso. Lasso selects variables that are important in predicting the response variables by setting less important variables to zero. Variants of Lasso can prompt many different properties in the coefficient while maintaining its sparsity. For example, sparse group Lasso can induce sparsity on the coefficients while maintaining group-wise sparsity. Hierarchical group Lasso can further incorporate hierarchy within its coefficients. We proposed similar work in the longitduinal data analysis. In longitudinal data, there are two coefficients that control the response variable: coefficient β and autoregressive coefficient ϕ, which allows time dependent effect to the response variable. First model we used is double Lasso where we applied Lasso on β and ϕ. This model is intended to give sparsity to both β and ϕ. Second model we investigated is hierarchical Lasso penalization model. Similar to the hierarchical group lasso, we intend to induce chain-like hierarchy on ϕ. We provide the details of how this algorithm is applied, showing the optimization process that uses

iterative steps to first optimize for $\boldsymbol{\beta}$ and subsequently, $\boldsymbol{\phi}$. Also, we discuss the parameters and settings used on `glmnet` and `SPAMS` package in order to run the algorithm as intended. We demonstrated the results of these two algorithms on a simulated dataset, with varying size n and time points T. To correctly select optimal hyper-parameters, we used a modified cross validation (CV) approach to account for the temporal aspect of this dataset. Lastly, we test the results of these algorithms on a real dataset to examine which variable selection approach outperforms than others in terms of having a better prediction accuracy and a more interpretability of the resulting models. In the demonstration study, we aim to model the frequency of hospital visits among those with incident dementia in the province of Saskatchewan during 2000-2006. We offer step-by-step details about the selection process and interpretation based on the results of these two models. Overall, base on both simulated dataset and and demonstration studies we conclude that these penalized models outperform than unpenalized ones and the double Lasso model provides more stable and robust results.

Bibliography

[Bańbura et al., 2010] Bańbura, M., Giannone, D., and Reichlin, L. (2010). Large bayesian vector auto regressions. *Journal of applied Econometrics*, 25(1):71–92.

[Beck and Teboulle, 2009] Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202.

[Cardot and Sarda, 2005] Cardot, H. and Sarda, P. (2005). Estimation in generalized linear models for functional data via penalized likelihood. *Journal of Multivariate Analysis*, 92(1):24–41.

[Efron et al., 2004] Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al. (2004). Least angle regression. *The Annals of statistics*, 32(2):407–499.

[Friedman et al., 2010a] Friedman, J., Hastie, T., and Tibshirani, R. (2010a). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1.

[Friedman et al., 2010b] Friedman, J., Hastie, T., and Tibshirani, R. (2010b). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

[Groll and Tutz, 2014] Groll, A. and Tutz, G. (2014). Variable selection for generalized linear mixed models by l 1-penalized estimation. *Statistics and Computing*, 24(2):137–154.

[Gutterman et al., 1999] Gutterman, E. M., Markowitz, J. S., Lewis, B., and Fillit, H. (1999). Cost of alzheimer's disease and related dementia in managed-medicare. *Journal of the American Geriatrics Society*, 47(9):1065–1071.

[Jenatton et al., 2011] Jenatton, R., Audibert, J.-Y., and Bach, F. (2011). Structured variable selection with sparsity-inducing norms. *The Journal of Machine Learning Research*, 12:2777–2824.

[Jenatton et al., 2012] Jenatton, R., Gramfort, A., Michel, V., Obozinski, G., Eger, E., Bach, F., and Thirion, B. (2012). Multiscale mining of fmri data with hierarchical structured sparsity. *SIAM Journal on Imaging Sciences*, 5(3):835–856.

[Jenatton et al., 2010] Jenatton, R., Mairal, J., Obozinski, G., and Bach, F. R. (2010). Proximal methods for sparse hierarchical dictionary learning.

[Johnson et al., 2008] Johnson, B. A., Lin, D., and Zeng, D. (2008). Penalized estimating functions and variable selection in semiparametric regression models. *Journal of the American Statistical Association*, 103(482):672–680.

[Kim and Xing, 2010] Kim, S. and Xing, E. P. (2010). Tree-guided group lasso for multi-task regression with structured sparsity. In *ICML*, volume 2, page 1. Citeseer.

[Kunik et al., 2003] Kunik, M. E., Snow, A. L., Molinari, V. A., Menke, T. J., Souchek, J., Sullivan, G., and Ashton, C. M. (2003). Health care utilization in dementia patients with psychiatric comorbidity. *The Gerontologist*, 43(1):86–91.

[Liang and Zeger, 1986] Liang, K.-Y. and Zeger, S. L. (1986). Longitudinal data analysis using generalized linear models. *Biometrika*, 73(1):13–22.

[Mairal, 2014] Mairal, J. (2014). Spams: a sparse modeling software, v2. 5.

[Mairal et al., 2010] Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2010). Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(1).

[Mairal et al., 2011] Mairal, J., Jenatton, R., Obozinski, G., and Bach, F. (2011). Convex and network flow optimization for structured sparsity. *Journal of Machine Learning Research*, 12(9).

[Martins et al., 2011] Martins, A. F., Smith, N. A., Figueiredo, M., and Aguiar, P. (2011). Structured sparsity in structured prediction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1500–1511.

[Nicholson et al., 2017] Nicholson, W., Matteson, D., and Bien, J. (2017). Bigvar: Tools for modeling sparse high-dimensional multivariate time series. *arXiv preprint arXiv:1702.07094*.

[Obozinski et al., 2011] Obozinski, G., Jacob, L., and Vert, J.-P. (2011). Group lasso with overlaps: the latent group lasso approach. *arXiv preprint arXiv:1110.0413*.

[Rapaport et al., 2008] Rapaport, F., Barillot, E., and Vert, J.-P. (2008). Classification of arraycgh data using fused svm. *Bioinformatics*, 24(13):i375–i382.

[She, 2012] She, Y. (2012). An iterative algorithm for fitting nonconvex penalized generalized linear models with grouped predictors. *Computational Statistics & Data Analysis*, 56(10):2976–2990.

[Simon et al., 2013] Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2013). A sparse-group lasso. *Journal of computational and graphical statistics*, 22(2):231–245.

[Song and Bickel, 2011] Song, S. and Bickel, P. J. (2011). Large vector auto regressions. *arXiv preprint arXiv:1106.3915*.

[Stiratelli et al., 1984] Stiratelli, R., Laird, N., and Ware, J. H. (1984). Random-effects models for serial observations with binary response. *Biometrics*, pages 961–971.

[Tibshirani, 1996] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

[Wang et al., 2007] Wang, H., Li, G., and Tsai, C.-L. (2007). Regression coefficient and autoregressive order shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(1):63–78.

[Wang et al., 2012] Wang, L., Zhou, J., and Qu, A. (2012). Penalized generalized estimating equations for high-dimensional longitudinal data analysis. *Biometrics*, 68(2):353–360.

[Yuan and Lin, 2006] Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67.

[Zeng and Cook, 2007] Zeng, L. and Cook, R. J. (2007). Transition models for multivariate longitudinal binary data. *Journal of the American Statistical Association*, 102(477):211–223.

[Zhao and Yu, 2006] Zhao, P. and Yu, B. (2006). On model selection consistency of lasso. *Journal of Machine learning research*, 7(Nov):2541–2563.

[Zou, 2006] Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429.

[Zou and Hastie, 2005] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320.