Title

Joint Modeling of Longitudinal and Survival Data via Multivariate Mixed Effects State Space Model

Author

Luo, Ya

Publication Date

2018

University of California
Santa Barbara

Joint Modeling of Longitudinal and Survival Data via Multivariate Mixed Effects State Space Model

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Statistics and Applied Probability

by

Ya Luo

Committee in charge:

Professor Yuedong Wang, Chair
Professor S. Rao Jammalamadaka
Professor Wendy Meiring

December 2018

The Dissertation of Ya Luo is approved.

Professor S. Rao Jammalamadaka

Professor Wendy Meiring

Professor Yuedong Wang, Committee Chair

December 2018

Joint Modeling of Longitudinal and Survival Data via
Multivariate Mixed Effects State Space Model

by

Ya Luo

To my parents

Acknowledgements

I'm so lucky to have Professor Yuedong Wang as my PhD advisor. Professor Wang is a mentor and a friend, guiding me patiently and wisely through challenging problems. He has given me so much help and support throughout the PhD program and the writing of this dissertation. I wouldn't be where I am right now without his help.

I want to thank my committee members, Professor Wendy Meiring and Professor S. Rao Jammalamadaka, for their advice on my dissertation, and for their help over the past five years.

This research was supported by a grant from the National Science Foundation (DMS-1507620). We acknowledge support from the Center for Scientific Computing from the CNSI, MRL: an NSF MRSEC (DMR-1121053).

We thank Fresenius Medical Care North America for providing de-identified data.

I want to express my gratefulness towards all my professors for their teaching and advice during the past five years at UCSB.

I would also like to thank my piano professor Robert Koenig from the music department. Professor Koenig has immense passion for music and tremendous love for students. I not only took piano lessons with him, but learned to be a better person. His influence on me was life-changing.

I want to thank my friends Nhan, Ti, Zach, Laura, Redilyn, Bret, Jiaye, Shelly, Qianyu, Aditya, Yuanbo, and Youhong, for their friendship and support. Their company made the difficult times a lot warmer and happier.

And last, I want to thank my parents, for always being there and being supportive.

Curriculum Vitæ
Ya Luo

Education

2013 - 2018 Ph.D. in Statistics, University of California, Santa Barbara.

Honors: Abraham Wald Memorial Prize, 2014 (Awarded to a graduate student for excellence in graduate studies as selected by the faculty of the Department of Statistics and Applied Probability in memory of Dr. Abraham Wald, eminent American statistician.)

2009 - 2013 B.S. in Mathematics, Beihang University.

Honors: National Scholarship, 2010 (Issued by the Ministry of Education of P. R. China)

Publications

- "Intradialytic Hypertension is Associated with Low Intradialytic Arterial Oxygen Saturation". Anna Meyring-Wösten, Ya Luo, Hanjie Zhang, Stephan Thijssen, Yuedong Wang, Peter Kotanko. Nephrology Dialysis Transplantation, Volume 33, Issue 6, 1 June 2018, Pages 1040-1045, https://doi.org/10.1093/ndt/gfx309
- Abstract "Association Between Peridialytic Systolic Blood Pressure Changes and Arterial Oxygen Saturation: Results from a Large U.S. Hemodialysis Cohort" accepted by American Society of Nephrology (ASN) for poster presentation (Nov. 2016). Authors: Anna Meyring-Wösten, Ya Luo, Hanjie Zhang, Stephan Thijssen, Yuedong Wang, Peter Kotanko.

Abstract

Joint Modeling of Longitudinal and Survival Data via
Multivariate Mixed Effects State Space Model

by

Ya Luo

State space models are powerful in modeling dynamic processes and at the same time have clear interpretations. Due to their flexibility and interpretability, mixed effects state space models have been studied in the literature for the modeling of multivariate longitudinal data. In a multivariate mixed effects state space model, the population effects and subject random deviations of any variable can be modeled by different stochastic processes. These processes can differ between variables, allowing great flexibility in the modeling. In addition, the model provides multiple ways to characterize interactions between the variables. However, the expensive computational cost is a major hindrance to the application of the mixed effects state space model to data with large numbers of individuals. Let m be the number of individuals. The current most efficient version of the Kalman filter, the univariate treatment, has time complexity $\mathrm{O}(m^3)$ and space complexity $\mathrm{O}(m^2)$. The univariate treatment can handle only a few hundred individuals at a high computational cost. We discover special structures in the Kalman filter of the mixed effects state space model and develop a new algorithm to exploit these structures. This reduces both time and space complexity to $\mathrm{O}(m)$ and enables easy modeling of hundreds of thousands of individuals without parallel computing, although it is also highly parallelizable. We further extend the mixed effects state space model to a joint modeling framework, in which a mixed effects state space model characterizes longitudinal data and a logistic regression models the survival probability. The true values of the longitudinal variables, modeled by the latent state of the state space model, are used as predictors in the logistic regression. Our joint model can (i) characterize the evolution of longitudinal variables and interactions between them, (ii) model the relationship between the survival probability and longitudinal variables/external covariates, and (iii) perform online predictions for longitudinal variables and survival probability. We develop another efficient algorithm for the computation of the maximum likelihood estimates of parameters in the joint model with time and space complexity both linear in m.

Contents

Chapter 1

Introduction

1.1 Mixed Effects Models for Longitudinal Data

Longitudinal data contain repeated measurements of individuals over time. Compared to cross-sectional data in which the measurements of different subjects are collected at different time points, longitudinal data eliminate sample difference by track- ing the same subjects, hence more accurately characterize the change in variables over time. Longitudinal data are often modeled by mixed effects models, in which subject deviations are modeled by random effects and the population mean is modeled by fixed effects. A linear mixed effects model (Laird and Ware [1]) assumes that

$$y_i(t_{ij}) = \boldsymbol{x}_i^T(t_{ij})\boldsymbol{\beta} + \boldsymbol{z}_i^T(t_{ij})\boldsymbol{b}_i + \epsilon_i(t_{ij}), \tag{1.1}$$

where $y_i(t_{ij})$ is the observation of the response variable at time t_{ij} from subject i, $i = 1, \ldots, m$, $j = 1, \ldots, n_i$; $\boldsymbol{x}_i(t_{ij})$ is a $p \times 1$ vector of covariates associated with the fixed effects $\boldsymbol{\beta}$; $\boldsymbol{z}_i(t_{ij})$ is a $q \times 1$ vector associated with the random effects $\boldsymbol{b}_i \overset{iid}{\sim} N_q(\boldsymbol{0}, D)$; and $\epsilon_i(t_{ij})$ are random errors with $\boldsymbol{\epsilon}_i = (\epsilon_i(t_{i1}), \ldots, \epsilon_i(t_{in_i}))^T \sim N(\boldsymbol{0}, R_i)$. Given the random effects $\boldsymbol{b}_i$, the observations of subject i at time t_{ij} fluctuate around $\boldsymbol{x}_i(t_{ij})\boldsymbol{\beta} + \boldsymbol{z}_i(t_{ij})\boldsymbol{b}_i$, hence the random effects characterize between-subject variation.

Model (1.1) can be written in a vector form. Let $\boldsymbol{y}_i = (y_i(t_{i1}), \ldots, y_i(t_{in_i}))^T$ be the observation vector of subject i,

$$X_i = \begin{pmatrix} \boldsymbol{x}_i^T(t_{i1}) \\ \vdots \\ \boldsymbol{x}_i^T(t_{in_i}) \end{pmatrix}, \quad Z_i = \begin{pmatrix} \boldsymbol{z}_i^T(t_{i1}) \\ \vdots \\ \boldsymbol{z}_i^T(t_{in_i}) \end{pmatrix}, \quad \text{and } \boldsymbol{\epsilon}_i = \begin{pmatrix} \epsilon_i(t_{i1}) \\ \vdots \\ \epsilon_i(t_{in_i}) \end{pmatrix}, \tag{1.2}$$

then model (1.1) can be written as

$$\boldsymbol{y}_i = X_i\boldsymbol{\beta} + Z_i\boldsymbol{b}_i + \boldsymbol{\epsilon}_i, \quad i = 1, \ldots, m. \tag{1.3}$$

Furthermore, one can stack the observations of all subjects and write the model in a general linear mixed effects form. Let

$$\boldsymbol{y} = \begin{pmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_m \end{pmatrix}, \quad \boldsymbol{X} = \begin{pmatrix} X_1 \\ \vdots \\ X_m \end{pmatrix}, \quad Z = \begin{pmatrix} Z_1 & & 0 \\ & \ddots & \\ 0 & & Z_m \end{pmatrix},$$

$$\boldsymbol{b} = \begin{pmatrix} \boldsymbol{b}_1 \\ \vdots \\ \boldsymbol{b}_m \end{pmatrix}, \quad \text{and } \boldsymbol{\epsilon} = \begin{pmatrix} \boldsymbol{\epsilon}_1 \\ \vdots \\ \boldsymbol{\epsilon}_m \end{pmatrix},$$

we have

$$\boldsymbol{y} = X\boldsymbol{\beta} + Z\boldsymbol{b} + \boldsymbol{\epsilon}, \tag{1.4}$$

where $\boldsymbol{y}$ is an observation vector of dimension $n \times 1$, $n = \sum_{i=1}^m n_i$, X is an $n \times p$ design matrix for the fixed effects $\boldsymbol{\beta}$, Z is an $n \times qm$ design matrix for the random effects

$\boldsymbol{b}$, $\boldsymbol{b} \sim N(\mathbf{0}, G)$, $G = \text{diag}\{D, \ldots, D\}$, $\boldsymbol{\epsilon} \sim N(\mathbf{0}, R)$ is a vector of random errors, and $R = \text{diag}\{R_1, \ldots, R_m\}$.

Suppose that the covariance matrices G and R depend on a parameter vector $\boldsymbol{\theta}$. We need to estimate parameters $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$. According to (1.4),

$$\boldsymbol{y} \sim N(X\boldsymbol{\beta}, W^{-1}), \tag{1.5}$$

where $W^{-1} = ZGZ^T + R$ depends on $\boldsymbol{\theta}$. Maximizing the log-likelihood

$$l(\boldsymbol{\beta}, \boldsymbol{\theta}) = \log\, p(\boldsymbol{y}|\boldsymbol{\beta}, \boldsymbol{\theta}) = \text{constant} + \frac{1}{2}\log|W| - \frac{1}{2}(\boldsymbol{y} - X\boldsymbol{\beta})^T W(\boldsymbol{y} - X\boldsymbol{\beta}) \tag{1.6}$$

gives the maximum likelihood estimate (MLE) of $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$. For a fixed $\boldsymbol{\theta}$, maximizing the likelihood with respect to $\boldsymbol{\beta}$ results in an analytical expression of $\boldsymbol{\beta}$ as a function of $\boldsymbol{\theta}$,

$$\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}(\boldsymbol{\theta}) = (X^T W X)^{-1} X^T W \boldsymbol{y}, \tag{1.7}$$

which is the same as the generalized least squares (GLS) estimate. Plugging (1.7) back into the log-likelihood (1.6) gives the profiled likelihood

$$l(\hat{\boldsymbol{\beta}}, \boldsymbol{\theta}) = \text{constant} + \frac{1}{2}\log|W| - \frac{1}{2}(\boldsymbol{y} - X\hat{\boldsymbol{\beta}})^T W(\boldsymbol{y} - X\hat{\boldsymbol{\beta}}), \tag{1.8}$$

which is a function of $\boldsymbol{\theta}$ only. The MLE of $\hat{\boldsymbol{\theta}}$ is the maximizer of (1.8), and the MLE of $\boldsymbol{\beta}$ is $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}(\hat{\boldsymbol{\theta}})$. Usually there is no closed form solution for $\boldsymbol{\theta}$ and a numerical optimization procedure such as the EM algorithm or Newton-Raphson is employed.

The maximum likelihood approach, however, results in biased estimates of the variance components $\boldsymbol{\theta}$ due to loss in the degrees of freedom for estimating $\boldsymbol{\beta}$, especially when the dimension of $\boldsymbol{\beta}$ is large relative to the number of observations. Restricted maximum likelihood (REML) corrects this bias problem by eliminating fixed effects $\boldsymbol{\beta}$ from the likelihood and estimating $\boldsymbol{\theta}$ based on the reduced data.

Suppose that the $n \times p$ design matrix X is of rank r. Let O be an $(n-r) \times n$ matrix of full row rank satisfying $OX = 0$, we have

$$O\boldsymbol{y} \sim N(\mathbf{0}, OW^{-1}O^T), \tag{1.9}$$

in which the distribution of $O\boldsymbol{y}$ is independent of $\boldsymbol{\beta}$. The likelihood of $O\boldsymbol{y}$ is referred to as the restricted likelihood and is used to estimate $\boldsymbol{\theta}$.

The joint density of $\boldsymbol{y}$ and $\boldsymbol{b}$ is

$$\begin{aligned} p(\boldsymbol{y}, \boldsymbol{b}|\boldsymbol{\theta}) &= p(\boldsymbol{y}|\boldsymbol{b}, \boldsymbol{\theta})p(\boldsymbol{b}|\boldsymbol{\theta}) \\ &= (2\pi)^{-\frac{n+qm}{2}}|R|^{-\frac{1}{2}}|D|^{-\frac{1}{2}}\exp\{-\tfrac{1}{2}[(\boldsymbol{y} - X\boldsymbol{\beta} - Z\boldsymbol{b})^T R^{-1}(\boldsymbol{y} - X\boldsymbol{\beta} - Z\boldsymbol{b}) + \\ &\quad \boldsymbol{b}^T G^{-1}\boldsymbol{b}]\}. \end{aligned} \tag{1.10}$$

By taking the logarithm of (1.10), one obtains the joint log-likelihood of $\boldsymbol{\beta}$ and $\boldsymbol{b}$ (Henderson [2]),

$$l(\boldsymbol{\beta}, \boldsymbol{b}) = \text{constant} - \frac{1}{2}\log|R| - \frac{1}{2}\log|D| - \frac{1}{2}[(\boldsymbol{y} - X\boldsymbol{\beta} - Z\boldsymbol{b})^T R^{-1}(\boldsymbol{y} - X\boldsymbol{\beta} - Z\boldsymbol{b}) + \boldsymbol{b}^T G^{-1}\boldsymbol{b}]. \tag{1.11}$$

Differentiating (1.11) with respect to $\boldsymbol{\beta}$ and $\boldsymbol{b}$ and equating to zero, we have

$$\begin{aligned} X^T R^{-1} X\boldsymbol{\beta} + X^T R^{-1} Z\boldsymbol{b} &= X^T R^{-1}\boldsymbol{y}, \\ Z^T R^{-1} X\boldsymbol{\beta} + (Z^T R^{-1} Z + G^{-1})\boldsymbol{b} &= Z^T R^{-1}\boldsymbol{y}. \end{aligned} \tag{1.12}$$

The solutions for $\boldsymbol{\beta}$ and $\boldsymbol{b}$ are

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= (X^T W X)^{-1} X^T W \boldsymbol{y}, \\ \hat{\boldsymbol{b}} &= (Z^T R^{-1} Z + G^{-1})^{-1} Z^T R^{-1} (\boldsymbol{y} - X\hat{\boldsymbol{\beta}}). \end{aligned} \tag{1.13}$$

The parametric assumption of fixed and random effects may be too restrictive in some cases. More flexible semi-parametric and non-parametric mixed effects models have been developed ([3], [4], [5], [6], [7], [8], [9]).

1.2 Joint Modeling of Longitudinal and Survival Data

Longitudinal studies track subjects over time and collect various types of information. The measurements include exogenous variables whose values are independent of other variables in the system and are usually measured accurately with negligible error, endogenous variables whose existence and values depend on other variables, and time to an event such as death. Both exogenous and endogenous variables can change over time. Of interest are the interconnections between these variables, specifically, (a) the within-subject trajectories of endogenous longitudinal variables and their association with exogenous variables and other endogenous variables, and (b) the relationship between time-to-event and longitudinal endogenous/exogenous variables.

Traditional survival analysis, such as the Cox proportional hazard model, characterizes the relationship between time-to-event and observed covariates under the ideal assumptions that data are available at all times and are measured without errors. However, these assumptions rarely hold in practice. First, the longitudinal variables are usually observed intermittently. Naive imputations such as Last Value Carried Forward (LVCF) result in biased parameter estimates ([10]). Second, observed longitudinal variables are usually not the true values—there are measurement errors and biological variations. Third, the distributions of longitudinal variables may change when the event is about to happen.

The complications in practice and the potential for biased inference gave rise to the method of joint modeling, which assumes that longitudinal and survival data depend on a common set of latent processes. A joint model typically consists of a submodel for longitudinal data and a submodel for survival data (see [11] and [12] for a review). Let T_i, C_i, $\boldsymbol{x}_i$, and $m_i(t)$ be the event time, censoring time, the vector of exogenous variables which may be time dependent, and the latent true longitudinal process for subject i, respectively. The actual observations are $V_i = \min\{T_i, C_i\}$, the event indicator $\triangle_i = I(T_i \leq C_i)$, and the observation $y_i(t)$ of latent process $m_i(t)$ with error at intermittent time points $t \in \{t_{i1}, \ldots, t_{in_i}\}$.

There are three main models for the latent process $m_i(t)$. The simplest is

$$m_i(t) = \alpha_{0i} + \alpha_{1i} t, \tag{1.14}$$

specifying the latent process as a linear function of time t. A more flexible model depicts the latent process as a smooth trajectory

$$m_i(t) = \boldsymbol{f}(t)\boldsymbol{\alpha}_i, \tag{1.15}$$

where $\boldsymbol{\alpha}_i$ is a vector of subject-specific time-invariant effects and $\boldsymbol{f}(t)$ is a vector of functions of time t. Other work ([13], [14], [15], [16], [17]) accounts for autocorrelation over time by adding a zero-mean stochastic process $U_i(t)$ to (1.15), namely,

$$m_i(t) = \boldsymbol{f}(t)\boldsymbol{\alpha}_i + U_i(t). \tag{1.16}$$

The actual observation of a longitudinal variable is modeled by

$$y_i(t) = m_i(t) + \epsilon_i(t), \tag{1.17}$$

where $\epsilon_i(t)$ is the measurement error and/or biological variation, either serially independent or has a covariance structure if autocorrelation is present and is not included in $m_i(t)$.

Parametric accelerate failure time models and semi-parametric proportional hazard models have been considered for the survival submodel. For example, a proportional hazard survival submodel assumes that

$$h_i(M_i(t), \boldsymbol{x}_i) = h_0(t)\text{exp}\{\boldsymbol{\gamma}^T\boldsymbol{x}_i + \alpha m_i(t)\}, \quad t > 0, \tag{1.18}$$

where $M_i(t)$ is the history of the latent process up to time t, and $h_0(t)$ is the baseline hazard function. The standard Cox model ([18]) in survival analysis leaves the baseline hazard $h_0(t)$ completely unspecified. However, in a joint modeling framework, such a semi-parametric approach often leads to underestimation of the standard errors of parameter estimates ([19]). It is therefore preferable to use a known parametric distribution for $h_0(t)$, such as Weibull, log-normal, and Gamma. Alternatively, one may model $h_0(t)$ non-parametrically using step functions or splines ([20], [21], [22]). The hazard $h_i(M_i(t), \boldsymbol{x}_i)$ may also depend on the first derivative of the latent process, in this case, the exponent in (1.18) will have an additional regression term for $m_i'(t)$.

Parameters in a joint model can be estimated by maximum likelihood. However, the evaluation of the joint likelihood of the two submodels involves numerical integration, which is computationally expensive and is thus a main limitation of joint models.

Extensions to multivariate longitudinal data have also been studied in the literature ([23], [24], [25]). Let $m_{ik}(t)$ be the latent process of the kth longitudinal variable at time t for subject i, $k = 1, \ldots, q$. Xu and Zeger [24] considered a linear mixed effects model

$$m_{ik}(t) = X_{ik}(t)\boldsymbol{\beta}_k + Z_{ik}(t)\boldsymbol{b}_{ik} + \epsilon_{ik}(t), \tag{1.19}$$

where $\boldsymbol{\beta}_k$ is a $p_k \times 1$ vector of fixed effects, $\boldsymbol{b}_{ik}$ is a $v_k \times 1$ vector of random effects, and $X_{ik}(t)$ and $Z_{ik}(t)$ are design matrices that can be time-dependent.

For each subject, there are two sources of correlation structures in the multivariate case: (i) correlation among repeated measurements for each variable, and (ii) correlation between different variables.

To account for the two correlation structures, Xu and Zeger [24] assumed that

$$\boldsymbol{b}_i = (\boldsymbol{b}_{i1}^T, \ldots, \boldsymbol{b}_{iq}^T)^T \overset{iid}{\sim} N(\mathbf{0}, G), \tag{1.20}$$

where random effects for different variables may be correlated. They also assumed that random errors are serially independent and mutually independent between different variables, i.e., $\epsilon_{ik}(t) \overset{independent}{\sim} N(0, \sigma_k^2)$. Correlation among repeated measurements within each variable is also accounted for by random effects.

Chi and Ibrahim [23] and Song et al. [25] also consider the mixed effects model (1.19) for multivariate longitudinal data with a different way to characterize correlation between variables:

$$\boldsymbol{\epsilon}_i(t) = (\boldsymbol{\epsilon}_{i1}(t), \ldots, \boldsymbol{\epsilon}_{iq}(t))^T \overset{iid}{\sim} N_q(\mathbf{0}, \Sigma) \text{ and } \boldsymbol{b}_{ik} \overset{independent}{\sim} N_{v_k}(\mathbf{0}, G_k), \tag{1.21}$$

$\boldsymbol{\epsilon}_i(t)$ are serially independent, and $\boldsymbol{\epsilon}_i(t)$ and $\boldsymbol{b}_{ik}$ are mutually independent. Covariance matrix Σ characterizes correlation between variables, and $\boldsymbol{b}_{ik}$ accounts for correlation among repeated measurements.

Chi and Ibrahim [23] argued that assumption (1.20) carries both of the two correlation structures in G alone, making it less straightforward to perform separate inferences about the different dependence structures. In addition, a model with assumption (1.20) has more parameters compared to a model with assumption (1.21), with a difference of $\frac{1}{2}\sum_{k\neq k',1\leq k,k'\leq q}^{q} v_k v_{k'} - \frac{1}{2}q(q-1)$, which increases with the dimension of random effects and the number of longitudinal variables.

The survival submodel in the multivariate case is similar to (1.18), where the exponent is now a linear combination of $\boldsymbol{x}_i$ and $m_{i1}(t), \ldots, m_{iq}(t)$.

1.3 State Space Models

This section provides an overview of the state space model, mainly based on Durbin and Koopman [26]. The state space model is a modeling approach that treats a variety of problems in which the system is dynamic and evolves over time. It assumes that the development of the system depends on a latent unobserved process $\boldsymbol{\alpha}_t$, where $t = 1, \ldots, n$ are observation time points, and the observations $\boldsymbol{y}_t$ are reflections of the latent process with added noise. Both $\boldsymbol{\alpha}_t$ and $\boldsymbol{y}_t$ can be either scalars or vectors, and are not constraint to have the same dimensionality.

The most general form of the state space model is given by

$$\begin{aligned}
\boldsymbol{y}_t &\sim p(\boldsymbol{y}_t|\boldsymbol{\alpha}_t), \\
\boldsymbol{\alpha}_{t+1} &\sim p(\boldsymbol{\alpha}_{t+1}|\boldsymbol{\alpha}_t), \\
\boldsymbol{\alpha}_1 &\sim p(\boldsymbol{\alpha}_1),
\end{aligned} \tag{1.22}$$

where $p(\boldsymbol{y}_t|\boldsymbol{\alpha}_t)$ is the conditional density of the observation $\boldsymbol{y}_t$ given the state $\boldsymbol{\alpha}_t$, specifying the relation between the observation and the state; $p(\boldsymbol{\alpha}_{t+1}|\boldsymbol{\alpha}_t)$ is the conditional density of the next state given the current state, characterizing the state evolution process; $p(\boldsymbol{\alpha}_1)$ is the distribution of the initial state, which can be known or unknown. The model is said to be linear when

$$\begin{aligned} \boldsymbol{y}_t &= Z_t\boldsymbol{\alpha}_t + \boldsymbol{\epsilon}_t, \\ \boldsymbol{\alpha}_{t+1} &= T_t\boldsymbol{\alpha}_t + R_t\boldsymbol{\eta}_t. \end{aligned} \tag{1.23}$$

If at least one of the dynamic equations for $\boldsymbol{y}_t$ and $\boldsymbol{\alpha}_t$ is not linear, the model is nonlinear. If $p(\boldsymbol{y}_t|\boldsymbol{\alpha}_t)$, $p(\boldsymbol{\alpha}_{t+1}|\boldsymbol{\alpha}_t)$, and $p(\boldsymbol{\alpha}_1)$ are all Gaussian, the model is said to be Gaussian. If at least one of $p(\boldsymbol{y}_t|\boldsymbol{\alpha}_t)$, $p(\boldsymbol{\alpha}_{t+1}|\boldsymbol{\alpha}_t)$, and $p(\boldsymbol{\alpha}_1)$ is not Gaussian, the model is non-Gaussian.

A particular type of non-Gaussian state space models comes up frequently in practice—models with a linear Gaussian signal, which assumes that

$$\begin{aligned} \boldsymbol{y}_t &\sim p(\boldsymbol{y}_t|Z_t\boldsymbol{\alpha}_t), \\ \boldsymbol{\alpha}_{t+1} &= T_t\boldsymbol{\alpha}_t + R_t\boldsymbol{\eta}_t, \quad \boldsymbol{\eta}_t \sim N(\mathbf{0}, Q_t), \end{aligned} \tag{1.24}$$

where $\boldsymbol{\theta}_t = Z_t\boldsymbol{\alpha}_t$ is referred to as the signal. The density $p(\boldsymbol{y}_t|Z_t\boldsymbol{\alpha}_t)$ can be non-Gaussian, and the relationship between $\boldsymbol{y}_t$ and $\boldsymbol{\theta}_t$ can be non-linear.

This thesis focuses on linear Gaussian state space models. A linear Gaussian state space model has the form

$$\begin{aligned} \boldsymbol{y}_t &= Z_t\boldsymbol{\alpha}_t + \boldsymbol{\epsilon}_t, & \boldsymbol{\epsilon}_t &\sim N(\mathbf{0}, H_t), \\ \boldsymbol{\alpha}_{t+1} &= T_t\boldsymbol{\alpha}_t + R_t\boldsymbol{\eta}_t, & \boldsymbol{\eta}_t &\sim N(\mathbf{0}, Q_t), \end{aligned} \tag{1.25}$$

for $t = 1, \ldots, n$, with initial distribution $\boldsymbol{\alpha}_1 \sim N(\boldsymbol{a}_1, P_1)$, where a_1 and P_1 can be known or unknown; $\boldsymbol{y}_t$ is a $p \times 1$ observation vector and $\boldsymbol{\alpha}_t$ is an unobserved $m \times 1$ state vector, Z_t, T_t, H_t, R_t, Q_t are system matrices, which may contain unknown parameters, and disturbance terms $\boldsymbol{\epsilon}_t$ and $\boldsymbol{\eta}_t$ are assumed to be mutually and serially independent. The first equation in (1.25) is referred to as the observation equation, and the second the state equation.

Occasionally, a state space model may contain mean adjustments, given by

$$\begin{aligned} \boldsymbol{y}_t &= Z_t\boldsymbol{\alpha}_t + \boldsymbol{d}_t + \boldsymbol{\epsilon}_t, & \boldsymbol{\epsilon}_t &\sim N(\mathbf{0}, H_t), \\ \boldsymbol{\alpha}_{t+1} &= T_t\boldsymbol{\alpha}_t + \boldsymbol{c}_t + R_t\boldsymbol{\eta}_t, & \boldsymbol{\eta}_t &\sim N(\mathbf{0}, Q_t), \\ & & \boldsymbol{\alpha}_1 &\sim N(a_1, P_1) \end{aligned} \tag{1.26}$$

for $t = 1, \ldots, n$, where $\boldsymbol{d}_t$ and $\boldsymbol{c}_t$ are known, but may depend on parameters.

1.3.1 Initialization

In many cases, the initial distribution of $\boldsymbol{\alpha}_1 \sim N(\boldsymbol{a}_1, P_1)$ is completely or partially unknown. In this situation, the initial state vector $\boldsymbol{\alpha}_1$ is written as

$$\boldsymbol{\alpha}_1 = \boldsymbol{a} + A\boldsymbol{\delta} + R_0\boldsymbol{\eta}_0, \quad \boldsymbol{\eta}_0 \sim N(\mathbf{0}, Q_0), \tag{1.27}$$

where the $m \times 1$ vector $\boldsymbol{a}$ contains the known constant part of $\boldsymbol{\alpha}_1$, $\boldsymbol{\delta}$ is a $q \times 1$ vector of unknown quantities, and $\boldsymbol{\eta}_0$ is an $(m-q)\times 1$ vector whose distribution is known. Matrices A and R_0 are selection matrices, i.e., columns of the identity matrix I_m, satisfying $A^T R_0 = 0$.

One may treat $\boldsymbol{\delta}$ as a random vector with distribution

$$\boldsymbol{\delta} \sim N(\mathbf{0}, \kappa I_q), \tag{1.28}$$

where $\kappa \to \infty$, and $\boldsymbol{\delta}$ is said to have a diffuse distribution. One may also treat $\boldsymbol{\delta}$ as an unknown deterministic vector and estimate it by maximum likelihood, using the augmented Kalman filter ([27], [26], [28]). These two approaches give the same numerical results (Durbin and Koopman [26]).

1.3.2 The Kalman filter

Kalman filter is an algorithm that aims at filtering out noise in the observation to estimate the unobserved latent state. At each time point, it accomplishes two things: (1) given the current observation, update the estimate for the current state, and (2) make predictions for the next state. Let $\boldsymbol{Y}_t$ be the vector of history observations up to time t, $\boldsymbol{Y}_t = (\boldsymbol{y}_1', \ldots, \boldsymbol{y}_t')'$, $t = 1, 2, \ldots, n$. Let $\boldsymbol{a}_t = \mathrm{E}(\boldsymbol{\alpha}_t | \boldsymbol{Y}_{t-1})$, $\boldsymbol{a}_{t|t} = \mathrm{E}(\boldsymbol{\alpha}_t | \boldsymbol{Y}_t)$, $P_t = \mathrm{Var}(\boldsymbol{\alpha}_t | \boldsymbol{Y}_{t-1})$, and $P_{t|t} = \mathrm{Var}(\boldsymbol{\alpha}_t | \boldsymbol{Y}_t)$. Since the initial state and the disturbances are all normally distributed, we have $\boldsymbol{\alpha}_t | \boldsymbol{Y}_{t-1} \sim N(\boldsymbol{a}_t, P_t)$ and $\boldsymbol{\alpha}_t | \boldsymbol{Y}_t \sim N(\boldsymbol{a}_{t|t}, P_{t|t})$. The vectors $\boldsymbol{a}_t$ and $\boldsymbol{a}_{t|t}$ are referred to as the one-step-ahead prediction and the filtering estimate of $\boldsymbol{\alpha}_t$, respectively. At each time t, $t = 1, \ldots, n$, given $\boldsymbol{a}_t$ and P_t, the Kalman filter calculates $\boldsymbol{a}_{t|t}, P_{t|t}, \boldsymbol{a}_{t+1}$, and P_{t+1} via the following filtering equations:

$$\begin{array}{rclrcl}
\boldsymbol{w}_t &=& \boldsymbol{y}_t - Z_t \boldsymbol{a}_t, & F_t &=& Z_t P_t Z_t' + H_t, \\
\boldsymbol{a}_{t|t} &=& \boldsymbol{a}_t + P_t Z_t' F_t^{-1} \boldsymbol{w}_t, & P_{t|t} &=& P_t - P_t Z_t' F_t^{-1} Z_t P_t, \\
\boldsymbol{a}_{t+1} &=& T_t \boldsymbol{a}_t + K_t \boldsymbol{w}_t, & P_{t+1} &=& T_t P_t (T_t - K_t Z_t)' + R_t Q_t R_t',
\end{array} \tag{1.29}$$

for $t = 1, \ldots, n$, where $K_t = T_t P_t Z_t' F_t^{-1}$. In (1.29), $\boldsymbol{w}_t$ is the one-step-ahead prediction error given $\boldsymbol{Y}_{t-1}$, that is, $\boldsymbol{w}_t = \boldsymbol{y}_t - \mathrm{E}(\boldsymbol{y}_t | \boldsymbol{Y}_{t-1})$; F_t is the variance matrix of the one-step-ahead prediction, $F_t = \mathrm{Var}(\boldsymbol{w}_t | \boldsymbol{Y}_{t-1})$; and K_t is called the Kalman gain. Alternatively,

$\boldsymbol{a}_{t+1}$ and P_{t+1} can be directly computed from $\boldsymbol{a}_{t|t}$ and $P_{t|t}$,

$$\boldsymbol{a}_{t+1} = T_t \boldsymbol{a}_{t|t}, \quad P_{t+1} = T_t P_{t|t} T_t^T + R_t Q_t R_t^T. \tag{1.30}$$

For the model with mean adjustments given in (1.26), the Kalman filter recursion is given by

$$\begin{aligned} \boldsymbol{w}_t &= \boldsymbol{y}_t - Z_t \boldsymbol{a}_t - \boldsymbol{d}_t, & F_t &= Z_t P_t Z_t' + H_t, \\ \boldsymbol{a}_{t|t} &= \boldsymbol{a}_t + P_t Z_t' F_t^{-1} \boldsymbol{w}_t, & P_{t|t} &= P_t - P_t Z_t' F_t^{-1} Z_t P_t, \\ \boldsymbol{a}_{t+1} &= T_t \boldsymbol{a}_{t|t} + \boldsymbol{c}_t, & P_{t+1} &= T_t P_{t|t} T_t^T + R_t Q_t R_t'. \end{aligned} \tag{1.31}$$

1.3.3 Likelihood

Let $\boldsymbol{\psi}$ be the vector of all parameters in the state space model. When the initial state distribution $N(\boldsymbol{a}_1, P_1)$ is known, the likelihood is

$$L(\boldsymbol{\psi}) = p(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n) = \prod_{t=1}^{n} p(\boldsymbol{y}_t | \boldsymbol{Y}_{t-1}), \tag{1.32}$$

where $p(\boldsymbol{y}_1|\boldsymbol{Y}_0) = p(\boldsymbol{y}_1)$ and $\boldsymbol{Y}_0$ is an empty set. Taking logarithm, the log-likelihood is

$$l(\boldsymbol{\psi}) = \sum_{t=1}^{n} \log\, p(\boldsymbol{y}_t | \boldsymbol{Y}_{t-1}). \tag{1.33}$$

Since $\mathrm{E}(\boldsymbol{y}_t|\boldsymbol{Y}_{t-1}) = Z_t \boldsymbol{a}_t$, $F_t = \mathrm{Var}(\boldsymbol{y}_t|\boldsymbol{Y}_{t-1})$, $p(\boldsymbol{y}_t|\boldsymbol{Y}_{t-1})$ is Gaussian, and the prediction error $\boldsymbol{w}_t = \boldsymbol{y_t} - Z_t \boldsymbol{a}_t$, we have

$$l(\boldsymbol{\psi}) = \sum_{t=1}^{n} \log\, p(\boldsymbol{w}_t) = -\frac{np}{2} \log\, 2\pi - \frac{1}{2} \sum_{t=1}^{n} (\log |F_t| + \boldsymbol{w}_t^T F_t^{-1} \boldsymbol{w}_t), \tag{1.34}$$

where $\boldsymbol{w}_t$ and F_t are recursively calculated by the Kalman filter. Since the likelihood is equal to the joint density of $\boldsymbol{w}_t$, $t = 1, \ldots, n$, (1.34) is referred to as the prediction error decomposition ([26], [28]).

When the distribution of the initial state $\boldsymbol{\alpha}_1$ is partially unknown and $\boldsymbol{\delta}$ has a diffuse prior as described in Section 1.3.1, the likelihood (1.34) will contain a term $-\frac{1}{2}q\log 2\pi\kappa$, which goes to $-\infty$ as $\kappa \to \infty$. In this situation, the diffuse likelihood is used,

$$L_d(\boldsymbol{\psi}) = \lim_{\kappa\to\infty} \kappa^{\frac{q}{2}} L(\boldsymbol{\psi}). \tag{1.35}$$

1.3.4 Connection to the linear mixed effects model

A linear Gaussian state space model with a zero mean prior for the initial state can be written in the form of a general linear mixed effects model ([29], [28]). Consider the linear Gaussian state space model

$$\begin{aligned} \boldsymbol{y}_t &= Z_t\boldsymbol{\alpha}_t + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon_t} \sim N(\mathbf{0}, H_t), \\ \boldsymbol{\alpha}_{t+1} &= T_t\boldsymbol{\alpha}_t + R_t\boldsymbol{\eta}_t, \quad \boldsymbol{\eta}_t \sim N(\mathbf{0}, Q_t), \end{aligned} \tag{1.36}$$

for $t = 1, \ldots, n$. Partition the $q \times 1$ initial state $\boldsymbol{\alpha}_1$ into two parts, $\boldsymbol{\alpha}_1 = (\boldsymbol{\alpha}_{11}^T, \boldsymbol{\alpha}_{12}^T)^T$, where the $q_1 \times 1$ vector $\boldsymbol{\alpha}_{11}$ has a diffuse distribution and the $q_2 \times 1$ vector $\boldsymbol{\alpha}_{12}$ has a proper zero mean normal distribution.

Let $\boldsymbol{\alpha}_t^* = \boldsymbol{\alpha}_t - \prod_{j=1}^{t-1} T_j\boldsymbol{\alpha}_1$ for $t = 2, \ldots, n$, and $\boldsymbol{\alpha}_1^* = \mathbf{0}$. Then $\boldsymbol{\alpha}_t^*$ satisfies the recursion equation

$$\boldsymbol{\alpha}_{t+1}^* = T_t\boldsymbol{\alpha}_t^* + \boldsymbol{\eta}_t. \tag{1.37}$$

Rewrite the observation equation in (1.36) as

$$\boldsymbol{y}_t = (Z_t \prod_{j=1}^{t-1} T_j)\boldsymbol{\alpha}_1 + Z_t\boldsymbol{\alpha}_t^* + \boldsymbol{\epsilon}_t. \tag{1.38}$$

Let $X_t = (Z_t \prod_{j=1}^{t-1} T_j)^T$ and partition it into $X_t = (X_{1t}, X_{2t})$ where X_{1t} has q_1 columns and X_{2t} has q_2 columns.

Then (1.38) becomes

$$\boldsymbol{y}_t = X_{1t}\boldsymbol{\alpha}_{11} + X_{2t}\boldsymbol{\alpha}_{12} + Z_t\boldsymbol{\alpha}_t^* + \boldsymbol{\epsilon}_t, \quad t = 1, \ldots, n. \tag{1.39}$$

Writing the state space model in vector form, we have

$$\boldsymbol{y} = X_1\boldsymbol{\alpha}_{11} + X_2\boldsymbol{\alpha}_{12} + Z\boldsymbol{\alpha}^* + \boldsymbol{\epsilon} \tag{1.40}$$

where

$$\boldsymbol{y} = \begin{pmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_n \end{pmatrix}, \quad X_1 = \begin{pmatrix} X_{11} \\ \vdots \\ X_{1n} \end{pmatrix}, \quad X_2 = \begin{pmatrix} X_{21} \\ \vdots \\ X_{2n} \end{pmatrix}, \tag{1.41}$$

$$Z = \begin{pmatrix} Z_1 & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & Z_n \end{pmatrix}, \quad \boldsymbol{\alpha}^* = \begin{pmatrix} \boldsymbol{\alpha}_1^* \\ \vdots \\ \boldsymbol{\alpha}_n^* \end{pmatrix}, \quad \text{and } \boldsymbol{\epsilon} = \begin{pmatrix} \boldsymbol{\epsilon}_1 \\ \vdots \\ \boldsymbol{\epsilon}_n \end{pmatrix}. \tag{1.42}$$

Model (1.39) is a linear mixed effects model of the form (1.4), where X_1 and X_2 are design matrices, $\boldsymbol{\alpha}_{11}$ is the vector of fixed effects, $\boldsymbol{\alpha}_{12}$ is the vector of random effects with a proper zero mean normal distribution, and $Z\boldsymbol{\alpha}^* + \boldsymbol{\epsilon}$ is the error vector following a zero mean normal distribution.

Suppose that the variance components of random effects and random errors in model (1.40) depend on a parameter vector $\boldsymbol{\theta}$. From a Bayesian point of view, in a general linear mixed effects model of the form (1.4), if the data is sufficiently informative that the prior of $(\boldsymbol{\beta}, \boldsymbol{\theta})$ is flat relative to the likelihood, the REML estimate of $\boldsymbol{\theta}$ is identical to the Bayesian estimate of $\boldsymbol{\theta}$ using the full data (Harville [30]).

The full likelihood, (1.34), of the state space model calculated by Kalman filter, is the posterior joint density of fixed effects $\boldsymbol{\alpha}_{11}$ and variance component $\boldsymbol{\theta}$. Since $\boldsymbol{\alpha}_{11}$

has a diffuse prior, the MLEs of parameters in the state space model that are variance components of (1.40) are identical to those obtained using REML.

1.3.5 State smoothing

While the Kalman filter calculates filtering estimates and one-step-ahead predictions for the latent state, state smoothing calculates the posterior estimates of the latent state and the covariance matrices, given all observations, that is, $\mathrm{E}(\boldsymbol{\alpha}_t|\boldsymbol{Y}_n)$ and $\mathrm{Var}(\boldsymbol{\alpha}_t|\boldsymbol{Y}_n)$, for $t = 1, \dots, n$.

Let $\hat{\boldsymbol{\alpha}}_t = \mathrm{E}(\boldsymbol{\alpha}_t|\boldsymbol{Y}_n)$ and $V_t = \mathrm{Var}(\boldsymbol{\alpha}_t|\boldsymbol{Y}_n)$, the state smoothing recursion is given by

$$\begin{array}{rclrcl} \boldsymbol{r}_{t-1} & = & Z_t' F_t^{-1} \boldsymbol{w}_t + L_t' \boldsymbol{r}_t, & N_{t-1} & = & Z_t' F_t^{-1} Z_t + L_t' N_t L_t, \\ \hat{\boldsymbol{\alpha}}_t & = & \boldsymbol{a}_t + P_t \boldsymbol{r}_{t-1}, & V_t & = & P_t - P_t N_{t-1} P_t, \end{array} \tag{1.43}$$

for $t = n, \dots, 1$ with $\boldsymbol{r}_n = \boldsymbol{0}$ and $N_n = 0$.

1.3.6 Disturbance smoothing

The smoothing estimates of the disturbances $\boldsymbol{\epsilon}_t$ and $\boldsymbol{\eta}_t$ are useful for diagnostic checking. The computation of $\hat{\boldsymbol{\epsilon}}_t = \mathrm{E}(\boldsymbol{\epsilon}_t|\boldsymbol{Y}_n)$ and $\hat{\boldsymbol{\eta}}_t = \mathrm{E}(\boldsymbol{\eta}_t|\boldsymbol{Y}_n)$ is given by the recursion

$$\begin{array}{rcl} \hat{\boldsymbol{\epsilon}}_t & = & H_t \boldsymbol{u}_t, \\ \hat{\boldsymbol{\eta}}_t & = & Q_t R_t^T \boldsymbol{r}_t, \\ \boldsymbol{u}_t & = & F_t^{-1} \boldsymbol{w}_t - K_t^T \boldsymbol{r}_t, \\ \boldsymbol{r}_{t-1} & = & Z_t^T \boldsymbol{u}_t + T_t^T \boldsymbol{r}_t, \end{array} \tag{1.44}$$

for $t = n, \dots, 1$, with $\boldsymbol{r}_n = \mathbf{0}$. The computation of corresponding variance matrices $\mathrm{Var}(\boldsymbol{\epsilon}_t|\boldsymbol{Y}_n)$ and $\mathrm{Var}(\boldsymbol{\eta}_t|\boldsymbol{Y}_n)$ is given by the recursion

$$\begin{aligned} D_t &= F_t^{-1} + K_t^T N_t K_t, \\ \mathrm{Var}(\boldsymbol{\epsilon}_t|\boldsymbol{Y}_n) &= H_t - H_t D_t H_t, \\ \mathrm{Var}(\boldsymbol{\eta}_t|\boldsymbol{Y}_n) &= Q_t - Q_t R_t^T N_t R_t Q_t, \\ N_{t-1} &= Z_t^T D_t Z_t + T_t^T N_t T_t - Z_t^T K_t^T N_t T_t - T_t^T N_t K_t Z_t, \end{aligned} \tag{1.45}$$

for $t = n, \dots, 1$, with $N_n = 0$.

1.3.7 Fast state smoothing

In state smoothing, if one is only interested in $\hat{\boldsymbol{\alpha}}_t$, but not the covariance matrix V_t, a computationally more efficient recursion can be used:

$$\hat{\boldsymbol{\alpha}}_{t+1} = T_t \hat{\boldsymbol{\alpha}}_t + R_t Q_t R_t^T \boldsymbol{r}_t, \tag{1.46}$$

for $t = 1, \dots, n$, with $\hat{\boldsymbol{\alpha}}_1 = \boldsymbol{a}_1 + P_1 \boldsymbol{r}_0$. That is, for $t = 1, \dots, n$, do filtering recursion (1.29), then for $t = n, \dots, 1$, do disturbance smoothing recursion (1.44), after that, for $t = 1, \dots, n$, do fast state smoothing recursion (1.46) to obtain $\hat{\boldsymbol{\alpha}}_t$.

1.3.8 Univariate treatment of multivariate time series

The Kalman filter and smoother takes in the whole observation vector $\boldsymbol{y}_t$ at each time point. A computationally more efficient algorithm was developed by Durbin and Koopman [26], which brings in one observation at a time, converting the multivariate series into a univariate series.

The univariate treatment significantly reduces the computational cost of filtering and smoothing by avoiding the inversion of the usually high dimensional matrix F_t. This approach also allows the dimension of the observation vector $\boldsymbol{y}_t$ to vary over time. Assuming that the dimension of $\boldsymbol{y}_t$ at time t is $p_t \times 1$, write

$$\boldsymbol{y}_t = \begin{pmatrix} y_{t,1} \\ \vdots \\ y_{t,p_t} \end{pmatrix}, \quad \boldsymbol{\epsilon}_t = \begin{pmatrix} \epsilon_{t,1} \\ \vdots \\ \epsilon_{t,p_t} \end{pmatrix}, \quad Z_t = \begin{pmatrix} \boldsymbol{Z}_{t,1} \\ \vdots \\ \boldsymbol{Z}_{t,p_t} \end{pmatrix},$$

where $\boldsymbol{Z}_{t,i}$ is the ith row of Z_t, $i = 1, \ldots, p_t$. Assume that the error covariance matrix H_t is diagonal, that is, $H_t = \text{diag}\{\sigma^2_{t,1}, \ldots, \sigma^2_{t,p_t}\}$. If H_t is not diagonal, one can diagonalize it by Cholesky decomposition

$$H_t = C_t H_t^* C_t^T,$$

where H_t^* is a diagonal matrix and C_t is a lower triangular matrix with diagonal elements equal to one; then transform the observation equation into

$$\boldsymbol{y}_t^* = Z_t^* \boldsymbol{\alpha}_t + \boldsymbol{\epsilon}_t^*, \quad \boldsymbol{\epsilon}_t^* \sim N(\boldsymbol{0}, H_t^*),$$

where $\boldsymbol{y}_t^* = C_t^{-1}\boldsymbol{y}_t$, $Z_t^* = C_t^{-1} Z_t$, and $\boldsymbol{\epsilon}_t^* = C_t^{-1}\boldsymbol{\epsilon}_t$.

Let each element $y_{t,i}$ of $\boldsymbol{y}_t$ come into the system one at a time, $i = 1, \ldots, p_t$. The observation series now becomes

$$y_{1,1}, \ldots, y_{1,p_1}, \ldots, y_{n,1}, \ldots, y_{n,p_n},$$

which is univariate over $\sum_{t=1}^n p_t$ time points.

The observation equation is given by

$$y_{t,i} = \boldsymbol{Z}_{t,i}\boldsymbol{\alpha}_{t,i} + \epsilon_{t,i}, \quad i = 1, \ldots, p_t, \quad t = 1, \ldots, n,$$

where $\boldsymbol{Z}_{t,i}$ is the ith row of Z_t, $\epsilon_{t,i}$ is the ith element of $\boldsymbol{\epsilon}_t$, and $\boldsymbol{\alpha}_{t,i}$ equals $\boldsymbol{\alpha}_t$ in the original model (1.25). The state equation is given by

$$\begin{aligned} \boldsymbol{\alpha}_{t,i+1} &= \boldsymbol{\alpha}_{t,i}, && i = 1, \ldots, p_t - 1, \\ \boldsymbol{\alpha}_{t+1,1} &= T_t \boldsymbol{\alpha}_{t,p_t} + R_t \boldsymbol{\eta}_t, && t = 1, \ldots, n, \end{aligned}$$

with $\boldsymbol{\alpha}_{1,1} = \boldsymbol{\alpha}_1 \sim N(\boldsymbol{a}_1, P_1)$.

Let

$$\begin{aligned} \boldsymbol{a}_{t,i} &= E(\boldsymbol{\alpha}_{t,i} | \boldsymbol{Y}_{t-1}, y_{t,1}, \ldots, y_{t,i-1}), \\ P_{t,i} &= \mathrm{Var}(\boldsymbol{\alpha}_{t,i} | \boldsymbol{Y}_{t-1}, y_{t,1}, \ldots, y_{t,i-1}), \quad i = 2, \ldots, p_t, \end{aligned}$$

be the one-step-ahead prediction mean and variance of the state vector, and

$$\begin{aligned} \boldsymbol{a}_{t,1} &= E(\boldsymbol{\alpha}_{t,1} | \boldsymbol{Y}_{t-1}), \\ P_{t,1} &= \mathrm{Var}(\boldsymbol{\alpha}_{t,1} | \boldsymbol{Y}_{t-1}). \end{aligned}$$

The filtering recursion for the univariate model is given by

$$\begin{aligned} v_{t,i} &= y_{t,i} - \boldsymbol{Z}_{t,i} \boldsymbol{a}_{t,i}, \\ F_{t,i} &= \boldsymbol{Z}_{t,i} P_{t,i} \boldsymbol{Z}_{t,i}^T + \sigma_{t,i}^2, \\ \boldsymbol{K}_{t,i} &= P_{t,i} \boldsymbol{Z}_{t,i}^T F_{t,i}^{-1}, \\ \boldsymbol{a}_{t,i+1} &= \boldsymbol{a}_{t,i} + \boldsymbol{K}_{t,i} v_{t,i}, \\ P_{t,i+1} &= P_{t,i} - \boldsymbol{K}_{t,i} F_{t,i} \boldsymbol{K}_{t,i}^T, \end{aligned} \tag{1.47}$$

for $i = 1, \ldots, p_t$ and $t = 1, \ldots, n$, where $v_{t,i}$ and $F_{t,i}$ are scalars, $\boldsymbol{K}_{t,i}$ is a column vector, $\boldsymbol{Z}_{t,i}$ is a row vector, and $\boldsymbol{a}_{t,i}$ and $P_{t,i+1}$ have the same dimensions as the original $\boldsymbol{a}_t$ and P_t. In (1.47), at time t, one obtains $\boldsymbol{a}_{t,p_t+1}$ and P_{t,p_t+1}; their transition to time $t+1$ is given by

$$\begin{aligned} \boldsymbol{a}_{t+1,1} &= T_t \boldsymbol{a}_{t,p_t+1}, \\ P_{t+1,1} &= T_t P_{t,p_t+1} T_t^T + R_t Q_t R_t^T. \end{aligned}$$

The $\boldsymbol{a}_{t+1,1}$ and $P_{t+1,1}$ calculated from this recursion are the same as $\boldsymbol{a}_{t+1}$ and P_{t+1} in the original Kalman filter.

The smoothing algorithm is given by

$$\begin{aligned} L_{t,i} &= I - \boldsymbol{K}_{t,i} \boldsymbol{Z}_{t,i}, \\ \boldsymbol{r}_{t,i-1} &= \boldsymbol{Z}_{t,i}^T F_{t,i}^{-1} v_{t,i} + L_{t,i}^T \boldsymbol{r}_{t,i}, \\ N_{t,i-1} &= \boldsymbol{Z}_{t,i}^T F_{t,i}^{-1} \boldsymbol{Z}_{t,i} + L_{t,i}^T N_{t,i} L_{t,i}, \end{aligned} \tag{1.48}$$

for $i = p_t, \ldots, 1$ and $t = n, \ldots, 1$, with initial values $\boldsymbol{r}_{n,p_n} = \boldsymbol{0}$ and $N_{n,p_n} = 0$, $\boldsymbol{r}_{t,i}$ a column vector, and $L_{t,i}$ a square matrix of the same dimension as $N_{t,i}$. In (1.48), at each time point t, one will obtain $\boldsymbol{r}_{t,0}$ and $N_{t,0}$; the transition to time $t-1$ is given by

$$\begin{aligned} \boldsymbol{r}_{t-1,p_{t-1}} &= T_{t-1}^T \boldsymbol{r}_{t,0}, \\ N_{t-1,p_{t-1}} &= T_{t-1}^T N_{t,0} T_{t-1}. \end{aligned} \tag{1.49}$$

The $\boldsymbol{r}_{t,0}$ and $N_{t,0}$ are equal to $\boldsymbol{r}_{t-1}$ and N_{t-1} in the original Kalman filter, $t = n, \ldots, 1$.

After obtaining $\boldsymbol{a}_t, P_t, \boldsymbol{r}_{t-1}, N_{t-1}$ from the univariate treatment filtering and smoothing recursions, one can use the last two equations in (1.43) to calculate the smoothed mean and covariance estimate of state vectors.

1.3.9 Ensemble Kalman filter

In a linear Gaussian state space model, when the dimension of the state vector is extremely high, say, millions, the Kalman filter ([31]) and particle filters for low-dimensional state space models are infeasible due to high computational cost and degeneracy. In this setting, there is an approximate method called the ensemble Kalman filter ([32], [33], [34], [35], [36], [37], [38], [39]), which keeps track of the current state distribution by a collection (ensemble) of sample state vectors and update them with a linear shift. Ensemble Kalman filter has been successfully applied to geophysical data assimilation, where the dimensions are usually high. However, it is an approximation method that is suboptimal compared to Kalman filter. In addition, while the time complexity is linear in the dimension of states, it is quadratic in the dimension of the observation vector ([40], [41]). We note that the time complexity is $O(p^2n)$, where p is the dimension of the observation vector and n is the dimension of the state vector, rather than $O(pn)$ (stated in [41]). Therefore, the computational cost of the ensemble Kalman filter is high when p is large and it cannot handle extremely high dimensional observation vectors.

1.3.10 Regression estimation

The observation equation in the state space model (1.25) can be extended to include covariates,

$$\boldsymbol{y}_t = Z_t\boldsymbol{\alpha}_t + X_t\boldsymbol{\beta} + \boldsymbol{\epsilon}_t, \tag{1.50}$$

where X_t is a $p \times k$ design matrix of covariates and $\boldsymbol{\beta}$ is a time-independent vector of coefficients of dimension $k \times 1$. There are two ways to handle (1.50). The first approach is to include $\boldsymbol{\beta}$ in the state vector, so the state space model is given by

$$\begin{aligned} \boldsymbol{y}_t &= \begin{pmatrix} Z_t & X_t \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha}_t \\ \boldsymbol{\beta}_t \end{pmatrix} + \boldsymbol{\epsilon}_t \\ \begin{pmatrix} \boldsymbol{\alpha}_{t+1} \\ \boldsymbol{\beta}_{t+1} \end{pmatrix} &= \begin{pmatrix} T_t & 0 \\ 0 & I_k \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha}_t \\ \boldsymbol{\beta}_t \end{pmatrix} + \begin{pmatrix} R_t \\ 0 \end{pmatrix} \boldsymbol{\eta}_t \end{aligned} \tag{1.51}$$

for $t = 1, \ldots, n$.

The second approach is via the augmentation of the Kalman filter. The log-likelihood of $\boldsymbol{\beta}$ is given by $-\sum_{t=1}^{n} \boldsymbol{w}_t^T F_t^{-1} \boldsymbol{w}_t$+constant, where constant means that it is independent of $\boldsymbol{\beta}$. One estimates $\boldsymbol{\beta}$ by minimizing

$$\sum_{t=1}^{n} \boldsymbol{w}_t^T F_t^{-1} \boldsymbol{w}_t. \tag{1.52}$$

For a given $\boldsymbol{\beta}$, (1.50) can be rewritten as

$$\boldsymbol{y}_t - X_t \boldsymbol{\beta} = Z_t \boldsymbol{\alpha}_t + \boldsymbol{\epsilon}_t. \tag{1.53}$$

Let $\boldsymbol{w}_t$ be the one-step-ahead forecast error for model (1.53) in which $\boldsymbol{y}_t - X_t\boldsymbol{\beta}$ is treated as the observation vector; $\boldsymbol{w}_t$ can be obtained in the following way. Write $X_t = (\boldsymbol{x}_{1,t}, \ldots, \boldsymbol{x}_{k,t})$, where $\boldsymbol{x}_{i,t}$ is the ith column of X_t, $i = 1, \ldots, k$. Recall that in model (1.25), we apply Kalman filter with $\boldsymbol{y}_t$ as the observation vector. Here, analogously, we apply Kalman filter with each of $\boldsymbol{y}_t, \boldsymbol{x}_{1,t}, \ldots, \boldsymbol{x}_{k,t}$ as the observation vector, but using the same updating equations for F_t, $P_{t|t}$, and P_t as in model (1.25). Denote the corresponding one-step-ahead prediction errors as $\boldsymbol{w}_t^*, \boldsymbol{W}_{1,t}^*, \ldots, \boldsymbol{W}_{k,t}^*$. The one-step-ahead forecast errors for the observation vector $\boldsymbol{y}_t - X_t\boldsymbol{\beta}$ is given by $\boldsymbol{w}_t = \boldsymbol{w}_t^* - W_t^*\boldsymbol{\beta}$, where $W_t^* = (\boldsymbol{W}_{1,t}^*, \ldots, \boldsymbol{W}_{k,t}^*)$. Hence we have

$$\sum_{t=1}^{n} \boldsymbol{w}_t^T F_t^{-1} \boldsymbol{w}_t = \sum_{t=1}^{n} (\boldsymbol{w}_t^* - W_t^*\boldsymbol{\beta})^T F_t^{-1} (\boldsymbol{w}_t^* - W_t^*\boldsymbol{\beta}), \tag{1.54}$$

which has an analytical solution for its minimizer $\boldsymbol{\beta}$:

$$\hat{\boldsymbol{\beta}} = (\sum_{t=1}^{n} X_t^{*T} F_t^{-1} X_t^*)^{-1} \sum_{t=1}^{n} X_t^{*T} F_t^{-1} \boldsymbol{w}_t^*. \tag{1.55}$$

When diffuse initialization is used, the exact initial Kalman filter can be applied, see

Durbin and Koopman [26].

1.4 Mixed Effects State Space Models

Linear mixed effects models have been widely used in the modeling of longitudinal data. Some authors propose to write certain special cases of linear mixed effects models in a state space form and use the Kalman filter to compute the likelihood, see, for example, [42].

Recent literature realizes that the power of the state space model is far more than just a computational tool. A state space model has an explicit expression for the underly evolving process, which in turn can model a wide variety of parametric or non-parametric processes. It is this property of both interpretability and flexibility that the state space model has attracted much attention, and numerous studies have used it to directly model longitudinal data ([43], [44], [45], [46], [47], [48], [49], [50], [51], [52]).

Liu et al. (2011) [48] consider the following mixed effects state space model:

$$\begin{aligned} \boldsymbol{y}_i(t_{ij}) &= Z(\boldsymbol{\theta}_i)\boldsymbol{\alpha}_i(t_{ij}) + \boldsymbol{\epsilon}_i(t_{ij}), \quad \boldsymbol{\epsilon}_i(t_{ij}) \sim N(\mathbf{0}, H), \\ \boldsymbol{\alpha}_i(t_{i,j+1}) &= T(\boldsymbol{\theta}_i)\boldsymbol{\alpha}_i(t_{ij}) + \boldsymbol{\eta}_i(t_{ij}), \quad \boldsymbol{\eta}_i(t_{ij}) \sim N(\mathbf{0}, Q), \end{aligned} \tag{1.56}$$

where $\boldsymbol{y}_i(t_{ij})$ is a $q \times 1$ observation vector for subject i at time t_{ij}, $i = 1, \ldots, m$, $j = 1, \ldots, n_i$, $\boldsymbol{\alpha}_i(t_{ij})$ is a $q \times 1$ state vector for subject i, and $\boldsymbol{\epsilon}_i(t_{ij})$ and $\boldsymbol{\eta}_i(t_{ij})$ are error and disturbance vectors. The system matrices $Z(\boldsymbol{\theta}_i)$ and $T(\boldsymbol{\theta}_i)$ are parameterized by a random vector $\boldsymbol{\theta}_i$, modeled by

$$\boldsymbol{\theta}_i = \boldsymbol{\theta} + \boldsymbol{b}_i, \quad \boldsymbol{b}_i \sim N(\mathbf{0}, D), \tag{1.57}$$

where $\boldsymbol{\theta}$ is the population fixed effect parameter vector, and $\boldsymbol{b}_i$ is the random effect vector. Model (1.56) assumes that all subjects share common time-invariant disturbance covariance matrices H and Q, which characterize within-subject variation. System matrices $Z(\boldsymbol{\theta}_i)$ and $T(\boldsymbol{\theta}_i)$ depend on a subject-specific vector $\boldsymbol{\theta}_i$, which is modeled by a simple linear mixed-effects model. Therefore, $Z(\boldsymbol{\theta}_i)$ and $T(\boldsymbol{\theta}_i)$ characterize between-subject variation. Model (1.56) integrates both the mixed effects idea of longitudinal models and the dynamic modeling power of state space models. One limitation is that one must know how the system matrices $Z(\boldsymbol{\theta}_i)$ and $T(\boldsymbol{\theta}_i)$ are parameterized in terms of $\boldsymbol{\theta}_i$, which is often not the case in practice.

Liu (2010) [47] proposed another form of state space models for longitudinal data,

$$\begin{aligned}
y_i(t_{ij}) &= X_i(t_{ij})\boldsymbol{\beta}(t_{ij}) + Z_i(t_{ij})\boldsymbol{b}_i(t_{ij}) + \epsilon_i(t_{ij}) \\
\begin{pmatrix} \boldsymbol{\beta}(t_{ij}) \\ \boldsymbol{b}_i(t_{ij}) \end{pmatrix} &= \begin{pmatrix} F_\beta(t_{ij}) & 0 \\ 0 & F_b(t_{ij}) \end{pmatrix} \begin{pmatrix} \boldsymbol{u}(t_{ij}) \\ \boldsymbol{v}_i(t_{ij}) \end{pmatrix}, \\
\begin{pmatrix} \boldsymbol{u}(t_{ij}) \\ \boldsymbol{v}_i(t_{ij}) \end{pmatrix} &= \begin{pmatrix} T_u(t_{ij}) & 0 \\ T_{uv}(t_{ij}) & T_v(t_{ij}) \end{pmatrix} \begin{pmatrix} \boldsymbol{u}(t_{i,j-1}) \\ \boldsymbol{v}_i(t_{i,j-1}) \end{pmatrix} + \\
& \quad \begin{pmatrix} R_u(t_{ij}) & 0 \\ R_{uv}(t_{ij}) & R_v(t_{ij}) \end{pmatrix} \begin{pmatrix} \boldsymbol{\eta}_u(t_{ij}) \\ \boldsymbol{\eta}_{vi}(t_{ij}) \end{pmatrix},
\end{aligned} \tag{1.58}$$

with observation equation random error $\epsilon_i(t_{ij}) \overset{iid}{\sim} N(0, \sigma_\epsilon^2)$, state equation disturbance

$$\begin{pmatrix} \boldsymbol{\eta}_u(t_{ij}) \\ \boldsymbol{\eta}_{vi}(t_{ij}) \end{pmatrix} \sim N(\begin{pmatrix} \boldsymbol{\mu}_u(t_{ij}) \\ \boldsymbol{\mu}_v(t_{ij}) \end{pmatrix}, \begin{pmatrix} Q_u(t_{ij}) & 0 \\ 0 & Q_v(t_{ij}) \end{pmatrix}), \tag{1.59}$$

and initial state distribution

$$\begin{pmatrix} \boldsymbol{u}(t_{i0}) \\ \boldsymbol{v}_i(t_{i0}) \end{pmatrix} \sim N(\mathbf{0}, \begin{pmatrix} P_{u0} & 0 \\ 0 & P_{v0} \end{pmatrix}). \tag{1.60}$$

In model (1.58), $\boldsymbol{\beta}(t_{ij})$ and $\boldsymbol{b}_i(t_{ij})$ are the population and individual effects, and $\boldsymbol{u}(t_{ij})$ and $\boldsymbol{v}_i(t_{ij})$ are their corresponding latent state vectors, transformed through $F_\beta(t_{ij})$ and $F_b(t_{ij})$. The initial state $\boldsymbol{u}(t_{i0})$ has a diffuse prior and $\boldsymbol{v}_i(t_{i0})$ has a zero mean proper prior. The third equation in (1.58) defines the latent process for $\boldsymbol{u}(t_{ij})$ and $\boldsymbol{v}_i(t_{ij})$, in which the system matrices can have non-zero off-diagonal blocks $T_{uv}(t_{ij})$ and $R_{uv}(t_{ij})$, allowing the population fixed effects to have an influence on subject random effects. Model (1.58) is very flexible in that population fixed effects $\boldsymbol{\beta}(t)$ and subject random effects $\boldsymbol{b}_i(t)$ can be time-dependent and can be any combination of parametric or non-parametric curves. Note that the random effects stochastic process $\boldsymbol{b}_i(t)$ must have mean zero to ensure identifiability.

Compared to (1.56) which includes time-invariant random effects in the system matrices and requires a known parameterization form of the transition matrices, (1.58) models mixed effects in a more direct way by specifying the observation as the sum of fixed and random effects.

1.5 Outline of the Thesis

To model the longitudinal trajectory, linear mixed effects models, non-parametric spline based methods, mixed effects functional models, and state space models have been used in the literature. Among these four approaches, only the state space model is capable of modeling the dynamic interactions between the longitudinal variables from one time point to the next.

We extend the mixed effects state space model (1.58) of Liu et al. [47] to (a) include a regression term of covariates in the observation equation, (b) allow the random errors from different variables to have an unstructured covariance matrix, and (c) joint model longitudinal data with survival data. The computation of the proposed models remains a severe limitation when the number of subjects is large. The mixed effects state space models have high dimensional observation vectors and state vectors. Current computational methods such as the univariate treatment of Kalman filter and ensemble Kalman filter are infeasible for such high dimensional data. Denote by m the number of subjects, n the number of time points, and q the number of variables. The time complexity is $O(m^3q^3n)$ for both the original Kalman filter and the univariate treatment, despite the fact that the univariate treatment is more efficient by avoiding the inversion of $mq \times mq$ matrices. The time complexity of the ensemble Kalman filter is $O(m^2q^2N^2n)$, where N is the ensemble size that is usually at least 40. In addition, while the ensemble approach is applicable when the number of subjects is not too large, it is always more desirable to use an exact method whenever possible. We propose in this thesis a new algorithm of time complexity $O(mq^3n)$ that gives exactly the same numerical results as the original Kalman filter.

The remaining part of the thesis is organized as follows. Chapter 2 presents a mixed effects state space model and develops an algorithm efficient in both time and space for the model computation. Chapter 3 presents a non-Gaussian mixed effects state space model to jointly model multivariate longitudinal variables and survival time. Evaluation of the joint likelihood requires numerical simulations, for which we present an algorithm to simulate from high dimensional multivariate normal distributions. Chapter 4 presents simulation studies, in which we (i) compare the univariate treatment of the Kalman filter with the new algorithm in terms of computation time and numerical accuracy, and (ii) apply the new algorithm to the joint model. Chapter 5 presents real data applications, where we apply the joint model to a dialysis data set to answer some clinical questions. Chapter 6 discusses limitations and future work.

Chapter 2

Multivariate State Space Mixed Effects Models

2.1 The Model

A mixed effects state space model is used to model longitudinal data, for its flexibility and interpretability. Longitudinal data naturally evolves continuously over time, but the observations can only be made intermittently. The model is similar to (1.58) with some modifications.

Denote the observations for subject i as $\{(\boldsymbol{x}_i(t_{ij}),\ \boldsymbol{y}_i(t_{ij})),\ j = 1, \ldots, n_i\}$, where $\boldsymbol{x}_i(t_{ij})$ and $\boldsymbol{y}_i(t_{ij})$ are vectors of covariates and longitudinal variables measured at time t_{ij}, $j = 1, \ldots, n_i$. Suppose there are q longitudinal variables and p covariates. We assume the following mixed effects state space model

$$\begin{aligned}
\boldsymbol{y}_i(t_{ij}) &= Z_u(t_{ij})\boldsymbol{u}(t_{ij}) + Z_v(t_{ij})\boldsymbol{v}_i(t_{ij}) + X_i(t_{ij})\boldsymbol{\beta} + \boldsymbol{\epsilon}_i(t_{ij}), & \boldsymbol{\epsilon}_i(t_{ij}) &\sim N(\mathbf{0}, \Sigma_\epsilon(t_{ij})), \\
\boldsymbol{u}(t_{i,j+1}) &= T_u(t_{ij})\boldsymbol{u}(t_{ij}) + R_u(t_{ij})\boldsymbol{\eta}_u(t_{ij}), & \boldsymbol{\eta}_u(t_{ij}) &\sim N(\mathbf{0}, \Sigma_u(t_{ij})), \\
\boldsymbol{v}_i(t_{i,j+1}) &= T_v(t_{ij})\boldsymbol{v}_i(t_{ij}) + R_v(t_{ij})\boldsymbol{\eta}_{vi}(t_{ij}), & \boldsymbol{\eta}_{vi}(t_{ij}) &\sim N(\mathbf{0}, \Sigma_v(t_{ij})),
\end{aligned} \tag{2.1}$$

where $\boldsymbol{y}_i(t_{ij})$ is an observation vector of longitudinal variables at time t_{ij} from subject i, $Z_u(t_{ij})$ is a $q \times d_u$ design matrix, $Z_v(t_{ij})$ is a $q \times d_v$ design matrix, $X_i(t_{ij}) = \text{diag}\{\boldsymbol{x}_i^T(t_{ij}), \ldots, \boldsymbol{x}_i^T(t_{ij}\}$ is a $q \times pq$ design matrix, $\boldsymbol{u}(t_{ij})$ is a $d_u \times 1$ vector of population effects, $\boldsymbol{v}_i(t_{ij})$ is a $d_v \times 1$ vector of random effects for subject i, $\boldsymbol{\beta}$ is a $pq \times 1$ vector of parameters, $\boldsymbol{\epsilon}_i(t_{ij})$ is a vector of random errors, $T_u(t_{ij})$ and $T_v(t_{ij})$ are state transition matrices, $\boldsymbol{\eta}_u(t_{ij})$ and $\boldsymbol{\eta}_{vi}(t_{ij})$ are disturbance terms, and $R_u(t_{ij})$ and $R_v(t_{ij})$ are selection matrices.

The first equation in model (2.1) describes the observation vector as the sum of four parts: a population mean, a subject deviation, an external influence from covariates, and a random error. For simplicity, we use the same set of covariates for all longitudinal variables. Extensions to the case where covariates are different for different variables are straight forward. The error term $\boldsymbol{\epsilon}_i(t_{ij})$ is assumed to be serially independent and i.i.d. for all subjects at the same time points. $\boldsymbol{\epsilon}_i(t_{ij})$ models fluctuations around the dominant trend $Z_u(t_{ij})\boldsymbol{u}(t_{ij}) + Z_v(t_{ij})\boldsymbol{v}_i(t_{ij}) + X_i(t_{ij})\boldsymbol{\beta}$ and describes within-subject variation, which is usually due to measurement errors and/or biological variation. In addition, the correlation structure between variables can be modeled by the covariance matrix $\Sigma_\epsilon(t_{ij})$ of $\boldsymbol{\epsilon}_i(t_{ij})$. For each variable, the corresponding component of individual effects $\boldsymbol{v}_i(t_{ij})$ specifies a subject deviation curve, so $\boldsymbol{v}_i(t_{ij})$ describes between-subject variation. Meanwhile, since $\boldsymbol{v}_i(t_{ij})$ evolves continuously over time, it also accounts for correlation among repeated measurements for each variable. The population mean $\boldsymbol{u}(t_{ij})$ and subject deviation $\boldsymbol{v}_i(t_{ij})$ are modeled as latent states, defined by the second and third equation, respectively. The state transition matrices $T_u(t_{ij})$ and $T_v(t_{ij})$ can model dynamic interactions between variables at population and individual level. Therefore, three components can be used to model the interactions between variables: $\Sigma_\epsilon(t_{ij})$, $T_u(t_{ij})$, and $T_v(t_{ij})$, where $\Sigma_\epsilon(t_{ij})$ characterizes the correlation structure of the combination of population effects and subject random effects. Depending on the mechanism of the system and the purpose of the study, one can use one or a combination of these three components.

Comparing to model (1.58), model (2.1) has an additional regression term $X_i(t_{ij})\boldsymbol{\beta}$ in the observation equation. In addition, the random errors $\epsilon_i(t_{ij})$ from different variables measured at the same time are allowed to be correlated. Another major difference is that we do not consider the influence of the population effects on the individual effects, which is modeled by $T_{uv}(t_{ij})$ in the state transition matrix in (1.58). Other minor differences include setting the $F_\beta(t_{ij})$ and $F_b(t_{ij})$ in (1.58) to identity matrices, so that the population and individual effects are directly modeled by latent states.

2.1.1 Models for latent states

There is a wide selection of models for latent processes $\boldsymbol{u}(t_{ij})$ and $\boldsymbol{v}_i(t_{ij})$, specified by the state equations in (2.1). Any stochastic process that can be represented by a state space model can be used to model these latent processes. Note that $\boldsymbol{u}(t_{ij})$ and $\boldsymbol{v}_i(t_{ij})$ can be modeled as different processes; components of $\boldsymbol{u}(t_{ij})$ and $\boldsymbol{v}_i(t_{ij})$ that correspond to different variables can also be modeled as different processes. We provide details for cubic spline and Ornstein-Uhlenbeck (OU) process models below.

Cubic spline model

For simplicity, we consider the case when $q = 1$ and both the population mean and subject deviations are modeled by cubic splines. Then the corresponding mixed effects state space model is given by

$$
\begin{aligned}
y_i(t_{ij}) &= (1 \quad 0)\begin{pmatrix} f(t_{ij}) \\ f'(t_{ij}) \end{pmatrix} + (1 \quad 0)\begin{pmatrix} b_i(t_{ij}) \\ b_i'(t_{ij}) \end{pmatrix} + X_i(t_{ij})\boldsymbol{\beta} + \epsilon_i(t_{ij}), \\
\begin{pmatrix} f(t_{ij}) \\ f'(t_{ij}) \end{pmatrix} &= \begin{pmatrix} 1 & \triangle t_{ij} \\ 0 & 1 \end{pmatrix}\begin{pmatrix} f(t_{i,j-1}) \\ f'(t_{i,j-1}) \end{pmatrix} + R_u(t_{ij})\boldsymbol{\eta}_u(t_{ij}), \qquad (2.2) \\
\begin{pmatrix} b_i(t_{ij}) \\ b_i'(t_{ij}) \end{pmatrix} &= \begin{pmatrix} 1 & \triangle t_{ij} \\ 0 & 1 \end{pmatrix}\begin{pmatrix} b_i(t_{i,j-1}) \\ b_i'(t_{i,j-1}) \end{pmatrix} + R_v(t_{ij})\boldsymbol{\eta}_{vi}(t_{ij}),
\end{aligned}
$$

where $f(t_{ij})$ is the population mean, $b_i(t_{ij})$ is the deviation of subject i, $X_i(t_{ij})\boldsymbol{\beta}$ is the covariates effect, $\epsilon_i(t_{ij}) \overset{iid}{\sim} N(0, \sigma_\epsilon^2)$, $\boldsymbol{\eta}_u(t_{ij}) \overset{iid}{\sim} N(\mathbf{0}, \zeta\Lambda)$, $\boldsymbol{\eta}_{vi}(t_{ij}) \overset{iid}{\sim} N(\mathbf{0}, \lambda\Lambda)$, $R_u(t_{ij})$ and $R_v(t_{ij})$ are identity matrices, ζ and λ are population level and subject level smoothing parameters,

$$
\Lambda = \begin{pmatrix} \triangle t_{ij}^3/3 & \triangle t_{ij}^2/2 \\ \triangle t_{ij}^2/2 & \triangle t_{ij} \end{pmatrix},
$$

and $\triangle t_{ij} = t_{ij} - t_{i,j-1}$ is the time difference between two consecutive observations. The population level latent state is

$$
\boldsymbol{u}(t_{ij}) = \begin{pmatrix} f(t_{ij}) \\ f'(t_{ij}) \end{pmatrix},
$$

and the subject level latent state is

$$\boldsymbol{v}_i(t_{ij}) = \begin{pmatrix} b_i(t_{ij}) \\ b_i'(t_{ij}) \end{pmatrix}.$$

The population and subject level state equations have the same transition matrix

$$T_u(t_{ij}) = T_v(t_{ij}) = \begin{pmatrix} 1 & \triangle t_{ij} \\ 0 & 1 \end{pmatrix}.$$

Generalization to the multivariate case is trivial. Suppose there are q longitudinal variables, $\boldsymbol{y}_i(t_{ij}) = (y_{i1}(t_{ij}), \ldots, y_{iq}(t_{ij}))^T$ is the vector of all longitudinal variables of subject i. The population level state is

$$\boldsymbol{u}(t_{ij}) = \begin{pmatrix} f_1(t_{ij}) \\ f_1'(t_{ij}) \\ \vdots \\ f_q(t_{ij}) \\ f_q'(t_{ij}) \end{pmatrix},$$

and the subject level state is

$$\boldsymbol{v}_i(t_{ij}) = \begin{pmatrix} b_{i1}(t_{ij}) \\ b_{i1}'(t_{ij}) \\ \vdots \\ b_{iq}(t_{ij}) \\ b_{iq}'(t_{ij}) \end{pmatrix},$$

where $f_k(t_{ij})$ and $b_{ik}(t_{ij})$ are the population mean and subject deviation of the kth longitudinal variable, $k = 1, \ldots, q$, $Z_u(t_{ij}) = Z_v(t_{ij}) = I_q \otimes (1 \quad 0)$; $T_u(t_{ij}) = T_v(t_{ij}) = I_q \otimes T_0(t_{ij})$, with

$$T_0(t_{ij}) = \begin{pmatrix} 1 & \triangle t_{ij} \\ 0 & 1 \end{pmatrix};$$

$R_u(t_{ij}) = R_{vi}(t_{ij}) = I_{2q}$; $\boldsymbol{\eta}_u(t_{ij}) \sim N(\mathbf{0}, \text{diag}\{\zeta_1\Lambda, \ldots, \zeta_q\Lambda\})$ and $\boldsymbol{\eta}_{vi}(t_{ij}) \sim N(\mathbf{0},$ $\text{diag}\{\lambda_1\Lambda, \ldots, \lambda_q\Lambda\})$ are mutually and serially independent, where $\zeta_1, \ldots, \zeta_q$ and $\lambda_1, \ldots, \lambda_q$ are population level and subject level smoothing parameters for the q variables.

Ornstein-Uhlenbeck process

Another choice for the latent processes is the Ornstein-Uhlenbeck (OU) process, which is often used to model biological variation. A continuous time OU process $X(t)$ is given by

$$dX(t) = \xi[\mu - X(t)]dt + \nu dW(t), \tag{2.3}$$

where $\xi > 0, \mu$, and $\nu > 0$ are parameters, and $W(t)$ is a Wiener process. The analytical solution for (2.3) is given by ([53])

$$X(t) = \mu + e^{-\xi(t-s)}[X(s) - \mu] + \nu \int_s^t e^{-\xi(t-s)} dW(u). \tag{2.4}$$

To write the OU process (2.4) in a discrete time state space form, suppose the discrete time points are $t_1 < t_2 < \cdots < t_n$. In (2.4), let $t = t_{j+1}$, $s = t_j$, and $\triangle t_j = t_{j+1} - t_j$, $j = 1, \ldots, n-1$, we have

$$\begin{aligned} X(t_{j+1}) &= \mu(1 - e^{-\xi\triangle t_j}) + e^{-\xi\triangle t_j} X(t_j) + \nu \textstyle\int_{t_j}^{t_{j+1}} e^{-\xi(t_{j+1}-u)} dW(u) \\ &= \mu(1 - e^{-\xi\triangle t_j}) + e^{-\xi\triangle t} X(t_j) + \nu e^{-\xi t_{j+1}} \textstyle\int_{t_j}^{t_{j+1}} e^{\xi u} dW(u) \\ &= \mu(1 - e^{-\xi\triangle t_j}) + e^{-\xi\triangle t_j} X(t_j) + \eta(t_j), \end{aligned}$$

with $\eta(t_j) = \nu e^{-\xi t_{j+1}} \int_{t_j}^{t_{j+1}} e^{\xi u} dW(u)$. It can be shown that $\int_{t_j}^{t_{j+1}} e^{\xi u} dW(u)$ is a martingale, hence

$$E[\int_{t_j}^{t_{j+1}} e^{\xi u} dW(u)] = 0. \tag{2.5}$$

By the Itô isometry, we have

$$E[\int_{t_j}^{t_{j+1}} e^{\xi u} dW(u)]^2 = \int_{t_j}^{t_{j+1}} (e^{\xi u})^2 du = \frac{1}{2\xi}(e^{2\xi t_{j+1}} - e^{2\xi t_j}).$$

Therefore,

$$\mathrm{Var}[\eta(t_j)] = \nu^2 e^{-2\xi t_{j+1}} \frac{1}{2\xi}(e^{2\xi t_{j+1}} - e^{2\xi t_j}) = \frac{\nu^2}{2\xi}(1 - e^{-2\xi \triangle t_j}).$$

Since $W(t)$ is Guassian, $\int_{t_j}^{t_{j+1}} e^{\xi u} dW(u)$ is also Gaussian, we have

$$\eta(t_j) \sim N(0, \frac{\nu^2}{2\xi}(1 - e^{-2\xi \triangle t_j})).$$

Putting everything together, the discretized OU process $X(t)$ is given by

$$X(t_{j+1}) = \mu(1 - e^{-\xi \triangle t_j}) + e^{-\xi \triangle t_j} X(t_j) + \eta(t_j), \quad \eta(t_j) \sim N(0, \frac{\nu^2}{2\xi}(1 - e^{-2\xi \triangle t_j})). \tag{2.6}$$

If, for example, we model the individual effects $\boldsymbol{v}_i(t_{ij})$ as an OU process, in the univariate case, the state equation for $\boldsymbol{v}_i(t_{ij}) = v_i(t_{ij})$ is

$$v_i(t_{ij}) = T_v(t_{ij}) v_i(t_{i,j-1}) + c(t_{ij}) + R_v(t_{ij}) \eta_{vi}(t_{ij}),$$

where $T_v(t_{ij}) = e^{-\xi \triangle t_{ij}}$, $R_v(t_{ij}) = 1$, $\eta_{vi}(t_{ij}) \sim N(0, \frac{\nu^2}{2\xi}(1 - e^{-2\xi \triangle t_{ij}}))$, and $c(t_{ij}) = \mu(1 - e^{-\xi \triangle t_{ij}})$ is the mean adjustment. We set $\mu = 0$ so that the random effects have mean zero. Generalization to the multivariate case is similar to that of the cubic spline model.

2.1.2 Vector form of the mixed effects state space model

The equations in (2.1) specify the dynamic mixed-effects model for each subject. In this section, observations and state vectors of all subjects are stacked to provide a state space form. In the remainder of this dissertation, we assume that all subjects share the same time points, but the time points do not have to be equally spaced. That is, $t_{ij} = t_j$ and $\triangle t_j = t_{j+1} - t_j$, $j = 1, \ldots, n$. For now, assume there is no missing data. Suppose there are m subjects and n time points. Stack the observations of all subjects at time t_j together and write

$$\boldsymbol{y}(t_j) = \begin{pmatrix} \boldsymbol{y}_1(t_j) \\ \vdots \\ \boldsymbol{y}_m(t_j) \end{pmatrix},$$

where $\boldsymbol{y}_i(t_j) = (y_{i1}(t_j), \ldots, y_{iq}(t_j))^T$ is the vector of q variables observed at time t_{ij} for subject i, $i = 1, \ldots, m$. Stack the population-level state and subject deviation states together into one state vector

$$\boldsymbol{\alpha}(t_j) = \begin{pmatrix} \boldsymbol{u}(t_j) \\ \boldsymbol{v}_1(t_j) \\ \vdots \\ \boldsymbol{v}_m(t_j) \end{pmatrix},$$

where $\boldsymbol{u}(t_j)$ is a $d_u \times 1$ population level state vector, and each $\boldsymbol{v}_i(t_j)$ is a $d_v \times 1$ subject level state vector, $i = 1, \ldots, m$. The initial distribution of the state vector is

$$\boldsymbol{\alpha}(t_1) \sim N(\boldsymbol{a}(t_1), P(t_1)), \tag{2.7}$$

where $\boldsymbol{a}(t_1)$ and $P(t_1)$ can be known or partially/completely unknown. The dimension of $\boldsymbol{y}(t_j)$ is $qm \times 1$ and the dimension of $\boldsymbol{\alpha}(t_j)$ is $(d_u + md_v) \times 1$. When the number of subjects m is large, both the observation vector and state vector have high dimensions. Let

$$Z(t_j) = \begin{pmatrix} Z_u(t_j) & Z_v(t_j) & & 0 \\ \vdots & & \ddots & \\ Z_u(t_j) & 0 & & Z_v(t_j) \end{pmatrix}, \tag{2.8}$$

which is a column block consisting of m identical matrices $Z_u(t_j)$ combined with a block diagonal matrix containing m identical matrices $Z_v(t_j)$. $Z_u(t_j)$ and $Z_v(t_j)$ are the system matrices defined in (2.1). Stacking the design matrices of all covariates together, write

$$X(t_j) = \begin{pmatrix} X_1(t_j) \\ \vdots \\ X_m(t_j) \end{pmatrix}, \tag{2.9}$$

where $X_i(t_j)$ is the design matrix defined in (2.1), $i = 1, \ldots, m$. Write the error term as

$$\boldsymbol{\epsilon}(t_j) = \begin{pmatrix} \boldsymbol{\epsilon}_1(t_j) \\ \vdots \\ \boldsymbol{\epsilon}_m(t_j) \end{pmatrix},$$

where $\boldsymbol{\epsilon}_i(t_j)$ is the vector of errors of the q variables for each subject, $i = 1, \ldots, m$. Then $\boldsymbol{\epsilon}(t_j) \sim N(\mathbf{0}, H(t_j))$ where

$$H(t_j) = \begin{pmatrix} \Sigma_\epsilon(t_j) & & 0 \\ & \ddots & \\ 0 & & \Sigma_\epsilon(t_j) \end{pmatrix}. \tag{2.10}$$

Let

$$T(t_j) = \begin{pmatrix} T_u(t_j) & \cdots & \cdots & 0 \\ \vdots & T_v(t_j) & \cdots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & & T_v(t_j) \end{pmatrix}, \tag{2.11}$$

Stack all state disturbance terms into one vector

$$\boldsymbol{\eta}(t_j) = \begin{pmatrix} \boldsymbol{\eta}_u(t_j) \\ \boldsymbol{\eta}_{v1}(t_j) \\ \vdots \\ \boldsymbol{\eta}_{vm}(t_j) \end{pmatrix}.$$

Then $\boldsymbol{\eta}(t_j) \sim N(\mathbf{0}, Q(t_j))$ where

$$Q(t_j) = \begin{pmatrix} \Sigma_u(t_j) & \cdots & \cdots & 0 \\ \vdots & \Sigma_v(t_j) & \cdots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & & \Sigma_v(t_j) \end{pmatrix}. \tag{2.12}$$

The vector form of the mixed effects state space model is

$$\begin{aligned} \boldsymbol{y}(t_j) &= Z(t_j)\boldsymbol{\alpha}(t_j) + X(t_j)\boldsymbol{\beta} + \boldsymbol{\epsilon}(t_j), & \boldsymbol{\epsilon}(t_j) &\sim N(\mathbf{0}, H(t_j)), \\ \boldsymbol{\alpha}(t_{j+1}) &= T(t_j)\boldsymbol{\alpha}(t_j) + R(t_j)\boldsymbol{\eta}(t_j), & \boldsymbol{\eta}(t_j) &\sim N(\mathbf{0}, Q(t_j)), \\ & & \boldsymbol{\alpha}(t_1) &\sim N(\boldsymbol{a}(t_1), P(t_1)), \end{aligned} \tag{2.13}$$

where

$$R(t_j) = \begin{pmatrix} R_u(t_j) & & & 0 \\ & R_v(t_j) & & \\ & & \ddots & \\ 0 & & & R_v(t_j) \end{pmatrix}, \tag{2.14}$$

is a selection matrix. In multivariate case, $R_u(t_j)$ and $R_v(t_j)$ can be used to model interactions among the latent states of different variables. The system matrices $Z(t_j)$, $T(t_j)$, $H(t_j)$, $Q(t_j)$, and $R(t_j)$ can contain unknown parameters. Depending on the initial state distribution, $\boldsymbol{a}(t_1)$ and $P(t_1)$ may also contain parameters.

2.2 Computations and Challenges

The unknown parameters in the state space model are estimated by maximizing the marginal likelihood of the observations, which is calculated via the Kalman filter. One of our main interests is the estimation of latent state vectors. This section details the steps of the Kalman filter, the computation of likelihood, the smoothing algorithm, and the computational challenges. The algorithms discussed in this section does not include the regression term $X(t_{ij})\boldsymbol{\beta}$ in the model. The regression term will later be added into the model, using the augmented Kalman filter.

2.2.1 Filtering

Let $\boldsymbol{Y}(t_j) = (\boldsymbol{y}^T(t_1), \ldots, \boldsymbol{y}^T(t_j))^T$ be historical observations up to time t_j. Let $\boldsymbol{a}(t_j) = \mathrm{E}(\boldsymbol{\alpha}(t_j)|\boldsymbol{Y}(t_{j-1}))$ and $P(t_j) = \mathrm{Var}(\boldsymbol{\alpha}(t_j)|\boldsymbol{Y}(t_{j-1}))$ be the one-step-ahead predictions of the mean and covariance matrix of the state vector $\boldsymbol{\alpha}(t_j)$, and $\boldsymbol{a}(t_j|t_j) = \mathrm{E}(\boldsymbol{\alpha}(t_j)|\boldsymbol{Y}(t_j))$ and $P(t_j|t_j) = \mathrm{Var}(\boldsymbol{\alpha}(t_j)|\boldsymbol{Y}(t_j))$ be the filtering estimates of the mean and covariance matrix of $\boldsymbol{\alpha}(t_j)$. The filtering equations for the state space model (2.13) are as follows:

$$\begin{aligned}
\boldsymbol{w}(t_j) &= \boldsymbol{y}(t_j) - Z(t_j)\boldsymbol{a}(t_j), \\
F(t_j) &= Z(t_j)P(t_j)Z^T(t_j) + H(t_j), \\
\boldsymbol{a}(t_j|t_j) &= \boldsymbol{a}(t_j) + P(t_j)Z^T(t_j)F^{-1}(t_j)\boldsymbol{w}(t_j), \\
P(t_j|t_j) &= P(t_j) - P(t_j)Z^T(t_j)F^{-1}(t_j)Z(t_j)P(t_j), \\
\boldsymbol{a}(t_{j+1}) &= T(t_j)\boldsymbol{a}(t_j|t_j), \\
P(t_{j+1}) &= T(t_j)P(t_j|t_j)T^T(t_j) + R(t_j)Q(t_j)R^T(t_j),
\end{aligned} \tag{2.15}$$

for $j = 1, \ldots, n$, where n is the total number of time points.

2.2.2 Likelihood

Assume for the moment that the initial condition $\boldsymbol{\alpha}(t_1) \sim N(\boldsymbol{a}(t_1), P(t_1))$ is known. The likelihood is calculated as

$$L(\boldsymbol{Y}(t_n)) = p(\boldsymbol{y}(t_1), \ldots, \boldsymbol{y}(t_n)) = \prod_{j=1}^{n} p(\boldsymbol{y}(t_j)|\boldsymbol{Y}(t_{j-1})), \tag{2.16}$$

with $p(\boldsymbol{y}(t_1)|\boldsymbol{Y}(t_0)) = p(\boldsymbol{y}(t_1))$. The density $p(\boldsymbol{y}(t_j)|\boldsymbol{Y}(t_{j-1}))$ is Gaussian with mean $\mathrm{E}(\boldsymbol{y}(t_j)|\boldsymbol{Y}(t_{j-1})) = Z(t_j)\boldsymbol{a}(t_j)$ and covariance matrix $\mathrm{Var}(\boldsymbol{y}(t_j)|\boldsymbol{Y}(t_{j-1})) = \boldsymbol{F}(t_j)$. The log-likelihood can be rewritten as

$$l(\boldsymbol{Y}(t_n)) = \log \sum_{j=1}^{n} p(\boldsymbol{w}(t_j)) = -\frac{nqm}{2}\log 2\pi - \frac{1}{2}\sum_{j=1}^{n}[\log|F(t_j)| + \boldsymbol{w}(t_j)^T F^{-1}(t_j)\boldsymbol{w}(t_j)], \tag{2.17}$$

where $\boldsymbol{w}(t_j)$ is the one-step-ahead prediction error, $\boldsymbol{w}(t_j) = \boldsymbol{y}(t_j) - Z(t_j)\boldsymbol{a}(t_j)$.

2.2.3 State smoothing

Let $\hat{\boldsymbol{\alpha}}(t_j) = \mathrm{E}(\boldsymbol{\alpha}(t_j)|\boldsymbol{Y}(t_n))$ and $V(t_j) = \mathrm{Var}(\boldsymbol{\alpha}(t_j)|\boldsymbol{Y}(t_n))$ be the posterior mean and covariance matrix of the state vector $\boldsymbol{\alpha}(t_j)$, $j = 1, \ldots, n$, given observations from all n time points. Then $\hat{\boldsymbol{\alpha}}(t_j)$ and $\mathrm{Var}(\boldsymbol{\alpha}(t_j))$ can be evaluated recursively via the following state smoothing recursion:

$$\begin{aligned} \boldsymbol{r}(t_{j-1}) &= Z^T(t_j)F^{-1}(t_j)\boldsymbol{w}(t_j) + L^T(t_j)\boldsymbol{r}(t_j), \\ N(t_{j-1}) &= Z^T(t_j)F^{-1}(t_j)Z(t_j) + L^T(t_j)N(t_j)L(t_j), \\ \hat{\boldsymbol{\alpha}}(t_j) &= \boldsymbol{a}(t_j) + P(t_j)\boldsymbol{r}_{t_{j-1}}, \\ V(t_j) &= P(t_j) - P(t_j)N(t_{j-1})P(t_j), \end{aligned} \tag{2.18}$$

for $j = n, \ldots, 1$, with initial values $\boldsymbol{r}(t_n) = \mathbf{0}$ and $N(t_n) = 0$; $L(t_j) = T(t_j) - K(t_j)Z(t_j)$ and $K(t_j) = T(t_j)P(t_j)Z^T(t_j)F^{-1}(t_j)$ can be calculated from the filtering step; $F^{-1}(t_j), \boldsymbol{w}(t_j), \boldsymbol{a}(t_j)$, and $P(t_j)$ are also computed during the filtering recursion.

2.2.4 Disturbance smoothing

Let $\hat{\boldsymbol{\epsilon}}(t_j) = E[\boldsymbol{\epsilon}(t_j)|\boldsymbol{Y}(t_n)]$ and $\hat{\boldsymbol{\eta}}(t_j) = E[\boldsymbol{\eta}(t_j)|\boldsymbol{Y}(t_n)]$ be the smoothing estimates of the mean of the disturbance vectors $\boldsymbol{\epsilon}(t_j)$ and $\boldsymbol{\eta}(t_j)$. They are calculated by the recursion

$$\begin{aligned} \boldsymbol{u}(t_j) &= F^{-1}(t_j)\boldsymbol{w}(t_j) - K^T(t_j)\boldsymbol{r}(t_j) \\ \hat{\boldsymbol{\epsilon}}(t_j) &= H(t_j)\boldsymbol{u}(t_j), \\ \hat{\boldsymbol{\eta}}(t_j) &= Q(t_j)R^T(t_j)\boldsymbol{r}(t_j), \\ \boldsymbol{r}(t_{j-1}) &= Z^T(t_j)\boldsymbol{u}(t_j) + T^T(t_j)\boldsymbol{r}(t_j). \end{aligned} \tag{2.19}$$

for $j = n, \dots, 1$, with $\boldsymbol{r}(t_n) = \boldsymbol{0}$. The smoothing estimates of the covariance matrices of $\boldsymbol{\epsilon}(t_j)$ and $\boldsymbol{\eta}(t_j)$ are calculated by the recursion

$$\begin{aligned}
D(t_j) &= F^{-1}(t_j) + K^T(t_j)N(t_j)K(t_j) \\
\mathrm{Var}[\boldsymbol{\epsilon}(t_j)|\boldsymbol{Y}(t_n)] &= H(t_j) - H(t_j)D(t_j)H(t_j), \\
\mathrm{Var}[\boldsymbol{\eta}(t_j)|\boldsymbol{Y}(t_n)] &= Q(t_j) - Q(t_j)R^T(t_j)N(t_j)R(t_j)Q(t_j), \\
N(t_{j-1}) &= Z^T(t_j)D(t_j)Z(t_j) + T^T(t_j)N(t_j)T(t_j) - \\
&\quad Z^T(t_j)K^T(t_j)N(t_j)T(t_j) - T^T(t_j)N(t_j)K(t_j)Z(t_j).
\end{aligned} \tag{2.20}$$

for $j = n, \dots, 1$, with $N(t_n) = 0$. (2.19) and (2.20) together are called disturbance smoothing recursion. The matrices/vectors $F^{-1}(t_j), \boldsymbol{w}(t_j)$, and $K(t_j)$ in (2.19) and (2.20) are obtained in the filtering step.

2.2.5 Fast state smoothing

In state smoothing, if one is only interested in obtaining the smoothed mean of the state vector but not the smoothed covariance matrix estimate, one can use fast state smoothing, which is computationally more efficient, given by

$$\begin{aligned}
\hat{\boldsymbol{\alpha}}(t_{j+1}) &= T(t_j)\hat{\boldsymbol{\alpha}}(t_j) + R(t_j)\hat{\boldsymbol{\eta}}(t_j) \\
&= T(t_j)\hat{\boldsymbol{\alpha}}(t_j) + R(t_j)Q(t_j)R^T(t_j)\boldsymbol{r}(t_j)
\end{aligned} \tag{2.21}$$

for $t = 1, \dots, n$, with $\hat{\boldsymbol{\alpha}}(t_1) = \boldsymbol{a}(t_1) + P(t_1)\boldsymbol{r}(t_0)$, where $\boldsymbol{r}(t_0)$ is obtained from the recursion specified in (2.19). The procedure of fast state smoothing is outlined as follows:

Step 1: for $j = 1, \dots, n$, perform the filtering recursion (2.15).

Step 2: for $j = n, \dots, 1$, carry out the disturbance smoothing recursion for disturbance means given in (2.19).

Step 3: for $j = 1, \dots, n$, use the fast state smoothing recursion (2.21) to compute smoothed state.

2.2.6 Computational difficulties

The dimension of $\boldsymbol{y}(t_j)$ is qm and the dimension of $\boldsymbol{\alpha}(t_j)$ is $d_u + md_v$. In the Kalman filter equations (2.15), $F(t_j)$ is calculated by

$$F(t_j) = Z(t_j)P(t_j)Z^T(t_j) + H(t_j), \tag{2.22}$$

where $Z(t_j)$ is $qm \times (d_u + md_v)$ and $P(t_j)$ is $(d_u + md_v) \times (d_u + md_v)$. So the time complexity of matrix multiplication $Z(t_j)P(t_j)$ is $qm(d_u + md_v)^2$. Since d_u and d_v are multiples of q, the time complexity of (2.22) is hence $O(q^3m^3)$. The equation to compute $P(t_j|t_j)$ in the Kalman filter is given by

$$P(t_j|t_j) = P(t_j) - P(t_j)Z^T(t_j)F^{-1}(t_j)\boldsymbol{w}(t_j), \tag{2.23}$$

where $P(t_j)$ is $(d_u + md_v) \times (d_u + md_v)$ and $F^{-1}(t_j)$ is $qm \times qm$. The time complexity of inverting the $qm \times qm$ matrix $F(t_j)$ varies from $O((qm)^{2.373})$ to $O((qm)^3)$, depending on the matrix inversion algorithm ([54]). Matrix multiplication $P(t_j)Z^T(t_j)F^{-1}(t_j)$ are of time complexity $O(q^3m^3)$. So (2.23) has time complexity $O(q^3m^3)$. Other parts of the Kalman filter have the same or less time complexity. In total there are n time points, so the time complexity of the Kalman filter (2.15) is $O(q^3m^3n)$. The same time complexity applies to the smoothing algorithm. In computing the likelihood, the time complexity for calculating the determinant of $F(t_j)$ ranges from $O((qm)^{2.373})$ to $O((qm)!)$ (Laplace expansion), depending on the algorithm ([54]).

Under the univariate treatment of the state space model, in the filtering recursion, each step involves multiplication between a $(d_u + md_v) \times (d_u + md_v)$ matrix and a $(d_u +$

$md_v)\times 1$ vector or similar computation, which is of time complexity $O(q^2m^2)$. Meanwhile, there are in total qmn "time points" for the univariate series. So the over all time complexity is $O(m^3q^3n)$, the same as the original Kalman filter. But the univariate treatment is more efficient by avoiding the inversion of the $qm \times qm$ matrix $F(t_j)$.

Both the original Kalman filter and the univariate treatment require the storage of matrices of dimensions on the order of $qm \times qm$ at each time point, thus the space complexity is $O(q^2m^2)$. The smoothing algorithm requires the storage of matrices from all time points, hence the space complexity is $O(q^2m^2n)$.

According to our simulation study, the computations of both the Kalman filter and the univariate treatment become impossible when m is only moderately large.

2.3 An Efficient New Algorithm

We present in this section a new algorithm for the mixed effects state space model, reducing time complexity to $O(mq^3n)$. We first list three assumptions made in the proposed mixed effects state space model. The new algorithm is based on these assumptions.

Assumption 1 *The random errors* $\boldsymbol{\epsilon}_i(t_j) \overset{iid}{\sim} N(\mathbf{0}, \Sigma_\epsilon(t_j))$, $i = 1, \ldots, m$. *The state disturbance terms* $\boldsymbol{\eta}_u(t_j) \overset{iid}{\sim} N(\mathbf{0}, \Sigma_u(t_j))$ *and* $\boldsymbol{\eta}_{vi}(t_j) \overset{iid}{\sim} N(\mathbf{0}, \Sigma_v(t_j))$, $i = 1, \ldots, m$. *And* $\boldsymbol{\epsilon}_i(t_j)$, $\boldsymbol{\eta}_u(t_j)$, *and* $\boldsymbol{\eta}_{vi}(t_j)$ *are mutually and serially independent.*

Assumption 2 $\boldsymbol{u}(t_1)$ *and* $\boldsymbol{v}_i(t_1)$ *are mutually independent, with* $\boldsymbol{u}(t_1) \sim N(\boldsymbol{\mu}_u, P_u)$ *and* $\boldsymbol{v}_i(t_1) \overset{iid}{\sim} N(\mathbf{0}, P_v)$, $i = 1, \ldots, m$.

Assumption 3 *The subject-deviation state components* $\boldsymbol{v}_i(t_j)$, $i = 1, \ldots, m$, *of the state vector* $\boldsymbol{\alpha}(t_j) = (\boldsymbol{u}^T(t_j), \boldsymbol{v}_1(t_j), \ldots, \boldsymbol{v}_m(t_j))^T$, *evolve in the same way over time for all subjects. That is, the state equation for subject deviations,* $\boldsymbol{v}_i(t_{j+1}) = T_v(t_j)\boldsymbol{v}_i(t_j) + \boldsymbol{\eta}_{vi}(t_j)$, *share a common system matrix* $T_v(t_j)$ *for all* $i = 1, \ldots, m$.

Under Assumptions 1 - 3, a close examination of the vectors and matrices in the filtering equations (2.15) unveils some special structures, presented in Theorem 1.

Theorem 1 *(a) $P(t_j)$ has the structure*

$$P(t_j) = \begin{pmatrix} P_0(t_j) & \mathbf{1}_m^T \otimes P_1(t_j) \\ \mathbf{1}_m \otimes P_1^T(t_j) & I_m \otimes P_2(t_j) + \mathbf{1}_{m\times m} \otimes P_3(t_j) \end{pmatrix}, \tag{2.24}$$

for $j = 1, \ldots, n$, where $\mathbf{1}_m$ is an $m \times 1$ vector with all elements equal to 1, $\mathbf{1}_{m\times m}$ is an $m \times m$ matrix with all elements equal to 1, $P_0(t_j)$ is a $d_u \times d_u$ matrix, $P_1(t_j)$ is a $d_u \times d_v$ matrix, $P_2(t_j)$ and $P_3(t_j)$ are $d_v \times d_v$ matrices. In addition, $P_0(t_j)$, $P_2(t_j)$, and $P_3(t_j)$ are symmetric, and $P_0(t_j)$ and $P_2(t_j) + P_3(t_j)$ are positive semi-definite.

(b) $F^{-1}(t_j)$ has the structure

$$F^{-1}(t_j) = I_m \otimes F_1(t_j) + \mathbf{1}_{m\times m} \otimes F_2(t_j), \tag{2.25}$$

where $F_1(t_j)$ and $F_2(t_j)$ are $q \times q$ square matrices.

(c) $P(t_j|t_j)$ has the structure

$$P(t_j|t_j) = \begin{pmatrix} P_0(t_j|t_j) & \mathbf{1}_m^T \otimes P_1(t_j|t_j) \\ \mathbf{1}_m \otimes P_1^T(t_j|t_j) & I_m \otimes P_2(t_j|t_j) + \mathbf{1}_{m\times m} \otimes P_3(t_j|t_j) \end{pmatrix}, \tag{2.26}$$

where $P_0(t_j|t_j)$ is a $d_u \times d_u$ matrix, $P_1(t_j|t_j)$ is a $d_u \times d_v$ matrix, and $P_2(t_j|t_j)$ and $P_3(t_j|t_j)$ are $d_v \times d_v$ matrices. $P_0(t_j|t_j)$, $P_2(t_j|t_j)$, and $P_3(t_j|t_j)$ are symmetric, and $P_0(t_j|t_j)$ and $P_2(t_j|t_j) + P_3(t_j|t_j)$ are positive semi-definite.

We prove Theorem 1 by induction.

Proof: For $t_j = t_1$, by Assumption 2,

$$P(t_1) = \begin{pmatrix} P_u & & & 0 \\ & P_v & & \\ & & \ddots & \\ 0 & & & P_v \end{pmatrix}, \tag{2.27}$$

so $P(t_1)$ is of the form (2.24), where $P_0(t_1) = P_u$, $P_1(t_1) = 0$, $P_3(t_1) = 0$, and $P_2(t_1) = P_v$. It is for the convenience of computation and representation that matrix P_v is split into the sum of $P_2(t_1)$ and $P_3(t_1)$.

$F(t_1)$ is calculated by the filtering recursion $F(t_1) = Z(t_1)P(t_1)Z^T(t_1) + H(t_1)$, where $Z(t_1)$ and $H(t_1)$ are given in (2.8) and (2.10) with $j = 1$. It is straight forward to show that

$$F(t_1) = I_m \otimes A(t_1) + \mathbf{1}_{m\times m} \otimes B(t_1), \tag{2.28}$$

where

$$\begin{aligned} A(t_1) &= Z_v(t_1)P_2(t_1)Z_v^T(t_1) + \Sigma_\epsilon(t_1), \\ B(t_1) &= Z_u(t_1)P_0(t_1)Z_u^T(t_1) + Z_u(t_1)P_1(t_1)Z_v^T(t_1) + \\ &\quad Z_v(t_1)P_1^T(t_1)Z_u^T(t_1) + Z_v(t_1)P_3(t_1)Z_v^T(t_1), \end{aligned} \tag{2.29}$$

$A(t_1)$ and $B(t_1)$ are both $q \times q$. To calculate $F^{-1}(t_1)$, rewrite $F(t_1)$ as

$$F(t_1) = I_m \otimes A(t_1) + [\mathbf{1}_m \otimes B(t_1)][\mathbf{1}_m^T \otimes I_q]. \tag{2.30}$$

By the Woodbury formula,

$$\begin{aligned} F_t^{-1}(t_1) &= I_m \otimes A^{-1}(t_1) - [I_m \otimes A^{-1}(t_1)][\mathbf{1}_m \otimes B(t_1)]\{I_q^{-1}+ \\ &\quad (\mathbf{1}_m^T \otimes I_q)[I_m \otimes A^{-1}(t_1)][\mathbf{1}_m \otimes B(t_1)]\}^{-1}(\mathbf{1}_m^T \otimes I_q)[I_m \otimes A^{-1}(t_1)] \\ &= I_m \otimes A^{-1}(t_1) - \mathbf{1}_{m\times m} \otimes \{A^{-1}(t_1)B(t_1)[I_q + mA^{-1}(t_1)B(t_1)]^{-1}A^{-1}(t_1)\} \\ &= I_m \otimes F_1(t_1) + \mathbf{1}_{m\times m} \otimes F_2(t_1), \end{aligned} \tag{2.31}$$

where

$$\begin{aligned} F_1(t_1) &= A^{-1}(t_1), \\ F_2(t_1) &= -A^{-1}(t_1)B(t_1)[I_q + mA^{-1}(t_1)B(t_1)]^{-1}A^{-1}(t_1) \\ &= -A^{-1}(t_1)B(t_1)[A(t_1) + mB(t_1)]^{-1}. \end{aligned} \tag{2.32}$$

$P(t_1|t_1)$ is calculated by the equation

$$P(t_1|t_1) = P(t_1) - P(t_1)Z^T(t_1)F^{-1}(t_1)Z(t_1)P(t_1). \tag{2.33}$$

Based on (2.8) and (2.27)-(2.32), we have

$$P(t_1)Z^T(t_1)F^{-1}(t_1)Z(t_1)P(t_1) = \begin{pmatrix} M_0(t_1) & \mathbf{1}_m^T \otimes M_1(t_1) \\ \mathbf{1}_m \otimes M_1^T(t_1) & I_m \otimes M_2(t_1) + \mathbf{1}_{m\times m} \otimes M_3(t_1) \end{pmatrix}, \tag{2.34}$$

where

$$\begin{aligned} M_0(t_1) &= m[P_0(t_1)Z_u^T(t_1) + P_1(t_1)Z_v^T(t_1)][F_1(t_1)+ \\ &\quad mF_2(t_1)][Z_u(t_1)P_0(t_1) + Z_v(t_1)P_1^T(t_1)], \\ M_1(t_1) &= [P_0(t_1)Z_u^T(t_1) + P_1(t_1)Z_v^T(t_1)][F_1(t_1) + mF_2(t_1)][mZ_u(t_1)P_1(t_1)+ \\ &\quad Z_v(t_1)P_2(t_1) + mZ_v(t_1)P_3(t_1)], \\ M_2(t_1) &= P_2(t_1)Z_v^T(t_1)F_1(t_1)Z_v(t_1)P_2(t_1), \\ M_3(t_1) &= P_2(t_1)Z_v^T(t_1)F_2(t_1)Z_v(t_1)P_2(t_1)+ \\ &\quad P_2(t_1)Z_v^T(t_1)[F_1(t_1) + mF_2(t_1)][Z_u(t_1)P_1(t_1) + Z_v(t_1)P_3(t_1)]+ \\ &\quad [P_1^T(t_1)Z_u^T(t_1) + P_3(t_1)Z_v^T(t_1)][F_1(t_1) + mF_2(t_1)]Z_v(t_1)P_2(t_1)+ \\ &\quad m[P_1^T(t_1)Z_u^T(t_1) + P_3(t_1)Z_v^T(t_1)][F_1(t_1) + mF_2(t_1)][Z_u(t_1)P_1(t_1)+ \\ &\quad Z_v(t_1)P_3(t_1)]. \end{aligned} \tag{2.35}$$

So we have

$$P(t_1|t_1) = \begin{pmatrix} P_0(t_1|t_1) & \mathbf{1}_m^T \otimes P_1(t_1|t_1) \\ \mathbf{1}_m \otimes P_1^T(t_1|t_1) & I_m \otimes P_2(t_1|t_1) + \mathbf{1}_{m\times m} \otimes P_3(t_1|t_1) \end{pmatrix}, \tag{2.36}$$

where

$$\begin{aligned} P_0(t_1|t_1) &= P_0(t_1) - M_0(t_1), \\ P_1(t_1|t_1) &= P_1(t_1) - M_1(t_1), \\ P_2(t_1|t_1) &= P_2(t_1) - M_2(t_1), \\ P_3(t_1|t_1) &= P_3(t_1) - M_3(t_1). \end{aligned} \tag{2.37}$$

So far we have proved that Theorem 1 holds for $t_j = t_1$. Suppose that Theorem 1 holds for $t_1, \ldots, t_j$, $j < n$, We prove that it is also true for t_{j+1}.

$P(t_{j+1})$ is given by the Kalman filter equation

$$P(t_{j+1}) = T(t_j)P(t_j|t_j)T^T(t_j) + R(t_j)Q(t_j)R^T(t_j), \tag{2.38}$$

where $T(t_j)$, $Q(t_j)$, and $R(t_j)$ are given in (2.11), (2.12), and (2.14), and

$$P(t_j|t_j) = \begin{pmatrix} P_0(t_j|t_j) & \mathbf{1}_m^T \otimes P_1(t_j|t_j) \\ \mathbf{1}_m \otimes P_j^t(t_j|t_j) & I_m \otimes P_2(t_j|t_j) + \mathbf{1}_{m\times m} \otimes P_3(t_j|t_j) \end{pmatrix}. \tag{2.39}$$

Plugging (2.39) into (2.38), it is not difficult to show that

$$P(t_{j+1}) = \begin{pmatrix} P_0(t_{j+1}) & \mathbf{1}_m^T \otimes P_1(t_{j+1}) \\ \mathbf{1}_m \otimes P_1^T(t_{j+1}) & I_m \otimes P_2(t_{j+1}) + \mathbf{1}_{m\times m} \otimes P_3(t_{j+1}) \end{pmatrix}, \tag{2.40}$$

where

$$\begin{aligned} P_0(t_{j+1}) &= T_u(t_j)P_0(t_j|t_j)T_u^T(t_j) + R_u(t_j)\Sigma_u(t_j)R_u^T(t_j), \\ P_1(t_{j+1}) &= T_u(t_j)P_1(t_j|t_j)T_v^T(t_j), \\ P_2(t_{j+1}) &= T_v(t_j)P_2(t_j|t_j)T_v^T(t_j) + R_v(t_j)\Sigma_v(t_j)R_v^T(t_j), \\ P_3(t_{j+1}) &= T_v(t_j)P_3(t_j|t_j)T_v^T(t_j). \end{aligned} \tag{2.41}$$

Similar to (2.28)-(2.45), one can show that

$$F(t_{j+1}) = I_m \otimes A(t_{j+1}) + \mathbf{1}_{m\times m} \otimes B(t_{j+1}), \tag{2.42}$$

where

$$\begin{aligned} A(t_{j+1}) &= Z_v(t_{j+1})P_2(t_{j+1})Z_v^T(t_{j+1}) + \Sigma_\epsilon(t_{j+1}), \\ B(t_{j+1}) &= Z_u(t_{j+1})P_0(t_{j+1})Z_u^T(t_{j+1}) + Z_u(t_{j+1})P_1(t_{j+1})Z_v^T(t_{j+1}) + \\ &\quad Z_v(t_{j+1})P_1^T(t_{j+1})Z_u^T(t_{j+1}) + Z_v(t_{j+1})P_3(t_{j+1})Z_v^T(t_{j+1}), \end{aligned} \tag{2.43}$$

and

$$F_t^{-1}(t_{j+1}) = I_m \otimes F_1(t_{j+1}) + \mathbf{1}_{m\times m} \otimes F_2(t_{j+1}), \tag{2.44}$$

where

$$\begin{aligned} F_1(t_{j+1}) &= A^{-1}(t_{j+1}), \\ F_2(t_{j+1}) &= -A^{-1}(t_{j+1})B(t_{j+1})[I_q + mA^{-1}(t_{j+1})B(t_{j+1})]^{-1}A^{-1}(t_{j+1}) \\ &= -A^{-1}(t_{j+1})B(t_{j+1})[A(t_{j+1}) + mB(t_{j+1})]^{-1}. \end{aligned} \tag{2.45}$$

Similar to (2.33)-(2.47), one can show that

$$P(t_{j+1}|t_{j+1}) = \begin{pmatrix} P_0(t_{j+1}|t_{j+1}) & \mathbf{1}_m^T \otimes P_1(t_{j+1}|t_{j+1}) \\ \mathbf{1}_m \otimes P_1^T(t_{j+1}|t_{j+1}) & I_m \otimes P_2(t_{j+1}|t_{j+1}) + \mathbf{1}_{m\times m} \otimes P_3(t_{j+1}|t_{j+1}) \end{pmatrix}, \tag{2.46}$$

where

$$\begin{aligned} P_0(t_{j+1}|t_{j+1}) &= P_0(t_{j+1}) - M_0(t_{j+1}), \\ P_1(t_{j+1}|t_{j+1}) &= P_1(t_{j+1}) - M_1(t_{j+1}), \\ P_2(t_{j+1}|t_{j+1}) &= P_2(t_{j+1}) - M_2(t_{j+1}), \\ P_3(t_{j+1}|t_{j+1}) &= P_3(t_{j+1}) - M_3(t_{j+1}), \end{aligned} \tag{2.47}$$

and

$$
\begin{aligned}
M_0(t_{j+1}) &= m[P_0(t_{j+1})Z_u^T(t_{j+1}) + P_1(t_{j+1})Z_v^T(t_{j+1})][F_1(t_{j+1})+ \\
&\quad mF_2(t_{j+1})][Z_u(t_{j+1})P_0(t_{j+1}) + Z_v(t_{j+1})P_1^T(t_{j+1})], \\
M_1(t_{j+1}) &= [P_0(t_{j+1})Z_u^T(t_{j+1}) + P_1(t_{j+1})Z_v^T(t_{j+1})][F_1(t_{j+1})+ \\
&\quad mF_2(t_{j+1})][mZ_u(t_{j+1})P_1(t_{j+1})+ \\
&\quad Z_v(t_{j+1})P_2(t_{j+1}) + mZ_v(t_{j+1})P_3(t_{j+1})], \\
M_2(t_{j+1}) &= P_2(t_{j+1})Z_v^T(t_{j+1})F_1(t_{j+1})Z_v(t_{j+1})P_2(t_{j+1}), \\
M_3(t_{j+1}) &= P_2(t_{j+1})Z_v^T(t_{j+1})F_2(t_{j+1})Z_v(t_{j+1})P_2(t_{j+1})+ \qquad (2.48) \\
&\quad P_2(t_{j+1})Z_v^T(t_{j+1})[F_1(t_{j+1})+ \\
&\quad mF_2(t_{j+1})][Z_u(t_{j+1})P_1(t_{j+1}) + Z_v(t_{j+1})P_3(t_{j+1})]+ \\
&\quad [P_1^T(t_{j+1})Z_u^T(t_{j+1}) + P_3(t_{j+1})Z_v^T(t_{j+1})][F_1(t_{j+1})+ \\
&\quad mF_2(t_{j+1})]Z_v(t_{j+1})P_2(t_{j+1})+ \\
&\quad m[P_1^T(t_{j+1})Z_u^T(t_{j+1}) + P_3(t_{j+1})Z_v^T(t_{j+1})][F_1(t_{j+1})+ \\
&\quad mF_2(t_{j+1})][Z_u(t_{j+1})P_1(t_{j+1}) + Z_v(t_{j+1})P_3(t_{j+1})].
\end{aligned}
$$

■

2.3.1 Kalman filter utilizing the special structure

Based on Theorem 1, it turns out that every matrix or vector in the Kalman filter recursion has a special structure. For the high dimensional matrices, one only needs to keep track of a few small matrices whose dimensions are on the order of $q \times q$. The vectors can be partitioned into either m components corresponding to m subjects, or with one extra component for population effects. These components share some common terms and can be computed independently. The Kalman filter algorithm is thus tremendously simplified.

In (2.15), the one-step-ahead prediction vector $\boldsymbol{w}(t_j)$ can be written as

$$\boldsymbol{w}(t_j) = \begin{pmatrix} \boldsymbol{w}_1(t_j) \\ \vdots \\ \boldsymbol{w}_m(t_j) \end{pmatrix} \tag{2.49}$$

consisting of m components corresponding to the m subjects, where each $\boldsymbol{w}_i(t_j)$ is $q \times 1$. The vectors $\boldsymbol{a}(t_j)$ and $\boldsymbol{a}(t_j|t_j)$ can be written as

$$\boldsymbol{a}(t_j) = \begin{pmatrix} \boldsymbol{a}_0(t_j) \\ \boldsymbol{a}_1(t_j) \\ \vdots \\ \boldsymbol{a}_m(t_j) \end{pmatrix} \quad \text{and} \quad \boldsymbol{a}(t_j|t_j) = \begin{pmatrix} \boldsymbol{a}_0(t_j|t_j) \\ \boldsymbol{a}_1(t_j|t_j) \\ \vdots \\ \boldsymbol{a}_m(t_j|t_j) \end{pmatrix}, \tag{2.50}$$

containing $m+1$ components, where the first component is at the population level and the rest m components correspond to the m subjects. $\boldsymbol{a}_0(t_j)$ and $\boldsymbol{a}_0(t_j|t_j)$ are $d_u \times 1$ vectors, and $\boldsymbol{a}_i(t_j)$ and $\boldsymbol{a}_i(t_j|t_j)$ are $d_v \times 1$ vectors for $i = 1, \dots, m$.

Given $Z(t_j)$ specified in (2.8), the first recursion equation in the Kalman filter (2.15) can be computed as

$$\boldsymbol{w}(t_j) = \boldsymbol{y}(t_j) - Z(t_j)\boldsymbol{a}(t_j) = \begin{pmatrix} \boldsymbol{y}_1(t_j) \\ \vdots \\ \boldsymbol{y}_m(t_j) \end{pmatrix} - \begin{pmatrix} Z_u(t_j)\boldsymbol{a}_0(t_j) + Z_v(t_j)\boldsymbol{a}_1(t_j) \\ \vdots \\ Z_u(t_j)\boldsymbol{a}_0(t_j) + Z_v(t_j)\boldsymbol{a}_m(t_j) \end{pmatrix}, \tag{2.51}$$

that is, instead of multiplying the $qm \times (d_u + md_v)$ matrix $Z(t_j)$ with the $(d_u + md_v) \times 1$ vector $\boldsymbol{\alpha}(t_j)$, we only need to compute $Z_u(t_j)\boldsymbol{a}_0(t_j)$ and $Z_v(t_j)\boldsymbol{a}_i(t_j)$ for $i = 1, \dots, m$. In addition, the calculation of $Z_v(t_j)\boldsymbol{a}_i(t_j)$ and $Z_u(t_j)\boldsymbol{a}_0(t_j) + Z_v(t_j)\boldsymbol{a}_i(t_j)$ for $i = 1, \dots, m$ is parallelizable. The time complexity of calculating $\boldsymbol{w}(t_j)$ is reduced from $O(q^2m^2)$ to

$O(q^2m)$. Since we do not need to store $Z(t_j)$, but $Z_u(t_j)$ and $Z_v(t_j)$ instead, and we need to store $\boldsymbol{a}(t_j)$ and $\boldsymbol{w}(t_j)$, the space complexity of (2.51) is reduced from $O(q^2m^2)$ to $O(q^2 + qm)$.

In the second equation of the Kalman filter (2.15),

$$F(t_j) = Z(t_j)P(t_j)Z^T(t_j) + H(t_j), \tag{2.52}$$

according to Theorem 1, we have

$$F(t_j) = I_m \otimes A(t_j) + \mathbf{1}_{m\times m} \otimes B(t_j), \tag{2.53}$$

and

$$F^{-1}(t_j) = I_m \otimes F_1(t_j) + \mathbf{1}_{m\times m}F_2(t_j), \tag{2.54}$$

where $A(t_j)$ and $B(t_j)$ are given in (2.43), and $F_1(t_j)$ and $F_2(t_j)$ are given in (2.45). Therefore, for $F^{-1}(t_j)$, we only need to compute the two $q \times q$ matrices $F_1(t_j)$ and $F_2(t_j)$. Also, instead of the $qm \times qm$ matrix $F(t_j)$, we only need to store $F_1(t_j)$ and $F_2(t_j)$. The time complexity of calculating (2.52) and $F^{-1}(t_j)$ is reduced from $O(q^3m^3)$ to $O(q^3m)$. The space complexity is reduced from $O(q^2m^2)$ to $O(q^2)$.

In the third equation of the Kalman filter (2.15),

$$\boldsymbol{a}(t_j|t_j) = \boldsymbol{a}(t_j) + P(t_j)Z^T(t_j)F^{-1}(t_j)\boldsymbol{w}(t_j), \tag{2.55}$$

it can be shown that

$$P(t_j)Z^T(t_j)F^{-1}(t_j) = \begin{pmatrix} \mathbf{1}_m^T \otimes C_0(t_j) \\ I_m \otimes C_1(t_j) + \mathbf{1}_{m\times m} \otimes C_2(t_j) \end{pmatrix}, \tag{2.56}$$

where

$$\begin{aligned} C_0(t_j) &= [P_0(t_j)Z_u^T(t_j) + P_1(t_j)Z_v^T(t_j)][F_1(t_j) + mF_2(t_j)], \\ C_1(t_j) &= P_2(t_j)Z_v^T(t_j)F_1(t_j), \\ C_2(t_j) &= P_2(t_j)Z_v^T(t_j)F_2(t_j) + [P_1^T(t_j)Z_u^T(t_j) + P_3(t_j)Z_v^T(t_j)][F_1(t_j) + mF_2(t_j)]. \end{aligned} \tag{2.57}$$

The multiplication $P(t_j)Z^T(t_j)F^{-1}(t_j)$ of three matrices of dimensions on the order of $qm \times qm$ is now simplified into computing the $d_u \times q$ matrix $C_0(t_j)$ and the $d_v \times q$ matrices $C_1(t_j)$ and $C_2(t_j)$. The time complexity of (2.55) is reduced from $O(q^3m^3)$ to $O(q^3m)$. The space complexity is reduced from $O(q^2m^2)$ to $O(q^2)$. Vector $\boldsymbol{a}(t_j|t_j)$ in (2.55) is hence given by

$$\boldsymbol{a}(t_j|t_j) = \begin{pmatrix} \boldsymbol{a}_0(t_j) \\ \boldsymbol{a}_1(t_j) \\ \vdots \\ \boldsymbol{a}_m(t_j) \end{pmatrix} + \begin{pmatrix} C_0(t_j)\sum_{i=1}^m \boldsymbol{w}_i(t_j) \\ C_1(t_j)\boldsymbol{w}_1(t_j) + C_2(t_j)\sum_{i=1}^m \boldsymbol{w}_i(t_j) \\ \vdots \\ C_1(t_j)\boldsymbol{w}_m(t_j) + C_2(t_j)\sum_{i=1}^m \boldsymbol{w}_i(t_j) \end{pmatrix}. \tag{2.58}$$

In the fourth equation in the Kalman filter (2.15),

$$P(t_j|t_j) = P(t_j) - P(t_j)Z^T(t_j)F^{-1}(t_j)Z(t_j)P(t_j), \tag{2.59}$$

it can be shown that

$$P(t_j)Z^T(t_j)F^{-1}(t_j)Z(t_j)P(t_j) = \begin{pmatrix} M_0(t_j) & \mathbf{1}_m^T \otimes M_1(t_j) \\ \mathbf{1}_m \otimes M_1^T(t_j) & I_m \otimes M_2(t_j) + \mathbf{1}_{m\times m} \otimes M_3(t_j) \end{pmatrix}, \tag{2.60}$$

where

$$\begin{aligned}
M_0(t_j) &= m[P_0(t_j)Z_u^T(t_j) + P_1(t_j)Z_v^T(t_j)][F_1(t_j)+ \\
&\quad mF_2(t_j)][Z_u(t_j)P_0(t_j) + Z_v(t_j)P_1^T(t_j)], \\
M_1(t_j) &= [P_0(t_j)Z_u^T(t_j) + P_1(t_j)Z_v^T(t_j)][F_1(t_j) + mF_2(t_j)][mZ_u(t_j)P_1(t_j)+ \\
&\quad Z_v(t_j)P_2(t_j) + mZ_v(t_j)P_3(t_j)], \\
M_2(t_j) &= P_2(t_j)Z_v^T(t_j)F_1(t_j)Z_v(t_j)P_2(t_j), \\
M_3(t_j) &= P_2(t_j)Z_v^T(t_j)F_2(t_j)Z_v(t_j)P_2(t_j)+ \\
&\quad P_2(t_j)Z_v^T(t_j)[F_1(t_j) + mF_2(t_j)][Z_u(t_j)P_1(t_j) + Z_v(t_j)P_3(t_j)]+ \\
&\quad [P_1^T(t_j)Z_u^T(t_j) + P_3(t_j)Z_v^T(t_j)][F_1(t_j) + mF_2(t_j)]Z_v(t_j)P_2(t_j)+ \\
&\quad m[P_1^T(t_j)Z_u^T(t_j) + P_3(t_j)Z_v^T(t_j)][F_1(t_j)+ \\
&\quad mF_2(t_j)][Z_u(t_j)P_1(t_j) + Z_v(t_j)P_3(t_j)].
\end{aligned} \tag{2.61}$$

So (2.59) becomes

$$P(t_j|t_j) = \begin{pmatrix} P_0(t_j|t_j) & \mathbf{1}_m^T \otimes P_1(t_j|t_j) \\ \mathbf{1}_m \otimes P_1^T(t_j|t_j) & I_m \otimes P_2(t_j|t_j) + \mathbf{1}_{m\times m} \otimes P_3(t_j|t_j) \end{pmatrix}, \tag{2.62}$$

where

$$\begin{aligned}
P_0(t_j|t_j) &= P_0(t_j) - M_0(t_j), \\
P_1(t_j|t_j) &= P_1(t_j) - M_1(t_j), \\
P_2(t_j|t_j) &= P_2(t_j) - M_2(t_j), \\
P_3(t_j|t_j) &= P_3(t_j) - M_3(t_j).
\end{aligned} \tag{2.63}$$

As stated before, the multiplications of the matrices with dimensions on the order of $qm \times qm$ is now reduced to computing the matrices $P_0(t_j|t_j)$, $P_1(t_j|t_j)$, $P_2(t_j|t_j)$, and $P_3(t_j|t_j)$, whose dimensions are on the order of $q \times q$, since d_u and d_v are multiples of q. And the required space for storage is now independent of m.

The fifth equation in the Kalman filter (2.15) becomes

$$\boldsymbol{a}(t_{j+1}) = \begin{pmatrix} T_u(t_j)\boldsymbol{a}_0(t_j|t_j) \\ T_v(t_j)\boldsymbol{a}_1(t_j|t_j) \\ \vdots \\ T_v(t_j)\boldsymbol{a}_m(t_j|t_j). \end{pmatrix} \tag{2.64}$$

In the sixth equation of the Kalman filter (2.15), we have

$$P(t_{j+1}) = \begin{pmatrix} P_0(t_{j+1}) & \mathbf{1}_m^T \otimes P_1(t_{j+1}) \\ \mathbf{1}_m \otimes P_1^T(t_{j+1}) & I_m \otimes P_2(t_{j+1}) + \mathbf{1}_{m\times m} \otimes P_3(t_{j+1}) \end{pmatrix}, \tag{2.65}$$

where

$$\begin{aligned} P_0(t_{j+1}) &= T_u(t_j)P_0(t_j|t_j)T_u^T(t_j) + R_u(t_j)\Sigma_u(t_j)R_u^T(t_j), \\ P_1(t_{j+1}) &= T_u(t_j)P_1(t_j|t_j)T_v^T(t_j), \\ P_2(t_{j+1}) &= T_v(t_j)P_2(t_j|t_j)T_v^T(t_j) + R_v(t_j)\Sigma_v(t_j)R_v^T(t_j), \\ P_3(t_{j+1}) &= T_v(t_j)P_3(t_j|t_j)T_v^T(t_j). \end{aligned} \tag{2.66}$$

As can be seen from the above algorithm, for matrix multiplications in the filtering recursion, instead of calculating $P(t_j), P(t_j|t_j)$, and $F(t_j)$, we only need to compute small matrices $P_0(t_j), P_1(t_j), P_2(t_j), P_3(t_j),\ P_0(t_j|t_j), P_1(t_j|t_j), P_2(t_j|t_j), P_3(t_j|t_j),\ F_1(t_j)$, and $F_2(t_j)$, whose dimensions are independent of the number of subjects. The computation complexity of the vectors $\boldsymbol{w}(t_j)$, $\boldsymbol{a}(t_j|t_j)$, and $\boldsymbol{a}(t_{j+1})$ is also significantly reduced. The time complexity of this algorithm is $O(q^3m)$. The space complexity is $O(q^2+qm)$, in contrast to the space complexity of the univariate treatment, $O(q^2m^2)$.

2.3.2 Computation of likelihood using the special structure

The computation of the log-likelihood in (2.17) can also be significantly simplified. Since

$$\log|F(t_j)| = \log|F^{-1}(t_j)|^{-1} = -\log|F^{-1}(t_j)|, \tag{2.67}$$

we compute $|F^{-1}(t_j)|$ instead of $|F(t_j)|$ here. But note that both $|F(t_j)|$ and $|F^{-1}(t_j)|$ can be computed easily using the special structure.

Suppose that the dimension of the square matrix $F_1(t_j)$ is $q \times q$. Since $|F^{-1}(t_j)|$ has the form (2.54), we have

$$|F^{-1}(t_j)| \;=\; \begin{vmatrix} F_1(t_j)+F_2(t_j) & \cdots & F_2(t_j) \\ & \ddots & \\ F_2(t_j) & \cdots & F_1(t_j)+F_2(t_j) \end{vmatrix}. \tag{2.68}$$

Adding all blocks to the first block row, we have

$$
\begin{aligned}
|F^{-1}(t_j)| &= \begin{vmatrix} F_1(t_j)+mF_2(t_j) & \cdots & F_1(t_j)+mF_2(t_j) \\ & \ddots & \\ F_2(t_j) & \cdots & F_1(t_j)+F_2(t_j) \end{vmatrix} \\
&= \begin{vmatrix} F_1(t_j)+mF_2(t_j) & 0 & \cdots & 0 \\ 0 & I_q & & \\ & & \ddots & \\ 0 & & & I_q \end{vmatrix} \cdot \\
&\quad \begin{vmatrix} I_q & I_q & \cdots & I_q \\ F_2(t_j) & F_1(t_j)+F_2(t_j) & & F_2(t_j) \\ & & \ddots & \\ F_2(t_j) & & & F_1(t_j)+F_2(t_j) \end{vmatrix} \\
&= |F_1(t_j)+mF_2(t_j)| \begin{vmatrix} I_q & I_q & \cdots & I_q \\ 0 & F_1(t_j) & & 0 \\ & & \ddots & \\ 0 & & & F_1(t_j) \end{vmatrix} \\
&= |F_1(t_j)+mF_2(t_j)| \begin{vmatrix} I_q & 0 & \cdots & 0 \\ 0 & F_1(t_j) & & 0 \\ & & \ddots & \\ 0 & & & F_1(t_j) \end{vmatrix} \\
&= |F_1(t_j)+mF_2(t_j)||F_1(t_j)|^{m-1}.
\end{aligned}
\tag{2.69}
$$

Hence, the calculation of $|F(t_j)|$ is reduced to computations related to $q \times q$ matrices, where q is the number of variables and is independent of the number of subjects.

The calculation of term $\boldsymbol{w}^T(t_j)F^{-1}(t_j)\boldsymbol{w}(t_j)$ in the log-likelihood (2.17) is simplified to

$$
\begin{aligned}
&\boldsymbol{w}^T(t_j)F^{-1}(t_j)\boldsymbol{w}(t_j)\\
&= \begin{pmatrix} \boldsymbol{w}_1^T(t_j) & \cdots & \boldsymbol{w}_m^T(t_j) \end{pmatrix} \begin{pmatrix} F_1(t_j)+F_2(t_j) & \cdots & F_2(t_j) \\ \vdots & \ddots & \vdots \\ F_2(t_j) & \cdots & F_1(t_j)+F_2(t_j) \end{pmatrix} \begin{pmatrix} \boldsymbol{w}_1(t_j) \\ \vdots \\ \boldsymbol{w}_m(t_j) \end{pmatrix}\\
&= \textstyle\sum_{i=1}^m [\boldsymbol{w}_i^T(t_j)F_1(t_j)\boldsymbol{w}_i(t_j)] + [\sum_{i=1}^m \boldsymbol{w}_i(t_j)]^T F_2(t_j)[\sum_{i=1}^m \boldsymbol{w}_i(t_j)],
\end{aligned}
\tag{2.70}
$$

the time complexity of which is linear in m.

2.3.3 State smoothing recursion using the special structure

In the state smoothing recursion (2.18), $\boldsymbol{r}(t_j)$ can be decomposed into $m+1$ components:

$$
\boldsymbol{r}(t_j) = \begin{pmatrix} \boldsymbol{r}_0(t_j) \\ \boldsymbol{r}_1(t_j) \\ \vdots \\ \boldsymbol{r}_m(t_j) \end{pmatrix}. \tag{2.71}
$$

The matrix $K(t_j) = T(t_j)P(t_j)Z^T(t_j)F^{-1}(t_j)$ can be computed during the filtering step and is given by

$$
K(t_j) = T(t_j)P(t_j)Z^T(t_j)F^{-1}(t_j) = \begin{pmatrix} \mathbf{1}_m^T \otimes K_0(t_j) \\ I_m \otimes K_1(t_j) + \mathbf{1}_{m\times m} \otimes K_2(t_j) \end{pmatrix}, \tag{2.72}
$$

where

$$
\begin{aligned}
K_0(t_j) &= T_u(t_j)C_0(t_j),\\
K_1(t_j) &= T_v(t_j)C_1(t_j),\\
K_2(t_j) &= T_v(t_j)C_2(t_j),
\end{aligned}
\tag{2.73}
$$

with $C_0(t_j), C_1(t_j)$, and $C_2(t_j)$ given in (2.57).

The matrix $L(t_j) = T(t_j) - K(t_j)Z(t_j)$ is given by

$$L(t_j) = \begin{pmatrix} L_0(t_j) & \mathbf{1}'_m \otimes L_1(t_j) \\ \mathbf{1}_m \otimes L_2(t_j) & \boldsymbol{I}_m \otimes L_3(t_j) + \mathbf{1}_{m\times m} \otimes L_4(t_j) \end{pmatrix}, \tag{2.74}$$

where

$$\begin{aligned} L_0(t_j) &= T_u(t_j) - mK_0(t_j)Z_u(t_j), \\ L_1(t_j) &= -K_0(t_j)Z_v(t_j), \\ L_2(t_j) &= -[K_1(t_j) + mK_2(t_j)]Z_u(t_j), \\ L_3(t_j) &= T_v(t_j) - K_1(t_j)Z_v(t_j), \\ L_4(t_j) &= -K_2(t_j)Z_v(t_j). \end{aligned} \tag{2.75}$$

The computation of the $(d_u + md_v) \times (d_u + md_v)$ matrix $L(t_j)$ is reduced to computing $L_0(t_j), L_1(t_j), L_2(t_j), L_3(t_j)$, and $L_4(t_j)$, which are of dimensions $d_u \times d_u, d_u \times d_v, d_v \times d_u, d_v \times d_v$, and $d_v \times d_v$, respectively.

In the first equation $\boldsymbol{r}(t_{j-1}) = Z^T(t_j)F^{-1}(t_j)\boldsymbol{w}(t_j) + L^T(t_j)\boldsymbol{r}(t_j)$ of (2.18),

$$Z^T(t_j)F^{-1}(t_j)\boldsymbol{w}(t_j) = \begin{pmatrix} Z_u^T(t_j)[F_1(t_j) + mF_2(t_j)] \sum_{i=1}^m \boldsymbol{w}_i(t_j) \\ \\ Z_v^T(t_j)[F_1(t_j)\boldsymbol{w}_1(t_j) + F_2(t_j) \sum_{i=1}^m \boldsymbol{w}_i(t_j)] \\ \\ \vdots \\ \\ Z_v^T(t_j)[F_1(t_j)\boldsymbol{w}_m(t_j) + F_2(t_j) \sum_{i=1}^m \boldsymbol{w}_i(t_j)] \end{pmatrix}, \tag{2.76}$$

and

$$L^T(t_j)\boldsymbol{r}(t_j) = \begin{pmatrix} L_0^T(t_j)\boldsymbol{r}_0(t_j) + L_2^T(t_j)\sum_{i=1}^m \boldsymbol{r}_i(t_j) \\ L_1^T(t_j)\boldsymbol{r}_0(t_j) + L_3^T(t_j)\boldsymbol{r}_1(t_j) + L_4^T(t_j)\sum_{i=1}^m \boldsymbol{r}_i(t_j) \\ \vdots \\ L_1^T(t_j)\boldsymbol{r}_0(t_j) + L_3^T(t_j)\boldsymbol{r}_m(t_j) + L_4^T(t_j)\sum_{i=1}^m \boldsymbol{r}_i(t_j) \end{pmatrix}, \tag{2.77}$$

where $F_1(t_j), F_2(t_j), K_0(t_j), K_1(t_j)$, and $K_2(t_j)$ are computed in the filtering step. In (2.76), the matrix-matrix and matrix vector multiplications of dimensions on the order qm is reduced to computing the $m+1$ components on the right hand side using the small blocks $F_1(t_j), F_2(t_j), K_0(t_j), K_1(t_j), K_2(t_j)$ and components of $\boldsymbol{w}(t_j)$.

In equation

$$\hat{\boldsymbol{\alpha}}(t_j) = \boldsymbol{a}(t_j) + P(t_j)\boldsymbol{r}_{t_{j-1}}, \tag{2.78}$$

we have

$$\hat{\boldsymbol{\alpha}}(t_j) = \begin{pmatrix} \boldsymbol{a}_0(t_j) \\ \boldsymbol{a}_1(t_j) \\ \vdots \\ \boldsymbol{a}_m(t_j) \end{pmatrix} + \begin{pmatrix} P_0(t_j)\boldsymbol{r}_0(t_{j-1}) + P_1(t_j)\sum_{i=1}^m \boldsymbol{r}_i(t_{j-1}) \\ P_1^T(t_j)\boldsymbol{r}_0(t_{j-1}) + P_2(t_j)\boldsymbol{r}_1(t_{j-1}) + P_3(t_j)\sum_{i=1}^m \boldsymbol{r}_i(t_{j-1}) \\ \vdots \\ P_1^T(t_j)\boldsymbol{r}_0(t_{j-1}) + P_2(t_j)\boldsymbol{r}_m(t_{j-1}) + P_3(t_j)\sum_{i=1}^m \boldsymbol{r}_i(t_{j-1}) \end{pmatrix}, \tag{2.79}$$

where $P_0(t_j), P_1(t_j), P_2(t_j), P_3(t_j)$ are computed in the filtering step.

In equation

$$N(t_{j-1}) = Z^T(t_j)F^{-1}(t_j)Z(t_j) + L^T(t_j)N(t_j)L(t_j), \tag{2.80}$$

similar to the proof of Theorem 1, $N(t_j)$ can be shown by induction to have the form

$$N(t_j) = \begin{pmatrix} N_0(t_j) & \mathbf{1}_m^T \otimes N_1(t_j) \\ \mathbf{1}_m \otimes N_1^T(t_j) & \boldsymbol{I}_m \otimes N_2(t_j) + \mathbf{1}_{m\times m} \otimes N_3(t_j) \end{pmatrix}, \tag{2.81}$$

where $N_0(t_n) = N_1(t_n) = N_2(t_n) = N_3(t_n)$ are initialized to zero matrices. In (2.80),

$$Z^T(t_j)F^{-1}(t_j)Z(t_j) = \begin{pmatrix} U_0(t_j) & \mathbf{1}_m^T \otimes U_1(t_j) \\ \mathbf{1}_m \otimes U_1^T(t_j) & I_m \otimes U_2(t_j) + \mathbf{1}_{m\times m} \otimes U_3(t_j) \end{pmatrix}, \tag{2.82}$$

where

$$\begin{aligned} U_0(t_j) &= mZ_u^T(t_j)[F_1(t_j) + mF_2(t_j)]Z_u(t_j), \\ U_1(t_j) &= Z_u^T(t_j)[F_1(t_j) + mF_2(t_j)]Z_v(t_j), \\ U_2(t_j) &= Z_v^T(t_j)F_1(t_j)Z_v(t_j), \\ U_3(t_j) &= Z_v^T(t_j)F_2(t_j)Z_v(t_j), \end{aligned} \tag{2.83}$$

and

$$L^T(t_j)N(t_j)L(t_j) = \begin{pmatrix} X_0(t_j) & \mathbf{1}_m^T \otimes X_1(t_j) \\ \mathbf{1}_m \otimes X_1^T & \boldsymbol{I}_m \otimes X_2(t_j) + \mathbf{1}_{m\times m} X_3(t_j) \end{pmatrix}, \tag{2.84}$$

where

$$\begin{aligned}
X_0(t_j) &= L_0^T(t_j)N_0(t_j)L_0(t_j) + mL_2^T(t_j)N_1^T(t_j)L_0(t_j)+\\
&\quad mL_0^T(t_j)N_1(t_j)L_2(t_j) + mL_2^T(t_j)[N_2(t_j)+mN_3(t_j)]L_2(t_j),\\
X_1(t_j) &= L_0^T(t_j)N_0(t_j)L_1(t_j) + mL_2^T(t_j)N_1^T(t_j)L_1(t_j) + L_0^T N_1(t_j)[L_3(t_j)+\\
&\quad mL_4(t_j)] + L_2^T[N_2(t_j)+mN_3(t_j)][L_3(t_j)+mL_4(t_j)],\\
X_2(t_j) &= L_3^T(t_j)N_2(t_j)L_3(t_j),\\
X_3(t_j) &= L_1^T(t_j)N_0(t_j)L_1(t_j) + [L_3^T(t_j)+mL_4^T(t_j)]N_1^T(t_j)L_1(t_j)+\\
&\quad L_1^T(t_j)N_1(t_j)[L_3(t_j)+mL_4(t_j)] + [L_3^T(t_j)+mL_4^T(t_j)]N_3(t_j)[L_3(t_j)+\\
&\quad mL_4(t_j)] + L_3^T N_2(t_j)L_4(t_j) + L_4^T(t_j)N_2(t_j)L_3(t_j) + mL_4^T N_2(t_j)L_4(t_j).
\end{aligned} \tag{2.85}$$

Adding up the two terms, $N(t_{j-1})$ is given by

$$N(t_{j-1}) = \begin{pmatrix} N_0(t_{j-1}) & \mathbf{1}_m^T \otimes N_1(t_{j-1}) \\ \mathbf{1}_m \otimes N_1^T(t_{j-1}) & \boldsymbol{I}_m \otimes N_2(t_{j-1}) + \mathbf{1}_{m\times m} \otimes N_3(t_{j-1}) \end{pmatrix}, \tag{2.86}$$

where

$$\begin{aligned}
N_0(t_{j-1}) &= U_0(t_j) + X_0(t_j),\\
N_1(t_{j-1}) &= U_1(t_j) + X_1(t_j),\\
N_2(t_{j-1}) &= U_2(t_j) + X_2(t_j),\\
N_3(t_{j-1}) &= U_3(t_j) + X_3(t_j).
\end{aligned} \tag{2.87}$$

The time complexity of (2.80) is thus reduced from $O(q^3m^3)$ to $O(q^3m)$. In equation $V(t_j) = P(t_j) - P(t_j)N(t_{j-1})P(t_j)$,

$$P(t_j)N(t_{j-1})P(t_j) = \begin{pmatrix} W_0(t_j) & \mathbf{1}_m^T \otimes W_1(t_j) \\ \mathbf{1}_m \otimes W_1^T(t_j) & \boldsymbol{I}_m \otimes W_2(t_j) + \mathbf{1}_{m\times m} W_3(t_j) \end{pmatrix}, \tag{2.88}$$

where

$$\begin{aligned}
W_0(t_j) &= P_0(t_j)N_0(t_{j-1})P_0(t_j) + mP_1(t_j)N_1^T(t_{j-1})P_0(t_j) + \\
&\quad mP_0(t_j)N_1(t_{j-1})P_1^T(t_j) + mP_1(t_j)[N_2(t_{j-1}) + mN_3(t_{j-1})]P_1^T(t_j), \\
W_1(t_j) &= P_0(t_j)N_0(t_{j-1})P_1(t_j) + mP_1(t_j)N_1^T(t_{j-1})P_1(t_j) + \\
&\quad P_0(t_j)N_1(t_{j-1})[P_2(t_j) + \\
&\quad mP_3(t_j)] + P_1(t_j)[N_2(t_{j-1}) + mN_3(t_{j-1})][P_2(t_j) + mP_3(t_j)], \\
W_2(t_j) &= P_2(t_j)N_2(t_{j-1})P_2(t_j), \\
W_3(t_j) &= P_1^T(t_j)N_0(t_{j-1})P_1(t_j) + [P_2(t_j) + mP_3(t_j)]N_1^T(t_{j-1})P_1(t_j) + \\
&\quad P_1^T N_1(t_{j-1})[P_2(t_j) + mP_3(t_j)] + \\
&\quad [P_2(t_j) + mP_3(t_j)]N_3(t_{j-1})[P_2(t_j) + mP_3(t_j)] + P_3N_2(t_{j-1})P_2(t_j) + \\
&\quad P_2(t_j)N_2(t_{j-1})P_3(t_j) + mP_3(t_j)N_2(t_{j-1})P_3(t_j).
\end{aligned} \tag{2.89}$$

So $V(t_j)$ is given by

$$V(t_j) = \begin{pmatrix} V_0(t_j) & \mathbf{1}_m^T \otimes V_1(t_j) \\ \mathbf{1}_m \otimes V_1^T(t_j) & I_m \otimes V_2(t_j) + \mathbf{1}_{m\times m} \otimes V_3(t_j) \end{pmatrix}, \tag{2.90}$$

where

$$\begin{aligned}
V_0(t_j) &= P_0(t_j) - W_0(t_j), \\
V_1(t_j) &= P_1(t_j) - W_1(t_j), \\
V_2(t_j) &= P_2(t_j) - W_2(t_j), \\
V_3(t_j) &= P_3(t_j) - W_3(t_j).
\end{aligned} \tag{2.91}$$

The new algorithm reduces the time complexity of the state smoothing recursion from $O(q^3m^3)$ to $O(q^3m)$. The space complexity of the original state smoothing is $O(q^2m^2n)$, where n comes from storing the filtering quantities. In contrast, the new algorithm for state smoothing is of space complexity $O((q^2 + qm)n)$.

2.3.4 Disturbance smoothing using the special structure

The computation of quantities in disturbance smoothing (2.19) and (2.20) can also be simplified using the special structures.

In the first equation in (2.19),

$$\boldsymbol{u}(t_j) = F^{-1}(t_j)\boldsymbol{w}(t_j) - K^T(t_j)\boldsymbol{r}(t_j), \tag{2.92}$$

the two terms are given by

$$F^{-1}(t_j)\boldsymbol{w}(t_j) = \begin{pmatrix} F_1(t_j)\boldsymbol{w}_1(t_j) + F_2(t_j)\sum_{i=1}^m \boldsymbol{w}_i(t_j) \\ \vdots \\ F_1(t_j)\boldsymbol{w}_m + F_2(t_j)\sum_{i=1}^m \boldsymbol{w}_i(t_j) \end{pmatrix}, \tag{2.93}$$

and

$$K'(t_j)\boldsymbol{r}(t_j) = \begin{pmatrix} K_0^T(t_j)\boldsymbol{r}_0(t_j) + K_1^T(t_j)\boldsymbol{r}_1(t_j) + K_2^T(t_j)\sum_{i=1}^m \boldsymbol{r}_i(t_j) \\ \vdots \\ K_0^T(t_j)\boldsymbol{r}_0(t_j) + K_1^T(t_j)\boldsymbol{r}_m(t_j) + K_2^T(t_j)\sum_{i=1}^m \boldsymbol{r}_i(t_j) \end{pmatrix}. \tag{2.94}$$

In $\hat{\boldsymbol{\epsilon}}(t_j) = H(t_j)\boldsymbol{u}(t_j)$, write $\boldsymbol{u}(t_j)$ as

$$\boldsymbol{u}(t_j) = \begin{pmatrix} \boldsymbol{u}_1(t_j) \\ \vdots \\ \boldsymbol{u}_m(t_j) \end{pmatrix}. \tag{2.95}$$

then $\hat{\boldsymbol{\epsilon}}(t_j)$ is given by

$$\hat{\boldsymbol{\epsilon}}(t_j) = H(t_j)\boldsymbol{u}(t_j) = \begin{pmatrix} \Sigma_\epsilon(t_j)\boldsymbol{u}_1(t_j) \\ \vdots \\ \Sigma_\epsilon(t_j)\boldsymbol{u}_m(t_j) \end{pmatrix}, \tag{2.96}$$

where $H(t_j) = \text{diag}\{\Sigma_\epsilon(t_j), \ldots, \Sigma_\epsilon(t_j)\}$. $\hat{\boldsymbol{\eta}}(t_j)$ is given by

$$\begin{aligned} &\hat{\boldsymbol{\eta}}(t_j) \\ = \quad & Q(t_j)R^T(t_j)\boldsymbol{r}(t_j) \\ = \quad & \begin{pmatrix} \Sigma_u(t_j) & 0 & & 0 \\ 0 & \Sigma_v(t_j) & & \\ & & \ddots & \\ 0 & & & \Sigma_v(t_j) \end{pmatrix} \begin{pmatrix} R_u^T(t_j) & 0 & & 0 \\ 0 & R_v^T(t_j) & & \\ & & \ddots & \\ 0 & & & R_v^T(t_j) \end{pmatrix} \begin{pmatrix} \boldsymbol{r}_0(t_j) \\ \boldsymbol{r}_1(t_j) \\ \vdots \\ \boldsymbol{r}_m(t_j) \end{pmatrix} \\ = \quad & \begin{pmatrix} \Sigma_u(t_j)R_u^T(t_j)\boldsymbol{r}_0(t_j) \\ \Sigma_v(t_j)R_v^T(t_j)\boldsymbol{r}_1(t_j) \\ \vdots \\ \Sigma_v(t_j)R_v^T(t_j)\boldsymbol{r}_m(t_j) \end{pmatrix}. \end{aligned} \tag{2.97}$$

$$\boldsymbol{r}(t_{j-1}) = Z^T(t_j)\boldsymbol{u}_t + T^T(t_j)\boldsymbol{r}(t_j) = \begin{pmatrix} Z_u^T(t_j)\sum_{i=1}^m \boldsymbol{u}_i(t_j) \\ Z_v^T(t_j)\boldsymbol{u}_1(t_j) \\ \vdots \\ Z_v^T(t_j)\boldsymbol{u}_m(t_j) \end{pmatrix} + \begin{pmatrix} T_u^T(t_j)\boldsymbol{r}_0(t_j) \\ T_v^T(t_j)\boldsymbol{r}_1(t_j) \\ \vdots \\ T_v^T(t_j)\boldsymbol{r}_m(t_j) \end{pmatrix}. \tag{2.98}$$

For the disturbance smoothing recursion of the variance matrices (2.20), in $D(t_j) = F^{-1}(t_j) + K^T(t_j)N(t_j)K(t_j)$, $K(t_j)$ is given in (2.72) and $N(t_j)$ is given in (2.86), thus we have

$$K^T(t_j)N(t_j)K(t_j) = I_m \otimes E_1(t_j) + \mathbf{1}_{m\times m} \otimes E_2(t_j), \tag{2.99}$$

where

$$\begin{aligned} E_1(t_j) &= K_1^T(t_j)N_2(t_j)K_1(t_j), \\ E_2(t_j) &= K_0^T(t_j)N_0(t_j)K_0(t_j) + [K_1^T(t_j) + mK_2^T(t_j)]N_1^T(t_j)K_0(t_j) + \\ &\quad K_0^T(t_j)N_1(t_j)[K_1(t_j) + mK_2(t_j)] + [K_1^T(t_j) + \\ &\quad mK_2^T(t_j)]N_3(t_j)[K_1(t_j) + mK_2(t_j)] + \\ &\quad K_1^T(t_j)N_2(t_j)K_2(t_j) + K_2^T(t_j)N_2(t_j)K_1(t_j) + mK_2^T(t_j)N_2(t_j)K_2(t_j). \end{aligned} \tag{2.100}$$

Therefore,

$$D(t_j) = I_m \otimes D_1(t_j) + \mathbf{1}_{m\times m} \otimes D_2(t_j), \tag{2.101}$$

where

$$\begin{aligned} D_1(t_j) &= F_1(t_j) + E_1(t_j), \\ D_2(t_j) &= F_2(t_j) + E_2(t_j). \end{aligned} \tag{2.102}$$

We have

$$\mathrm{Var}[\boldsymbol{\epsilon}(t_j)|\boldsymbol{Y}(t_n)] = H(t_j) - H(t_j)D(t_j)H(t_j) = I_m \otimes G_1(t_j) + \mathbf{1}_{m\times m} \otimes G_2(t_j), \tag{2.103}$$

where

$$\begin{aligned} G_1(t_j) &= \Sigma_\epsilon(t_j) - \Sigma_\epsilon(t_j)D_1(t_j)\Sigma_\epsilon(t_j), \\ G_2(t_j) &= -\Sigma_\epsilon(t_j)D_2(t_j)\Sigma_\epsilon(t_j), \end{aligned} \tag{2.104}$$

and

$$\begin{aligned}\mathrm{Var}[\boldsymbol{\eta}(t_j)|\boldsymbol{Y}(t_n)] &= Q(t_j) - Q(t_j)R^T(t_j)N(t_j)R(t_j)Q(t_j)\\ &= \begin{pmatrix} J_0(t_j) & \mathbf{1}_m^T \otimes J_1(t_j) \\ \mathbf{1}_m \otimes J_1^T(t_j) & I_m \otimes J_2(t_j) + \mathbf{1}_{m\times m} \otimes J_3(t_j) \end{pmatrix},\end{aligned} \tag{2.105}$$

where

$$\begin{aligned} J_0(t_j) &= \Sigma_u(t_j) - \Sigma_u(t_j)R_u^T(t_j)N_0(t_j)R_u(t_j)\Sigma_u(t_j),\\ J_1(t_j) &= -\Sigma_u(t_j)R_u^T(t_j)N_1(t_j)R_v(t_j)\Sigma_v(t_j),\\ J_2(t_j) &= \Sigma_v(t_j) - \Sigma_v(t_j)R_v^T(t_j)N_2(t_j)R_v(t_j)\Sigma_v(t_j),\\ J_3(t_j) &= -\Sigma_v(t_j)R_v^T(t_j)N_3(t_j)R_v(t_j)\Sigma_v(t_j). \end{aligned} \tag{2.106}$$

2.3.5 Fast state smoothing using the special structure

In (2.21),

$$\hat{\boldsymbol{\alpha}}(t_{j+1}) = T(t_j)\hat{\boldsymbol{\alpha}}(t_j) + \hat{\boldsymbol{\eta}}(t_j). \tag{2.107}$$

Write $\hat{\boldsymbol{\alpha}}(t_j)$ as

$$\hat{\boldsymbol{\alpha}}(t_j) = \begin{pmatrix} \hat{\boldsymbol{\alpha}}_0(t_j) \\ \hat{\boldsymbol{\alpha}}_1(t_j) \\ \vdots \\ \hat{\boldsymbol{\alpha}}_m(t_j) \end{pmatrix}, \tag{2.108}$$

then

$$T(t_j)\hat{\boldsymbol{\alpha}}(t_j) = \begin{pmatrix} T_u(t_j)\hat{\boldsymbol{\alpha}}_0(t_j) \\ T_v(t_j)\hat{\boldsymbol{\alpha}}_1(t_j) \\ \vdots \\ T_v(t_j)\hat{\boldsymbol{\alpha}}_m(t_j) \end{pmatrix}. \tag{2.109}$$

For the initial value $\hat{\boldsymbol{\alpha}}(t_1) = \boldsymbol{a}(t_1) + P(t_1)\boldsymbol{r}(t_0)$, we have

$$
\begin{aligned}
P(t_1)\boldsymbol{r}(t_0) &= \begin{pmatrix} P_0(t_1) & \mathbf{1}_m^T \otimes P_1(t_1) \\ \mathbf{1}_m \otimes P_1^T(t_1) & \boldsymbol{I}_m \otimes P_2(t_1) + \mathbf{1}_{m\times m} \otimes P_3(t_1) \end{pmatrix} \begin{pmatrix} \boldsymbol{r}_0(t_0) \\ \boldsymbol{r}_1(t_0) \\ \vdots \\ \boldsymbol{r}_m(t_0) \end{pmatrix} \\
&= \begin{pmatrix} P_0(t_1)\boldsymbol{r}_0(t_0) + P_1(t_1)\sum_{i=1}^m \boldsymbol{r}_i(t_0) \\ P_1^T(t_1)\boldsymbol{r}_0(t_0) + P_2(t_1)\boldsymbol{r}_1(t_0) + P_3(t_1)\sum_{i=1}^m \boldsymbol{r}_i(t_0) \\ \vdots \\ P_1^T(t_1)\boldsymbol{r}_0(t_0) + P_2(t_1)\boldsymbol{r}_m(t_0) + P_3(t_1)\sum_{i=1}^m \boldsymbol{r}_i(t_0) \end{pmatrix},
\end{aligned}
\tag{2.110}
$$

where $P_0(t_1), P_1(t_1), P_2(t_1)$, and $P_3(t_1)$ are the components of $P(t_1)$, obtained from the filtering step, and $\boldsymbol{r}(t_0)$ is obtained from disturbance smoothing (2.19).

The time complexity and space complexity of the original Kalman filter (KF) / univariate treatment and the new algorithm are summarized in Table 2.1.

Algorithm	Filtering		Smoothing	
	Time	Space	Time	Space
KF/Univariate treatment	$O(q^3m^3n)$	$O(q^2m^2)$	$O(q^3m^3n)$	$O(q^2m^2n)$
New Algorithm	$O(q^3mn)$	$O(q^2+qm)$	$O(q^3mn)$	$O((q^2+qm)n)$

Table 2.1: Time and space complexity of the univariate treatment and the new algorithm.

2.4 Dealing with Missing Values

The original Kalman filter and the univariate treatment can handle any kind of missing values. Our new algorithm, as far as we know for now, can only cope with one kind of missing values: subject dropouts, but different subjects can drop out at different time points. That is, the new algorithm cannot deal with the case where there are intermittent missing values, because the special structure would be disrupted. In practice intermittent missing values may be imputed.

2.4.1 Filtering algorithm with subject dropouts

In the filtering part of the new algorithm, at time t_j, in the observation vector $\boldsymbol{y}(t_j) = (\boldsymbol{y}_1^T(t_j), \ldots, \boldsymbol{y}_m^T(t_j))^T$, one simply replaces missing observations with NA. That is, if $\boldsymbol{y}_i^T(t_j)$ is missing for some $1 \leq i \leq m$, replace it with a vector of NA's of the same length. Note that we cannot deal with the case where only a part of a subject's variables are missing, the entire vector $\boldsymbol{y}_i(t_j)$ is either missing or non-missing. According to the first equation in the Kalman filter (2.15), the components of $\boldsymbol{w}(t_j) = (\boldsymbol{w}_1^T(t_j), \ldots, \boldsymbol{w}_m^T(t_j))^T$ that correspond to missing subjects will be NA. All matrices and vectors in the Kalman filter recursion still have the same special structures, except that their evolutions will only be contributed by non-missing subjects. Let

$$A_j = \{i : \text{subject } i\text{'s longitudinal variables are observed at time } t_j, 1 \leq i \leq m\}$$

be the index set of subjects who are observed at time t_j. Denote by $|A_j|$ the number of elements in A_j. Then the quantities that we keep track of, i.e., the small matrices, are computed by replacing m with $m(t_j) = |A_j|$, the number of non-missing subjects at time t_j. For example, in (2.57), m will be replaced by $m(t_j)$, and in (2.58), $\sum_{i=1}^m \boldsymbol{w}_i(t_j)$ will be replaced by $\sum_{i \in A_j} \boldsymbol{w}_i(t_j)$.

It should be stressed that, in implementation, missing subjects cannot simply be removed from the system, they must be kept in their positions as NA's—it is important to retain the ordering of all subjects, no matter whether they are missing or not, otherwise we will lose track of them. Hence we keep the NA's in $\boldsymbol{w}(t_j)$, $\boldsymbol{a}(t_j|t_j)$, and $\boldsymbol{a}(t_{j+1})$, instead of reducing the dimension of the vectors by removing missing values.

2.4.2 Smoothing algorithm with missing values

For smoothing, currently, our new algorithm can only deal with the case where there is no subject dropout, but the subjects can come into the system at different time points. That is, once a subject comes into the system, it must exist to the end.

2.5 Initialization

According to the connection between a linear Gaussian state space model and a general mixed effects model, elaborated in Section (1.3.4), in the initial state vector

$$\boldsymbol{\alpha}(t_1) = \begin{pmatrix} \boldsymbol{\alpha}_0(t_1) \\ \boldsymbol{\alpha}_1(t_1) \\ \vdots \\ \boldsymbol{\alpha}_m(t_1) \end{pmatrix}, \tag{2.111}$$

the fixed effects $\boldsymbol{\alpha}_0(t_1)$ correspond to $\boldsymbol{\alpha}_{11}$ in (1.40) and has a diffuse prior, and the random effects $\boldsymbol{\alpha}_i(t_1)$, $i = 1, \dots, m$ correspond to $\boldsymbol{\alpha}_{12}$ and are independent identically normally distributed with mean zero and a finite covariance matrix.

To cope with the diffuse and proper priors for the components of the initial state, as described in Section 1.3.1, write the initial state vector as

$$\boldsymbol{\alpha}(t_1) = \boldsymbol{a} + A\boldsymbol{\delta} + R_0\boldsymbol{\eta}_0, \quad \boldsymbol{\eta}_0 \sim N(\mathbf{0}, Q_0), \tag{2.112}$$

in which, for our mixed effects state space model, $\boldsymbol{a} = \mathbf{0}$, $\boldsymbol{\delta}$ is a $d_u \times 1$ random vector with a diffuse prior $\boldsymbol{\delta} \sim N(\mathbf{0}, \kappa_0 I)$ where $\kappa_0 \to \infty$, $\boldsymbol{\eta}_0$ is a $md_v \times 1$ random vector whose covariance matrix Q_0 we assume to be $\kappa_1 I$, where κ_1 is an unknown finite parameter to be estimated, and

$$A = \begin{pmatrix} I_{d_u} \\ \mathbf{0}_{md_v \times d_u} \end{pmatrix}_{(d_u+md_v)\times d_u} \quad \text{and } R_0 = \begin{pmatrix} \mathbf{0}_{d_u \times md_v} \\ I_{md_v} \end{pmatrix} \tag{2.113}$$

are selection matrices.

The diffuse prior for $\boldsymbol{\delta}$ can be implemented by either setting κ_0 to an arbitrarily large number, or using the exact initial Kalman filter [26]; the former often leads to large rounding errors and the latter is algebraically complicated. As stated in Section 1.3.1, an equivalent algorithm is to treat $\boldsymbol{\delta}$ as a fixed unknown parameter vector and estimate it using the augmented Kalman filter ([27], [26], [28]). The parameters $\boldsymbol{\delta}$ can be concentrated out of the likelihood and estimated independently and analytically, given the other parameters. Usually, the exact initial Kalman filter is more efficient than the augmented Kalman filter. However, in our case, using the special structure, the augmented Kalman filter does not add much extra computation and is algebraically simpler. Details for this approach are provided below.

By (2.112), the distribution of the initial state vector $\boldsymbol{\alpha}(t_1)$ is normal with mean $\mathrm{E}[\boldsymbol{\alpha}(t_1)] = \boldsymbol{a}(t_1) = \boldsymbol{a} + A\boldsymbol{\delta}$ and covariance matrix $\mathrm{Var}[\boldsymbol{\alpha}(t_1)] = P(t_1) = R_0 Q_0 R_0^T$. Since filtering operations are linear, we have

$$\boldsymbol{a}(t_j) = \boldsymbol{a}_a(t_j) + A_A(t_j)\boldsymbol{\delta} \tag{2.114}$$

for all t_j, $j = 1, \ldots, n$. $\boldsymbol{a}_a(t_j)$ is the $\boldsymbol{a}(t_j)$ obtained from a Kalman filter with $\boldsymbol{a}$ as $\boldsymbol{a}(t_1)$ and $\boldsymbol{y}(t_j)$ as the observation vector. $A_A(t_j)$ is a $(d_u + md_v) \times d_u$ matrix. The kth column of $A_A(t_j)$ is the $\boldsymbol{a}(t_j)$ obtained from a Kalman filter with the kth column of A as the initial state mean $\boldsymbol{a}(t_1)$ and $\mathbf{0}$ as the observation vector $\boldsymbol{y}(t_j)$, $j = 1, \ldots, n$. Similarly, we have

$$\boldsymbol{a}(t_j|t_j) = \boldsymbol{a}_a(t_j|t_j) + A_A(t_j|t_j)\boldsymbol{\delta}, \tag{2.115}$$

and

$$\boldsymbol{w}(t_j) = \boldsymbol{w}_a(t_j) + W_A(t_j)\boldsymbol{\delta}, \tag{2.116}$$

for $j = 1, \ldots, n$. The log-likelihood of $\boldsymbol{\delta}$ is given by

$$l(\boldsymbol{\delta}) = \log[p(\boldsymbol{Y}_n|\boldsymbol{\delta})] = \sum_{j=1}^{n} p(\boldsymbol{w}(t_j)) = -\boldsymbol{b}^T(t_n)\boldsymbol{\delta} - \frac{1}{2}\boldsymbol{\delta}^T S_A(t_n)\boldsymbol{\delta} + \text{constant}, \tag{2.117}$$

where $\boldsymbol{b}(t_n) = \sum_{j=1}^{n} W_A^T(t_j)F^{-1}(t_j)\boldsymbol{w}_a(t_j)$, $S_A(t_n) = \sum_{j=1}^{n} W_A^T(t_j)F^{-1}(t_j)W_A(t_j)$, and constant means it is independent of $\boldsymbol{\delta}$.

Given other parameters, $\boldsymbol{\delta}$ is estimated by minimizing

$$\boldsymbol{b}^T(t_j)\boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^T S_A(t_j)\boldsymbol{\delta}, \tag{2.118}$$

for which there is an analytical solution, given by

$$\hat{\boldsymbol{\delta}} = -S_A^{-1}(t_n)\boldsymbol{b}(t_n). \tag{2.119}$$

The $1 + d_u$ Kalman filter recursions used to obtain $(\boldsymbol{w}_a(t_j), W_A(t_j)), (\boldsymbol{a}_a(t_j|t_j), A_A(t_j|t_j))$, and $(\boldsymbol{a}_a(t_{j+1}), A_A(t_{j+1}))$ share common covariance matrices $P(t_j), P(t_j|t_j)$, and $F(t_j)$, initialized with $P(t_1) = R_0 Q_0 R_0^T$. Therefore, one can apply an augmented Kalman filter to an observation matrix $(\boldsymbol{y}(t_j), \boldsymbol{0}_{qm \times d_u})$, with initial state mean matrix $(\boldsymbol{a}_a(t_1), A_A(t_1)) =$

$(\boldsymbol{a}, A)$. Updates for the augmented matrices are given by

$$\begin{aligned}
(\boldsymbol{w}_a(t_j), W_A(t_j)) &= (\boldsymbol{y}(t_j), \mathbf{0}_{qm\times d_u}) - Z(t_j)(\boldsymbol{a}_a(t_j), A_A(t_j)) \\
(\boldsymbol{a}_a(t_j|t_j), A_A(t_j|t_j)) &= (\boldsymbol{a}_a(t_j), A_A(t_j)) + P(t_j)Z^T(t_j)F^{-1}(t_j)(\boldsymbol{w}_a(t_j), W_A(t_j)) \\
(\boldsymbol{a}_a(t_{j+1}), A_A(t_{j+1})) &= T_t(t_j)(\boldsymbol{a}_a(t_j|t_j), A_A(t_j|t_j)).
\end{aligned} \tag{2.120}$$

The updates for the covariance matrices remain the same as in section 2.3.1. There are special structures in (2.120), so the fast algorithm also applies. The updates for $\boldsymbol{w}_a(t_j), \boldsymbol{a}_a(t_j|t_j)$, and $\boldsymbol{a}_a(t_{j+1})$ are the same as the algorithm described in Section (2.3.1). The calculations for $W_A(t_j), A_A(t_j|t_j)$, and $A_A(t_{j+1})$ can be simplified as follows. $W_A(t_j)$ can be partitioned into m parts corresponding to the m subjects, and $A_A(t_j|t_j)$ and $A_A(t_j)$ can be partitioned into $m+1$ parts, corresponding to population effects and m subject random deviations, written as

$$W_A(t_j) = \begin{pmatrix} W_{A1}(t_j) \\ \vdots \\ W_{Am}(t_j) \end{pmatrix}, \quad A_A(t_j|t_j) = \begin{pmatrix} A_{A0}(t_j|t_j) \\ A_{A1}(t_j|t_j) \\ \vdots \\ A_{Am}(t_j|t_j) \end{pmatrix}, \quad A_A(t_j) = \begin{pmatrix} A_{A0}(t_j) \\ A_{A1}(t_j) \\ \vdots \\ A_{Am}(t_j) \end{pmatrix}.$$

The calculation of $W_A(t_j)$ in (2.120) can be simplified as

$$W_A(t_j) = \begin{pmatrix} W_{A1}(t_j) \\ \vdots \\ W_{Am}(t_j) \end{pmatrix} = -\begin{pmatrix} Z_u(t_j)A_{A0}(t_j) + Z_v(t_j)A_{A1}(t_j) \\ \vdots \\ Z_u(t_j)A_{A0}(t_j) + Z_v(t_j)A_{Am}(t_j) \end{pmatrix}.$$

In the second equation of (2.120), $P(t_j)Z^T(t_j)F^{-1}(t_j)$ has a special structure given in

(2.56), so $A_A(t_j|t_j)$ is calculated as

$$A_A(t_j|t_j) = \begin{pmatrix} A_{A0}(t_j) \\ A_{A1}(t_j) \\ \vdots \\ A_{Am}(t_j) \end{pmatrix} + \begin{pmatrix} C_0(t_j)\sum_{i=1}^m W_{Ai}(t_j) \\ C_1(t_j)W_{A1}(t_j) + C_2(t_j)\sum_{i=1}^m W_{Ai}(t_j) \\ \vdots \\ C_1(t_j)W_{Am}(t_j) + C_2(t_j)\sum_{i=1}^m W_{Ai}(t_j) \end{pmatrix}.$$

And $A_A(t_{j+1})$ is given by

$$A_A(t_{j+1}) = \begin{pmatrix} T_u(t_j)A_{A0}(t_j|t_j) \\ T_v(t_j)A_{A1}(t_j|t_j) \\ \vdots \\ T_v(t_j)A_{Am}(t_j|t_j) \end{pmatrix}.$$

Applying the special structure of $F^{-1}(t_j)$ in (2.54), we have

$$S_A(t_n) = \sum_{t=1}^n \left\{ \sum_{i=1}^m \left[W_{Ai}^T(t_j)F_1(t_j)W_{Ai}(t_j) \right] + \left[\sum_{i=1}^m W_{Ai}^T(t_j) \right] F_2(t_j) \left[\sum_{i=1}^m W_{Ai}(t_j) \right] \right\},$$

and

$$\boldsymbol{b}(t_n) = \sum_{t=1}^n \left\{ \sum_{i=1}^m \left[W_{Ai}^T(t_j)F_1(t_j)\boldsymbol{w}_{ai}(t_j) \right] + \left[\sum_{i=1}^m W_{Ai}^T(t_j) \right] F_2(t_j) \left[\sum_{i=1}^m \boldsymbol{w}_{ai}(t_j) \right] \right\},$$

where $\boldsymbol{w}_{ai}(t_j)$, $i = 1, \ldots, m$, are the components of

$$\boldsymbol{w}_a(t_j) = \begin{pmatrix} \boldsymbol{w}_{a1}(t_j) \\ \vdots \\ \boldsymbol{w}_{am}(t_j) \end{pmatrix}.$$

2.6 Adding the Regression Term

In the mixed effects state space model (2.13), the regression coefficient vector $\boldsymbol{\beta}$ can be concentrated out and estimated separately, given other parameters; this treatment considerably simplifies the numerical optimization procedure of maximum likelihood estimation.

In the observation equation in (2.13), moving the regression term to the left hand side, we have

$$\boldsymbol{y}(t_j) - X(t_j)\boldsymbol{\beta} = Z(t_j)\boldsymbol{\alpha}(t_j) + \boldsymbol{\epsilon}(t_j), \tag{2.121}$$

in which, for a fixed $\boldsymbol{\beta}$, one can treat $\boldsymbol{y}(t_j) - X(t_j)\boldsymbol{\beta}$ as the observation vector. Given the other parameters in the model, the log-likelihood of $\boldsymbol{\beta}$ is

$$l(\boldsymbol{\beta}) = -\frac{1}{2}\sum_{j=1}^{n} \boldsymbol{w}^T(t_j)F^{-1}(t_j)\boldsymbol{w}(t_j) + \text{constant}, \tag{2.122}$$

where $\boldsymbol{w}(t_j)$ is the one-step-ahead prediction error obtained from a Kalman filter with $\boldsymbol{y}(t_j) - X(t_j)\boldsymbol{\beta}$ as the observation vector, $F^{-1}(t_j)$ is the same as that obtained from the Kalman filter for a model without the regression term, and "constant" means independent of $\boldsymbol{\beta}$. Since the filtering operations are linear, we have

$$\boldsymbol{w}(t_j) = \boldsymbol{w}^*(t_j) - W^*(t_j)\boldsymbol{\beta}, \tag{2.123}$$

with $\boldsymbol{w}^*(t_j)$ a $qm \times 1$ vector and $W^*(t_j)$ a $qm \times pq$ matrix, where $\boldsymbol{w}^*(t_j)$ is the one-step-ahead prediction error $\boldsymbol{w}(t_j)$ obtained from a Kalman filter with $\boldsymbol{y}(t_j)$ as the observation vector, and the kth column of $W^*(t_j)$ is the $\boldsymbol{w}(t_j)$ obtained from a Kalman filter with the kth column of $X(t_j)$ as the observation vector, $j = 1, \ldots, pq$. The $pq+1$ Kalman filtering recursions share common covariance matrices $F(t_j), P(t_j|t_j)$, and $P(t_j)$, the difference is in $\boldsymbol{w}(t_j), \boldsymbol{a}(t_j)$, and $\boldsymbol{a}(t_j|t_j)$. Therefore, one can apply Kalman filter to the observation matrix $(\boldsymbol{y}(t_j), X(t_j))$ instead of an observation vector, and use the same recursion for covariance matrices as in Section 2.3.1. This is the augmented Kalman filter discussed in Section 2.5. The parameter vector $\boldsymbol{\beta}$ is estimated by minimizing

$$\sum_{j=1}^{n} \boldsymbol{w}^T(t_j)F^{-1}(t_j)\boldsymbol{w}(t_j) = [\boldsymbol{w}^*(t_j) - W^*(t_j)\boldsymbol{\beta}]^T F^{-1}(t_j)[\boldsymbol{w}^*(t_j) - W^*(t_j)\boldsymbol{\beta}], \quad (2.124)$$

which has an analytical solution

$$\hat{\boldsymbol{\beta}} = [\sum_{j=1}^{n} W^{*T}(t_j)F^{-1}(t_j)W^*(t_j)]^{-1} \sum_{j=1}^{n} W^{*T}(t_j)F^{-1}(t_j)\boldsymbol{w}^*(t_j). \quad (2.125)$$

The new efficient algorithm also applies for the estimation procedure of $\boldsymbol{\beta}$, done in a similar manner as in section 2.5. The augmented Kalman filter has recursions for covariance matrices identical to that in Section 2.3.1, the difference lies in the updating equations for $\boldsymbol{w}(t_j), \boldsymbol{a}(t_j)$, and $\boldsymbol{a}(t_j|t_j)$, given by

$$\begin{aligned}
(\boldsymbol{w}^*(t_j), W^*(t_j)) &= (\boldsymbol{y}(t_j), X(t_j)) - Z(t_j)(\boldsymbol{a}^*(t_j), A^*(t_j)),\\
(\boldsymbol{a}^*(t_j|t_j), A^*(t_j|t_j)) &= (\boldsymbol{a}^*(t_j), A^*(t_j)) + P(t_j)Z^T(t_j)F^{-1}(t_j)(\boldsymbol{w}^*(t_j), W^*(t_j)),\\
(\boldsymbol{a}^*(t_{j+1}), A^*(t_{j+1})) &= T_t(\boldsymbol{a}^*(t_j|t_j), A^*(t_j|t_j)),
\end{aligned} \quad (2.126)$$

where $A^*(t_j)$ is a matrix, of which the kth column is the $\boldsymbol{a}(t_j)$ obtained from a Kalman filter with the kth column of $X(t_j)$ as the observation vector, and the kth column of $A^*(t_j|t_j)$ the corresponding $\boldsymbol{a}^*(t_j|t_j)$, $k = 1, \ldots, pq$.

The updates for $\boldsymbol{w}^*(t_j), \boldsymbol{a}^*(t_j|t_j)$, and $\boldsymbol{a}^*(t_{j+1})$ are the same as the algorithm described in Section (2.3.1). The calculations for $W^*(t_j), A^*(t_j|t_j)$, and $A^*(t_j)$ can be simplified as follows. $W^*(t_j)$ can be partitioned into m parts corresponding to the m subjects, and $A^*(t_j|t_j)$ and $A^*(t_{j+1})$ can be partitioned into $m+1$ parts, corresponding to a population level fixed effect and m subject random deviations, written as

$$W^*(t_j) = \begin{pmatrix} W_1^*(t_j) \\ \vdots \\ W_m^*(t_j) \end{pmatrix}, \quad A^*(t_j|t_j) = \begin{pmatrix} A_0^*(t_j|t_j) \\ A_1^*(t_j|t_j) \\ \vdots \\ A_m^*(t_j|t_j) \end{pmatrix}, \quad A^*(t_j) = \begin{pmatrix} A_0^*(t_j) \\ A_1^*(t_j) \\ \vdots \\ A_m^*(t_j) \end{pmatrix}. \tag{2.127}$$

The calculation of $W^*(t_j)$ in (2.126) can be simplified as

$$W^*(t_j) = \begin{pmatrix} W_1^*(t_j) \\ \vdots \\ W_m^*(t_j) \end{pmatrix} = \begin{pmatrix} X_1(t_j) \\ \vdots \\ X_m(t_j) \end{pmatrix} - \begin{pmatrix} Z_u(t_j)A_0^*(t_j) + Z_v(t_j)A_1^*(t_j) \\ \vdots \\ Z_u(t_j)A_0^*(t_j) + Z_v(t_j)A_m^*(t_j) \end{pmatrix} \tag{2.128}$$

In the second equation of (2.126), $P(t_j)Z^T(t_j)F^{-1}(t_j)$ has a special structure given in (2.56), so $A^*(t_j|t_j)$ is calculated as

$$A^*(t_j|t_j) = \begin{pmatrix} A_0^*(t_j) \\ A_1^*(t_j) \\ \vdots \\ A_m^*(t_j) \end{pmatrix} + \begin{pmatrix} C_0(t_j)\sum_{i=1}^m W_i^*(t_j) \\ C_1(t_j)W_1^*(t_j) + C_2(t_j)\sum_{i=1}^m W_i^*(t_j) \\ \vdots \\ C_1(t_j)W_m^*(t_j) + C_2(t_j)\sum_{i=1}^m W_i^*(t_j) \end{pmatrix}. \tag{2.129}$$

And $A^*(t_{j+1})$ is given by

$$A^*(t_{j+1}) = \begin{pmatrix} T_u(t_j)A_0^*(t_j|t_j) \\ T_v(t_j)A_1^*(t_j|t_j) \\ \vdots \\ T_v(t_j)A_m^*(t_j|t_j) \end{pmatrix}. \tag{2.130}$$

Applying the special structure of $F^{-1}(t_j)$ in (2.54), we have

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= \left(\textstyle\sum_{t=1}^{n}\left\{\sum_{i=1}^{m}\left[W_i^{*T}(t_j)F_1(t_j)W_i^*(t_j)\right]+\left[\sum_{i=1}^{m}W_i^{*T}(t_j)\right]F_2(t_j)\left[\sum_{i=1}^{m}W_i^*(t_j)\right]\right\}\right)^{-1}\\ &\quad \textstyle\sum_{t=1}^{n}\left\{\sum_{i=1}^{m}\left[W_i^{*T}(t_j)F_1(t_j)\boldsymbol{w}_i^*(t_j)\right]+\left[\sum_{i=1}^{m}W_i^{*T}(t_j)\right]F_2(t_j)\left[\sum_{i=1}^{m}\boldsymbol{w}_i^*(t_j)\right]\right\},\end{aligned} \tag{2.131}$$

where $\boldsymbol{w}_i^*(t_j)$, $i = 1, \ldots, m$, are the components of

$$\boldsymbol{w}^*(t_j) = \begin{pmatrix} \boldsymbol{w}_1^*(t_j) \\ \vdots \\ \boldsymbol{w}_m^*(t_j) \end{pmatrix}. \tag{2.132}$$

Chapter 3

Joint Modeling of Longitudinal and Time-to-Event Data

3.1 The Joint Model

For the modeling of longitudinal data, choices include linear mixed effects models, semi-parametric/non-parametric mixed effects models via splines or functional models, and state space models. Among these models, the state space model is capable of dynamically modeling the interactions between multiple longitudinal variables, and simultaneously enjoys interpretability and flexibility. To the best of our knowledge, the state space model has not been used to model longitudinal variables in joint models. Semi-parametric/non-parametric mixed effects models have rarely been used to model longitudinal variables in joint models, possibly because of the extra burden to the already high computational cost. The state space model with random effects also requires expensive computation. With the advancement of new technologies, an increasing amount of regularly measured data have emerged; and there is an urge for developing new efficient algorithms to extract valuable information from these large data sets. For example, wearable device data have been widely used in studies, where the data are measured regularly and densely. Activity trackers and heart rate monitors are two such examples. Another example is our dialysis data set, which contains 354,572 end stage renal disease patients who receive dialysis treatment regularly. Their clinical variables such as hemoglobin, albumin, and blood pressure are measured at each treatment or monthly. Interests are to find out (i) how longitudinal variables change dynamically over time and relate to other longitudinal variables and covariates, and (ii) the relationship between survival status and longitudinal variables/covariates.

With the new fast and efficient algorithm under our belt, a joint modeling framework using the state space model is developed in this chapter. The state space model for longitudinal variables and the submodel for time to event will share common latent state processes, thus naturally introduce correlation between longitudinal variables and time to event. For the survival submodel, a logistic regression model is assumed which does not require the restrictive proportional hazard assumption in a Cox regression model. The joint model aims to (i) model the evolution of each longitudinal variable and its association with covariates and other longitudinal variables, and (ii) perform online predictions for longitudinal variables and the event probability of each subject.

We assume that all subjects share common observation time points $t_j, j = 1, \ldots, n$. Let $\{(\boldsymbol{x}_i(t_j), \boldsymbol{y}_i(t_j), z_i(t_j), s_i(t_j)) : i = 1, \ldots, m; j = 1, \ldots, n_i\}$ be the observations for subject i at time t_j, where $\boldsymbol{x}_i(t_j)$ is a $p \times 1$ vector of covariates, $\boldsymbol{y}_i(t_j)$ is a $q \times 1$ vector of longitudinal variables, $z_i(t_j)$ is an event indicator, taking value 1 if the event happens during time interval $(t_{j-1}, t_j]$ and 0 otherwise, and $s_i(t_j)$ is a censoring indicator, taking value 1 if the subject is censored during time interval $(t_{j-1}, t_j]$ and 0 otherwise. We assume that the event is terminal such as death. There are no observations for subject i after t_j if $z_i(t_j) = 1$. Note that subjects can be censored. In this case, we have $s_i(t_j) = 1$, and no observations for subject i thereafter.

The longitudinal submodel is given by (2.1), with the vector form (2.13). In model (2.1), each longitudinal observation is decomposed into three parts: the influence of covariates, a latent process, and measurement error and/or biological variation. Correlation/interactions between longitudinal variables can be modeled by one or a combination of $\Sigma_\epsilon(t_j)$, $T_u(t_j)$, and $T_v(t_j)$. Population effects $\boldsymbol{u}(t_j)$ and individual random effects $\boldsymbol{v}_i(t_j)$ can be any process in the form of a state space model.

For each subject i, let t_{n_i} be the last time point its survival or censoring status is observed, and $n = \max_{1\leq i\leq m}\{n_i\}$ be the maximum number of time points. Then for any n_i, we have either $s_i(t_{n_i}) = 1$ if subject i is censored during time interval $(t_{n_i-1}, t_{n_i}]$, or $z_i(t_{n_i}) = 1$ if subject i experiences the event during time interval $(t_{n_i-1}, t_{n_i}]$.

Assume that all subjects are alive and not censored at time t_1 (otherwise the subject will not enter the study), that is, $z_i(t_1) = s_i(t_1) = 0$ for $i = 1, \ldots, m$. One of the goals of our joint model is to predict the value of $z_i(t_{j+1})$ given the history longitudinal observations and covariates up to time t_j, $j \geq 1$.

For subject i, let $\boldsymbol{x}_i(t_{1:j}) = (\boldsymbol{x}_i^T(t_1), \ldots, \boldsymbol{x}_i^T(t_j))^T$, $\boldsymbol{u}(t_{1:j}) = (\boldsymbol{u}^T(t_1), \ldots, \boldsymbol{u}^T(t_j))^T$, and $\boldsymbol{v}_i(t_{1:j}) = (\boldsymbol{v}_i^T(t_1), \ldots, \boldsymbol{v}_i^T(t_j))^T$ be historical covariates and latent state vectors. Survival status at the next time point $z_i(t_{j+1})$ is assumed to follow a Bernoulli distribution

$$z_i(t_{j+1})|[\boldsymbol{x}_i(t_{1:j}), \boldsymbol{u}(t_{1:j}), \boldsymbol{v}_i(t_{1:j})] \sim \text{Bernoulli}(\pi_i(t_{j+1})), \quad i = 1, \ldots, m,\ j = 1, \ldots, n_i, \tag{3.1}$$

with

$$\text{logit}[\pi_i(t_{j+1})] = \gamma_0 + \boldsymbol{x}_i^T(t_j)\boldsymbol{\gamma}_1 + (\boldsymbol{u}^T(t_j), \boldsymbol{v}_i^T(t_j))^T D^T \boldsymbol{\gamma}_2, \tag{3.2}$$

where γ_0, $\boldsymbol{\gamma}_1$, and $\boldsymbol{\gamma}_2$ are parameters, and D is a design matrix.

The survival model (3.1) and (3.2) can be easily modified to predict survival probabilities at a certain amount of time after t_j. For example, if we want to study the mortality rate k months from now, we simply change $z_i(t_{j+1})$ to $z_i(t_{j+k})$.

We note that it is not difficult to extend the following estimation methods to other logistic regression models.

3.2 Likelihood of the Joint Model

The joint model is based on the following assumptions:

1. observations from different subjects are independent of each other;
2. conditional on being alive and not censored at time t_{j-1}, $s_i(t_j)$ are independent of historical longitudinal observations.
3. all longitudinal variables and covariates are observed at t_j as long as $z_i(t_j) = 0$ and $s_i(t_j) = 0$, and are not observed when either $z_i(t_j) = 1$ or $s_i(t_j) = 1$;
4. when $s_i(t_j) = 1$, the survival status $z_i(t_j)$ is not observed;
5. when $z_i(t_j) = 1$, we have $s_i(t_j) = 0$.

At time t_j, $j = 2, \dots, n$, let

$$A_j = \{i : z_i(t_j) = 0 \text{ and } s_i(t_j) = 0, i \in A_{j-1}\}, \tag{3.3}$$

$$B_j = \{i : z_i(t_j) = 1, i \in A_{j-1}\}, \tag{3.4}$$

and

$$C_j = \{i : s_i(t_j) = 1, i \in A_{j-1}\}, \tag{3.5}$$

where A_1 is the index set of all subjects. That is, A_j is the index set of subjects who are alive and not censored at time t_j, B_j is the index set of subjects who experience the terminating event during time interval $(t_{j-1}, t_j]$, and C_j is the index set of subjects who are censored during time interval $(t_{j-1}, t_j]$. Once a subject drops out due to censoring or death, he/she will no longer be included in A_j, B_j, or C_j at later time points.

At time t_1, we assume that all subjects are alive and not censored, otherwise he/she will not enter the study. Therefore, we have $B_1 = \emptyset$ and $C_1 = \emptyset$. We have the following relationships:

1. $A_1 \supset A_2 \supset \cdots \supset A_n$, because subjects gradually drop out or experience the terminating event as time moves on;
2. $A_j \cap (B_j \cup C_j) = \emptyset$ for $j = 1, \ldots, n$;
3. $B_j \cap C_j = \emptyset$; and
4. $A_j \cup B_j \cup C_j = A_{j-1}$ for $j = 2, ..., n$.

Let $\boldsymbol{y}^{A_j}(t_j)$ be the stacked vector of $\{\boldsymbol{y}_i(t_j)\}_{i \in A_j}$ and $\boldsymbol{Y}^{A_{1:j}}(t_{1:j}) = (\boldsymbol{y}^{A_1}(t_1)^T, \ldots, \boldsymbol{y}^{A_j}(t_j)^T)^T$, $1 \leq j \leq n$. Let $\boldsymbol{\alpha}^{A_j}(t_j)$ be a subvector of the state vector $\boldsymbol{\alpha}(t_j)$ with components corresponding to subjects who are not in the set A_j removed. That is, $\boldsymbol{\alpha}^{A_j}(t_j) = (\boldsymbol{u}^T(t_j), \boldsymbol{v}_{i_1}^T(t_j), \ldots, \boldsymbol{v}_{i_{k_j}}^T(t_j))^T$ where $A_j = \{i_1, \ldots, i_{k_j}\}$ and $i_1 < \cdots < i_{k_j}$, $1 \leq j \leq n$. Similar definition applies for $\boldsymbol{\alpha}^{A_j \cup B_j}(t_{j-1})$, $2 \leq j \leq n$.

Denoting by $\boldsymbol{z}^{A_j}(t_j)$ the stacked vector of $\{z_i(t_j)\}_{i \in A_j}$, then according to the definition of A_j, we have $\boldsymbol{z}^{A_j}(t_j) = \boldsymbol{0}$ for $j = 1, \ldots, n$. Letting $\boldsymbol{z}^{B_j}(t_j)$ be the stacked vector of $\{z_i(t_j)\}_{i \in B_j}$, according to the definition of B_j, we have $\boldsymbol{z}^{B_j}(t_j) = \boldsymbol{1}$ for $2 \leq j \leq n$ and $\boldsymbol{z}^{B_1}(t_1) = \emptyset$. $\boldsymbol{z}^{C_j}(t_j) = \{z_i(t_j)\}_{i \in C_j} = \emptyset$ according to our assumption. Let $\boldsymbol{Z}^{A_{1:j}}(t_{1:j}) = (\boldsymbol{z}^{A_1}(t_1)^T, \ldots, \boldsymbol{z}^{A_j}(t_j))^T$ and $\boldsymbol{Z}^{B_{1:j}}(t_{1:j}) = (\boldsymbol{z}^{B_1}(t_1)^T, \ldots, \boldsymbol{z}^{B_j}(t_j))^T$, $1 \leq j \leq n$.

Let $\boldsymbol{s}^{A_{j-1}}(t_j)$ be the stacked vector of $\{s_i(t_j)\}_{i \in A_{j-1}}$. That is, $\boldsymbol{s}^{A_{j-1}}(t_j)$ is the censoring status at t_j for subjects who are alive at time t_{j-1}. Define A_0 as the index set of all subjects and we have $\boldsymbol{s}^{A_0}(t_1) = \boldsymbol{0}$. Let $\boldsymbol{S}^{A_{0:j-1}}(t_{1:j}) = (\boldsymbol{s}^{A_0}(t_1)^T, \ldots, \boldsymbol{s}^{A_{j-1}}(t_j)^T)^T$, $1 \leq j \leq n$.

Suppose that $\boldsymbol{\psi}$ is the vector of all parameters in the model. Since all subjects are alive at t_1, we have $\boldsymbol{z}^{A_1}(t_1) = \boldsymbol{0}$ with probability 1. At time t_j, $1 \leq j \leq n$, $\boldsymbol{y}^{A_j}(t_j)$ is the vector of current longitudinal observations, $(\boldsymbol{z}^{A_j}(t_j), \boldsymbol{z}^{B_j}(t_j))$ are current survival status observations, and $\boldsymbol{s}^{A_{j-1}}(t_j)$ are censoring status observations for all subjects in A_{j-1}. Therefore, the likelihood is given by

$$\begin{aligned} L(\boldsymbol{\psi}) &= p(\boldsymbol{Y}^{A_{1:n}}(t_{1:n}), \boldsymbol{Z}^{A_{1:n}}(t_{1:n}), \boldsymbol{Z}^{B_{1:n}}(t_{1:n}), \boldsymbol{S}^{A_{0:n-1}}(t_{1:n})) \\ &= p(\boldsymbol{y}^{A_1}(t_1), \underbrace{\boldsymbol{z}^{A_1}(t_1)}_{\boldsymbol{0}}, \underbrace{\boldsymbol{z}^{B_1}(t_1)}_{\emptyset}, \underbrace{\boldsymbol{s}^{A_0}(t_1)}_{\boldsymbol{0}}) \\ &\quad \textstyle\prod_{j=2}^{n} p(\boldsymbol{y}^{A_j}(t_j), \boldsymbol{z}^{A_j}(t_j), \boldsymbol{z}^{B_j}(t_j), \boldsymbol{s}^{A_{j-1}}(t_j)| \\ &\quad \boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1}), \boldsymbol{Z}^{A_{1:j-1}}(t_{1:j-1}), \boldsymbol{Z}^{B_{1:j-1}}(t_{1:j-1}), \boldsymbol{S}^{A_{0:j-2}}(t_{1:j-1})). \end{aligned} \tag{3.6}$$

Note that all densities are conditional on external covariates $\boldsymbol{x}_i(t_j)$, which is suppressed here for notational simplicity.

In (3.6), we have

$$\begin{aligned} & p(\boldsymbol{y}^{A_1}(t_1), \boldsymbol{z}^{A_1}(t_1), \boldsymbol{z}^{B_1}(t_1), \boldsymbol{s}^{A_0}(t_1)) \\ = \;& p(\boldsymbol{y}^{A_1}(t_1), \boldsymbol{z}^{A_1}(t_1) = \boldsymbol{0}, \boldsymbol{s}^{A_0}(t_1) = \boldsymbol{0}) \\ = \;& p(\boldsymbol{y}^{A_1}(t_1)). \end{aligned} \tag{3.7}$$

The second half of the likelihood in (3.6) can be rewritten as

$$
\begin{aligned}
& \textstyle\prod_{j=2}^{n} p(\boldsymbol{y}^{A_j}(t_j), \boldsymbol{z}^{A_j}(t_j), \boldsymbol{z}^{B_j}(t_j), \boldsymbol{s}^{A_{j-1}}(t_j)| \\
& \boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1}), \boldsymbol{Z}^{A_{1:j-1}}(t_{1:j-1}), \boldsymbol{Z}^{B_{1:j-1}}(t_{1:j-1}), \boldsymbol{S}^{A_{0:j-2}}(t_{1:j-1})) \\
= \; & \textstyle\prod_{j=2}^{n} p(\boldsymbol{y}^{A_j}(t_j)|\boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1}), \boldsymbol{Z}^{A_{1:j-1}}(t_{1:j-1}), \boldsymbol{Z}^{B_{1:j-1}}(t_{1:j-1}), \\
& \boldsymbol{S}^{A_{0:j-2}}(t_{1:j-1}), \boldsymbol{z}^{A_j}(t_j), \boldsymbol{z}^{B_j}(t_j), \boldsymbol{s}^{A_{j-1}}(t_j)) \\
& \textstyle\prod_{j=2}^{n} p(\boldsymbol{z}^{A_j}(t_j), \boldsymbol{z}^{B_j}(t_j)|\boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1}), \\
& \boldsymbol{Z}^{A_{1:j-1}}(t_{1:j-1}), \boldsymbol{Z}^{B_{1:j-1}}(t_{1:j-1}), \boldsymbol{S}^{A_{0:j-2}}(t_{1:j-1}), \boldsymbol{s}^{A_{j-1}}(t_j)) \\
& \textstyle\prod_{j=2}^{n} \underbrace{p(\boldsymbol{s}^{A_{j-1}}(t_j)|\boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1}), \boldsymbol{Z}^{A_{1:j-1}}(t_{1:j-1}), \boldsymbol{Z}^{B_{1:j-1}}(t_{1:j-1}), \boldsymbol{S}^{A_{0:j-2}}(t_{1:j-1}))}_{=p(\boldsymbol{s}^{A_{j-1}}(t_j)) \text{ by assumption}} \\
= \; & \textstyle\prod_{j=2}^{n} p(\boldsymbol{y}^{A_j}(t_j)|\boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1}), \boldsymbol{Z}^{A_{1:j}}(t_{1:j}), \boldsymbol{Z}^{B_{1:j}}(t_{1:j}), \boldsymbol{S}^{A_{0:j-1}}(t_{1:j})) \\
& \textstyle\prod_{j=2}^{n} p(\boldsymbol{z}^{A_j}(t_j), \boldsymbol{z}^{B_j}(t_j)|\boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1}), \boldsymbol{Z}^{A_{1:j-1}}(t_{1:j-1}), \boldsymbol{Z}^{B_{1:j-1}}(t_{1:j-1}), \boldsymbol{S}^{A_{0:j-1}}(t_{1:j})) \\
& \underbrace{\prod_{j=2}^{n} p(\boldsymbol{s}^{A_{j-1}}(t_j))}_{\text{constant}} \\
= \; & \textstyle\prod_{j=2}^{n} p(\boldsymbol{y}^{A_j}(t_j)|\boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1})) \prod_{j=2}^{n} p(\boldsymbol{z}^{A_j}(t_j), \boldsymbol{z}^{B_j}(t_j)|\boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1})) \cdot \text{constant.}
\end{aligned}
\tag{3.8}
$$

In (3.8), $\prod_{j=2}^{n} p(\boldsymbol{s}^{A_{j-1}}(t_j))$ is a constant because by assumption, given that subject i is alive and not censored at time t_{j-1}, the distribution of $s_i(t_j)$ is independent of the history of longitudinal variables. The third equality in (3.8) holds because in the first density, conditional on $\boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1})$, the longitudinal variables of subjects in A_j are independent of the historical survival and censoring status of subjects who are not in A_j; in addition, although the existence of $\boldsymbol{y}^{A_j}(t_j)$ conditions on $\boldsymbol{Z}^{A_j}(t_{1:j-1}) = \boldsymbol{0}$ and $\boldsymbol{S}^{A_j}(t_{1:j}) = \boldsymbol{0}$, these conditions are indicated by the superscript A_j of $\boldsymbol{y}^{A_j}(t_j)$, thus we suppress in notation the conditions $\boldsymbol{Z}^{A_j}(t_{1:j-1}) = \boldsymbol{0}$ and $\boldsymbol{S}^{A_j}(t_{1:j}) = \boldsymbol{0}$; and similar reasons apply for the second density.

Plugging (3.7) and (3.8) into (3.6), the likelihood is rewritten as

$$\begin{aligned} L(\boldsymbol{\psi}) &= \text{constant} \cdot p(\boldsymbol{y}^{A_1}(t_1)) \textstyle\prod_{j=2}^{n} p(\boldsymbol{y}^{A_j}(t_j)|\boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1})) \\ &\quad \textstyle\prod_{j=2}^{n} p(\boldsymbol{z}^{A_j}(t_j), \boldsymbol{z}^{B_j}(t_j)|\boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1})) \\ &= \text{constant} \cdot L_1(\boldsymbol{\psi}) L_2(\boldsymbol{\psi}), \end{aligned} \tag{3.9}$$

where

$$L_1(\boldsymbol{\psi}) = p(\boldsymbol{y}^{A_1}(t_1)) \prod_{j=2}^{n} p(\boldsymbol{y}^{A_j}(t_j)|\boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1})) \tag{3.10}$$

is the joint density of longitudinal data, and

$$L_2(\boldsymbol{\psi}) = \prod_{j=2}^{n} p(\boldsymbol{z}^{A_j}(t_j), \boldsymbol{z}^{B_j}(t_j)|\boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1})) \tag{3.11}$$

is the conditional joint survival density given longitudinal data. $L_1(\boldsymbol{\psi})$ is computed by the Kalman filter. The density $p(\boldsymbol{z}^{A_j}(t_j), \boldsymbol{z}^{B_j}(t_j)|\boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1}))$ in $L_2(\boldsymbol{\psi})$ is calculated by

$$\begin{aligned} &\textstyle\int p(\boldsymbol{z}^{A_j}(t_j), \boldsymbol{z}^{B_j}(t_j)|\boldsymbol{\alpha}^{A_{j-1}}(t_{j-1}), \boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1})) \\ &\quad p(\boldsymbol{\alpha}^{A_{j-1}}(t_{j-1})|\boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1}))d\boldsymbol{\alpha}^{A_{j-1}}(t_{j-1}) \\ = &\textstyle\int p(\boldsymbol{z}^{A_j}(t_j), \boldsymbol{z}^{B_j}(t_j)|\boldsymbol{\alpha}^{A_{j-1}}(t_{j-1}))p(\boldsymbol{\alpha}^{A_{j-1}}(t_{j-1})|\boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1}))d\boldsymbol{\alpha}^{A_{j-1}}(t_{j-1}), \end{aligned} \tag{3.12}$$

for $j = 2, \ldots, n$, in which the term $\boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1})$ is dropped from the first density because by definition, $\boldsymbol{z}^{A_j}(t_j)$ and $\boldsymbol{z}^{B_j}(t_j)$ are independent of $\boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1})$ given $\boldsymbol{\alpha}^{A_{j-1}}(t_{j-1})$ (recall that $A_j \cup B_j \subset A_{j-1}$). In addition, according to the definition of $z_i(t_j)$ in (3.1) and (3.2), the $z_i(t_j)$'s for different subjects are conditionally independent given covariates

$\boldsymbol{x}_i(t_{j-1})$ and the state vector $\boldsymbol{\alpha}(t_{j-1})$. Therefore, we have

$$p(\boldsymbol{z}^{A_j}(t_j), \boldsymbol{z}^{B_j}(t_j)|\boldsymbol{\alpha}^{A_{j-1}}(t_{j-1})) = \prod_{i \in A_j \cup B_j} p(z_i(t_j)|\boldsymbol{\alpha}^{A_{j-1}}(t_{j-1})). \tag{3.13}$$

The integral (3.12) does not have an explicit form because $p(\boldsymbol{z}^{(\cdot)}(\cdot)|\boldsymbol{\alpha}^{(\cdot)}(\cdot))$ is Bernoulli, and $p(\boldsymbol{\alpha}^{(\cdot)}(\cdot)|\boldsymbol{y}^{(\cdot)}(\cdot))$ is Gaussian. A numerical method is thus needed to evaluate (3.12).

To illustrate our approximation method for the integral (3.12), let us first simplify the notations and abstract the problem as

$$p(\boldsymbol{z}|\boldsymbol{y}) = \int p(\boldsymbol{z}|\boldsymbol{\alpha})p(\boldsymbol{\alpha}|\boldsymbol{y})d\boldsymbol{\alpha}, \tag{3.14}$$

where $\boldsymbol{z} = (z_1, \ldots, z_m)^T$ is a vector of m Bernoulli random variables that are conditionally independent given $\boldsymbol{\alpha}$, $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_0^T, \boldsymbol{\alpha}_1^T, \ldots, \boldsymbol{\alpha}_m^T)^T$, in which $\boldsymbol{\alpha}_0$ is a $d_u \times 1$ vector and $\boldsymbol{\alpha}_i$ are $d_v \times 1$ vectors for $i = 1, \ldots, m$; and $\boldsymbol{y} = (\boldsymbol{y}_1^T, \ldots, \boldsymbol{y}_m^T)^T$, wherein $\boldsymbol{y}_i$ are $q \times 1$ vectors, $i = 1, \ldots, m$. In (3.14), $p(\boldsymbol{z}|\boldsymbol{\alpha}) = \prod_{i=1}^m p(z_i|\boldsymbol{\alpha})$ is the joint conditional density of m Bernoulli random variables and is known, and $p(\boldsymbol{\alpha}|\boldsymbol{y})$ is a known multivariate Gaussian density $N(\boldsymbol{a}, P)$, where

$$\boldsymbol{a} = (\boldsymbol{a}_0^T, \boldsymbol{a}_1^T, \ldots, \boldsymbol{a}_m^T)^T \tag{3.15}$$

is partitioned in the same way as $\boldsymbol{\alpha}$, and P is of the structure

$$P = \begin{pmatrix} P_0 & \mathbf{1}_m^T \otimes P_1 \\ \mathbf{1}_m \otimes P_1^T & I_m \otimes P_2 + \mathbf{1}_{m \times m} \otimes P_3 \end{pmatrix}, \tag{3.16}$$

where P_0, P_2, and P_3 are symmetric matrices; and P_0 and $P_2 + P_3$ are positive semi-definite matrices.

The goal is to calculate $p(\boldsymbol{z}|\boldsymbol{y})$, which can be approximated by

$$p(\boldsymbol{z}|\boldsymbol{y}) \approx \frac{1}{M}\sum_{k=1}^{M} p(\boldsymbol{z}|\boldsymbol{\alpha}^{(k)}), \tag{3.17}$$

where $\boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(M)}$ are samples drawn from $p(\boldsymbol{\alpha}|\boldsymbol{y})$.

In integral (3.12), $N(\boldsymbol{a}, P)$ is the filtering distribution computed by Kalman filter, where $\boldsymbol{a}$ is of the form (3.15) and P is of the form (3.16). The dimension of P is $(d_u + md_v) \times (d_u + md_v)$, where m can be extremely large. It is therefore impractical to simulate directly from $N(\boldsymbol{a}, P)$, which requires a Cholesky decomposition of the extremely high-dimensional matrix P.

The trick we use here is sequential conditional sampling. Suppose we wish to generate a sample $\boldsymbol{\alpha}$ from $N(\boldsymbol{a}, P)$. We partition $\boldsymbol{\alpha}$ into $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_0^T, \boldsymbol{\alpha}_1^T, \ldots, \boldsymbol{\alpha}_m^T)^T$ and generate its components sequentially, each conditional on the already generated components. To explain, first, a lemma on multivariate Gaussian conditional distribution is presented below.

Lemma 1 *Suppose that the joint distribution of two random vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is normal, with mean and covariance matrix given by*

$$E\begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\mu_x} \\ \boldsymbol{\mu_y} \end{pmatrix}, \quad Var\begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix} = \begin{pmatrix} \Sigma_{\boldsymbol{xx}} & \Sigma_{\boldsymbol{xy}} \\ \Sigma_{\boldsymbol{xy}}^T & \Sigma_{\boldsymbol{yy}} \end{pmatrix},$$

then the conditional distribution of $\boldsymbol{x}$ given $\boldsymbol{y}$ is normal with conditional mean and covariance matrix

$$E(\boldsymbol{x}|\boldsymbol{y}) = \boldsymbol{\mu_x} + \Sigma_{\boldsymbol{xy}}\Sigma_{\boldsymbol{yy}}^{-1}(\boldsymbol{y} - \boldsymbol{\mu_y}), \quad Var(\boldsymbol{x}|\boldsymbol{y}) = \Sigma_{\boldsymbol{xx}} - \Sigma_{\boldsymbol{xy}}\Sigma_{\boldsymbol{yy}}^{-1}\Sigma_{\boldsymbol{xy}}^T. \quad \blacksquare$$

Using Lemma 1, $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_0^T, \boldsymbol{\alpha}_1^T, \dots, \boldsymbol{\alpha}_m^T)^T$ can be generated from $N(\boldsymbol{a}, P)$ in the following way. First, generate a $d_v \times 1$ random vector $\boldsymbol{\alpha}_m$ from $N(\boldsymbol{a}_m, P_2 + P_3)$. To generate $\boldsymbol{\alpha}_{m-1}$, note that the joint distribution of $\boldsymbol{\alpha}_{m-1}$ and $\boldsymbol{\alpha}_m$ is

$$N(\begin{pmatrix} \boldsymbol{a}_{m-1} \\ \boldsymbol{a}_m \end{pmatrix}, \begin{pmatrix} P_2 + P_3 & P_3 \\ P_3 & P_2 + P_3 \end{pmatrix}). \tag{3.18}$$

By Lemma 1, we have that the conditional distribution $\boldsymbol{\alpha}_{m-1}|\boldsymbol{\alpha}_m$ is normal with mean

$$\mathrm{E}(\boldsymbol{\alpha}_{m-1}|\boldsymbol{\alpha}_m) = \boldsymbol{a}_{m-1} + P_3(P_2 + P_3)^{-1}(\boldsymbol{\alpha}_m - \boldsymbol{a}_m)$$

and covariance matrix

$$\mathrm{Var}(\boldsymbol{\alpha}_{m-1}|\boldsymbol{\alpha}_m) = (P_2 + P_3) - P_3(P_2 + P_3)^{-1}P_3.$$

Generate $\boldsymbol{\alpha}_{m-2}, \dots, \boldsymbol{\alpha}_1$ sequentially using conditional distributions in a similar manner. In fact, suppose that $\boldsymbol{\alpha}_m, \boldsymbol{\alpha}_{m-1}, \dots, \boldsymbol{\alpha}_{k+1}$ have already been generated, $1 \le k \le m-2$, the next step is to generate $\boldsymbol{\alpha}_k$ conditional on $(\boldsymbol{\alpha}_{k+1}, \dots, \boldsymbol{\alpha}_m)$. The joint distribution of $\boldsymbol{\alpha}_k, \boldsymbol{\alpha}_{k+1}, \dots, \boldsymbol{\alpha}_m$ is

$$N(\begin{pmatrix} \boldsymbol{a}_k \\ \vdots \\ \boldsymbol{a}_m \end{pmatrix}, I_{m-k+1} \otimes P_2 + \mathbf{1}_{(m-k+1)\times(m-k+1)} \otimes P_3). \tag{3.19}$$

By Lemma 1, the conditional distribution $\boldsymbol{\alpha}_k|(\boldsymbol{\alpha}_{k+1}, \ldots, \boldsymbol{\alpha}_m)$ is normal with mean

$$\mathrm{E}(\boldsymbol{\alpha}_k|\boldsymbol{\alpha}_{k+1}, \ldots, \boldsymbol{\alpha}_m) = \boldsymbol{a}_k + (\mathbf{1}_{m-k}^T \otimes P_3)(I_{m-k} \otimes P_2 + \mathbf{1}_{(m-k)\times(m-k)} \otimes P_3)^{-1} \begin{pmatrix} \boldsymbol{\alpha}_{k+1} - \boldsymbol{a}_{k+1} \\ \vdots \\ \boldsymbol{\alpha}_m - \boldsymbol{a}_m \end{pmatrix}. \tag{3.20}$$

Using the Woodbury formula, similar to how $F^{-1}(t_j)$ is calculated in Theorem 1, we have

$$(I_{m-k} \otimes P_2 + \mathbf{1}_{(m-k)\times(m-k)} \otimes P_3)^{-1} = I_{m-k} \otimes W_{1,k} + \mathbf{1}_{(m-k)\times(m-k)} \otimes W_{2,k}, \tag{3.21}$$

where $W_{1,k} = P_2^{-1}$ and $W_{2,k} = -P_2^{-1}P_3[P_2 + (m-k)P_3]^{-1}$. So (3.20) becomes

$$\begin{aligned} & \mathrm{E}(\boldsymbol{\alpha}_k|\boldsymbol{\alpha}_{k+1}, \ldots, \boldsymbol{\alpha}_m) \\ = \;& \boldsymbol{a}_k + P_3[W_{1,k} + (m-k)W_{2,k}] \textstyle\sum_{i=k+1}^m (\boldsymbol{\alpha}_i - \boldsymbol{a}_i) \\ = \;& \boldsymbol{a}_k + P_3\{P_2^{-1} - (m-k)P_2^{-1}P_3[P_2 + (m-k)P_3]^{-1}\} \textstyle\sum_{i=k+1}^m (\boldsymbol{\alpha}_i - \boldsymbol{a}_i). \end{aligned}$$

The covariance matrix is

$$\begin{aligned} & \mathrm{Var}(\boldsymbol{\alpha}_k|\boldsymbol{\alpha}_{k+1}, \ldots, \boldsymbol{\alpha}_m) \\ = \;& (P_2 + P_3) - (\mathbf{1}_{m-k}^T \otimes P_3)(I_{m-k} \otimes P_2 + \mathbf{1}_{(m-k)\times(m-k)} \otimes P_3)^{-1}(\mathbf{1}_{m-k} \otimes P_3). \\ = \;& (P_2 + P_3) - (m-k)P_3[W_{1,k} + (m-k)W_{2,k}]P_3 \\ = \;& (P_2 + P_3) - (m-k)P_3\{P_2^{-1} - (m-k)P_2^{-1}P_3[P_2 + (m-k)P_3]^{-1}\}P_3. \end{aligned}$$

After obtaining $\boldsymbol{\alpha}_i$, $i = 1, \ldots, m$, the next step is to generate $\boldsymbol{\alpha}_0$ conditional on $(\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m)$. The joint distribution of $\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m$ is $N(\boldsymbol{a}, P)$. By Lemme 1, the conditional distribution $\boldsymbol{\alpha}_0|(\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m)$ is normal with mean

$$\begin{aligned}
\mathrm{E}(\boldsymbol{\alpha}_0|\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m) &= \boldsymbol{a}_0 + (\mathbf{1}_m^T \otimes P_3)(I_m \otimes P_2 + \mathbf{1}_{m\times m} \otimes P_3)^{-1} \begin{pmatrix} \boldsymbol{\alpha}_1 - \boldsymbol{a}_1 \\ \vdots \\ \boldsymbol{\alpha}_m - \boldsymbol{a}_m \end{pmatrix} \\
&= \boldsymbol{a}_0 + P_1[W_{1,0} + mW_{2,0}] \textstyle\sum_{i=1}^m (\boldsymbol{\alpha}_i - \boldsymbol{a}_i) \\
&= \boldsymbol{a}_0 + P_1\{P_2^{-1} - mP_2^{-1}P_3[P_2 + mP_3]^{-1}\} \textstyle\sum_{i=1}^m (\boldsymbol{\alpha}_i - \boldsymbol{a}_i),
\end{aligned} \tag{3.22}$$

where $W_{1,0} = P_2^{-1}$ and $W_{2,0} = -P_2^{-1}P_3[P_2 + mP_3]^{-1}$, and covariance matrix

$$\begin{aligned}
\mathrm{Var}(\boldsymbol{\alpha}_0|\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m) &= P_0 - (\mathbf{1}_m^T \otimes P_3)(I_m \otimes P_2 + \mathbf{1}_{m\times m} \otimes P_3)^{-1}(\mathbf{1}_m \otimes P_3) \\
&= P_0 - mP_1(W_{1,0} + mW_{2,0})P_1^T \\
&= P_0 - mP_1\{P_2^{-1} - mP_2^{-1}P_3[P_2 + mP_3]^{-1}\}P_1^T.
\end{aligned} \tag{3.23}$$

The simulation algorithm is summarized in Algorithm 1. In step 2 we need to calculate $\sum_{i=k+1}^m (\boldsymbol{\alpha}_i - \boldsymbol{a}_i)$ for $k = m-1, \ldots, 1$, making the time complexity quadratic in m. However, in implementation, for each k, we can store the current $\sum_{i=k+1}^m (\boldsymbol{\alpha}_i - \boldsymbol{a}_i)$. Then for $i = k-1$ we only need to add $\boldsymbol{\alpha}_k - \boldsymbol{a}_k$ to the previously stored quantity, hence reducing time complexity to $\mathrm{O}(m)$.

Algorithm 1: Generating a sample from $N(\boldsymbol{a}, P)$.

```
Input  : $\boldsymbol{a}$ and $P$, where $\boldsymbol{a}$ is of the structure (3.15) and $P$ of the structure (3.16)
Output: a sample $\boldsymbol{\alpha}$ generated from $N(\boldsymbol{a}, P)$
generate $\boldsymbol{\alpha}_m$ from $N(\boldsymbol{a}_m, P_2 + P_3)$;
for $k = m-1, \ldots, 1$ do
    generate $\boldsymbol{\alpha}_k$ from the multivariate normal conditional distribution
      $\boldsymbol{\alpha}_k|(\boldsymbol{\alpha}_{k+1}, \ldots, \boldsymbol{\alpha}_m)$ with mean $\mathrm{E}(\boldsymbol{\alpha}_k|\boldsymbol{\alpha}_{k+1}, \ldots, \boldsymbol{\alpha}_m) =$
      $\boldsymbol{a}_k + P_3\{P_2^{-1} - (m-k)P_2^{-1}P_3[P_2 + (m-k)P_3]^{-1}\}\sum_{i=k+1}^{m}(\boldsymbol{\alpha}_i - \boldsymbol{a}_i)$ and
      covariance matrix $\mathrm{Var}(\boldsymbol{\alpha}_k|\boldsymbol{\alpha}_{k+1}, \ldots, \boldsymbol{\alpha}_m) =$
      $(P_2 + P_3) - (m-k)P_3\{P_2^{-1} - (m-k)P_2^{-1}P_3[P_2 + (m-k)P_3]^{-1}\}P_3$;
end
generate $\boldsymbol{\alpha}_0$ from the multivariate normal conditional distribution $\boldsymbol{\alpha}_0|\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m$
  with mean
  $\mathrm{E}(\boldsymbol{\alpha}_0|\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m) = \boldsymbol{a}_0 + P_1\{P_2^{-1} - mP_2^{-1}P_3[P_2 + mP_3]^{-1}\}\sum_{i=1}^{m}(\boldsymbol{\alpha}_i - \boldsymbol{a}_i)$ and
  covariance matrix
  $\mathrm{Var}(\boldsymbol{\alpha}_0|\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m) = P_0 - mP_1\{P_2^{-1} - mP_2^{-1}P_3[P_2 + mP_3]^{-1}\}P_1^T$;
return $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_0^T, \boldsymbol{\alpha}_1^T, \ldots, \boldsymbol{\alpha}_m^T)^T$.
```

Antithetic variables Given one draw of $\boldsymbol{\alpha}$ from $N(\boldsymbol{a}, P)$, we can use antithetic variables to improve the efficiency in estimating (3.14). According to Durbin and Koopman [26], an antithetic variable is "a random draw of $\boldsymbol{\alpha}$ which is equiprobable with $\boldsymbol{\alpha}$ and which, when included together with $\boldsymbol{\alpha}$ in the estimate of the target function increases the efficiency of the estimation". Let $\breve{\boldsymbol{\alpha}} = 2\boldsymbol{a} - \boldsymbol{\alpha}$. Since $\breve{\boldsymbol{\alpha}} - \boldsymbol{a} = -(\boldsymbol{\alpha} - \boldsymbol{a})$ and $\boldsymbol{\alpha}$ is normal, we have that $\breve{\boldsymbol{\alpha}}$ and $\boldsymbol{\alpha}$ are equiprobable.

To approximate $L_2(\psi)$ in (3.11), at each time point t_j, $j = 2, \dots, n$, we draw N samples from the filtering distribution $p(\boldsymbol{\alpha}^{A_j \cup B_j}(t_{j-1}) | \boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1}))$ using Algorithm 1, where $p(\boldsymbol{\alpha}^{A_j \cup B_j}(t_{j-1}) | \boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1}))$ is a multivariate Gaussian distribution for subjects who are in the set $A_j \cup B_j$. Note that $\boldsymbol{\alpha}^{A_j \cup B_j}(t_{j-1})$ is used in place of $\boldsymbol{\alpha}^{A_{j-1}}(t_{j-1})$, where the former is a subvector of the latter. This is because we calculate survival likelihood only for subjects in $A_j \cup B_j$. Then $p(\boldsymbol{z}^{A_j}(t_j), \boldsymbol{z}^{B_j}(t_j) | \boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1}))$ is approximated by

$$p(\boldsymbol{z}^{A_j}(t_j), \boldsymbol{z}^{B_j}(t_j) | \boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1})) \approx \frac{1}{N} \sum_{k=1}^{N} p(\boldsymbol{z}^{A_j}(t_j), \boldsymbol{z}^{B_j}(t_j) | \boldsymbol{\alpha}^{A_j \cup B_j, (k)}(t_{j-1})), \quad (3.24)$$

where $\boldsymbol{\alpha}^{A_j \cup B_j, (k)}(t_{j-1})$, $k = 1, \dots, N$, are samples generated from the filtering distribution $p(\boldsymbol{\alpha}^{A_j \cup B_j}(t_{j-1}) | \boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1}))$.

Parameters are estimated by maximizing the logarithm of the joint likelihood (3.9). Since the likelihood surface of the state space model is not convex, we first use the genetic algorithm with R function *ga* in the GA package to find a set of reasonable initial values for the parameters, then use the Nelder-Mead method in the R function *optim* for the optimization procedure.

Chapter 4

Simulation Studies

4.1 Mixed Effects State Space Models for Longitudinal Data

In this section, we consider longitudinal data only, under three scenarios: (i) both population effects and subject random effects are generated by local level models, (ii) population effects are generated by cubic spline models and subject random effects by OU processes, and (iii) both population effects and subject random effects are generated by cubic spline models. Covariates or survival data are not considered in this section.

We set $q = 2$ and $n = 50$. To compare CPU time for one round of filtering using the univariate treatment and the new algorithm, we consider 5 choices of m: 50, 100, 200, 500, 1000, and 3 more values for the new algorithm: $m = 10^4, 10^5, 10^6$. We take the average CPU time over 10 replications for each setting. For evaluating the performance of the new algorithm, we use 5 choices of m: 100, 500, 1000, 5000, 10000. We use 100 replications for each setting.

The vector form of the mixed effects state space model is given by (2.13), rewritten here without the covariates regression term:

$$\begin{aligned}
\boldsymbol{y}(t_j) &= Z(t_j)\boldsymbol{\alpha}(t_j) + \boldsymbol{\epsilon}(t_j), & \boldsymbol{\epsilon}(t_j) &\sim N(\mathbf{0}, H(t_j)), \\
\boldsymbol{\alpha}(t_{j+1}) &= T(t_j)\boldsymbol{\alpha}(t_j) + R(t_j)\boldsymbol{\eta}(t_j), & \boldsymbol{\eta}(t_j) &\sim N(\mathbf{0}, Q(t_j)), \\
& & \boldsymbol{\alpha}(t_1) &\sim N(\boldsymbol{a}(t_1), P(t_1)),
\end{aligned} \tag{4.1}$$

for $j = 1, \ldots, n$, where we only consider $Z(t_j) = Z$ and $T(t_j) = T$ being time-invariant, and $R(t_j)$ is the identity matrix. In (4.1),

$$\boldsymbol{y}(t_j) = \begin{pmatrix} \boldsymbol{y}_1(t_j) \\ \vdots \\ \boldsymbol{y}_m(t_j) \end{pmatrix} \tag{4.2}$$

is a $qm \times 1$ vector of stacked observations for all subjects, where $\boldsymbol{y}_i(t_j) = (y_{i1}(t_j), \ldots, y_{iq}(t_j))^T$ is the vector of q longitudinal observations for subject i, $i = 1, \ldots, m$. The state vector

$$\boldsymbol{\alpha}(t_j) = \begin{pmatrix} \boldsymbol{u}(t_j) \\ \boldsymbol{v}_1(t_j) \\ \vdots \\ \boldsymbol{v}_m(t_j) \end{pmatrix} \tag{4.3}$$

is of dimension $(d_u + md_v) \times 1$, where $\boldsymbol{u}(t_j)$ is a $d_u \times 1$ vector and $\boldsymbol{v}_i(t_j)$ is a $d_v \times 1$ vector, $1 \leq i \leq m$. The values of d_u and d_v are proportional to q and depend on model specifications for population effects and subject random effects. The system matrix

$$Z(t_j) = Z = \begin{pmatrix} Z_u & Z_v & & 0 \\ \vdots & & \ddots & \\ Z_u & 0 & & Z_v \end{pmatrix} \tag{4.4}$$

is $qm \times (d_u + md_v)$, wherein Z_u is a $q \times d_u$ matrix and Z_v is a $q \times d_v$ matrix. The state transition matrix

$$T(t_j) = T = \begin{pmatrix} T_u & 0 & \cdots & 0 \\ 0 & T_v & \cdots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & & T_v \end{pmatrix} \tag{4.5}$$

is $(d_u + md_v) \times (d_u + md_v)$, in which T_u is $d_u \times d_u$ and T_v is $d_v \times d_v$ The random error

$$\boldsymbol{\epsilon}(t_j) = \begin{pmatrix} \boldsymbol{\epsilon}_1(t_j) \\ \vdots \\ \boldsymbol{\epsilon}_m(t_j) \end{pmatrix} \tag{4.6}$$

of the observation equation has covariance matrix

$$H(t_j) = H = \begin{pmatrix} \Sigma_\epsilon & & 0 \\ & \ddots & \\ 0 & & \Sigma_\epsilon \end{pmatrix}, \tag{4.7}$$

where Σ_ϵ is either a diagonal or unstructured $q \times q$ covariance matrix. The state disturbance term

$$\boldsymbol{\eta}(t_j) = \begin{pmatrix} \boldsymbol{\eta}_u(t_j) \\ \boldsymbol{\eta}_{v1}(t_j) \\ \vdots \\ \boldsymbol{\eta}_{vm}(t_j) \end{pmatrix} \tag{4.8}$$

has covariance matrix

$$Q(t_j) = Q = \begin{pmatrix} \Sigma_u & 0 & \cdots & 0 \\ 0 & \Sigma_v & \cdots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & & \Sigma_v \end{pmatrix}, \tag{4.9}$$

where Σ_u is $d_u \times d_u$ and Σ_v is $d_v \times d_v$; their structures are determined by model specifications for population effects and subject random effects.

The initial state is decomposed into

$$\boldsymbol{\alpha}(t_1) = \boldsymbol{a} + A\boldsymbol{\delta} + R_0\boldsymbol{\eta}_0, \quad \boldsymbol{\eta}_0 \sim N(\mathbf{0}, Q_0), \tag{4.10}$$

where $\boldsymbol{a}$ is a known constant vector, A and R_0 are selection matrices, $\boldsymbol{\delta}$ is an unknown vector, and $\boldsymbol{\eta}_0$ is a random vector whose distribution is known. The structures of A, R_0, and Q_0 depend on the model specification.

In our mixed effects state space model, $\boldsymbol{\delta}$ corresponds to the initial population effects which have a diffuse prior, and $\boldsymbol{\eta}_0$ corresponds to the initial individual effects which have a zero mean proper prior. Q_0 may contain unknown parameters that can be estimated by maximum likelihood. As stated earlier in Section (1.3.1), the diffuse initialization of $\boldsymbol{\delta}$ is equivalent to treating it as a fixed quantity and estimating it using maximum likelihood. Our implementation uses the maximum likelihood approach. Note that one can also assign a diffuse prior to $\boldsymbol{\delta}$ and apply the exact initial Kalman filter.

Treating $\boldsymbol{\delta}$ as a fixed quantity, the initial state has mean

$$\mathrm{E}(\boldsymbol{\alpha}(t_1)) = \boldsymbol{a} + A\boldsymbol{\delta} \tag{4.11}$$

and covariance matrix

$$\mathrm{Var}(\boldsymbol{\alpha}(t_1)) = R_0 Q_0 R_0^T. \tag{4.12}$$

For unstructured Σ_ϵ, to ensure the positive definiteness of the estimated Σ_ϵ, we use the LDL^T decomposition for Σ_ϵ:

$$\Sigma_\epsilon = LDL^T, \tag{4.13}$$

and estimate L and D instead, where L is a lower triangular matrix with ones on the diagonal and D is a diagonal matrix with positive diagonal elements. Let $\boldsymbol{l}$ be the parameters in L and $\boldsymbol{d}$ the parameters in D, then $\boldsymbol{l}$ is of length $\frac{q(q-1)}{2}$ and $\boldsymbol{d}$ of length q.

4.1.1 Scenario I: both population effects and subject random effects are local level models

The model is (4.1) with both population effects and subject random effects generated and modeled as local level models. Using the notations stated above, we have

$$d_u = d_v = q, \quad Z_u = Z_v = I_q, \quad T_u = T_v = I_q, \tag{4.14}$$

$$\Sigma_u = \begin{pmatrix} \sigma^2_{u_1} & & 0 \\ & \ddots & \\ 0 & & \sigma^2_{u_q} \end{pmatrix}, \quad \Sigma_v = \begin{pmatrix} \sigma^2_{v_1} & & 0 \\ & \ddots & \\ 0 & & \sigma^2_{v_q} \end{pmatrix}, \tag{4.15}$$

$$A = \begin{pmatrix} I_q \\ 0_{qm\times q} \end{pmatrix}, \quad R_0 = \begin{pmatrix} 0_{q\times qm} \\ I_{qm} \end{pmatrix}, \tag{4.16}$$

$\boldsymbol{\delta}$ is a $q \times 1$ parameter vector, and $Q_0 = \kappa_1 I$; both $\boldsymbol{\delta}$ and κ_1 are estimated by maximum likelihood.

Let $\boldsymbol{p}_u = (\sigma^2_{u_1}, \ldots, \sigma^2_{u_q})^T$ and $\boldsymbol{p}_v = (\sigma^2_{v_1}, \ldots, \sigma^2_{v_q})^T$, the collection of all parameters in the model is $\boldsymbol{\psi} = (\boldsymbol{p}_u, \boldsymbol{p}_v, \boldsymbol{d}, \boldsymbol{l}, \kappa_1, \boldsymbol{\delta})$, which is of length $0.5q^2 + 3.5q + 1$.

The parameter values are set as follows: $\sigma^2_{u1} = 0.7, \sigma^2_{u2} = 0.8, \sigma^2_{v1} = 0.2, \sigma^2_{v2} = 0.9$,

$$\Sigma_\epsilon = \begin{pmatrix} 0.2 & 0.1 \\ 0.1 & 0.8 \end{pmatrix} = LDL',$$

where $D = \text{diag}\{0.2, 0.75\}$ and

$$L = \begin{pmatrix} 1 & 0 \\ 0.5 & 1 \end{pmatrix}.$$

That is, $d_1 = 0.2$, $d_2 = 0.75$, and $l_1 = 0.5$. Parameters for the initial state are set as $\kappa_1 = 1$, and $\boldsymbol{\delta}$ is randomly generated from $N(\mathbf{0}, I_2)$ to make simulated data trajectories different. The parameters of interest are (σ^2_{u1}, σ^2_{u2}, σ^2_{v1}, σ^2_{v2}, d_1, d_2, l_1).

The univariate treatment and the new algorithm produce exactly the same numerical results. The only difference is in computation time. The CPU time is recorded on a node consisting of two 10-core processors (20 total cores) at 2.60GHz each and 128GB RAM. The functions for filtering and likelihood evaluation are coded in Rcpp.

We set $q = 2$, $n = 50$, and $m = 50, 100, 200, 500, 1000$ for the comparison of the univariate treatment and the new algorithm. Figure 4.1 and Table 4.1 report the average CPU time for one round of filtering over 10 replications of the univariate treatment and the new algorithm. One can see that the filtering time for the univariate treatment increases rapidly with the number of subjects: when $m = 1000$, the univariate treatment takes more than an hour, while the new algorithm takes only 0.005 seconds. We also tried out the new algorithm on larger sample sizes with $m = 10^4, 10^5, 10^6$. When $m = 10^5$, the time is still under one second. Even when $m = 10^6$, the new algorithm takes only 6 seconds. The univariate treatment could not be carried out for these larger sample sizes because it would took too long. We estimate the CPU time for the univariate treatment when $m = 10^4$, 10^5, and 10^6 by fitting a linear model with a cubic term of time. The estimates are presented in Table 4.1. The estimated CPU time using the univariate treatment when $m = 10^6$ is over $200,000$ years.

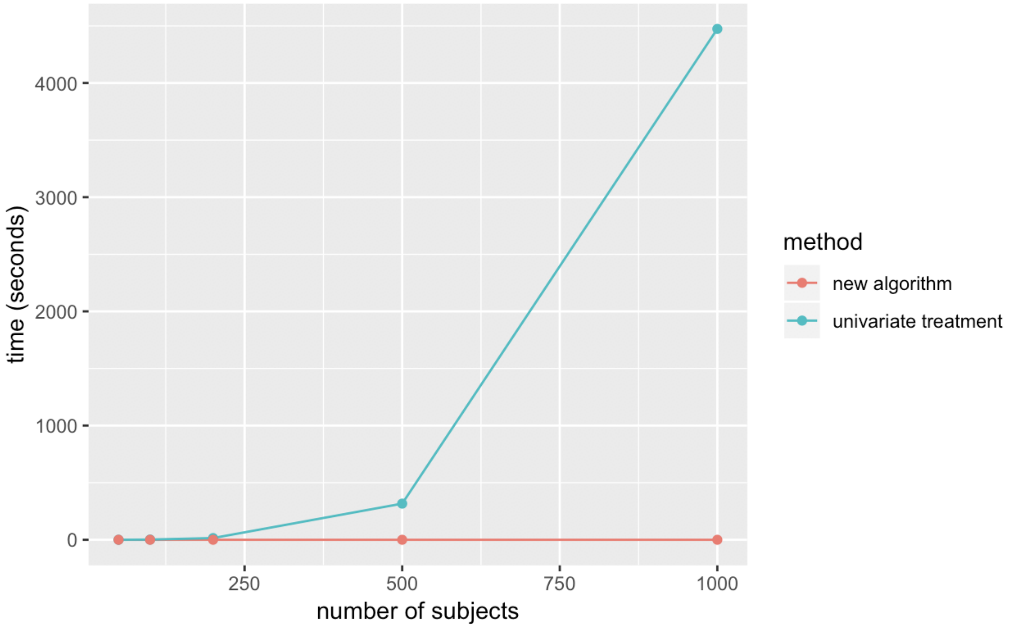

Figure 4.1: Average CPU time in seconds for one round of filtering over 10 replications of the univariate treatment (turquoise) and the new algorithm (red).

m	univariate treatment	new alogrithm
50	0.2140	0.0004
100	1.8380	0.0005
200	14.9410	0.0010
500	316.4980	0.0021
1000	4473.6270	0.0044
10000	est. > 82 days	0.0454
100000	est. > 200 years	0.5787
1000000	est. $> 2 \times 10^5$ years	6.8357

Table 4.1: Average CPU time in seconds for one round of filtering over 10 replications of the univariate treatment and the new algorithm.

Table 4.2 displays mean squared error (MSE), bias, and variance for parameter estimates when $m = 1000$, computed using the new algorithm over 100 replications.

Parameter	True value	MSE	Variance	Bias
σ^2_{u1}	0.7000	0.0168	0.0168	0.0017
σ^2_{u2}	0.8000	0.0193	0.0190	0.0161
σ^2_{v1}	0.2000	0.0000	0.0000	0.0005
σ^2_{v2}	0.9000	0.0002	0.0002	0.0023
d_1	0.2000	0.0000	0.0000	-0.0005
d_2	0.7500	0.0001	0.0001	-0.0010
l_1	0.5000	0.0002	0.0002	-0.0007

Table 4.2: Bias, variance, and MSE of the estimates of parameters.

Figure 4.2 displays the filtering estimation errors for population effects with $m = 100, 500, 1000, 5000, 10000$. The squared error of a population effects estimate is computed as

$$\mathrm{SE} = \frac{1}{n}\sum_{j=1}^{n}[\hat{u}(t_j) - u(t_j)]^2,$$

that is, the average squared error over n time points.

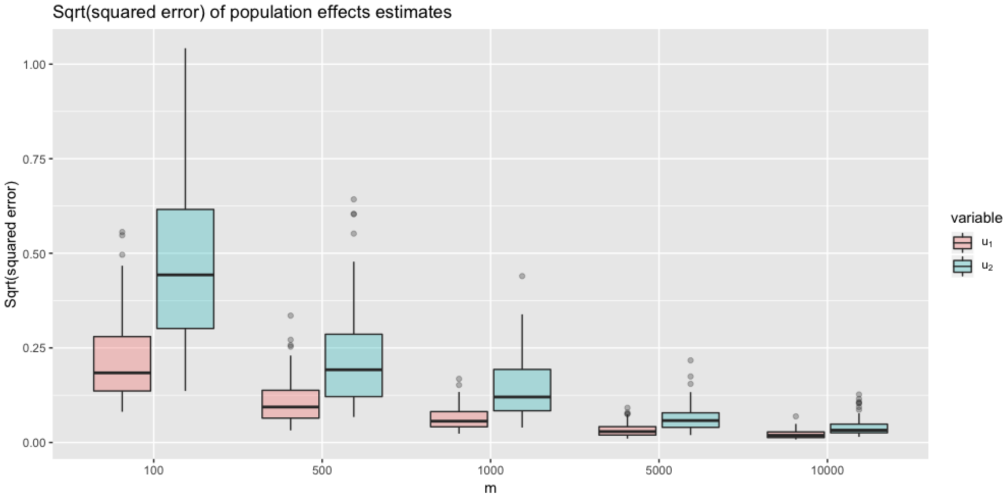

Figure 4.2: Boxplot of $\sqrt{\text{squared error}}$ of population effects filtering estimates.

Table 4.3 is the mean squared error (MSE) of population effects filtering estimates, which is defined as

$$\text{MSE} = \frac{1}{N}\sum_{k=1}^{N}\text{SE}_k,$$

where SE_k is the squared error of population effects estimate from the kth replication. The number of replications, as stated earlier, is $N = 100$.

m	u_1	u_2
100	0.0589	0.2691
500	0.0152	0.0650
1000	0.0051	0.0277
5000	0.0014	0.0053
10000	0.0006	0.0021

Table 4.3: MSE of population effects estimates.

Figure 4.3 displays the boxplots of $\sqrt{\text{squared error}}$ of the filtering estimates for individual random effects with $m = 100, 500, 1000, 5000, 10000$. The squared error of a random effects estimate is defined as

$$\text{SE} = \frac{1}{m}\sum_{i=1}^{m}\{\frac{1}{n}\sum_{j=1}^{n}[\hat{v}_i(t_j) - v_i(t_j)]^2\},$$

that is, the average squared error over all time points of all subjects.

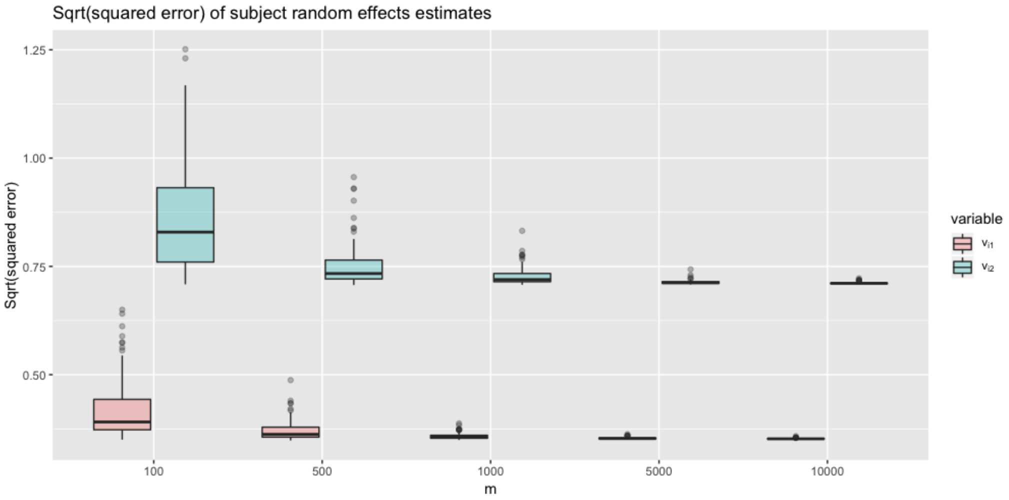

Figure 4.3: Boxplot: $\sqrt{\text{squared error}}$ of individual random effects filtering estimates.

Table 4.4 is the mean squared error (MSE) of subject random effects filtering estimates. MSE is defined as

$$\text{MSE} = \frac{1}{N}\sum_{k=1}^{N}\text{SE}_k,$$

where SE_k is the squared error of random effects estimates from the kth replication. The MSE of random effects do not converge to zero because the bias is not zero, i.e., the random effects estimates are not consistent.

m	v_{i1}	v_{i2}
100	0.1796	0.7597
500	0.1380	0.5673
1000	0.1283	0.5301
5000	0.1248	0.5089
10000	0.1240	0.5061

Table 4.4: MSE of random effects estimates.

Figure 4.4 displays the boxplots of $\sqrt{\text{squared error}}$ of parameter estimates for $m = 100, 500, 1000, 5000$, and 10000, with $q = 2$ and $n = 50$.

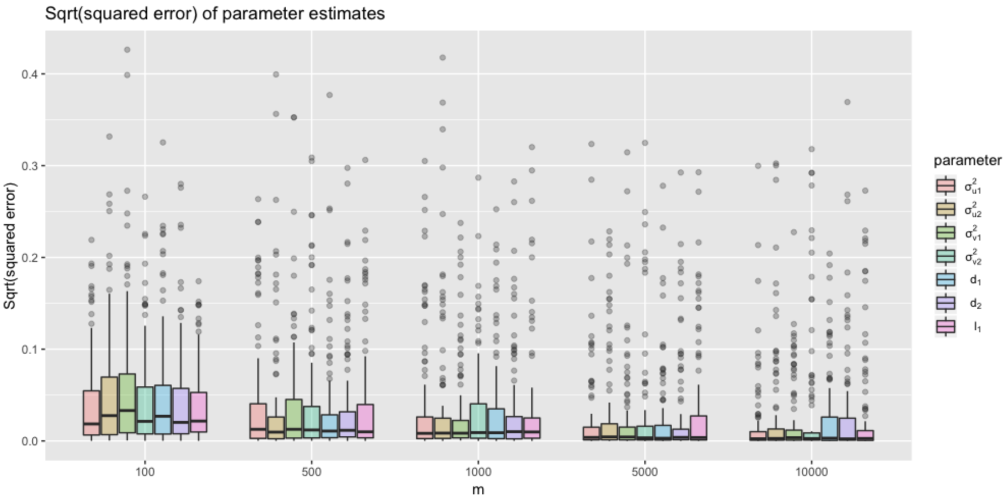

Figure 4.4: Boxplot: $\sqrt{\text{squared error}}$ of parameter estimates.

Table 4.5 displays the MSE of parameter estimates.

	m				
Parameter	100	500	1000	5000	10000
σ^2_{u1}	0.015988	0.018499	0.016801	0.012871	0.010193
σ^2_{u2}	0.020446	0.021263	0.019262	0.015862	0.017633
σ^2_{v1}	0.000093	0.000018	0.000010	0.000002	0.000001
σ^2_{v2}	0.002026	0.000321	0.000163	0.000032	0.000015
d_1	0.000067	0.000015	0.000006	0.000001	0.000001
d_2	0.001226	0.000232	0.000105	0.000026	0.000011
l_1	0.002601	0.000357	0.000215	0.000043	0.000025

Table 4.5: MSE of parameter estimates.

Figure 4.5 are selected filtering estimates of population effects for the first variable, corresponding to 5th and 95th percentiles of MSE when $m = 1000$.

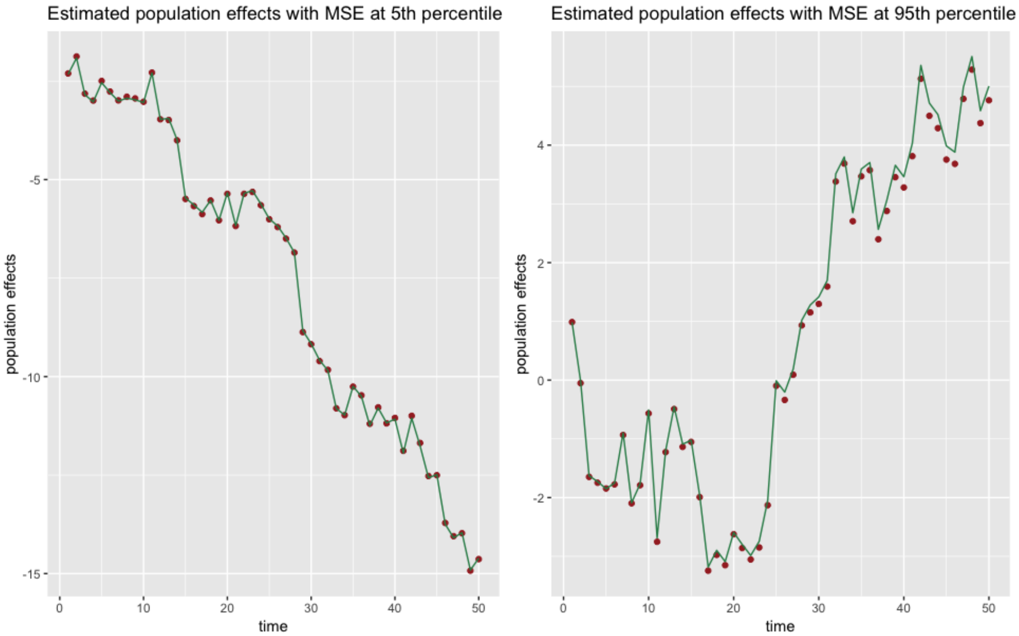

Figure 4.5: Selected filtering estimates of population effects for the first variable, corresponding to 5th and 95th percentiles of MSE when $m = 1000$. Brown dots: true population effects $u_1(t)$, green solid line: the filtering estimates of $u_1(t)$.

Figure 4.6 shows some randomly selected examples of the filtering estimates of individual random effects for the first variable when $m = 1000$.

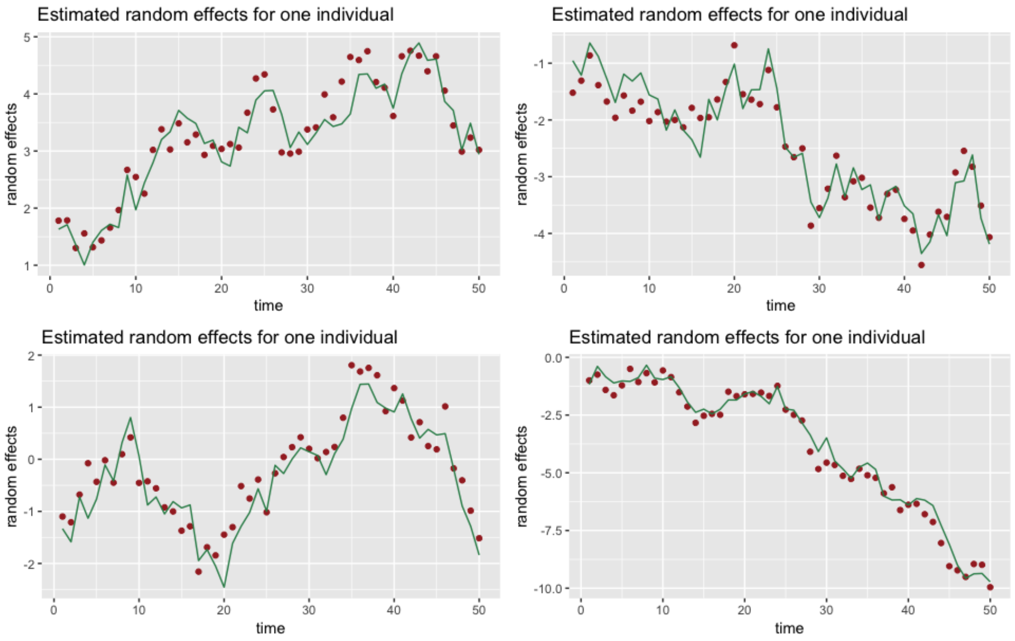

Figure 4.6: Randomly selected examples of the filtering estimates of individual random effects for the first variable when $m = 1000$. Brown dots: true subject random effects $v_{i1}(t)$, green solid line: the filtering estimates of $v_{i1}(t)$.

4.1.2 Scenario II: population effects are cubic splines and subject random effects are OU processes

We consider model (4.1) where population effects are cubic splines and sub random effects are OU processes. We use an unstructured Σ_ϵ to characterize the correlation between multiple variables. Since the random effects are assumed to have mean zero, the parameter μ in (2.6) equals zero. We have

$$d_u = 2q, \quad d_v = q, \quad Z_u = I_q \otimes (1 \quad 0), \quad Z_v = I_q, \tag{4.17}$$

$$T_u = I_q \otimes \begin{pmatrix} 1 & \triangle t \\ 0 & 1 \end{pmatrix}, \quad T_v = \begin{pmatrix} e^{-\xi_1 \triangle t} & & 0 \\ & \ddots & \\ 0 & & e^{-\xi_q \triangle t} \end{pmatrix}, \tag{4.18}$$

$$\Sigma_u = \begin{pmatrix} \zeta_1 \Lambda & & 0 \\ & \ddots & \\ 0 & & \zeta_q \Lambda \end{pmatrix}, \tag{4.19}$$

where

$$\Lambda = \begin{pmatrix} \triangle t^3/3 & \triangle t^2/2 \\ \triangle t^2/2 & \triangle t \end{pmatrix}, \tag{4.20}$$

$$\Sigma_v = \begin{pmatrix} \frac{\nu_1^2}{2\xi_1}(1 - e^{-2\xi_1 \triangle t}) & & 0 \\ & \ddots & \\ 0 & & \frac{\nu_q^2}{2\xi_q}(1 - e^{-2\xi_q \triangle t}) \end{pmatrix}, \tag{4.21}$$

$$A = \begin{pmatrix} I_{2q} \\ 0_{qm \times 2q} \end{pmatrix}, \quad R_0 = \begin{pmatrix} 0_{2q \times qm} \\ I_{qm} \end{pmatrix}, \tag{4.22}$$

$\boldsymbol{\delta}$ is a $2q \times 1$ parameter vector, and

$$Q_0 = I_m \otimes \begin{pmatrix} \frac{\nu_1^2}{2\xi_1} & & 0 \\ & \ddots & \\ 0 & & \frac{\nu_q^2}{2\xi_q} \end{pmatrix} \tag{4.23}$$

since OU process is a stationary AR(1) process when $\mu = 0$. Let $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_q)^T$, $\boldsymbol{p}_\nu = (\nu_1^2, \ldots, \nu_q^2)^T$, $\boldsymbol{p}_\xi = (\xi_1, \ldots, \xi_q)^T$, the collection of all parameters is $\boldsymbol{\psi} = (\boldsymbol{\zeta}, \boldsymbol{p}_\nu, \boldsymbol{p}_\xi, \boldsymbol{d}, \boldsymbol{l}, \boldsymbol{\delta})$, which is of length $0.5q^2 + 5.5q$. We set $q = 2$, $n = 50$, and $m = 50, 100, 200, 500, 1000$ for the comparison of the univariate treatment and the new algorithm. The parameter values are: population effects smoothing parameters $\zeta_1 = 0.4$, $\zeta_2 = 0.6$; OU process parameters $\nu_1^2 = 2$, $\nu_2^2 = 1$, $\xi_1 = 0.9$, $\xi_2 = 0.5$; the observation random error covariance matrix

$$\Sigma_\epsilon = \begin{pmatrix} 0.2 & 0.1 \\ 0.1 & 0.8 \end{pmatrix} = LDL', \tag{4.24}$$

where $D = \text{diag}\{0.2, 0.75\}$ and

$$L = \begin{pmatrix} 1 & 0 \\ 0.5 & 1 \end{pmatrix},$$

that is, $d_1 = 0.2$, $d_2 = 0.75$, and $l_1 = 0.5$; and the initial population state mean $\boldsymbol{\delta}$ is randomly generated from $N(\mathbf{0}, I_4)$ to make simulated data trajectories different from each other. The parameters of interest are (ν_1^2, ν_2^2, ξ_1, ξ_2, d_1, d_2, l_1).

Figure 4.7 and Table 4.6 report the average CPU time to perform one round of filtering over 10 replications of the univariate treatment and the new algorithm. The results are very similar to those in the local level model.

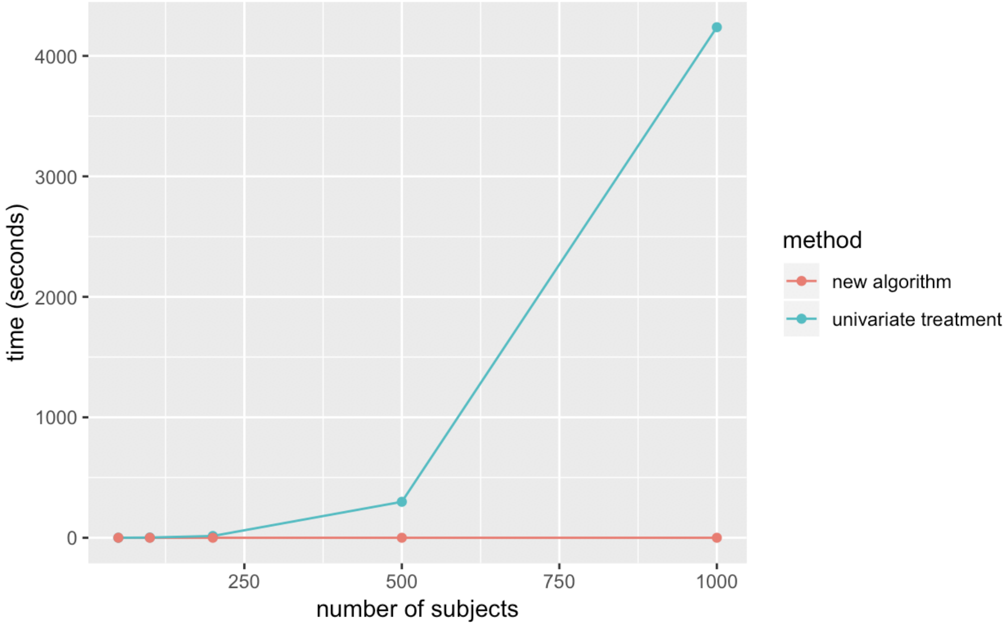

Figure 4.7: Average CPU time in seconds for one round of filtering over 10 replications of the univariate treatment (turquoise) and the new algorithm (red).

m	univariate treatment	new alogrithm
50	0.2310	0.0010
100	1.8470	0.0010
200	14.6410	0.0020
500	298.5740	0.0020
1000	4238.0650	0.0050
10000	est. $>$ 78 days	0.0470
100000	est. $>$ 227 years	0.5530
1000000	est. $> 2 \times 10^5$ years	6.0790

Table 4.6: Average CPU time in seconds for one round of filtering over 10 replications of the univariate treatment and the new algorithm.

Table 4.7 displays the MSE, bias, and variance for parameter estimates when $m = 1000$, computed using the new algorithm.

Parameter	True_value	MSE	Variance	Bias
ν_1^2	2.0000	0.0087	0.0087	0.0065
ν_2^2	1.0000	0.0024	0.0023	0.0110
ξ_1	0.9000	0.0006	0.0006	0.0022
ξ_2	0.5000	0.0003	0.0002	0.0032
d_1	0.2000	0.0006	0.0006	-0.0003
d_2	0.7500	0.0005	0.0005	-0.0051
l_1	0.5000	0.0051	0.0051	0.0055

Table 4.7: Parameter estimates when $m = 1000$.

Figure 4.8 displays the boxplots of the filtering estimation errors for population effects.

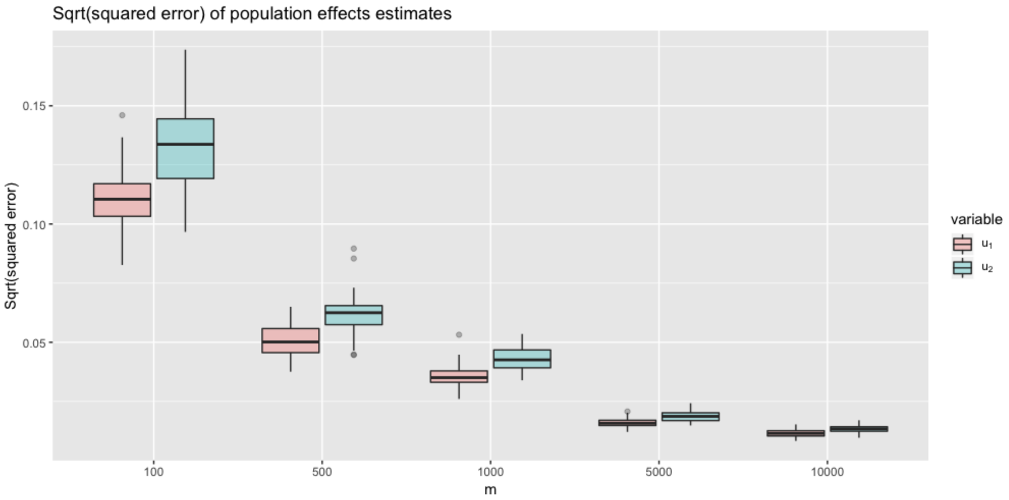

Figure 4.8: Boxplot of $\sqrt{\text{squared error}}$ of population effects filtering estimates.

Table 4.8 is the mean squared error (MSE) of the filtering estimates for population effects.

m	u_1	u_2
100	0.0123	0.0177
500	0.0026	0.0039
1000	0.0013	0.0019
5000	0.0003	0.0004
10000	0.0001	0.0002

Table 4.8: MSE of population effects filtering estimates.

Figure 4.9 displays the boxplots of the filtering estimation errors for individual random effects with $m = 100, 500, 1000, 5000, 10000$, $q = 2$ and $n = 50$.

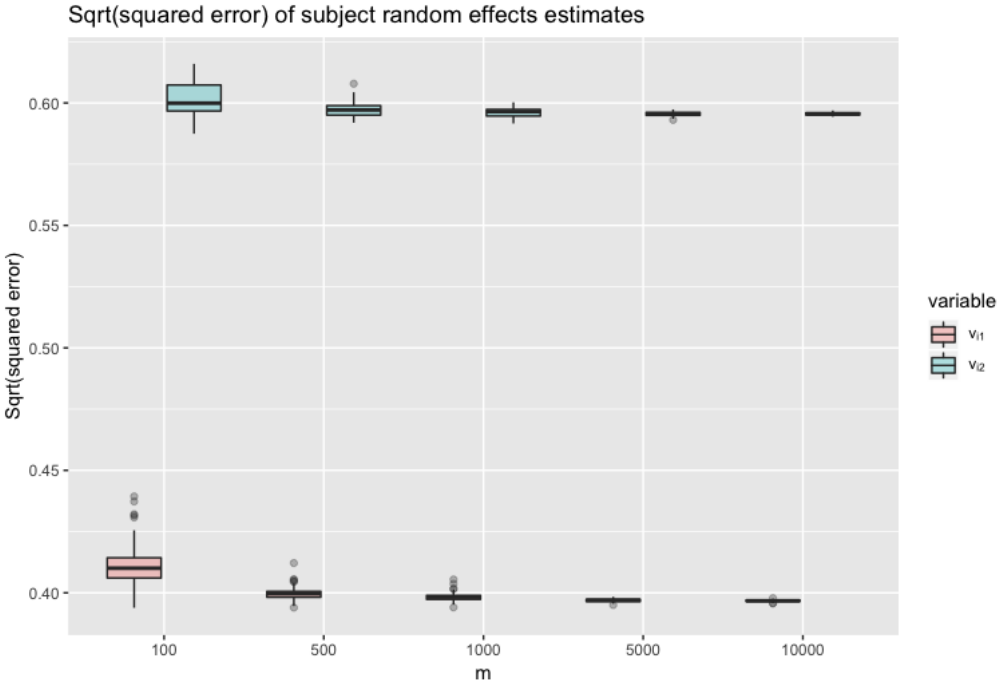

Figure 4.9: Boxplot: $\sqrt{\text{squared error}}$ of individual random effects filtering estimates.

Table 4.9 is the mean squared error (MSE) of subject random effects filtering estimates.

m	v_{i1}	v_{i2}
100	0.1734	0.3995
500	0.1636	0.3952
1000	0.1625	0.3944
5000	0.1615	0.3936
10000	0.1614	0.3937

Table 4.9: MSE of random effects filtering estimates.

Figure 4.10 displays the boxplots of estimation errors of parameter estimates for $m = 100, 500, 1000, 5000, 10000$.

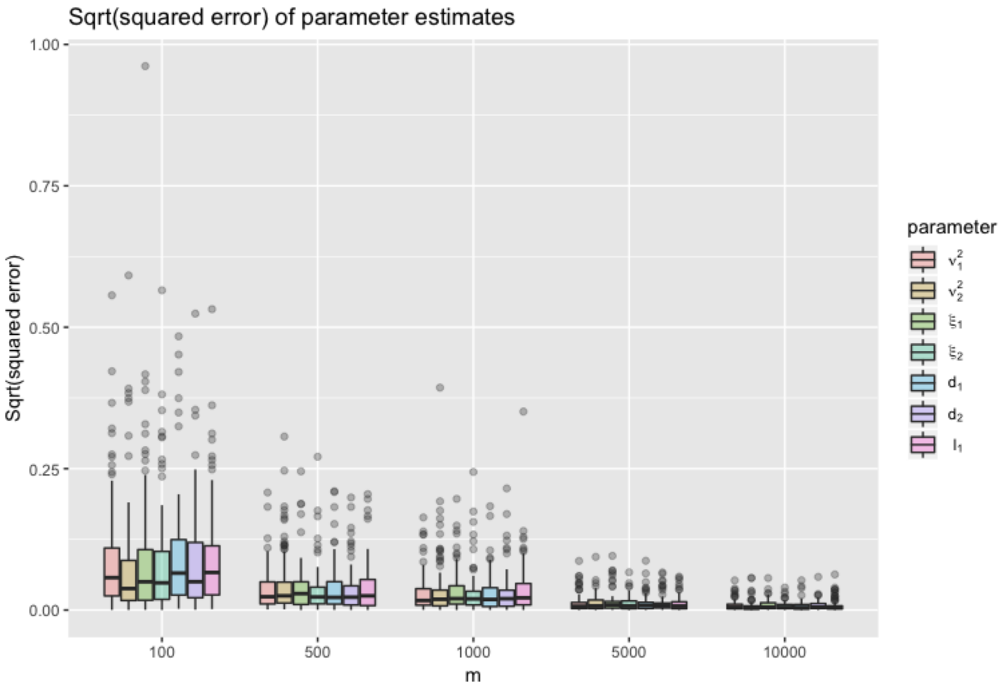

Figure 4.10: Boxplot: $\sqrt{\text{squared error}}$ of parameter estimates.

Table 4.10 displays the MSE of parameter estimates.

	m				
Parameter	100	500	1000	5000	10000
ν_1^2	0.0497	0.0104	0.0087	0.0011	0.0004
ν_2^2	0.0200	0.0045	0.0024	0.0004	0.0002
ξ_1	0.0036	0.0007	0.0006	0.0001	0.0000
ξ_2	0.0023	0.0004	0.0003	0.0000	0.0000
d_1	0.0040	0.0007	0.0006	0.0001	0.0000
d_2	0.0037	0.0008	0.0005	0.0001	0.0000
l_1	0.0373	0.0054	0.0051	0.0006	0.0002

Table 4.10: MSE of parameter estimates.

Figure 4.11 displays selected filtering estimates of population effects for the first variable, corresponding to the 5th and 95th percentiles of MSE when $m = 1000$.

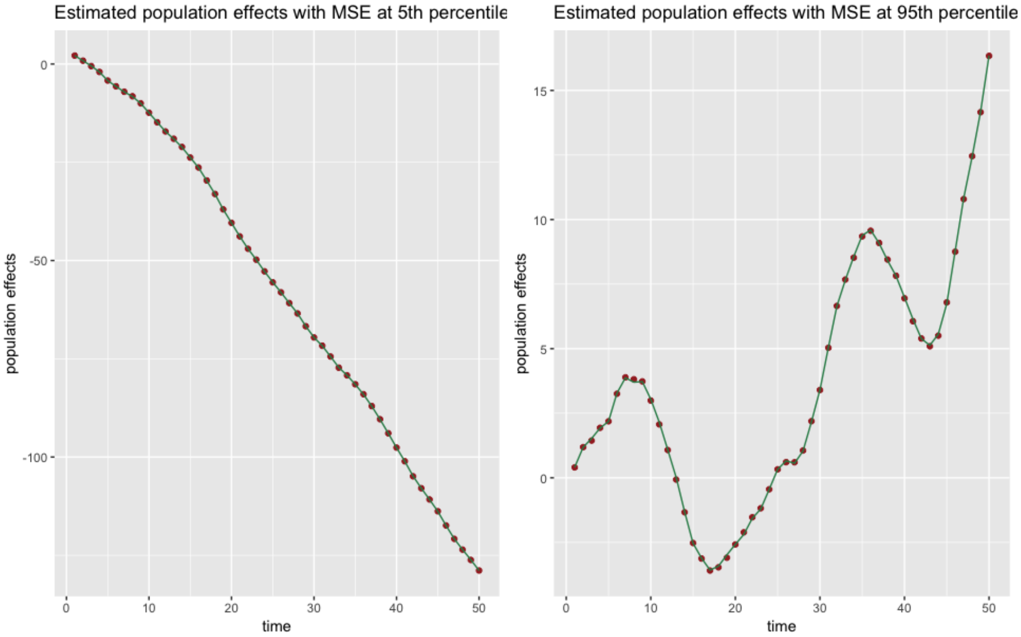

Figure 4.11: Selected filtering estimates of population effects for the first variable, corresponding to the 5th and 95th percentiles of MSE when $m = 1000$. Brown dots: true values of $u_1(t_j)$, green solid line: the filtering estimates of $u_1(t_j)$.

Figure 4.12 shows some randomly selected examples of the filtering estimates of individual random effects for the first variable when $m = 1000$.

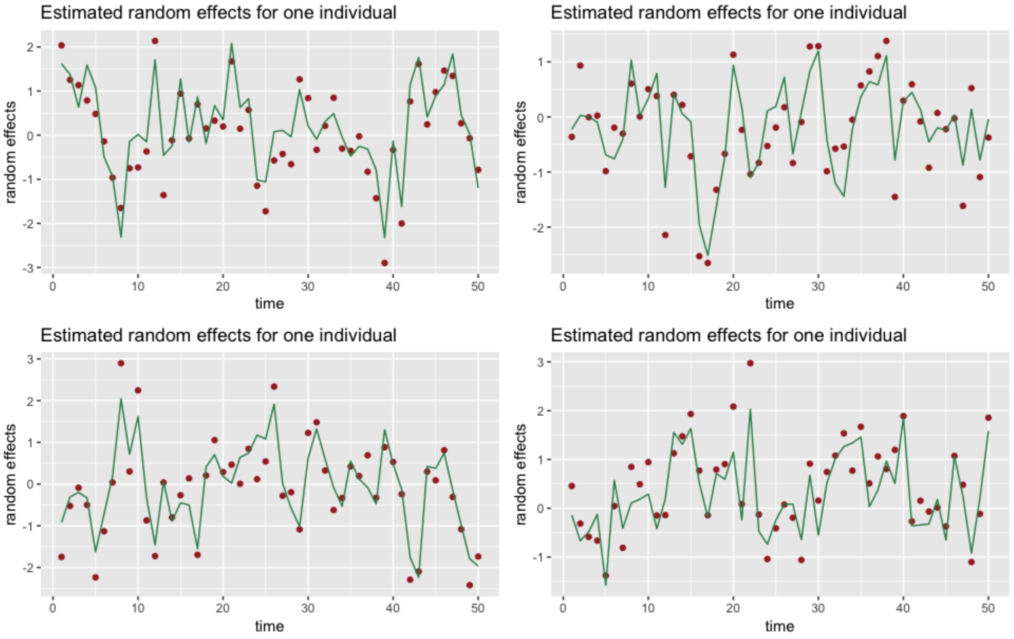

Figure 4.12: Randomly selected examples of the filtering estimates of individual random effects for the first variable when $m = 1000$. Brown dots: true values of $v_{i1}(t)$, green solid line: the filtering estimates of $v_{i1}(t)$.

4.1.3 Scenario III: both population effects and subject random effects are cubic splines

In model (4.1), when both population effects and subject random effects are modeled as cubic splines, we have

$$d_u = d_v = 2q, \quad Z_u = Z_v = I_q \otimes (1 \quad 0), \tag{4.25}$$

$$T_u = T_v = I_q \otimes \begin{pmatrix} 1 & \triangle t \\ 0 & 1 \end{pmatrix}, \tag{4.26}$$

$$\Sigma_u = \begin{pmatrix} \zeta_1\Lambda & & 0 \\ & \ddots & \\ 0 & & \zeta_q\Lambda \end{pmatrix}, \quad \Sigma_v = \begin{pmatrix} \lambda_1\Lambda & & 0 \\ & \ddots & \\ 0 & & \lambda_q\Lambda \end{pmatrix}, \tag{4.27}$$

where Λ is the same as in (4.20),

$$A = \begin{pmatrix} I_{2q} \\ 0_{2qm\times 2q} \end{pmatrix}, \quad R_0 = \begin{pmatrix} 0_{2q\times 2qm} \\ I_{2qm} \end{pmatrix}, \tag{4.28}$$

$\boldsymbol{\delta}$ is a $2q \times 1$ parameter vector, and $Q_0 = \kappa_1 I_{2qm}$, where κ_1 is a parameter. Here we use Σ_ϵ to model the correlation between multiple variables.

Let $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_q)^T$, $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_q)^T$, the collection of all parameters is $\boldsymbol{\psi} = (\boldsymbol{\zeta}, \boldsymbol{\lambda}, \boldsymbol{d}, \boldsymbol{l}, \kappa_1, \boldsymbol{\delta})$, which is of length $0.5q^2 + 4.5q + 1$. We set $q = 2$, $n = 50$, and $m = 50, 100, 200, 500, 1000$ to compare the univariate treatment and the new algorithm. The parameter values are: population effects smoothing parameters $\zeta_1 = 0.4$, $\zeta_2 = 0.6$; subject random effects smoothing parameters $\lambda_1 = 0.5$, $\lambda_2 = 0.2$; Σ_ϵ is the same as in (4.29), thus $d_1 = 0.2$, $d_2 = 0.75$. Initial subject random effects state variation is $\kappa_1 = 1$, and the initial population state mean $\boldsymbol{\delta}$ is randomly generated from $N(\mathbf{0}, I_4)$ to make population trajectories different for different data sets. The parameters of interest are (d_1, d_2, l_1).

Figure 4.13 and Table 4.11 report average CPU time in seconds for one round of filtering over 10 replications of the univariate treatment (turquoise) and the new algorithm (red), with $m = 50, 100, 200, 500, 1000$. The univariate treatment for the cubic spline model takes significantly longer time than for the local level model and the OU process model, since the cubic spline model is more complex. When $m = 1000$, the univariate treatment takes over 6 hours to do one round of filtering, while the new algorithm remains at 0.005 seconds.

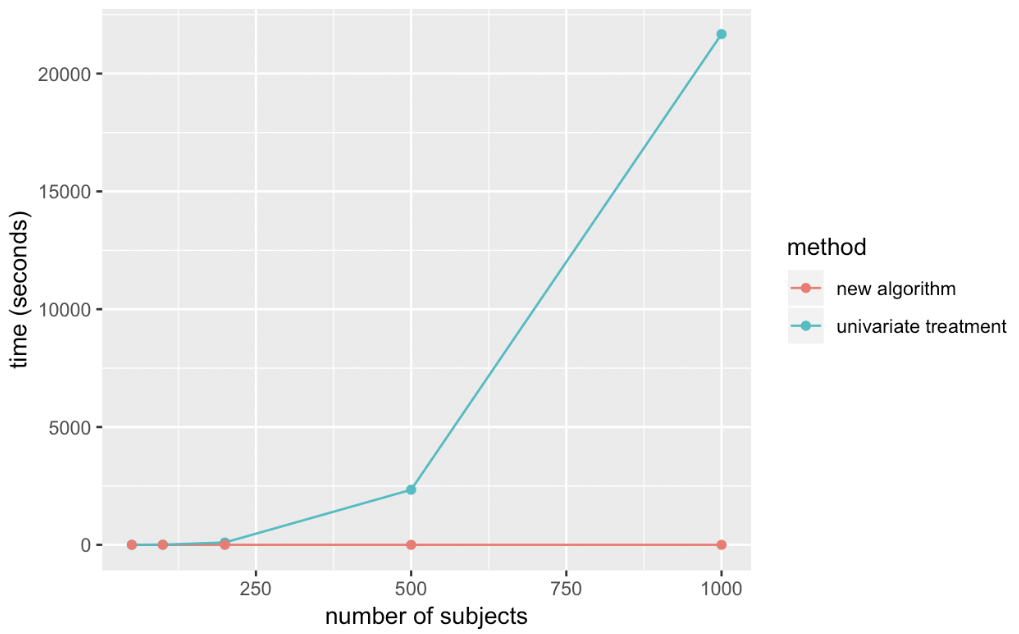

Figure 4.13: Average CPU time in seconds for one round of filtering over 10 replications of the univariate treatment (turquoise) and the new algorithm (red).

m	univariate treatment	new algorithm
50	1.1650	0.0010
100	9.0140	0.0010
200	99.7060	0.0020
500	2338.6310	0.0020
1000	21674.1180	0.0050
10000	est. $>$ 286 days	0.0560
100000	est. $>$ 796 years	0.6540
1000000	est. $> 7 \times 10^5$ years	7.5340

Table 4.11: Average CPU time in seconds for one round of filtering over 10 replications of the univariate treatment and the new algorithm.

Table 4.12 displays the MSE, bias, and variance for parameter estimates when $m = 1000$, computed using the new algorithm.

Parameter	True_value	MSE	Variance	Bias
d_1	0.200000	0.000004	0.000004	0.000039
d_2	0.750000	0.000050	0.000048	0.001324
l_1	0.500000	0.000133	0.000132	0.000364

Table 4.12: Parameter estimates when $m = 1000$.

Figure 4.14 displays the boxplots of the filtering estimation errors for population effects with $m = 100, 500, 1000, 5000, 10000$.

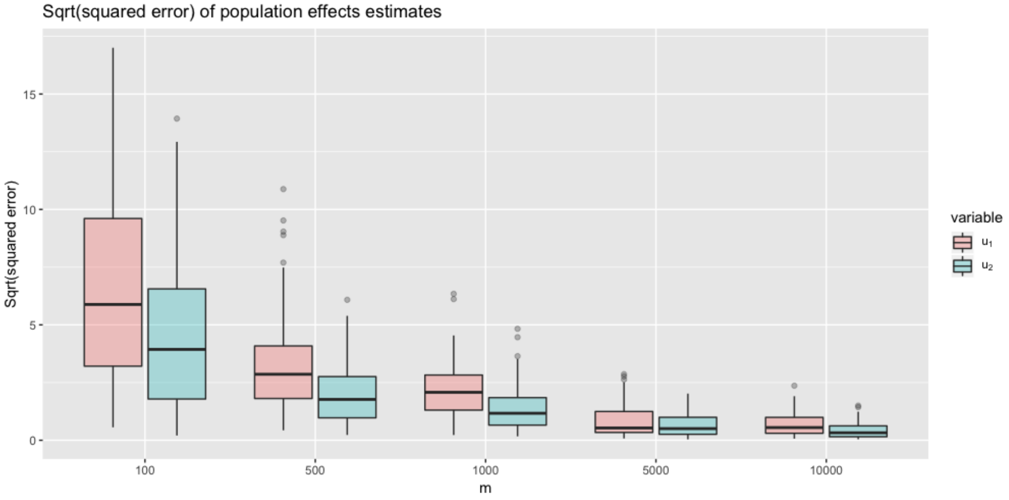

Figure 4.14: Boxplot of $\sqrt{\text{squared error}}$ of population effects filtering estimates.

Table 4.13 displays the mean squared error (MSE) of population effects filtering estimates.

m	u_1	u_2
100	63.5702	29.2267
500	14.1930	5.6186
1000	6.1664	2.7774
5000	1.0455	0.6033
10000	0.7226	0.2741

Table 4.13: MSE of population effects filtering estimates.

Figure 4.15 displays the boxplots of filtering estimation errors for individual random effects with $m = 100, 500, 1000, 5000, 10000$.

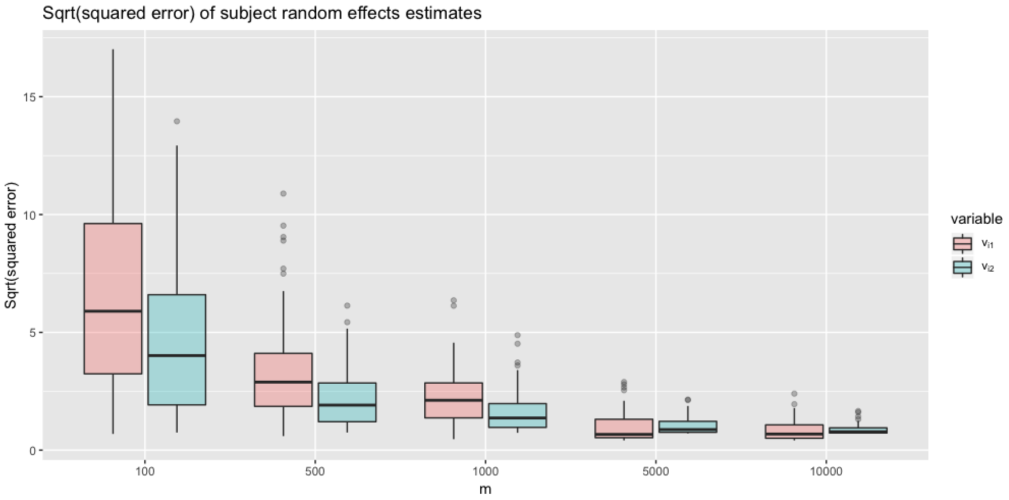

Figure 4.15: Boxplot: $\sqrt{\text{squared error}}$ of individual random effects filtering estimates.

Table 4.14 is the mean squared error (MSE) of subject random filtering estimates.

m	v_{i1}	v_{i2}
100	63.7375	29.7168
500	14.3551	6.1230
1000	6.3307	3.2837
5000	1.2095	1.1100
10000	0.8870	0.7805

Table 4.14: MSE of random effects filtering estimates.

Figure 4.16 displays the boxplots of the estimation errors for the parameters with $m = 100, 500, 1000, 5000, 10000$.

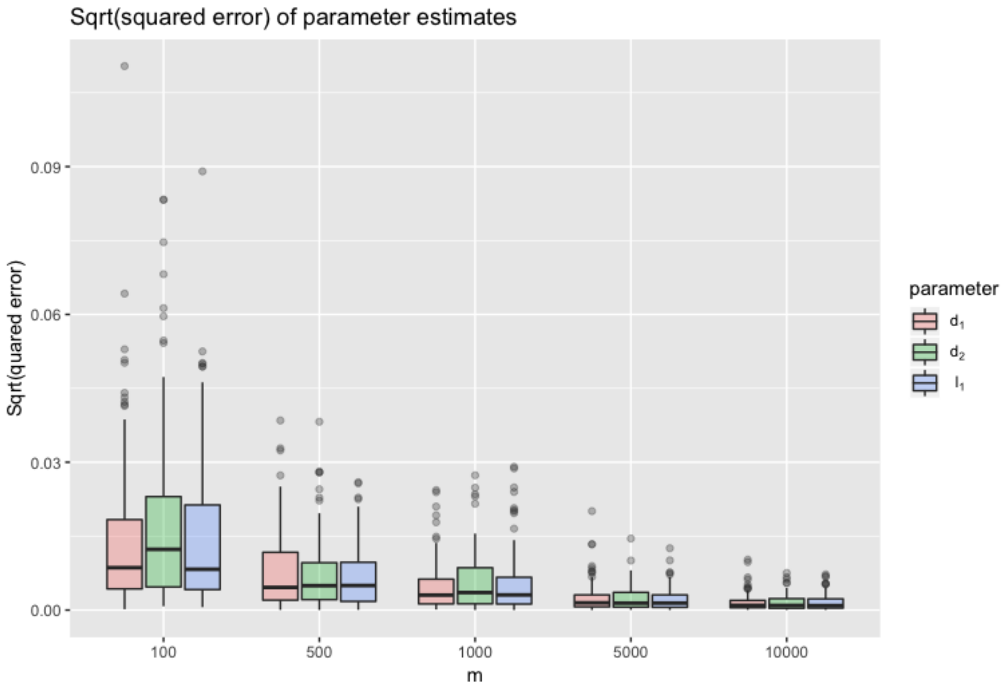

Figure 4.16: Boxplot: $\sqrt{\text{squared error}}$ of parameter estimates.

Table 4.15 displays the MSE of parameter estimates.

	m				
Parameters	100	500	1000	5000	10000
d_1	0.0000348	0.0000074	0.0000040	0.0000008	0.0000005
d_2	0.0003433	0.0000664	0.0000500	0.0000088	0.0000041
l_1	0.0013213	0.0002470	0.0001325	0.0000298	0.0000117

Table 4.15: MSE of parameter estimates.

Figure 4.17 are selected filtering estimates of population effects corresponding to the 5th and 95th percentiles of MSE for the first variable when $m = 1000$.

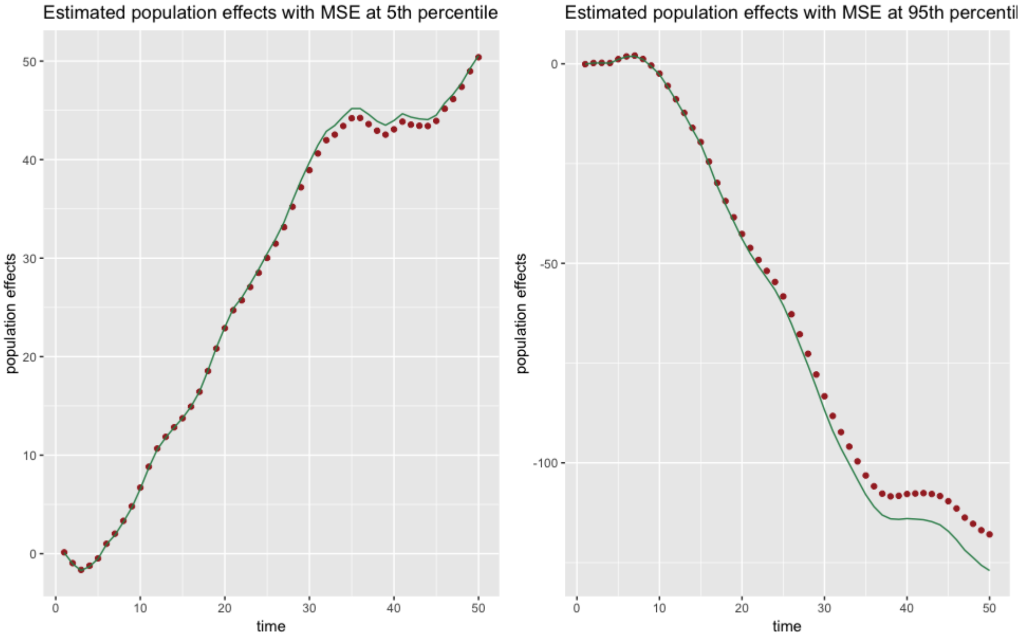

Figure 4.17: Selected filtering estimates of population effects corresponding to the 5th and 95th percentiles of MSE for the first variable when $m = 1000$. Brown dots: true values of $u_1(t)$, green solid line: the filtering estimates of $u_1(t)$.

Figure 4.18 shows some randomly selected examples of the filtering estimates of individual random effects for the first variable when $m = 1000$.

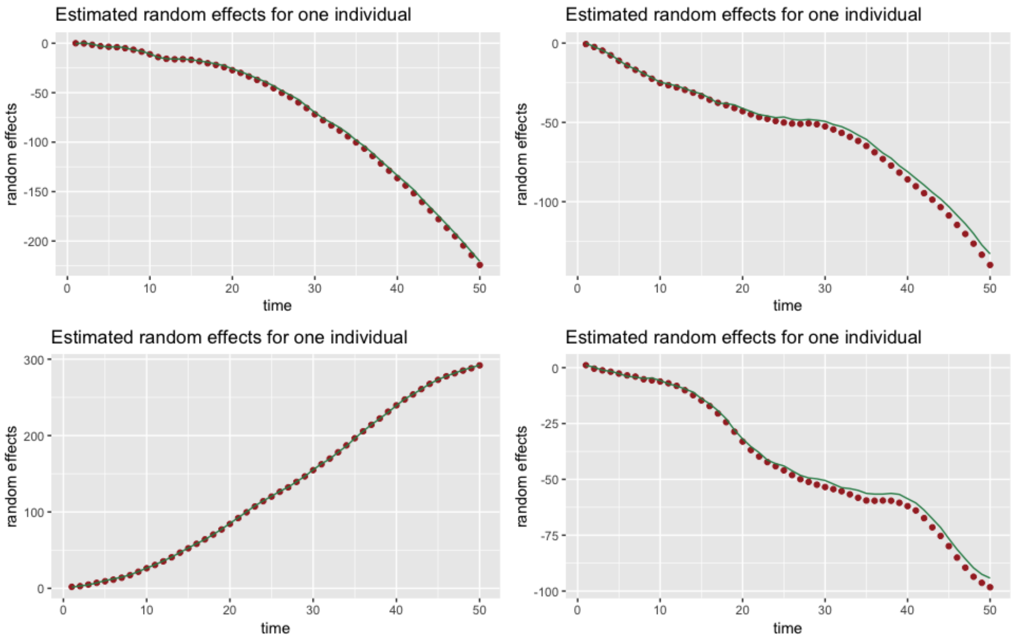

Figure 4.18: Randomly selected examples of the filtering estimates of individual random effects for the first variable when $m = 1000$. Brown dots: true values of $v_{i1}(t)$, green solid line: the filtering estimates of $v_{i1}(t)$.

4.2 Joint Modeling of Longitudinal Data and Survival Data

The longitudinal data are generated by model (4.1), with the number of longitudinal variables $q = 2$ and the number of covariates $p = 2$. That is, $\boldsymbol{y}_i(t_j) = (y_{i1}(t_j), y_{i2}(t_j))^T$ and $\boldsymbol{x}_i(t_j) = (x_{i1}(t_j), x_{i2}(t_j))^T$. The population effects are modeled by cubic splines, and the subject random effects are modeled by OU processes. We have

$$\boldsymbol{u}(t_j) = \begin{pmatrix} u_1(t_j) \\ u_1'(t_j) \\ u_2(t_j) \\ u_2'(t_j) \end{pmatrix} \quad \text{and} \quad \boldsymbol{v}_i(t_j) = \begin{pmatrix} v_{i1}(t_j) \\ v_{i2}(t_j) \end{pmatrix}.$$

The survival status at time t, $z_i(t)$, are generated by (3.1) and (3.2). In (3.2), $\boldsymbol{\gamma}_1 = (\gamma_{11}, \gamma_{12})^T$ is the coefficient vector for $\boldsymbol{x}_i(t_j)$. Given

$$(\boldsymbol{u}^T(t_j), \boldsymbol{v}_i^T(t_j))^T = \begin{pmatrix} u_1(t_j) \\ u_1'(t_j) \\ u_2(t_j) \\ u_2'(t_j) \\ v_{i1}(t_j) \\ v_{i2}(t_j) \end{pmatrix},$$

we set $D = (I_2 \otimes (1 \quad 0), \quad I_2)$, hence

$$D \begin{pmatrix} \boldsymbol{u}(t_j) \\ \boldsymbol{v}_i(t_j) \end{pmatrix} = \begin{pmatrix} u_1(t_j) + v_{i1}(t_j) \\ u_2(t_j) + v_{i2}(t_j) \end{pmatrix}$$

is the vector of the latent longitudinal values for subject i. The corresponding coefficient vector is $\boldsymbol{\gamma}_2 = (\gamma_{21}, \gamma_{22})^T$. We set $\gamma_0 = 0.5, \gamma_{11} = 0.1, \gamma_{12} = -0.2, \gamma_{21} = -0.5, \gamma_{22} = 0.3$. The parameter values in the longitudinal model are the same as in Scenario II in Section 4.1.2: population effects smoothing parameters $\zeta_1 = 0.4$, $\zeta_2 = 0.6$; OU process parameters $\nu_1^2 = 2$, $\nu_2^2 = 1$, $\xi_1 = 0.9$, $\xi_2 = 0.5$; the observation random error covariance matrix

$$\Sigma_\epsilon = \begin{pmatrix} 0.2 & 0.1 \\ 0.1 & 0.8 \end{pmatrix} = LDL', \tag{4.29}$$

where $D = \text{diag}\{0.2, 0.75\}$ and

$$L = \begin{pmatrix} 1 & 0 \\ 0.5 & 1 \end{pmatrix},$$

that is, $d_1 = 0.2$, $d_2 = 0.75$, and $l_1 = 0.5$; and the initial population state mean $\boldsymbol{\delta}$ is randomly generated from $N(\mathbf{0}, I_4)$ to make simulated data trajectories differentzz from each other. The censoring probability is set to 2% for all subjects from time t_2 to t_{50}. The censoring probability at time t_1 is 0. The parameters of interest are (ν_1^2, ν_2^2, ξ_1, ξ_2, d_1, d_2, l_1, γ_{11}, γ_{12}, γ_{21}, γ_{22}).

Table 4.16 displays the MSE, bias, and variance of parameter estimates with $m = 1000$ and $n = 50$. Note that for some simulated data sets, all subjects died/dropped out before the 50th time point.

Parameter	True_value	MSE	Variance	Bias
ν_1^2	2.0000	0.0047	0.0046	0.0096
ν_2^2	1.0000	0.0061	0.0059	0.0133
ξ_1	0.9000	0.0015	0.0014	0.0091
ξ_2	0.5000	0.0041	0.0039	0.0115
d_1	0.2000	0.0002	0.0002	0.0029
d_2	0.7500	0.0013	0.0013	0.0021
l_1	0.5000	0.0067	0.0067	0.0052
γ_{21}	-0.5000	0.0037	0.0037	0.0041
γ_{22}	0.3000	0.0031	0.0031	0.0034
γ_0	0.5000	0.0056	0.0048	0.0290
γ_{11}	0.1000	0.0049	0.0043	0.0236
γ_{12}	-0.2000	0.0096	0.0095	-0.0111

Table 4.16: Parameter estimates of the joint model.

Figure 4.19 and Table 4.17 report the average CPU time for computing the joint likelihood over 10 replications, with $n = 50$, $q = 2$, and $m = 100, 500, 1000, 5000, 10000$. The CPU time is linear in m, consistent with what we describe in Algorithm 1. The computation time was recorded from full data sets with no early dropouts, i.e., all subjects exist to the end time point. The survival likelihood at each time point is approximated using (3.24) by drawing 100 samples of $\boldsymbol{\alpha}$ from $p(\boldsymbol{\alpha}^{A_j \cup B_j}(t_{j-1}) | \boldsymbol{Y}^{A_{1:j-1}}(t_{1:j-1}))$, as described at the end of Section 3.2. Also, for each drawn $\boldsymbol{\alpha}$, we add an antithetic variable $\breve{\boldsymbol{\alpha}} = 2\boldsymbol{a} - \boldsymbol{\alpha}$, as described in Section 3.2, where $\boldsymbol{a}$ is the mean vector of the multivariate normal distribution from which $\boldsymbol{\alpha}$ is drawn.

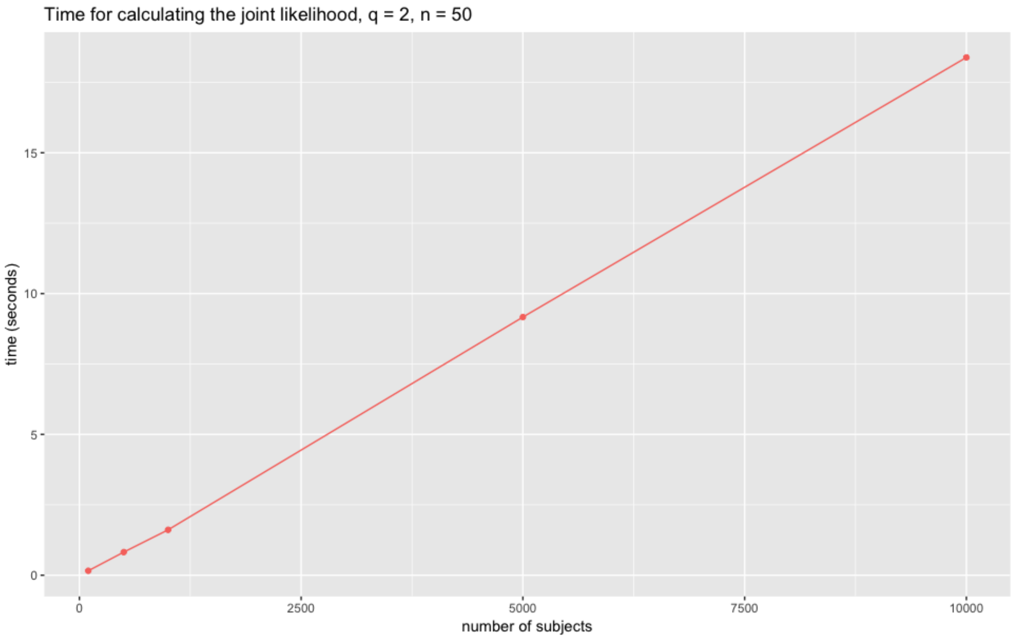

Figure 4.19: Average CPU time in seconds for calculating the joint likelihood over 10 replications.

Table 4.17 displays the corresponding values.

m	time
100	0.1569
500	0.8198
1000	1.6106
5000	9.1662
10000	18.3859

Table 4.17: Average CPU time in seconds for calculating the joint likelihood over 10 replications.

The filtering estimation errors of the population effects are calculated only at time points where there is at least one subject alive. The estimation errors for subject random effects are computed in a similar manner. We consider sample sizes $m = 100, 500, 1000, 5000$.

Figure 4.20 displays the boxplots of the estimation errors of population effects filtering estimates.

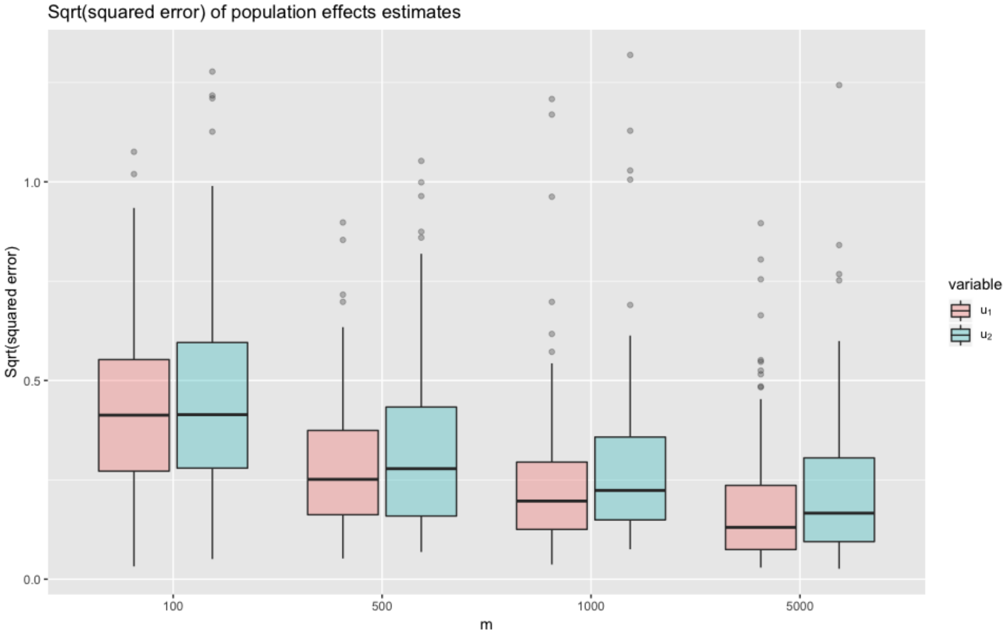

Figure 4.20: $\sqrt{\text{squared error}}$ of the population effects filtering estimates.

Table 4.18 displays the MSE of population effects filtering estimates.

m	u_1	u_2
100	0.2409	0.2847
500	0.1120	0.1639
1000	0.1040	0.1379
5000	0.0696	0.0921

Table 4.18: MSE of population effects filtering estimates.

Figure 4.21 displays the estimation errors of the individual random effects filtering estimates.

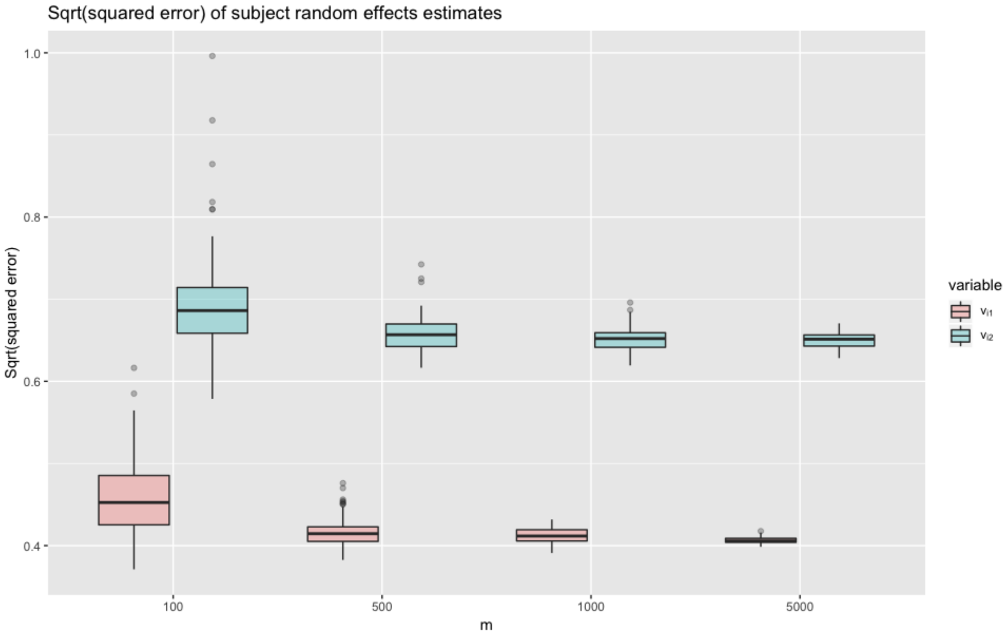

Figure 4.21: $\sqrt{\text{squared error}}$ of individual random effects filtering estimates.

Table 4.19 displays the MSE of individual random effects filtering estimates.

m	v_{i1}	v_{i2}
100	0.2143	0.4846
500	0.1746	0.4322
1000	0.1703	0.4238
5000	0.1653	0.4221

Table 4.19: MSE of individual random effects filtering estimates.

Figure 4.22 displays the box plots of parameter estimation errors. There are some extreme outliers for the $m = 100$ case. We truncate these outliers so the box plots of other samples sizes can be seen.

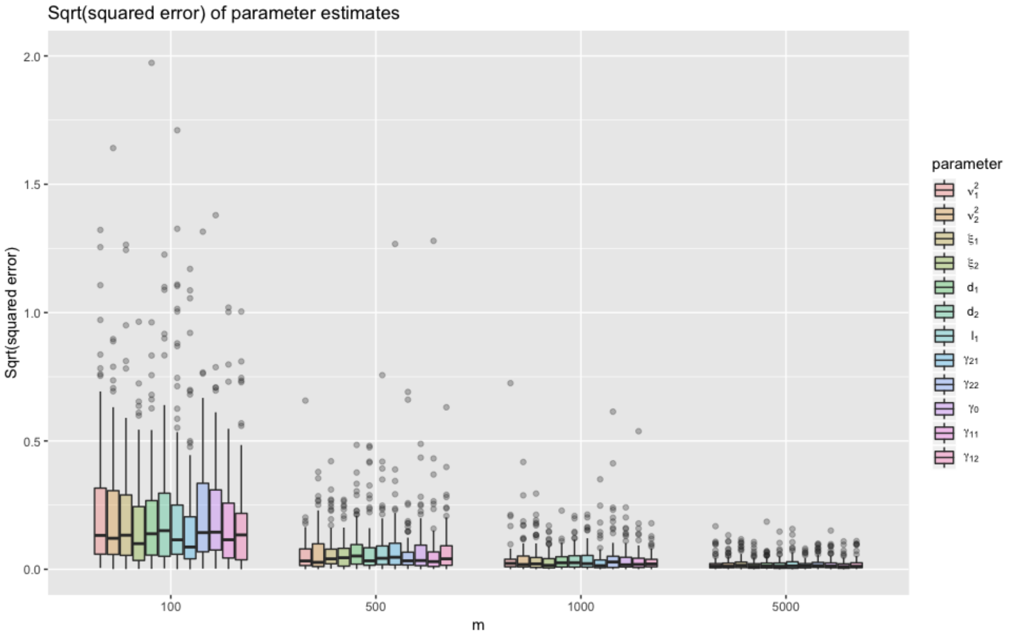

Figure 4.22: $\sqrt{\text{squared error}}$ of parameter estimates.

Table 4.20 displays the MSE of parameter estimates.

	100	500	1000	5000
ν_1^2	0.1424	0.0135	0.0047	0.0004
ν_2^2	0.0763	0.0056	0.0061	0.0003
ξ_1	0.0456	0.0036	0.0015	0.0002
ξ_2	2.1166	0.0032	0.0041	0.0001
d_1	0.0060	0.0007	0.0002	0.0001
d_2	0.0565	0.0067	0.0013	0.0002
l_1	0.4527	0.0184	0.0067	0.0014
γ_{21}	0.3306	0.0126	0.0037	0.0011
γ_{22}	0.1278	0.0158	0.0031	0.0013
γ_0	0.1719	0.0298	0.0056	0.0020
γ_{11}	0.3903	0.0448	0.0049	0.0021
γ_{12}	0.1852	0.0220	0.0096	0.0022

Table 4.20: MSE of parameter estimates.

Figure 4.23 displays two examples of filtering estimates for population effects $u_1(t_j)$ from the joint model. The two examples are selected from data sets where at least one subject lives to the end.

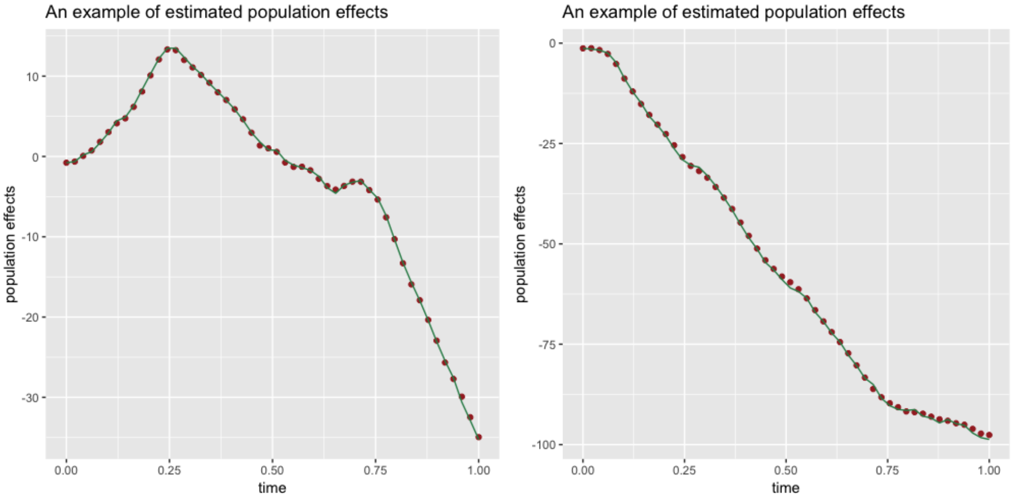

Figure 4.23: Examples of filtering estimates of population effects $u_1(t_j)$. Brown dots: true population effects, green solid line: the filtering estimates of population effects.

Figure 4.24 displays two examples of random effects filtering estimates from the joint model.

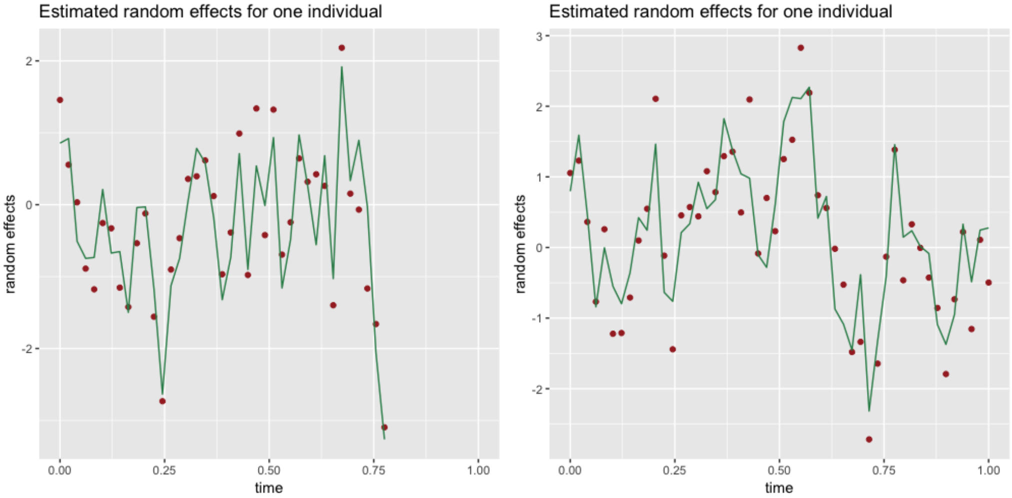

Figure 4.24: Two examples of random effects filtering estimates from the joint model. Brown dots: true random effects $v_{i1}(t)$, green solid line: the filtering estimates of $v_{i1}(t)$.

Chapter 5

Real Data Applications

We apply the joint model to a hemodialysis data set. The data set contains clinical observations of 354,572 end stage renal disease patients who were on dialysis treatment. The patients received dialysis treatment multiple times per week and were kept track of over five years from 2010 to 2014. The monthly averages of their clinical measurements were recorded in this data set.

The variables include: (i) external and internal covariates such as race (1: white, 0: otherwise), gender (1: male, 0: otherwise), diabetic status (1: has diabetes, 0: otherwise) and equilibrated Kt/V (EKTV), a measure of the intensity of a dialysis treatment; (ii) internal longitudinal variables, such as hemoglobin (g/dL), albumin (g/dL), serum sodium (mEq/L); and (iii) survival status (1: dead, 0: alive) and censoring status (1: censored, 0: otherwise).

The questions of clinical interest are (i) describing how multiple longitudinal variables evolve and interact with each other over time, and (ii) identifying risk factors of mortality. We present below two analyses for illustration.

5.1 Interrelationships Between Serum Sodium, Blood Pressure, and Inter-dialytic Weight Gain

For hemodialysis treatment patients, to individualize the dialysate sodium (DNa) prescription, research has been carried out to study its affect on patient outcomes ([55], [56], [57], [58], [59], [60]). DNa affects multiple aspects including serum sodium, blood pressure, and inter-dialytic weight gain. We present an example here to study the interactions between serum sodium (SNa), pre-dialysis systolic blood pressure (SBP), and inter-dialytic weight gain percentage (IDWG). To reduce potential inferential bias, we joint model these three longitudinal variables with survival status, and control for covariates albumin, age, gender, and diabetic status in the logistic regression survival model. Note that albumin is a longitudinal variable, but since we are not interested in modeling its trajectory or its interaction with other longitudinal variables in this analysis, we treat it as a time-variant covariate and include it in the survival model for control.

Figures 5.1 - 5.3 display the longitudinal observations of SNa, SBP, and IDWG from 100 randomly selected patients. Different colors correspond to different patients.

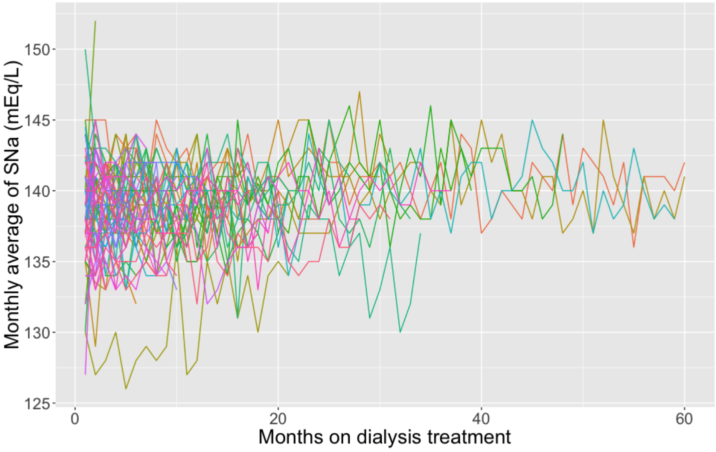

Figure 5.1: Monthly average of serum sodium from 100 randomly selected patients.

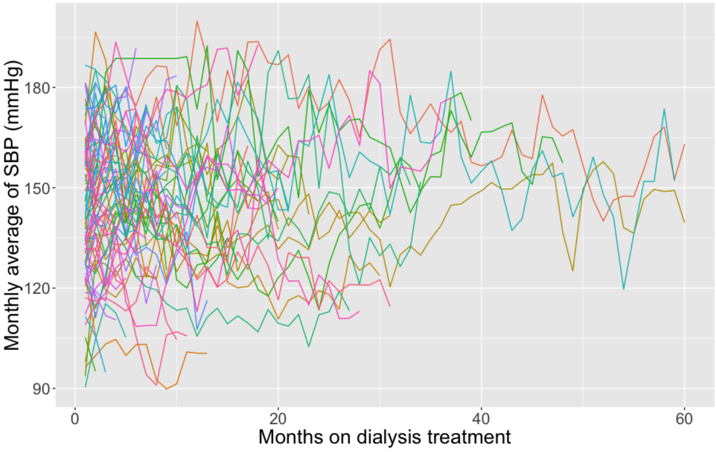

Figure 5.2: Monthly average of pre-dialytic systolic blood pressure from 100 randomly selected patients.

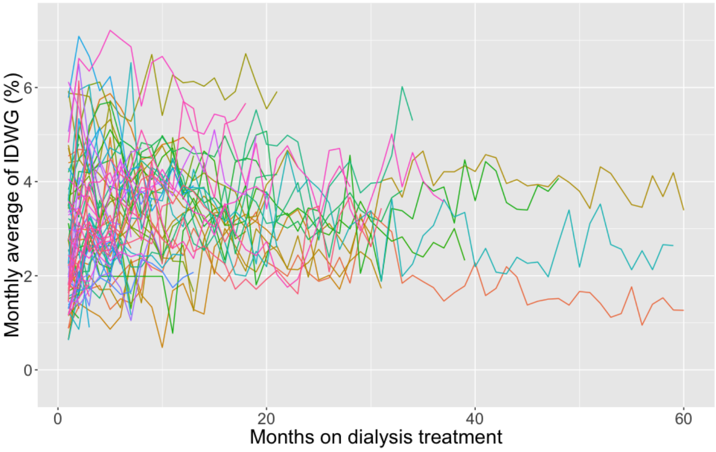

Figure 5.3: Monthly average of inter-dialytic weight gain percentage from 100 randomly selected patients.

Let $\boldsymbol{y}_i(t_j) = (y_{i1}(t_j), y_{i2}(t_j), y_{i3}(t_j))^T$, where $y_{i1}(t_j)$, $y_{i2}(t_j)$, and $y_{i3}(t_j)$ are SNa, SBP, and IDWG of subject i at time t_j, respectively. The longitudinal submodel is of the form (4.1). The population effects are modeled by cubic splines, and the subject random effects are modeled by OU processes. We use the subject level state transition matrix

$T_v(t_j)$ to model the interactions between the three longitudinal variables. We have

$$T_v(t_j) = T_v = \begin{pmatrix} e^{-\xi_1 \triangle t} & \phi_{12} & \phi_{13} \\ \phi_{21} & e^{-\xi_2 \triangle t} & \phi_{23} \\ \phi_{31} & \phi_{32} & e^{-\xi_3 \triangle t} \end{pmatrix},$$

where ξ_1, ξ_2, ξ_3 are the OU process parameters for each variable, same as the ξ in (2.6), and ϕ_{ij}, $i \neq j$, $1 \leq i, j \leq 3$, are the parameters to model the interactions between these three variables. $\Sigma_\epsilon(t_j)$ is assumed to be a time-invariant diagonal matrix, $\Sigma_\epsilon(t_j) = \Sigma_\epsilon = \text{diag}\{d_1, d_2, d_3\}$, $d_i > 0, i = 1, 2, 3$.

The population state is

$$\boldsymbol{u}(t_j) = \begin{pmatrix} u_1(t_j) \\ u_1'(t_j) \\ u_2(t_j) \\ u_2'(t_j) \\ u_3(t_j) \\ u_3'(t_j) \end{pmatrix},$$

and the subject random deviation state is

$$\boldsymbol{v}_i(t_j) = \begin{pmatrix} v_{i1}(t_j) \\ v_{i2}(t_j) \\ v_{i3}(t_j) \end{pmatrix}.$$

$\boldsymbol{v}_i(t_j)$ is a multivariate stationary AR(1) process, so the covariance matrix for the random effect components of the initial state is $Q_0 = I_m \otimes Q_{00}$, where

$$\text{vec}(Q_{00}) = (I_{q^2} - T_v \otimes T_v)^{-1}\text{vec}(\Sigma_v). \tag{5.1}$$

We provide details below for the derivation of (5.1). For each subject i, the evolution of subject random effects $\boldsymbol{v}_i(t_j)$ is described by the equation

$$\boldsymbol{v}_i(t_j) = T_v \boldsymbol{v}_i(t_{j-1}) + \boldsymbol{\eta}_{vi}(t_{j-1}), \quad \boldsymbol{\eta}_{vi}(t_{j-1}) \sim N(\mathbf{0}, \Sigma_v), \tag{5.2}$$

which is a zero mean multivariate AR(1) process. Since the subject deviations are i.i.d., for notational simplicity, we drop the i in the subscript and write

$$\boldsymbol{v}(t_j) = T_v \boldsymbol{v}(t_{j-1}) + \boldsymbol{\eta}_v(t_{j-1}), \quad \boldsymbol{\eta}_v(t_{j-1}) \sim N(\mathbf{0}, \Sigma_v). \tag{5.3}$$

Define $\Gamma_{j-k} = \mathrm{E}[\boldsymbol{v}(t_j)\boldsymbol{v}^T(t_k)]$. Then Γ_{j-k} is the covariance matrix between the two random vectors $\boldsymbol{v}(t_j)$ and $\boldsymbol{v}(t_k)$. We have $\Gamma_{j-k} = \Gamma_{k-j}^T$ and Γ_0 is a symmetric matrix. Multiplying both sides of (5.3) by $\boldsymbol{v}^T(t_{j-l})$, $l = 0, 1$, and taking expectation, we have

$$\mathrm{E}[\boldsymbol{v}(t_j)\boldsymbol{v}^T(t_{j-l})] = T_v \mathrm{E}[\boldsymbol{v}(t_{j-1})\boldsymbol{v}^T(t_{j-l})] + \mathrm{E}[\boldsymbol{\eta}_v(t_{j-1})\boldsymbol{v}^T(t_{j-l})], \tag{5.4}$$

where

$$\mathrm{E}[\boldsymbol{\eta}_v(t_{j-1})\boldsymbol{v}^T(t_{j-l})] = \begin{cases} \Sigma_v, & l = 0, \\ 0, & l = 1. \end{cases}$$

Therefore, (5.4) can be rewritten as

$$\Gamma_l - T_v \Gamma_{l-1} = \begin{cases} \Sigma_v, & l = 0, \\ 0, & l = 1. \end{cases} \tag{5.5}$$

That is,

$$\begin{aligned} \Gamma_0 - T_v \Gamma_1^T &= \Sigma_v, \\ \Gamma_1 - T_v \Gamma_0 &= 0, \end{aligned} \tag{5.6}$$

where Γ_{-1} is replaced by Γ_1^T. In (5.6), plugging $\Gamma_1 = T_v\Gamma_0$ into the first equation, we have

$$\Gamma_0 - T_v\Gamma_0 T_v^T = \Sigma_v. \tag{5.7}$$

Taking the vec$(\cdot)$ operation on both sides of (5.7), we have

$$\text{vec}(\Gamma_0) - (T_v \otimes T_v)\text{vec}(\Gamma_0) = (I - T_v \otimes T_v)\text{vec}(\Gamma_0) = \text{vec}(\Sigma_v), \tag{5.8}$$

since $\text{vec}(AXB) = (B^T A)\text{vec}(X)$. That is, $\text{vec}(\Gamma_0) = (I - T_v \otimes T_v)^{-1}\text{vec}(\Sigma_v)$. Γ_0 is the covariance matrix of $\boldsymbol{v}(t_j)$ for any $j \geq 1$, hence $Q_{00} = \text{Var}[\boldsymbol{v}(t_1)] = \Gamma_0$ and (5.1) holds.

The covariates are $\boldsymbol{x}_i(t_j) = (x_{i1}(t_j), x_{i2}(t_j), x_{i3}(t_j), x_{i4}(t_j))^T$, corresponding to albumin, age, gender, and diabetic status. In the survival submodel (3.2), we have $D = \left(I_3 \otimes (1 \quad 0), \quad I_3\right)$, so

$$D\begin{pmatrix} \boldsymbol{u}(t_j) \\ \boldsymbol{v}_i(t_j) \end{pmatrix} = \begin{pmatrix} u_1(t_j) + v_{i1}(t_j) \\ u_2(t_j) + v_{i2}(t_j) \\ u_3(t_j) + v_{i3}(t_j) \end{pmatrix}$$

are the latent values of the three longitudinal processes.

The survival submodel is given by

$$\begin{aligned} z_i(t_{j+1}) &\sim \text{Bernoulli}(\pi_i(t_{j+1})) \\ \text{logit}(\pi_i(t_{j+1})) &= \gamma_0 + \boldsymbol{x}_i^T(t_j)\boldsymbol{\gamma}_1 + (\boldsymbol{u}^T(t_j), \boldsymbol{v}_i^T(t_j))^T D^T \boldsymbol{\gamma}_2, \end{aligned}$$

where $\boldsymbol{\gamma}_1 = (\gamma_{11}, \gamma_{12}, \gamma_{13}, \gamma_{14})^T$ and $\boldsymbol{\gamma}_2 = (\gamma_{21}, \gamma_{22}, \gamma_{23})^T$.

We randomly select 3000 patients from a data set of 354,572 patients and fit the joint model to the selected sample. Tables 5.1 and 5.2 display parameter estimates and 95% bootstrap confidence intervals for the longitudinal and survival submodels, respectively.

We use 100 bootstrap samples, which are obtained by resampling the patients with replacement. The bootstrap confidence intervals for parameters are constructed by taking the 2.5% and 97.5% quantiles of bootstrap parameter estimates.

From Table 5.2, one can see that albumin, IDWG, and age are significantly associated with mortality. Larger values of albumin are associated with lower death probability, while larger values of IDWG and age are associated with higher death probability.

Parameter	Estimate	Lower	Upper
ζ_1 (SNa)	0.0000	0.0000	0.0000
ζ_2 (SBP)	0.1772	0.0673	0.4046
ζ_3 (IDWG)	0.0027	0.0019	0.0047
ν_1^2 (SNa)	0.4300	0.3514	0.5740
ν_2^2 (SBP)	50.9909	43.8295	58.0832
ν_3^2 (IDWG)	0.2292	0.1917	0.2623
ξ_1 (SNa)	0.0319	0.0240	0.0382
ξ_2 (SBP)	0.0619	0.0286	0.0739
ξ_3 (IDWG)	0.0882	0.0453	0.0990
d_1 (SNa)	5.0148	4.4001	5.6261
d_2 (SBP)	36.7646	32.5648	40.4381
d_3 (IDWG)	0.1590	0.1358	0.1819
ϕ_{12} (SNa, SBP)	0.0023	-0.0023	0.0134
ϕ_{13} (SNa, IDWG)	0.0235	-0.0716	0.3231
ϕ_{21} (SBP, SNa)	0.6017	-0.0102	0.7530
ϕ_{23} (SBP, IDWG)	0.0568	-0.0089	0.7955
ϕ_{31} (IDWG, SNa)	0.0138	-0.5121	0.0285
ϕ_{32} (IDWG, SBP)	0.0016	-0.0027	0.0070

Table 5.1: Longitudinal submodel parameter estimates and 95% bootstrap confidence intervals. $\zeta_{(\cdot)}$: smoothing parameter of population effects; $\xi_{(\cdot)}$ and $\nu^2_{(\cdot)}$: OU process parameters for subject random effects; $d_{(\cdot)}$: variance of the observation random errors; $\phi_{(\cdot)}$: interactions between two longitudinal variables, components of T_v.

Parameter	Estimate	Lower	Upper
γ_{21} (SNa)	-0.3135	-0.3828	0.2246
γ_{22} (SBP)	-0.0100	-0.2834	0.2949
γ_{23} (IDWG)	0.2170	0.0167	0.3992
γ_0 (intercept)	10.8541	4.9603	20.1125
γ_{11} (albumin)	-1.8559	-1.9868	-1.0841
γ_{12} (age)	0.1459	0.0472	0.3670
γ_{13} (male)	0.0440	-0.1040	0.9506
γ_{14} (diabetic)	0.1019	-0.2430	0.4913

Table 5.2: Survival submodel parameter estimates and 95% bootstrap confidence intervals. $\gamma_{(\cdot)}$: coefficient in the survival model.

The estimated T_v is

$$\hat{T}_v = \begin{pmatrix} 0.9686 & 0.0023 & 0.0235 \\ 0.6017 & 0.9399 & 0.0568 \\ 0.0138 & 0.0016 & 0.9156 \end{pmatrix},$$

in which the three longitudinal variables are SNa, SBP, IDWG in order. None of the off-diagonal elements of $\hat{T}_v$ are significantly different from 0, according to Table 5.1.

Figure 5.4 displays the filtering estimates of population effects for SNa, SBP, and IDWG, respectively, plotted against the mean of all observations at each time point. One can see that all three longitudinal variables increase rapidly during the first few months of dialysis treatment.

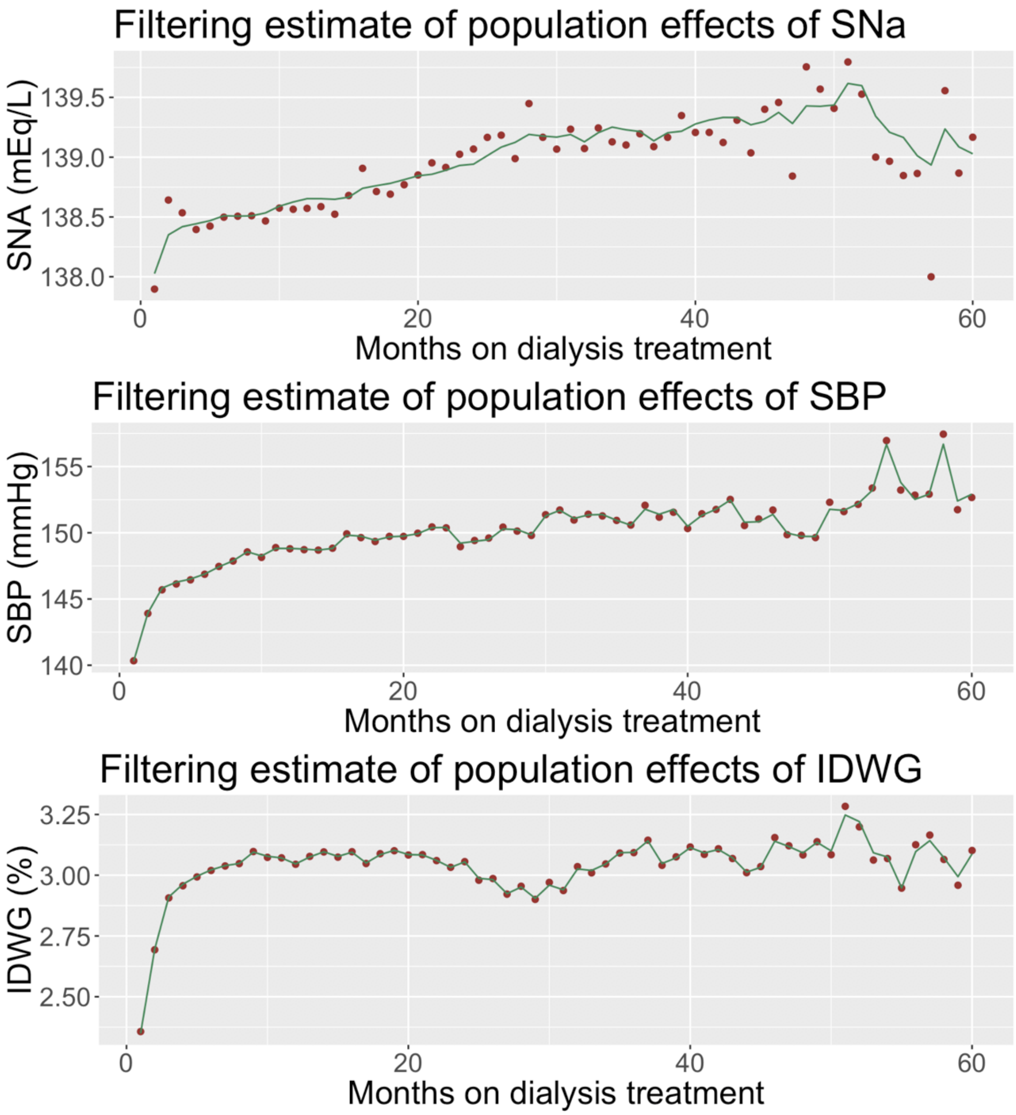

Figure 5.4: Filtering estimate of population effects for serum sodium, pre-dialytic systolic blood pressure, and inter-dialytic weight gain percentage. Brown dots: observation mean; green solid line: the filtering estimates of population effects.

Figure 5.5 shows the filtering estimates of longitudinal trajectories for a randomly selected subject, who died at the 50th month of dialysis treatment. One can see that during the last few months of the patient, his/her SBP was dramatically decreasing and IDWG was rapidly increasing.

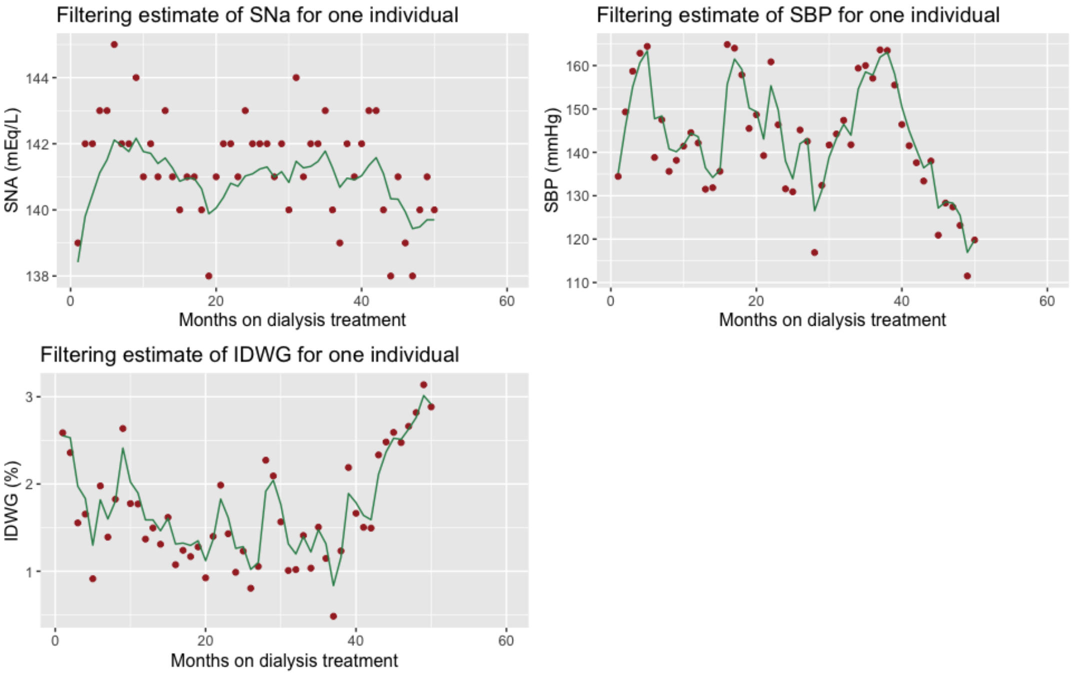

Figure 5.5: Filtering estimates of SNa, SBP, and IDWG for a randomly selected individual. Brown dots: observations. Green solid line: filtering estimates.

5.2 A Prediction Model for Mortality

Clinical measurements albumin, inter-dialytic weight gain, and blood pressure are potential risk factors of mortality. We construct a prediction model using these three and other variables. In the longitudinal submodel of the form (4.1), the three longitudinal variables are albumin, inter-dialytic weight gain percentage (IDWG), and pre-dialysis systolic blood pressure (SBP). For illustration we use a different approach from the previous example–here we use an unstructured time-invariant error covariance matrix to model the correlation between longitudinal variables. We have $\Sigma_\epsilon(t_j) = \Sigma_\epsilon = LDL^T$,

where

$$L = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix}$$

and

$$D = \begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{pmatrix}.$$

$l_{21}, l_{31}, l_{32}, d_1, d_2, d_3$ are parameters. The subject level state transition matrix is

$$T_v(t_j) = T_v = \begin{pmatrix} e^{-\xi_1 \triangle t} & 0 & 0 \\ 0 & e^{-\xi_2 \triangle t} & 0 \\ 0 & & e^{-\xi_3 \triangle t} \end{pmatrix},$$

where ξ_1, ξ_2, and ξ_3 are the OU process parameters for each variable. In the survival submodel of the form (3.2), the design matrix D and the coefficient vector $\boldsymbol{\gamma}_2$ are the same as those in the model in Section 5.1.

In the survival submodel, we control for covariates age, gender, diabetic status, EKTV, and the types of vascular access with three levels: AVF, AVG, and CATH, corresponding to arteriovenous (AV) fistula, AV graft, and catheter. For the types of vascular access, we set CATH as baseline. We have $\boldsymbol{\gamma}_2 = (\gamma_{21}, \gamma_{22}, \gamma_{23}, \gamma_{24}, \gamma_{25}, \gamma_{26})$, corresponding to age, male, diabetic, EKTV, AVF, and AVG, respectively.

Figures 5.6 - 5.8 display the longitudinal observations of albumin, SBP, and IDWG from 100 randomly selected patients. Different colors correspond to different patients.

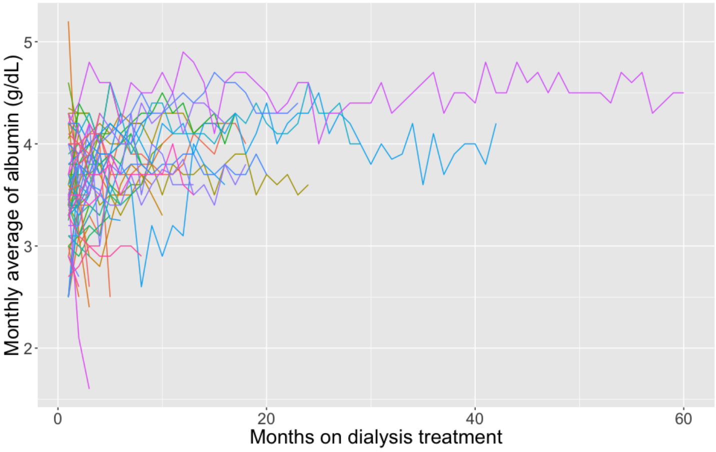

Figure 5.6: Monthly average of albumin from 100 randomly selected patients

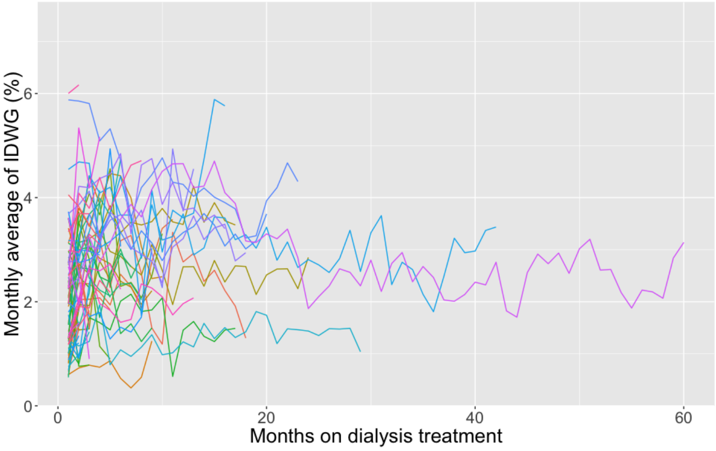

Figure 5.7: Monthly average of IDWG from 100 randomly selected patients.

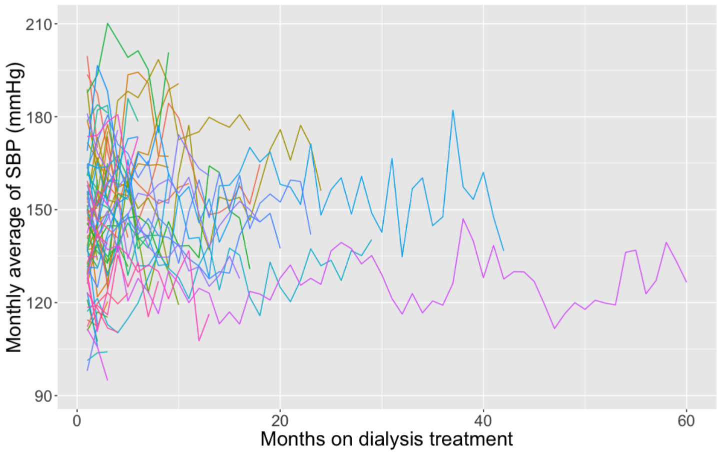

Figure 5.8: Monthly average of SBP from 100 randomly selected patients.

Tables 5.3 and 5.4 report the parameter estimates and 95% bootstrap confidence intervals for the longitudinal and survival submodels, respectively. The bootstrap confidence intervals are based on 100 bootstrap samples, which are obtained by resampling subjects with replacement. Albumin, IDWG, SBP, age, gender, EKTV, and AVF are significantly associated with mortality rate. Higher values of albumin, SBP, and EKTV are associated with lower probabilities of death. Male have lower death probability than female. Access type AV fistula are associated with lower probability of death compared to catheter, and there is no significant difference between access types AV graft and catheter. Higher values of IDWG and an older age are associated with higher probabilities of death.

Parameter	Estimate	Lower	Upper
ζ_1 (albumin)	0.0003	0.0003	0.0008
ζ_2 (IDWG)	0.0032	0.0028	0.0083
ζ_3 (SBP)	0.0298	0.0167	0.3992
ν_1^2 (albumin)	0.0142	0.0117	0.0179
ν_2^2 (IDWG)	0.2135	0.1963	0.2372
ν_3^2 (SBP)	59.1552	55.0777	69.1259
ξ_1 (albumin)	0.0401	0.0334	0.0487
ξ_2 (IDWG)	0.0874	0.0795	0.1003
ξ_3 (SBP)	0.0817	0.0743	0.0950
d_1 (albumin)	0.0275	0.0245	0.0294
d_2 (IDWG)	0.1242	0.1091	0.1340
d_3 (SBP)	28.4982	23.3930	30.5427
l_{21}	0.0640	0.0032	0.1188
l_{31}	2.4295	1.5982	3.6274
l_{32}	-0.8348	-1.3334	-0.2934

Table 5.3: Longitudinal submodel parameter estimates and 95% bootstrap confidence intervals. $\zeta_{(\cdot)}$: smoothing parameter of population effects; $\xi_{(\cdot)}$ and $\nu^2_{(\cdot)}$: OU process parameters for subject random effects; $d_{(\cdot)}$: variance of observation random errors; $l_{(\cdot)}$: elements of the lower triangular matrix L in the decomposition $\Sigma_\epsilon = LDL^T$, do not have particular meanings.

Parameter	Estimate	Lower	Upper
γ_{21} (albumin)	-1.6315	-1.9073	-1.4515
γ_{22} (IDWG)	0.1547	0.0473	0.2436
γ_{23} (SBP)	-0.0234	-0.0286	-0.0154
γ_0 (Intercept)	3.6138	2.2785	4.9840
γ_{11} (age)	0.0396	0.0317	0.0495
γ_{12} (male)	-0.2388	-0.4266	-0.0021
γ_{13} (EKTV)	-0.7230	-1.1406	-0.3828
γ_{14} (diabetic)	-0.0624	-0.2887	0.1745
γ_{15} (AVF)	-0.6118	-0.8253	-0.3645
γ_{16} (AVG)	-0.2034	-0.5913	0.1969

Table 5.4: Survival submodel parameter estimates and 95% bootstrap confidence intervals. $\gamma_{(\cdot)}$: coefficient in the survival model.

The estimated Σ_ϵ is

$$\hat{\Sigma} = \begin{pmatrix} 0.0275 & 0.0018 & 0.0667 \\ 0.0018 & 0.1244 & -0.0994 \\ 0.0667 & -0.0994 & 28.7469 \end{pmatrix},$$

in which the three longitudinal variables are albumin, IDWG, and SBP in order. Write Σ_ϵ as

$$\Sigma_\epsilon = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix},$$

where $\sigma_{12} = \sigma_{21}$, $\sigma_{13} = \sigma_{31}$, and $\sigma_{23} = \sigma_{32}$. Table 5.5 displays 95% bootstrap confidence intervals for the off diagonal elements of Σ_ϵ. IDWG and albumin are significantly positively correlated, SBP and albumin are significantly positively correlated, and SBP and IDWG are significantly negatively correlated.

At first sight, the negative correlation between SBP and IDWG seems conflicting with what we obtain in Section 5.1, where the estimates of ϕ_{23} and ϕ_{32} in the matrix T_v are positive, although not significant. There is also literature suggesting positive correlation between SBP and IDWG ([61], [62], [63]). However, in Figures 5.5 and 5.10, the SBP and IDWG values of an individual are often going in opposite directions. Recall that in our mixed effects state space model there are three places to model interactions between variables: Σ_ϵ, T_u, and T_v. Among these three choices, Σ_ϵ models the correlation structure of the combination of population effects and random effects, T_u models the correlation structure on the population effects level, and T_v models the correlation structure on the subject random effects level. Our findings about T_v in Section 5.1 suggest that SBP and IDWG are positively correlated on the subject level (although not significant), and our findings about Σ_ϵ in this section suggest that SBP and IDWG are negatively correlated after combining population effects and subject random effects. It would be interesting to see how the two variables correlate on the population level, modeled by T_u, which, due to time constraint, we do not include it here.

Parameter	Estimate	Lower	Upper
σ_{21} (IDWG, albumin)	0.0018	0.0001	0.0033
σ_{31} (SBP, albumin)	0.0667	0.0429	0.0980
σ_{32} (SBP, IDWG)	-0.0994	-0.1618	-0.0309

Table 5.5: Estimates of off-diagonal elements of Σ_ϵ and 95% bootstrap confidence intervals.

Figure 5.9 displays the filtering estimates of population effects for albumin, IDWG, and SBP, respectively, plotted against the mean of observations at each time point.

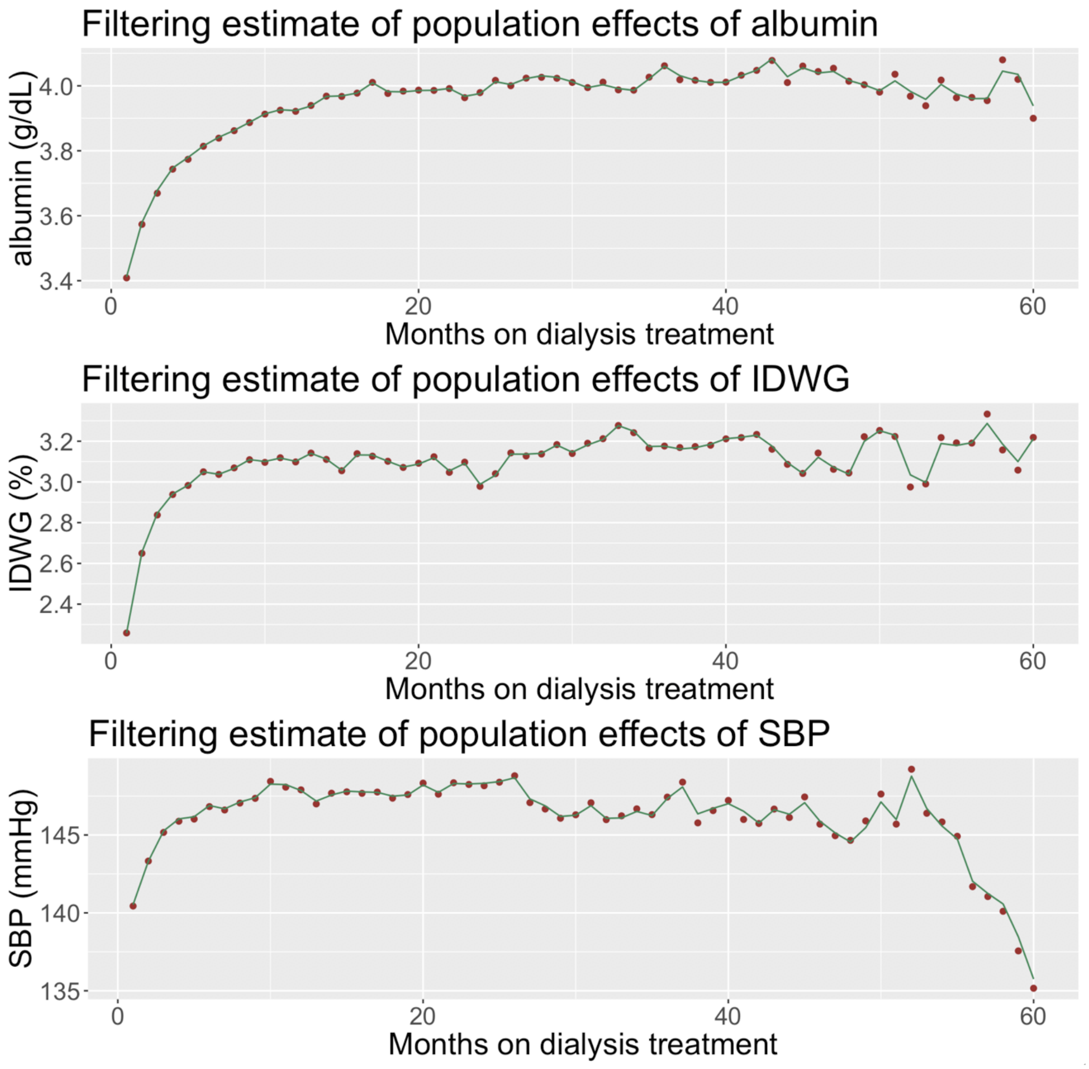

Figure 5.9: Filtering estimate of population effects for albumin, IDWG, and SBP. Brown dots: observation mean; green solid line: the filtering estimates of population effects.

Figure 5.10 shows the filtering estimates of longitudinal trajectories for a randomly selected individual, who was censored after the 59th month of dialysis treatment. One can see that SBP and IDWG are often going in opposite directions. Opposite patterns of SBP and IDWG can also be seen from Figure 5.5 in Section 5.1. In Figure 5.10, during the later months, this patient's IDWG was rapidly decreasing and SBP was increasing, the opposite of that what happened to the patient in Figure 5.5.

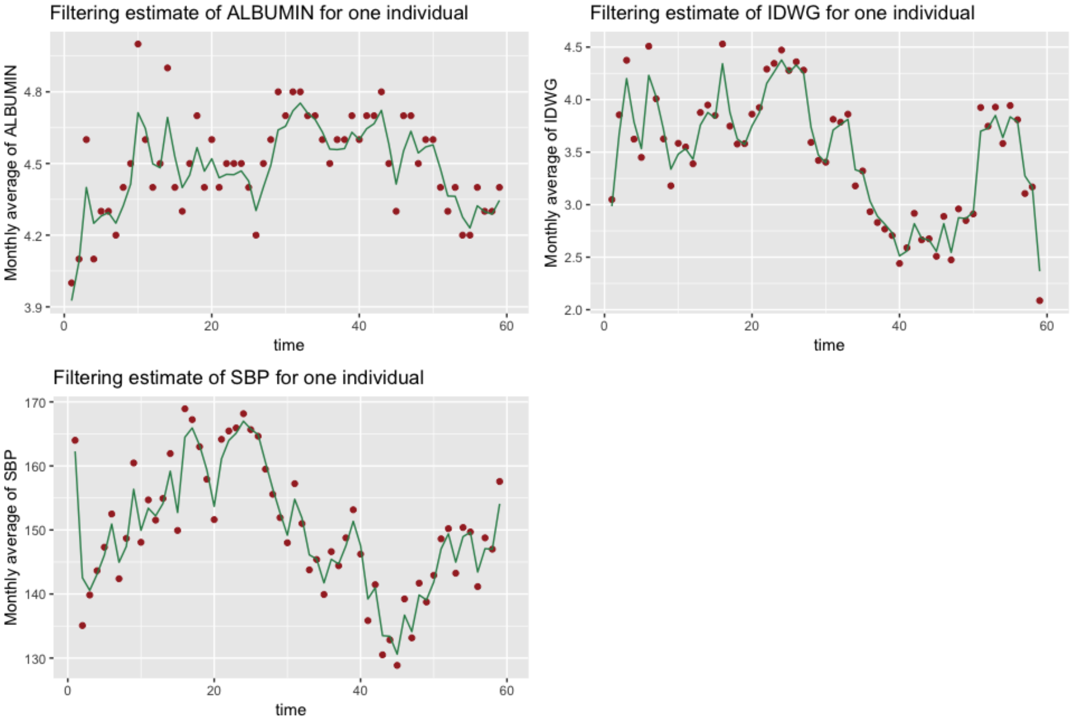

Figure 5.10: Filtering estimates of albumin, IDWG, and SBP for a randomly selected individual. Brown dots: observations. Green solid line: filtering estimates.

Chapter 6

Limitations and Future Work

6.1 Limitations of the New Smoothing Algorithm

Although our main goal is to perform online estimations and predictions which only use the filtering algorithm, posterior estimates of latent states, i.e., smoothing estimates, can also be of interest. Currently, the smoothing part of our new algorithm allows the subjects to come in at later time points, but cannot deal with the case where there are dropouts. We will work on dealing with missing data in the future.

6.2 Mixed-type Longitudinal Variables

Our new algorithm can potentially be applied to a mixed effects state space model with mixed types of longitudinal variables, as long as the model has a linear Gaussian signal as described in (1.24). A typical example of observations with a linear Gaussian signal is the exponential family random variables.

In (1.24), when y_t is an exponential family random variable, the signal $\theta_t = Z_t\boldsymbol{\alpha}_t$ is the linear predictor in a generalized linear model (GLM), and the error term covariance matrix H_t equals 0 for any non-Gaussian exponential family random variables.

To illustrate how our new algorithm can be applied to mixed types of longitudinal variables, suppose that for subject i we have longitudinal observations $\{y_{1i}(t_j), y_{2i}(t_j)\}$, $i = 1, \ldots, m$, where $y_{1i}(t_j)$ and $y_{2i}(t_j)$ are two types of exponential family random variables. Suppose that $y_{1i}(t_j) \sim p_1(y_{1i}(t_j)|\theta_{1i}(t_j))$ and $y_{2i}(t_j) \sim p_2(y_{2i}(t_j)|\theta_{2i}(t_j))$, we have

$$p(y_{ki}(t_j)|\theta_{ki}(t_j)) = \exp[y_{ki}(t_j)\theta_{ki}(t_j) - b(\theta_{ki}(t_j)) + c(y_{ki}(t_j))], \quad k = 1, 2,$$

where $b(\theta_{ki}(t_j))$ is twice differentiable.

Let $\boldsymbol{\theta}_1(t_j) = (\theta_{11}(t_j), \ldots, \theta_{1m}(t_j))^T$ be the stacked signals of $y_{1i}(t_j)$ for m subjects and $\boldsymbol{\theta}_2(t_j) = (\theta_{21}(t_j), \ldots, \theta_{2m}(t_j))^T$ be the stacked signals of $y_{2i}(t_j)$ for m subjects. The signals $\boldsymbol{\theta}_1(t_j)$ and $\boldsymbol{\theta}_2(t_j)$ depend on the latent state $\boldsymbol{\alpha}(t_j)$ via the equation

$$\underbrace{\begin{pmatrix} \boldsymbol{\theta}_1(t_j) \\ \boldsymbol{\theta}_2(t_j) \end{pmatrix}}_{\boldsymbol{\theta}(t_j)} = \underbrace{\begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix}}_{Z} \boldsymbol{\alpha}(t_j), \quad j = 1, \ldots, n. \tag{6.1}$$

The latent state $\boldsymbol{\alpha} = (\boldsymbol{u}^T(t_j), \boldsymbol{v}_1^T(t_j), \ldots, \boldsymbol{v}_m^T(t_j))^T$ contains a population effects component and m subject level random effects components. The population state $\boldsymbol{u}(t_j)$ contains both variables' latent states, that is,

$$\boldsymbol{u}(t_j) = \begin{pmatrix} u_1(t_j) \\ u_2(t_j) \end{pmatrix},$$

and so does the subject random effects component:

$$\boldsymbol{v}_i(t_j) = \begin{pmatrix} v_{i1}(t_j) \\ v_{i2}(t_j) \end{pmatrix}, \quad i = 1, \ldots, m.$$

$\boldsymbol{u}(t_j)$ and $\boldsymbol{v}_i(t_j)$ can also contain the first derivatives of the variables, depending on the specification of the stochastic process. The latent state $\boldsymbol{\alpha}(t_j)$ is connected to the signal $\boldsymbol{\theta}(t_j)$ through the system matrix $Z = (Z_1^T, Z_2^T)^T$, where

$$Z_1 = \begin{pmatrix} Z_{1u} & Z_{1v} & & 0 \\ \vdots & & \ddots & \\ Z_{1u} & 0 & & Z_{1v} \end{pmatrix}$$

and

$$Z_2 = \begin{pmatrix} Z_{2u} & Z_{2v} & & 0 \\ \vdots & & \ddots & \\ Z_{2u} & 0 & & Z_{2v} \end{pmatrix}.$$

The state equation for $\boldsymbol{\alpha}(t_j)$ is the same as in (2.13). Generalization to the case where there are multiple variables of each type is trivial.

Instead of letting the components of $\boldsymbol{\theta}(t_j)$ come into the system all at once or one at a time as in the univariate treatment, we let the components enter the system in two batches: $\boldsymbol{\theta}_1(t_j)$ and $\boldsymbol{\theta}_2(t_j)$. The equation for the signal, (6.1), thus becomes two sequential parts:

$$\boldsymbol{\theta}_1(t_j) = Z_1\boldsymbol{\alpha}(t_j), \quad \boldsymbol{\theta}_2(t_j) = Z_2\boldsymbol{\alpha}(t_j), \quad j = 1, \ldots, n.$$

In this way, the special structure of the model is preserved. In general, if there are k different types of variables, we will let them enter the system in k batches. To evaluate the likelihood, since the model is non-Gaussian, we will need importance sampling, see Durbin and Koopman [26] Section 11.6. A reasonable choice of the importance density is an approximating linear Gaussian model that has the same posterior mode for $\boldsymbol{\theta}(t_j)$, $j = 1, \ldots, n$. To find this approximating linear Gaussian model, we will use a technique called the mode estimation, described in Durbin and Koopman [26] Section 10.6. The approximation is achieved using the Newton-Raphson method, which iteratively updates the estimate of $\boldsymbol{\theta}(t_j)$, $j = 1, \ldots, n$. At each iteration, the next estimate of $\boldsymbol{\theta}(t_j)$ is obtained by applying the Kalman filter and smoother to the current approximating linear Gaussian state space model, to which we can apply our new algorithm. The observation error covariance matrix $H(t_j)$ of the current approximating linear Gaussian model is still block diagonal, but the blocks are no longer identical. To preserve the special structures of the Kalman filter and smoother, we let the subjects come into the system one at a time. At convergence, the approximating linear Gaussian model can be used as the importance density for likelihood evaluation. The time complexity will be $\mathrm{O}(q^3m^2n)$, still better than the univariate treatment's $\mathrm{O}(q^3m^3n)$. Currently we have this idea but have not started working on it, hereby we list it as future work.

We provide an example here. Suppose that $y_{1i}(t_j) \sim N(\mu_i(t_j), \sigma_i^2)$ and $y_{2i}(t_j) \sim \text{Poisson}(\lambda_i(t_j))$. A typical example of $y_{2i}(t_j)$ is the number of hospital admissions during time interval $(t_{j-1}, t_j]$. Then $\theta_{1i}(t_j) = \mu_i(t_j)$ and $\theta_{2i}(t_j) = \log(\lambda_i(t_j))$. Let $\boldsymbol{\lambda}(t_j) = (\lambda_1(t_j), \ldots, \lambda_m(t_j))^T$, we have

$$\begin{pmatrix} \boldsymbol{y}_1(t_j) \\ \log(\boldsymbol{\lambda}(t_j)) \end{pmatrix} = \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} \boldsymbol{\alpha}(t_j) + \begin{pmatrix} \boldsymbol{\epsilon}_1(t_j) \\ \mathbf{0} \end{pmatrix}, \quad \boldsymbol{\epsilon}_1(t_j) \sim N(\mathbf{0}, H_1(t_j)),$$

where $H_1(t_j) = \mathrm{diag}\{\sigma_1^2, \ldots, \sigma_m^2\}$.

Bibliography

[1] N. M. Laird and J. H. Ware, *Random-effects models for longitudinal data*, *Biometrics* (1982) 963–974.

[2] C. R. Henderson, O. Kempthorne, S. R. Searle, and C. Von Krosigk, *The estimation of environmental and genetic trends from records subject to culling*, *Biometrics* **15** (1959), no. 2 192–218.

[3] Y. Wang, *Mixed effects smoothing spline analysis of variance*, *Journal of the royal statistical society: Series b (statistical methodology)* **60** (1998), no. 1 159–174.

[4] C. Ke and Y. Wang, *Semiparametric nonlinear mixed-effects models and their applications*, *Journal of the American Statistical Association* **96** (2001), no. 456 1272–1298.

[5] D. Zhang, X. Lin, J. Raz, and M. Sowers, *Semiparametric stochastic mixed models for longitudinal data*, *Journal of the American Statistical Association* **93** (1998), no. 442 710–719.

[6] W. Guo, *Functional mixed effects models*, *Biometrics* **58** (2002), no. 1 121–128.

[7] J. S. Morris and R. J. Carroll, *Wavelet-based functional mixed models*, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68** (2006), no. 2 179–199.

[8] M. L. Mackenzie, C. Donovan, and B. McArdle, *Regression spline mixed models: A forestry example*, *Journal of agricultural, biological, and environmental statistics* **10** (2005), no. 4 394.

[9] J. A. Aston, J.-M. Chiou, and J. P. Evans, *Linguistic pitch analysis using functional principal component mixed effect models*, *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **59** (2010), no. 2 297–317.

[10] R. Prentice, *Covariate measurement errors and parameter estimation in a failure time regression model*, *Biometrika* **69** (1982), no. 2 331–342

.

[11] D. Rizopoulos, *Joint models for longitudinal and time-to-event data: With applications in R*. Chapman and Hall/CRC, 2012.

[12] A. A. Tsiatis and M. Davidian, *Joint modeling of longitudinal and time-to-event data: an overview*, *Statistica Sinica* (2004) 809–834.

[13] J. M. Taylor, W. Cumberland, and J. Sy, *A stochastic model for analysis of longitudinal aids data*, *Journal of the American Statistical Association* **89** (1994), no. 427 727–736.

[14] M. P. LAVALLEY and V. DEGRUTTOLA, *Models for empirical bayes estimators of longitudinal cd4 counts*, *Statistics in Medicine* **15** (1996), no. 21 2289–2305.

[15] R. Henderson, P. Diggle, and A. Dobson, *Joint modeling of longitudinal measurements and event time data*, *Biostatistics* **1** (2000), no. 4 465–480.

[16] Y. Wang and J. M. G. Taylor, *Jointly modeling longitudinal and event time data with application to acquired immunodeficiency syndrome*, *Journal of the American Statistical Association* **96** (2001), no. 455 895–905.

[17] J. Xu and S. L. Zeger, *Joint analysis of longitudinal data comprising repeated measures and times to events*, *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **50** (2001), no. 3 375–387.

[18] D. R. Cox, *Regression models and life-tables*, in *Breakthroughs in statistics*, pp. 527–541. Springer, 1992.

[19] F. Hsieh, Y.-K. Tseng, and J.-L. Wang, *Joint modeling of survival and longitudinal data: likelihood approach revisited*, *Biometrics* **62** (2006), no. 4 1037–1043.

[20] A. S. Whittemore and J. B. Keller, *Survival estimation using splines*, *Biometrics* (1986) 495–506.

[21] P. S. Rosenberg, *Hazard function estimation using b-splines*, *Biometrics* (1995) 874–887.

[22] J. E. Herndon and F. E. Harrell Jr, *The restricted cubic spline hazard model*, *Communications in Statistics-Theory and Methods* **19** (1990), no. 2 639–663.

[23] Y.-Y. Chi and J. G. Ibrahim, *Joint models for multivariate longitudinal and multivariate survival data*, *Biometrics* **62** (2006), no. 2 432–445.

[24] J. Xu and S. L. Zeger, *The evaluation of multiple surrogate endpoints*, *Biometrics* **57** (2001), no. 1 81–87.

[25] X. Song, M. Davidian, and A. A. Tsiatis, *An estimator for the proportional hazards model with multiple longitudinal covariates measured with error*, *Biostatistics* **3** (2002), no. 4 511–528.

[26] J. Durbin and S. J. Koopman, *Time series analysis by state space methods*, vol. 38. Oxford University Press, 2012.

[27] B. Rosenberg, *Random coefficients models: the analysis of a cross section of time series by stochastically convergent parameter regression*, in *Annals of Economic and Social Measurement, Volume 2, number 4*, pp. 399–428. NBER, 1973.

[28] A. C. Harvey, *Forecasting, structural time series models and the Kalman filter*. Cambridge university press, 1990.

[29] J. V. Tsimikas and J. Ledolter, *Mixed model representation of state space models: New smoothing results and their application to reml estimation*, *Statistica Sinica* (1997) 973–991.

[30] D. A. Harville, *Bayesian inference for variance components using only error contrasts*, *Biometrika* **61** (1974), no. 2 383–385.

[31] R. E. Kalman, *A new approach to linear filtering and prediction problems*, *Journal of basic Engineering* **82** (1960), no. 1 35–45.

[32] G. Evensen, *Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics*, *Journal of Geophysical Research: Oceans* **99** (1994), no. C5 10143–10162.

[33] P. Houtekamer and F. Zhang, *Review of the ensemble kalman filter for atmospheric data assimilation*, *Monthly Weather Review* **144** (2016), no. 12 4489–4532.

[34] M. Katzfuss, J. R. Stroud, and C. K. Wikle, *Understanding the ensemble kalman filter*, *The American Statistician* **70** (2016), no. 4 350–357.

[35] J. R. Stroud, M. Katzfuss, and C. K. Wikle, *A bayesian adaptive ensemble kalman filter for sequential state and parameter estimation*, *Monthly Weather Review* **146** (2018), no. 1 373–386.

[36] M. Katzfuss, J. R. Stroud, and C. K. Wikle, *Ensemble kalman methods for high-dimensional hierarchical dynamic space-time models*, *[stat.ME]* **arXiv:1704.06988** (2018)

.

[37] I. Szunyogh, E. J. Kostelich, G. Gyarmati, E. Kalnay, B. R. Hunt, E. Ott, E. Satterfield, and J. A. Yorke, *A local ensemble transform kalman filter data assimilation system for the ncep global model*, *Tellus A: Dynamic Meteorology and Oceanography* **60** (2008), no. 1 113–130.

[38] J. S. Whitaker, T. M. Hamill, X. Wei, Y. Song, and Z. Toth, *Ensemble data assimilation with the ncep global forecast system*, *Monthly Weather Review* **136** (2008), no. 2 463–482.

[39] P. Houtekamer, X. Deng, H. L. Mitchell, S.-J. Baek, and N. Gagnon, *Higher resolution in an operational ensemble kalman filter*, *Monthly Weather Review* **142** (2014), no. 3 1143–1162.

[40] M. Katzfuss, J. R. Stroud, and C. K. Wikle, *Extended ensemble kalman filters for high-dimensional hierarchical state-space models*, *arXiv preprint arXiv:1704.06988* (2017).

[41] S. Gillijns, O. B. Mendoza, J. Chandrasekar, B. De Moor, D. Bernstein, and A. Ridley, *What is the ensemble kalman filter and how well does it work*, in *American control conference*, vol. 6, IEEE, 2006.

[42] R. H. Jones, *Longitudinal data with serial correlation: a state-space approach.* Chapman and Hall/CRC, 1993.

[43] L. Fahrmeir and G. Tutz, *Multivariate statistical modelling based on generalized linear models.* Springer Science & Business Media, 2013.

[44] D. Y. Fong, *State space models and filtering methods in longitudinal studies*, *The University of Waterloo, Thesis* (1997).

[45] D. Gamerman and H. S. Migon, *Dynamic hierarchical models*, *Journal of the Royal Statistical Society. Series B (Methodological)* (1993) 629–642.

[46] B. Bakker and T. Heskes, *Learning and approximate inference in dynamic hierarchical models*, *Computational Statistics & Data Analysis* **52** (2007), no. 2 821–839.

[47] Z. Liu, *Modeling longitudinal data by state space method*, *University of Pennsylvania, Dissertation* (2010).

[48] D. Liu, T. Lu, X.-F. Niu, and H. Wu, *Mixed-effects state-space models for analysis of longitudinal dynamic systems*, *Biometrics* **67** (2011), no. 2 476–485.

[49] Z. Liu, A. R. Cappola, L. J. Crofford, and W. Guo, *Modeling bivariate longitudinal hormone profiles by hierarchical state space models*, *Journal of the American Statistical Association* **109** (2014), no. 505 108–118.

[50] J. Zhou and A. Tang, *Estimating linear mixed-effects state space model based on disturbance smoothing*, *arXiv preprint arXiv:1409.0391* (2014).

[51] J. Zhou, L. Han, and S. Liu, *Nonlinear mixed-effects state space models with applications to hiv dynamics*, *Statistics & Probability Letters* **83** (2013), no. 5 1448–1456.

[52] J. Zhou, A. Tang, and H. Feng, *Monte carlo likelihood estimation of mixed-effects state space models with application to hiv dynamics*, *Journal of Systems Science and Complexity* **29** (2016), no. 4 1160–1176.

[53] S. Cheng, *Analytic solution to ornstein-uhlenbeck sde*, *http://planetmath.org/analyticsolutiontoornsteinuhlenbecksde*.

[54] Wikipedia, *Computational complexity of mathematical operations*, *https://en.wikipedia.org/wiki/Computational_complexity_of_mathematical_operations*.

[55] S. F. Santos and A. J. Peixoto, *Revisiting the dialysate sodium prescription as a tool for better blood pressure and interdialytic weight gain management in hemodialysis patients*, *Clinical Journal of the American Society of Nephrology* **3** (2008), no. 2 522–530.

[56] C. Basile and C. Lomonte, *It is time to individualize the dialysate sodium prescription*, *Seminars in Dialysis* **29** (2016), no. 1 58–75.

[57] M. Hecking, A. Karaboyas, R. Saran, A. Sen, M. Inaba, H. Rayner, W. Hörl, R. Pisoni, B. Robinson, G. Sunder-Plassmann, and F. Port, *Dialysate sodium concentration and the association with interdialytic weight gain, hospitalization, and mortality*, *Clin J Am Soc Nephrol* **7** (2012) 92–100.

[58] G. Beduschi, L. Telini, J. Costa, T. Caramori, L. Martin, and P. Barretti, *Effect of dialysate sodium reduction on body water volume, blood pressure, and inflammatory markers in hemodialysis patients a prospective randomized controlled study*, *Renal Failure* **35** (2013), no. 5 742–747.

[59] C. Basile, A. Pisano, P. Lisi, L. Rossi, C. Lomonte, and D. Bolignano, *High versus low dialysate sodium concentration in chronic haemodialysis patients: a systematic review of 23 studies*, *Nephrol Dial Transplant* **31** (2016) 548–563

.

[60] J. Flythe and F. McCcausland, *Dialysate sodium: Rationale for evolution over time*, *Seminars in Dialysis* **30** (2017), no. 2 99–111.

[61] J. K. Inrig, U. D. Patel, B. S. Gillespie, V. Hasselblad, J. Himmelfarb, D. Reddan, R. M. Lindsay, J. F. Winchester, J. Stivelman, R. Toto, *et. al.*, *Relationship between interdialytic weight gain and blood pressure among prevalent hemodialysis patients*, *American Journal of Kidney Diseases* **50** (2007), no. 1 108–118.

[62] A. J. Luik, U. Gladziwa, J. P. Kooman, J. P. van Hooff, P. W. de Leeuw, L. M. B. van Bortel, and K. M. L. Leunissen, *Influence of interdialytic weight gain on blood pressure in hemodialysis patients*, *Blood purification* **12** (1994), no. 4-5 259–266.

[63] K. J. Ipema, J. Kuipers, R. Westerhuis, C. A. Gaillard, C. P. van der Schans, W. P. Krijnen, and C. F. Franssen, *Causes and consequences of interdialytic weight gain*, *Kidney and Blood Pressure Research* **41** (2016), no. 5 710–720.