STATISTICAL RESEARCH ON COVARIATE MATCHING, MONOTONE FUNCTIONAL DATA AND BINARY SPATIO-TEMPORAL DATA MODELING

A Dissertation

by

YEI EUN SHIN

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Jianhua Huang
Co-Chair of Committee,	Yu Ding
Committee Members,	Huiyan Sang
	Lan Zhou
Head of Department,	Valen Johnson

August 2017

Major Subject: Statistics

ABSTRACT

This dissertation consists of three studies in different fields. (1) The first study aims to evaluate the effect of wind turbine upgrades by devising a covariate matching method. The proposed method performs straightforward comparison between treated and non-treated outputs by re-organizing data records as if they were designed by randomized experiment. Also, it considers multi-dependencies of dynamic atmosphere conditions by taking into account the priority order and interaction effect of factors. (2) The second study proposes the functional data model to estimate a collection of monotone curves observed on an irregular and sparse grid. By integrating functional principal component analysis, not only does the model describe the variation of curves by few important functions but also it jointly estimates numerous monotone curves having a mean trend as well as individual-specific features. Simulation study validates its superiority to other classical approaches. (3) The last study investigates spatio-temporal binary data with a goal of describing infectious disease spreading pattern. An autologistic regressive model is proposed to illustrate spatial dependence and predict the progression over space and time. The accuracy of model estimation is verified by simulation study. Additionally, a hidden Markov network model is established from a slightly different standpoint on given data.

DEDICATION

To my parents and my husband Inwoo.

ACKNOWLEDGMENTS

First and foremost, I would like to express my special thanks to my advisor Dr. Huang for the continuous support and encouragement for my five-year study. The extensive experience in various fields he has guided has inspired me to seek a better insight on statistics. Besides my advisor, it has been a great honor for me to work with each of my committee members. My sincere thanks goes to Dr. Ding for his professional coaching and I would never forget writing my first manuscript under his direct guidance. Also, I deeply thank Dr. Zhou for her thoughtful and prudent advice. The lively discussions I and she had every week definitely make the second project more valuable. Lastly, Dr. Sang has my gratitude for her motivating and proactive instructions that enable the considerable progress in the last project of my Ph.D. study.

Behind every great daughter are truly amazing parents. Without the inspiration, drive, and support that my parents have given me, I might not be the person I am today. They deserve my heartfelt thanks.

So to conclude, I mostly want to acknowledge my beloved husband Inwoo who will stand by me as always. Ordinary days I and he lovingly have had together take the credit for the completion of this work.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professors Jianhua Huang (advisor), Lan Zhou and Huiyan Sang of the Department of Statistics and Professor Yu Ding (co-advisor) of the Department of Industrial and Systems Engineering.

The wind power data analyzed for Chapter 2 and 3 was provided by Professor Yu Ding, and the ALS patients data analyzed for Chapter 4 was provided by Biogen company.

All work conducted for the dissertation was completed by the student independently under the advisement of committee members; Professors Jianhua Huang and Yu Ding advised on Chapter 2 and Professor Lan Zhou did on Chapter 3. Chapter 4 was also guided by Professor Huiyan Sang as well as Professor Peter X. K. Song of the Department of Biostatistics in School of Public Health at University of Michigan and Biostatistician Dawei Liu of Biogen company.

Funding Sources

Graduate study for five years was fully supported by graduate assistantships from the Department of Statistics at Texas A&M University and partly supported by research fellowships from Biogen company.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

1. INTRODUCTION

This dissertation consists of three independent studies. The first and second studies are both motivated by wind power data while they have different perspectives and objectives. The third study is motivated by amyotrophic lateral sclerosis (ALS) disease. Each of the following chapters will describe each of these three studies, respectively, and we here briefly address what specific question arises, how challenging research is, which statistical field is applied, and so forth. The last chapter will provide summary and discussion of this dissertation.

Chapter 2: Covariate Matching Methods for Testing and Quantifying Wind Turbine Upgrades

In the wind industry, engineers perform retrofitting upgrades on in-service wind turbines for the purpose of improving power production capability. People often wonder about the upgrade effect: whether it indeed improves a turbine's performance, and if so, how much. Since atmosphere dynamics vary over time, specifically before and after upgrade, it is critical to have environmental effects controlled for while comparing power output difference. In this study, we devise a matching method to ensure the environmental covariates to have comparable distribution profiles before and after the upgrade.

Chapter 3: Joint Estimation of Monotone Curves via Functional Principal Component Analysis

The second study has a standpoint of functional data analysis in that wind power data formulate so-called wind power curves, which explain functional relationship between wind power output and wind speed input. A study on the estimation of power curves is important to describe turbine performance in wind farm management, however, it is challenging due to not only the monotonicity of power curves but also the large variance and irregularity of data observations. We develop a functional principal component model that can perform a joint estimation of a collection of monotone curves; also it can describe important modes of curve variation at the same time.

Chaper 4: Statistical Modeling on Spatio-temporal Binary Data for Describing Infectious Disease Spreading Pattern

Amyotrophic lateral sclerosis (ALS), also known as Lou Gehrig's disease, is a neurological disease in which motor nerve cells in the brain and the spinal cord that undergo degeneration. The onset of ALS tends to be focal, typically affecting first a particular group of muscles in one body region and then spreading to other regions as the disease progresses. Motivated by such ALS disease, the main interest in this study is to investigate the progression and spreading patterns of any infectious disease over space and time, provided that data are binary responses assessing whether infected or not. To that end, two modeling approaches are proposed, based on different perspectives.

2. COVARIATE MATCHING METHODS FOR TESTING AND QUANTIFYING WIND TURBINE UPGRADES

2.1 Introduction

Wind power is one of the fastest growing renewable energy resources (DOE, 2015). As large wind farms are built, cost considerations are essential for effective wind farm management (Byon et al., 2013). One of the costly management actions for in-service turbine fleet is to perform retrofitting upgrades, so that the outdated or malfunctioning wind turbines can restore or even improve their power generation capability (Khalfallah and Koliub, 2007). It is, therefore, not a surprise that operators want to know whether the benefits from an upgrade outweigh the expenses of doing it, including material and labor cost. This inquiry motivates researchers to scrutinize the turbine performances before and after the action of upgrade. It becomes the research question we aim to answer in this paper, and if an upgrade does indeed improve the turbine performances, we also want to quantify the improvement.

When it comes to comparing turbine performances between the periods before and after an upgrade, it is unreasonable to merely compare power output of the two periods because wind power generation is affected by an array of environmental covariates, such as wind speed, wind direction, temperature, air pressure and other atmosphere dynamics. Each of the environmental covariates observed before the upgrade may probabilistically distribute differently from the period after the upgrade. These incomparable input conditions cause different wind power outputs and could mislead the conclusion: for example, if too many windy days are there after the upgrade, high power generation might happen due to not only the upgrade effect but more so due to the high wind speed. For a fair comparison, therefore, these environmental effects need to be controlled for while comparing

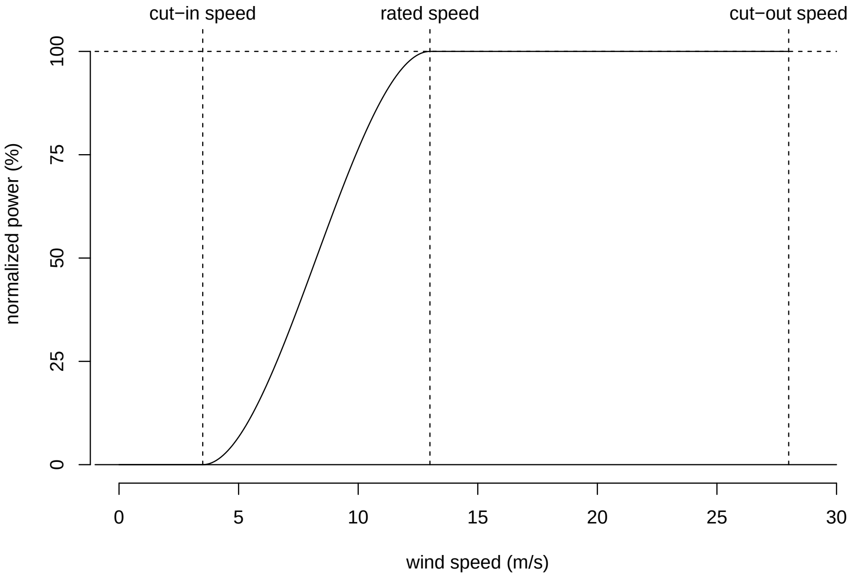


Figure 2.1: Wind power curve. Wind turbine produces higher power as wind speed increases. A turbine starts power production at the cut-in speed, reaches its full operation at the rated speed, and stops producing power at and beyond the cut-out speed. Power outputs are normalized by the rated power.

power outputs.

To handle the problem explained above, the dominating school of thought is to establish a model estimating wind power outputs conditioned on the observations of environmental covariates, so that the model can be used to compare the estimated power outputs between two periods by setting the same input conditions. Such a model, if taking wind speed as the single input, is known as the power curve, explaining the functional relationship between wind power output and wind speed input (Ackermann and Söder, 2005); Figure 2.1 presents an example.

To estimate the power curve using the actual wind speed and power observations, the International Electrotechnical Commission (IEC, 2005) recommended the use of a binning method, which discretizes wind speed into intervals of, say, 0.5 meters per second (m/s) width and then uses the wind powers and wind speeds, averaged in respective intervals, to fit a smooth curve. Other curve fitting methods are also developed for estimating the power curve based on wind speed (Yan et al., 2009; Kusiak et al., 2009; Uluyol et al., 2011; Osadciw et al., 2010; Albers, 2004), but they may be different from the binning method in specifics.

A common drawback of the IEC like approaches is that they all count the wind speed too heavily as the factor driving the power production. While it is true that wind speed is the most significant effect in wind power generation, other environmental effects cannot be ignored. In an effort to include other environmental factors into an extended power curve model, the effect of wind direction was incorporated, in addition to wind speed (Nielsen et al., 2002; Sanchez, 2006; Pinson et al., 2008; Jeon and Taylor, 2012; Wan et al., 2010). Most recently, Lee et al. (2015a) and Lee et al. (2015b) developed one of the first truly multivariate-dependency wind power models that allows all aforementioned environmental covariates to be included. Understandably, such a model, if fitted separately before and after an upgrade, could be used to compare a turbine's performance by setting the input conditions to the same values.

We in this paper advocate a different approach, and its basic idea is as follows. Suppose that one can select a large enough subset of wind turbine data before and after an upgrade, such that they have comparable distribution profiles of the environmental covariates. Then one can simply compare the wind power outputs of the two periods within that selected subset. The appeal of such a direct comparison approach is its simplicity. Unlike the model-based approaches (fitting a power curve is to estimate a model), it relies on fewer assumptions. Additionally, the direct comparison approach is quick to be carried out in practice, and its working mechanism is easy to be understood by engineers. The last point is important because a method is less likely to have real impact in practice until it is understood and thus accepted by practitioners.

Covariate matching methods are rooted in the statistical literature. In stabilizing the non-experimental discrepancy between the non-treated and treated subjects of observational data, Rubin (1973) adjusted the covariate distributions by selecting non-treated subjects that have a similar covariate condition as that of the treated ones. Through the process of matching, the non-treated and treated groups become only randomly different on all background covariates, as if these covariates were designed by experiments. As a result, the outcomes of the matched non-treated and treated groups, which keep the originally observed values, are comparable under the matched covariate conditions. For more discussion on covariate matching methods, please refer to Stuart (2010).

In this paper, we propose a covariate matching method tailored towards the wind application, in which, a turbine before the upgrade and the same turbine after the upgrade correspond to the non-treated and treated subjects, respectively. We follow the four key steps for a matching method, introduced in Stuart (2010), of which the first three represent the ‘design’ of a matching method, whereas the fourth represents the ‘analysis’ of the matched outcomes:

1. Define the measure of closeness;
2. Implement a matching method;
3. Diagnose the quality of the resulting matched samples;
4. Analyze the outcome and estimate the treatment effect.

Specifically in our approach, we use a Mahalanobis distance (Mahalanobis, 1936) in Step 1 to determine whether an individual is a good match to another. In Step 2, we adopt an idea of the $k:1$ nearest neighbor matching method (Rubin, 1973). In Step 3, we rely primarily on the density plots as our diagnostic tool. As the last step, we analyze the matched outcomes through paired *t*-tests and compute the improvement an upgrade makes.

We want to note that in the field of wind power analysis, there exists 'analog' technique, which has a similar idea with matching methods in that it searches and utilizes a set of observations that have the most similar weather condition to the specific time point. Since the analog approach aims at forecasting, it then estimates the probability distribution of the future state of the atmosphere (Delle Monache et al., 2013). The covariate matching methods discussed above, including our proposed one differ from the analog forecasting approach in that the covariate matching aims at investigating treatment effect, or specifically, the upgrade effect in our context. They do so without any estimation procedure. Another difference is that the analog method follows a timeline to find the most similar weather path to the time of interest, whereas the covariate matching methods break the time order of non-treated records to construct a counterpart of the treated ones.

The remainder of the paper is written in the following order. Section 2.2 first introduces the data structure. In Section 2.3, we describe the proposed matching method for handling the wind turbine data. Section 2.4 presents the outcome analysis, including the quantification of the upgrade effect. Section 2.5 performs a sensitivity analysis to verify our approach's capability in estimating the upgrade effect and to compare it with a power curve modeling approach. We make a few further remarks concerning the proposed matching method in Section 2.6.

2.2 Data Structure

In this study, we use the data obtained from the authors of Lee et al. (2015b). For this reason, we study the same two upgrade cases as in Lee et al. (2015b). We would like to explain briefly the setting under which the data are obtained.

This study involves two pairs of turbines, which are distant apart enough, so that one pair of turbines does not affect the other pair. Within a pair, one turbine is called a test turbine on which an upgrade is applied, while the other one is called a control turbine of which no change is made. We deem the two turbines in a pair are identical for practical considerations, as they are of the same type from the same manufacturer and started their service at the same time. Both turbines in each pair are also associated with a meteorological mast, which houses sensors to measure several environmental conditions. Figure 2.2, similar to Figure 5 in Lee et al. (2015b), illustrates the layout of the two turbine pairs and their associated mast.

As in Lee et al. (2015b), we consider two types of upgrade: one is known as the vortex generator installation (Øye, 1995) and the other one is a pitch angle adjustment (Wang et al., 2012); both actions are believed to make the upgraded turbine produce more wind power under the same environmental conditions. The vortex generator installation is physically carried out on the test turbine in the experimental pair, whereas the pitch angle adjustment is not physically carried out on a test turbine but simulated. We call the turbine pair with the simulated upgrade the mimicry pair.

The data modification is done to the test turbine data in the mimicry pair as follows: the actual wind turbine data, including both power production data and environmental mea-

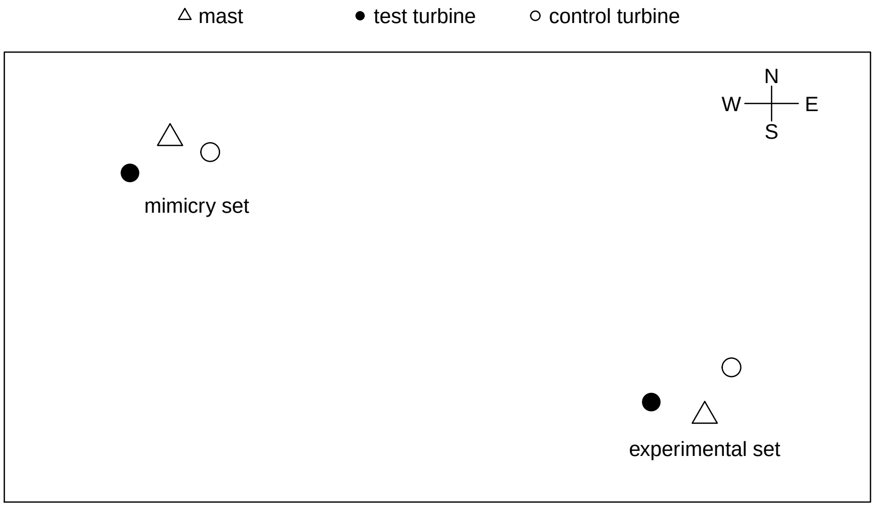


Figure 2.2: Wind farm layout. This layout shows the relative locations of the turbines and masts on the wind farm. Wind power production is measured at the respective turbine, and environmental conditions are measured by the sensors at the nearby meteorological mast. The experimental pair includes an actually-upgraded test turbine (a vortex generator installation) and its control turbine, while the mimicry pair includes an artificially-upgraded test turbine (a pitch angle adjustment) and its control turbine.

surements, are taken from the actual turbine pair operation. Then, for the designated test turbine, the power from that turbine on the range of wind speed over 9 m/s is increased by 5%, namely multiplied by a factor of 1.05; see Figure 2.3 for an illustration. This simulation of pitch angle adjustment is motivated by Wang et al. (2012). Including the simulated data set in our study helps us get a sense of how well a proposed method can detect a power production change due to an upgrade and how accurately it can quantify the change.

We denote the power produced by a turbine (in kilowatts) by P, so that P^{ctrl} and P^{test} are associated with a control turbine and a test turbine, respectively. In our reporting of the analysis, including in the power curve plots, the power values are normalized by the rated power, to protect the identity of the turbine manufacturer and the wind farm operator.

The environmental conditions directly measured at a meteorological mast are: wind

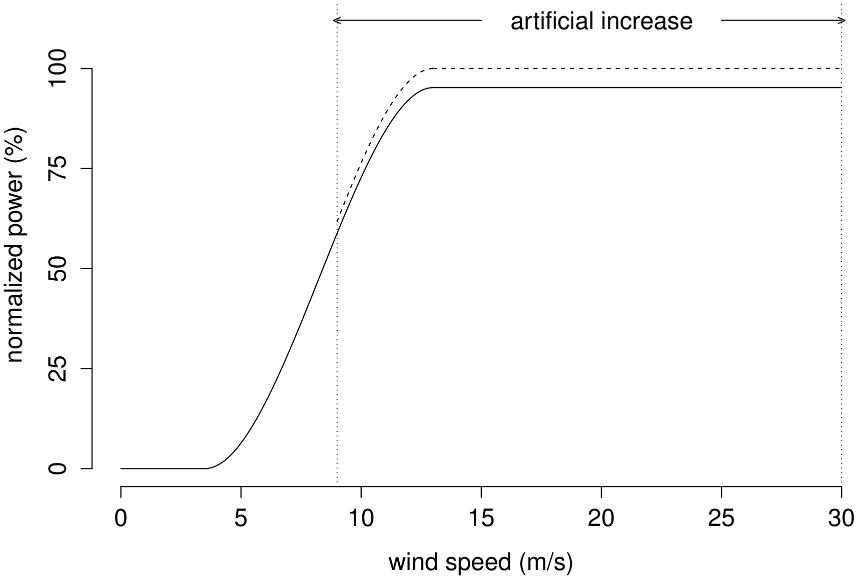


Figure 2.3: Modification in the mimicry test turbine data as if a pitch angle adjustment were applied. The power on the range of wind speed over 9 m/s is increased by 5%.

speed, V, wind direction, D, ambient temperature, T, and air pressure, Q. Using these measurements, values of additional environmental covariates can be computed, including air density, A, wind shear, W, and turbulence intensity, I, using the following formulas:

- air density, $A = \frac{Q}{R \cdot T}$ (kg/m^3), where $R = 287$ (Joule/(kg·K)) is a gas constant;
- wind shear, $W = \frac{\ln(V_2/V_1)}{\ln(g_2/g_1)}$, which represents the vertical variation of wind, where V_1 and V_2 are the wind speeds measured at heights $g_1 = 80$ m and $g_2 = 50$ m respectively;
- turbulence intensity, $I = \frac{\hat{\sigma}}{V}$, where $\hat{\sigma}$ is the standard deviation of wind speed in a 10-minute duration.

The covariates W and I measure certain aspects of the atmospheric dynamics that wind speed itself does not fully represent. Air density represents the combined effect of temperature and pressure; once air density is included to explain wind power output, temperature and pressure are no longer needed.

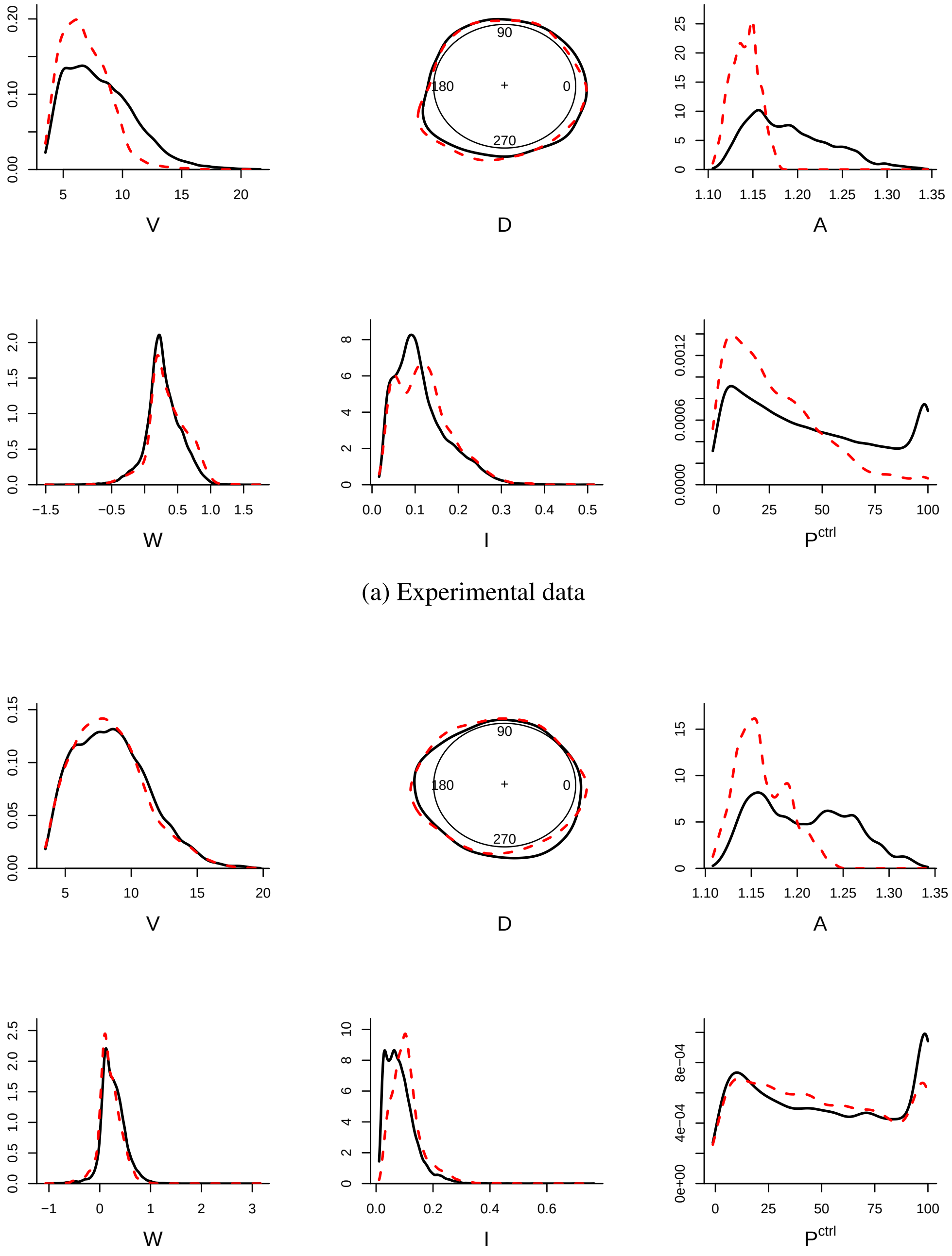
As such, each data set has five explanatory covariates, (V, D, A, W, I), and two power outcomes, $(P^{\text{ctrl}}, P^{\text{test}})$. Note that wind turbine data are arranged into 10-min blocks, so that the values of (V, D, A, W) are the averages of the 10-minute intervals and I is the ratio of the standard deviation of wind speed in a 10-min block over the average wind speed of the same block. This 10-min block data arrangement is commonly used in the wind industry.

For the experimental pair, we have 14 months worth of data in the non-treated period (i.e., before the upgrade) and 5 weeks worth of data in the treated period (i.e., after the upgrade), whereas for the mimicry pair, we have 8 months worth of data in the non-treated period and 7 weeks in the treated period. Note that it is preferable to have a much larger set in the non-treated period than the treated. It is because the sufficiently large candidate pool to match can avoid too many of repeatedly selected individuals, and therefore the matched subset of non-treated period reflects reality such as varying weather conditions.

2.3 Matching Methods

Our investigation starts off with exploring the discrepancy of the covariate distributions. Figure 2.4 demonstrates for each covariate the difference in empirically fitted density functions between the non-treated and treated periods. The last subplot in both the upper and lower panel is the density function of the power output of the respective control turbine. For a control turbine, as it is not modified, the distribution of its power output is supposed to be comparable, should the environmental conditions be maintained the same. But the data show otherwise, suggesting the existence of environmental influence, which confounds the upgrade effect in power output.

Let us introduce a few notations and terminologies. The environmental covariate vector is denoted by $\mathbf{X}$, and $\mathbf{X} = (V, D, A, W, I)^T$ in this study but it can include more



(a) Experimental data

(b) Mimicry data

Figure 2.4: Overlapped density functions of unmatched covariates and power output of control turbine; solid line = before upgrade (non-treated), dashed line = after upgrade (treated).

variables, should their measurements be available. The data pair $(\mathbf{X}, P)$ forms a data record, containing the value of the environmental covariates and its corresponding power output. The data record collected before the upgrade forms the non-treated data group, whereas those collected after the upgrade forms the treated group. Let S_{bef} and S_{aft} be the index set of the data records in the non-treated and treated group, respectively. Let Y_S denote the values of covariate Y for data indices in S. For example, $V_{S_{\text{bef}}}$ is the vector of all wind speed values that are observed before the upgrade.

This section presents the matching method to create a comparable distribution profiles of covariates. Before going through the four-step procedure of developing a matching method, as mentioned in Section 2.1, we first describe the preprocessing steps in Section 2.3.1 and 2.3.2. Then, Section 2.3.3 - 2.3.5 describes Step 1, 2 and 3, respectively. Step 4 is discussed in Section 2.4.

2.3.1 Hierarchical Subgrouping

The first action of preprocessing is to narrow down the set from which we will perform the data record matching subsequently. The reason for this preprocessing is to alleviate the computational demand arising from exhausting the pairwise combinations when comparing data records in two large size data sets.

This objective is fulfilled via a procedure we label as the hierarchical subgrouping. The idea goes as follows.

1. Locate a data record in the treated group, S_{aft}, and label it by the index j.

2. Select one of the covariates, for instance, wind speed, V, and designate it as the variable on which we measure similarity between two data records.

3. Go through the data records in the non-treated group, S_{bef}, by selecting the subset of data records such that the difference in terms of the designated covariate between the

data record j in S_{aft} and any one of the records in S_{bef} is smaller than a pre-specified threshold. When V is in fact the one designated in Step 2, the resulting subset is then labeled by placing V as the subscript to S, namely S_V.

4. Next, designate another covariate and use it to prune S_V in the same way as one prunes S_{bef} into S_V in Step 3. This produces a smaller subset nested within S_V. Then continue with another covariate until all covariates are used.

The order we apply the covariates in the above hierarchical subgrouping procedure is based on the importance of them in affecting wind power output, which, according to Lee et al. (2015a), is V, D, A, W, and I, from the most important to the least important. We will discuss more about the matching order of covariates in Section 2.6.1. Also note that wind direction D is a circular variable and the absolute difference between two angular degrees is between 0 and π; we then adopt a circular variable formula from Jammalamadaka and Sengupta (2001) to calculate the difference between two D values.

The above process can also be written in set representation. For data record j in S_{aft}, we define the subsets of data records in S_{bef}, hierarchically chosen, as

$$\begin{aligned}
S_V &:= \{i \in S_{\text{bef}} : |V_i - V_j| < \alpha_V \sigma(V_{S_{\text{bef}}})\};\\
S_D &:= \{i \in S_V : \pi - |\pi - |D_i - D_j|| < \alpha_D \sigma(D_{S_V})\};\\
S_A &:= \{i \in S_D : |A_i - A_j| < \alpha_A \sigma(A_{S_D})\};\\
S_W &:= \{i \in S_A : |W_i - W_j| < \alpha_W \sigma(W_{S_A})\};\\
S_I &:= \{i \in S_W : |I_i - I_j| < \alpha_I \sigma(I_{S_W})\},
\end{aligned}$$

where $\sigma(Y)$ is the standard deviation of Y and α_Y is the thresholding coefficient. We discuss how to determine α's in Section 2.3.5. This hierarchical subgrouping establishes the subsets nested as such, $S_I \subset S_W \subset S_A \subset S_D \subset S_V \subset S_{\text{bef}}$. Consequently, data records in the last hierarchical set S_I have the closest environmental conditions compared with data record j in S_{aft}.

This hierarchical subgrouping procedure shares certain similarity with the coarsened exact matching (CEM) approach (Iacus et al., 2012) in that it performs the data record matching on broader ranges of covariates and builds factor-sized strata. Unlike CEM, however, the strata from our procedure have a hierarchical, nested structure that CEM does not have.

2.3.2 Unmeasured Factors

There could be other environmental conditions, in addition to V, D, A, W and I, that affect wind power production, but they are not measured. For instance, humidity is one variable that was shown to have appreciable impact on wind power production for offshore wind turbines (Lee et al., 2015a) but for the wind farm data we worked with, the humidity was not measured.

The possible existence of unmeasured environmental factors presents the risk of causing distortion in power output comparison, even when the aforementioned measured environmental factors are matched between the treated and non-treated groups. In order to alleviate this risk, we make use of the power output of the control turbine in each turbine pair, P^{ctrl}. What we propose to do is to further narrow down from the most nested subset produced in Section 2.3.1, S_I, by selecting those which have a comparable P^{ctrl} value to that of data record j in S_{aft}. Specifically, this amounts to continuing the hierarchical subgrouping action in Section 2.3.1, producing a S_P, subset of S_I, based on P^{ctrl}, such that

$$S_P := \{i \in S_I : |P_i^{\text{ctrl}} - P_j^{\text{ctrl}}| < \alpha_P \sigma(P_{S_I}^{\text{ctrl}})\}.$$

We perform this procedure for all data records in the treated group so that each j in S_{aft} has its matched set $S_{P,j}$. In the case that $S_{P,j}$ is an empty set, we then discard the respective index j from S_{aft}. Because of this, S_{aft} may shrink after the subgrouping steps.

What we do in this subsection is essentially to use the control turbine to calibrate the conditions affecting the test turbine. A similar idea was tried by Albers (2004) but his approach is different from ours. Albers used a power curve based approach, in which the author fitted a *relative* power curve between the control and test turbines and hoped using that can calibrate the conditions for the test turbine. The rationale behind Albers's relative power curve is not as transparent as our subgrouping procedure and that approach is still model based rather than direct comparison; in fact, it involved more modeling steps in its analysis.

2.3.3 Mahalanobis Distance

$S_{P,j}$ is the set of candidate matches of data records in the non-treated group to data record j in the treated group. Our next goal is to choose the data record in $S_{P,j}$ that is the closest to data record j. For this purpose, we need to define a dissimilarity measure to quantify the closeness between two data records.

We decide to use the Mahalanobis distance (Mahalanobis, 1936) as our dissimilarity measure, which is popularly used in the context of multivariate analysis. It re-weighs the Euclidean distance between two covariate vectors with the reciprocal of a variance-covariance matrix. Before presenting the definition of the Mahalanobis distance between two wind turbine data records, we first introduce a transformed covariate vector, denoted by $\mathbf{X}^*$, such that

$$\mathbf{X}^* = (V \cos D, V \sin D, A, W, I)^T.$$

Using $\mathbf{X}^*$ makes it easier to deal with the circular wind direction variable D. The Mahalanobis distance ($\texttt{MD}_{ij}$) between data record j in S_{aft} and data record i in $S_{P,j}$ is defined as

$$\mathtt{MD}_{ij} := \sqrt{(\mathbf{X}_i^* - \mathbf{X}_j^*)^T \Sigma^{-1} (\mathbf{X}_i^* - \mathbf{X}_j^*)},$$

where $\Sigma = \mathrm{Cov}(\mathbf{X}^*_{S_{\text{bef}}})$. Obviously, the larger an $\mathtt{MD}$ value, the more dissimilar two data records.

Alternatively, propensity scores can be used as a dissimilarity measure (Rosenbaum and Rubin, 1983). Propensity scores have an advantage for a large number of covariates, whereas the Mahalanobis distance works quite well when there are fewer than eight continuous covariates (Zhao, 2004). Moreover, since the Mahalanobis distance can reflect an interaction among covariates, which indeed exists in our data as described in Section 2.6.1, we choose the Mahalanobis distance rather than the propensity scores.

2.3.4 One-to-one Matching

As the simplest form of $k : 1$ nearest neighbor matching, introduced by Rubin (1973), we perform $1 : 1$ matching; it selects for each treated subject j the non-treated subject with the smallest distance from j. There is a difference in our matching in that the size of matching candidates for each treated subject is primarily reduced by the previous steps, so there is no need to search within the entire non-treated group but within its subgroup.

In set representation, given $S_{P,j}$ and $\mathtt{MD}_{ij}$ from Section 2.3.2 and 2.3.3, respectively, we select data record i_j in $S_{P,j}$ that has the smallest Mahalanobis distance as the best match to data record j in S_{aft}. The data record i_j is found through

$$i_j = \arg\min_{i \in S_{P,j}} \mathtt{MD}_{ij},$$

for each j in S_{aft}. In case that two or more are tied for the smallest value, we choose one of them randomly. After this step, each data record j in the treated group has one non-treated counterpart i_j, with the exception of those already discarded during the subgrouping step.

We define an index set of the matched data records from the non-treated group as $S^*_{\text{bef}} := \{i_j \in S_{\text{bef}} \,|\, j \in S_{\text{aft}}\}$. As such, S_{aft} is now individually paired to S^*_{bef}.

It should be noted that we allow replacement to achieve a fair matching because there is no presumed order to be paired in advance among data records in S_{aft}. In other words, i_j is not eliminated from the candidate set S_P, even though it has matched to j once. When the next data record $j+1$ is selected from S_{aft}, the same non-treated data i is thus possible to be matched again. We will discuss further issues related to matching with replacement in Section 2.6.2.

2.3.5 Diagnostic

After performing the matching procedure, it is crucial to diagnose how much discrepancy of the covariate distributions has been removed, as compared to the unmatched data set. Only after the diagnostics signifies a sufficient improvement, an outcome analysis is then ready to perform in the next step.

We measure the discrepancy of distribution in two ways, numerically and graphically. For numerical diagnostics, the standardized difference of means (`SDM`) is used as a measure of the dissimilarity of a covariate between the treated and non-treated groups (Rosenbaum and Rubin, 1985):

$$\frac{\overline{Y}_{S_{\text{aft}}} - \overline{Y}_{S_{\text{bef}}}}{\sigma(Y_{S_{\text{aft}}})},$$

where Y is one of the covariates, and $\overline{Y}_S$ represents the average of Y in the set S. The `SDM` decreases if the matching procedure indeed reduces the discrepancy between the two groups. As shown in Table 2.1, `SDM` decreases significantly for all covariates. A previous study (Rubin, 2001) found that `SDM` should be less than 0.25 to be trustworthy adjustment. Otherwise, the differences between the distributions of covariates in the two groups are regarded as substantial.

For graphical diagnostics, we overlap the empirical density function of each covari-

Table 2.1: Numerical diagnostics; see the decrease of SDM after matching; the matching procedure indeed reduces the discrepancy between the two periods.

	V	D	A	W	I	P^{ctrl}
Unmatched	0.6685	0.0803	3.2715	0.2312	0.1382	0.8132
Matched	0.0142	0.0026	0.0589	0.0721	0.0003	0.0083

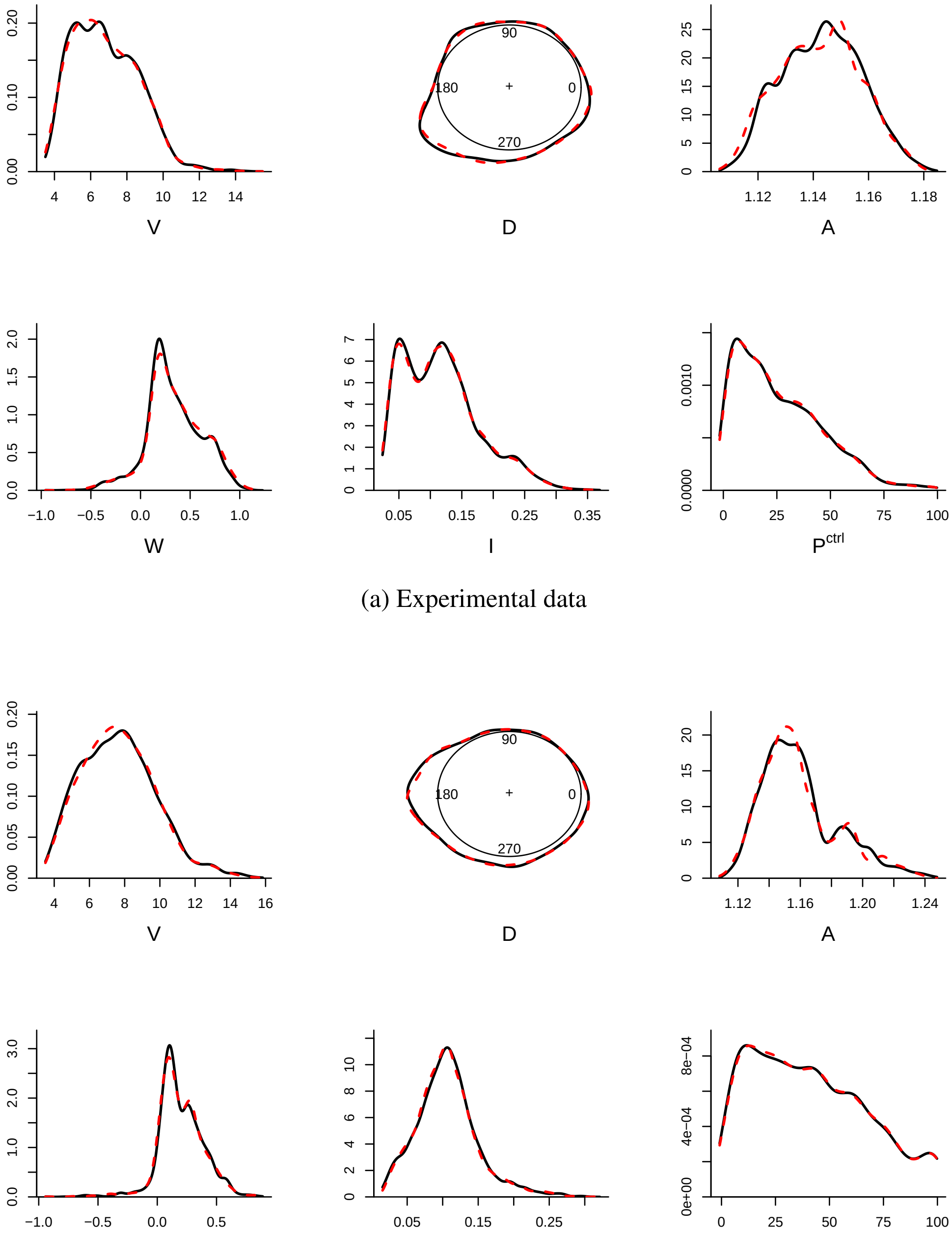
(a) Experimental data

	V	D	A	W	I	P^{ctrl}
Unmatched	0.0605	0.1647	1.6060	0.2759	0.4141	0.0798
Matched	0.0077	0.0029	0.0263	0.0158	0.0111	0.0036

(b) Mimicry data

ate as well as that of the control turbine power, associated with the treated group and the matched subset of non-treated group. We can visually inspect the discrepancy between the two density functions and see if they are similar enough. An example is shown in Figure 2.5, in which we observe well-matched distributions of covariates after the matching process. These improvements in term of distribution similarity are clear when compared to Figure 2.4, which demonstrates the dissimilarity in covariate distributions of the unmatched original set.

If the diagnostic procedure does not provide credible evidence to perform an outcome analysis, for instance, SDM increases or exceeds 0.25, or any non-overlapped bumps is notable in matched density plots, we then return to Section 2.3.1 and 2.3.2 and adjust the degree of truncation α's until well-matched set is obtained. It should also be noted that, if the size of S_{aft} after matching loses too many data records, and this can happen when too small α's are applied, we suggest to enlarge the size of S_{aft} prior to the matching process. It is because weather conditions of matched S_{aft} are desired to be representative of general atmosphere spectrum, not of specific state of the weather.



(a) Experimental data

(b) Mimicry data

Figure 2.5: Overlapped density functions of matched covariates as well as that of power output of control turbine; solid line = before upgrade (non-treated), dashed line = after upgrade (treated). Compare to Figure 2.4, notice the improvement in agreement between the pairs of density plots.

2.4 Outcome Analysis

This section describes the outcome analysis, Step 4 of a matching method as outlined in Section 2.1. It fulfills the research goal of testing the significance of the upgrade effect and quantifying the improvement in terms of extra power production under comparable environmental conditions.

2.4.1 Paired *t*-tests

From the matching procedure, we have the paired indices of data records of the two groups, (i_j, j) where $i_j \in S^*_{\text{bef}}$ and $j \in S_{\text{aft}}$. Using these paired indices, we can retrieve the paired test power outcomes, $(P^{\text{test}}_{i_j}, P^{\text{test}}_{j})$. The power output pair can be interpreted as repeated measurements under comparable environmental conditions, which makes the power output also comparable.

As such, we apply a paired t-test to analyze the difference of the two paired test outcomes, $D_j = P^{\text{test}}_{j} - P^{\text{test}}_{i_j}$, since the assumption of independence is met – this will be reviewed in Section 2.6.2. It tests the null hypothesis that the expected mean of differences is zero, $H_0 : E(\overline{D}) = 0$ where $\overline{D}$ is the sample mean of $\{D_j; j \in S_{\text{aft}}\}$. Accordingly, the test statistic is

$$t = \frac{\overline{D}}{s/\sqrt{n}}$$

where s and n is the sample standard deviation and the sample size of $\{D_j; j \in S_{\text{aft}}\}$, respectively. If the test concludes a significant positive mean difference, the upgrade on the test turbine is then concluded as effective.

In Table 2.2, the first and second cells show the results from the paired t-test. In both datasets, the tests show a significant effect due to upgrade with 0.05 level of significance. We know that the simulated pitch angle adjustment upgrade has an effect less than 5% and the VG effect on this particular pair of turbines could be even smaller. So the analysis sug-

Table 2.2: Outcome analysis; paired *t*-tests and upgrade quantification

t-stat	p-value	`UPG`
3.015	0.003	1.13%

(a) Experimental data

t-stat	p-value	`UPG`
7.447	< 0.0001	3.16%

(b) Mimicry data

gests that the proposed method is sensitive enough to a moderate change of that magnitude in power production.

2.4.2 Quantification

Reporting a percentage value representing the relative increase in power production is a typical way to quantify the improvement of the test turbine's performance after the upgrade. As such, we quantify the upgrade effect (`UPG`) by computing

$$\texttt{UPG} := \frac{\sum_{j \in S_{\text{aft}}} (P_j^{\text{test}} - P_{i_j}^{\text{test}})}{\sum_{j \in S_{\text{aft}}} P_{i_j}^{\text{test}}} \times 100,$$

where $i_j \in S_{\text{bef}}^*$ is the counterpart of $j \in S_{\text{aft}}$.

The quantification results are shown in the third cell of Table 2.2. Recall we have increased the test turbine power in the mimicry pair, by 5% for wind speed 9 m/s and above, which translates to a 3.11% increase for the whole wind spectrum. Our quantification shows an improvement of 3.16% overall, which appears to present a fair agreement with the simulated amount. If the quantification amount is to be trusted, the upgrade of vortex generator installation enables a turbine to produce 1.13% more wind power than without the upgrade.

2.4.3 Mean Comparison

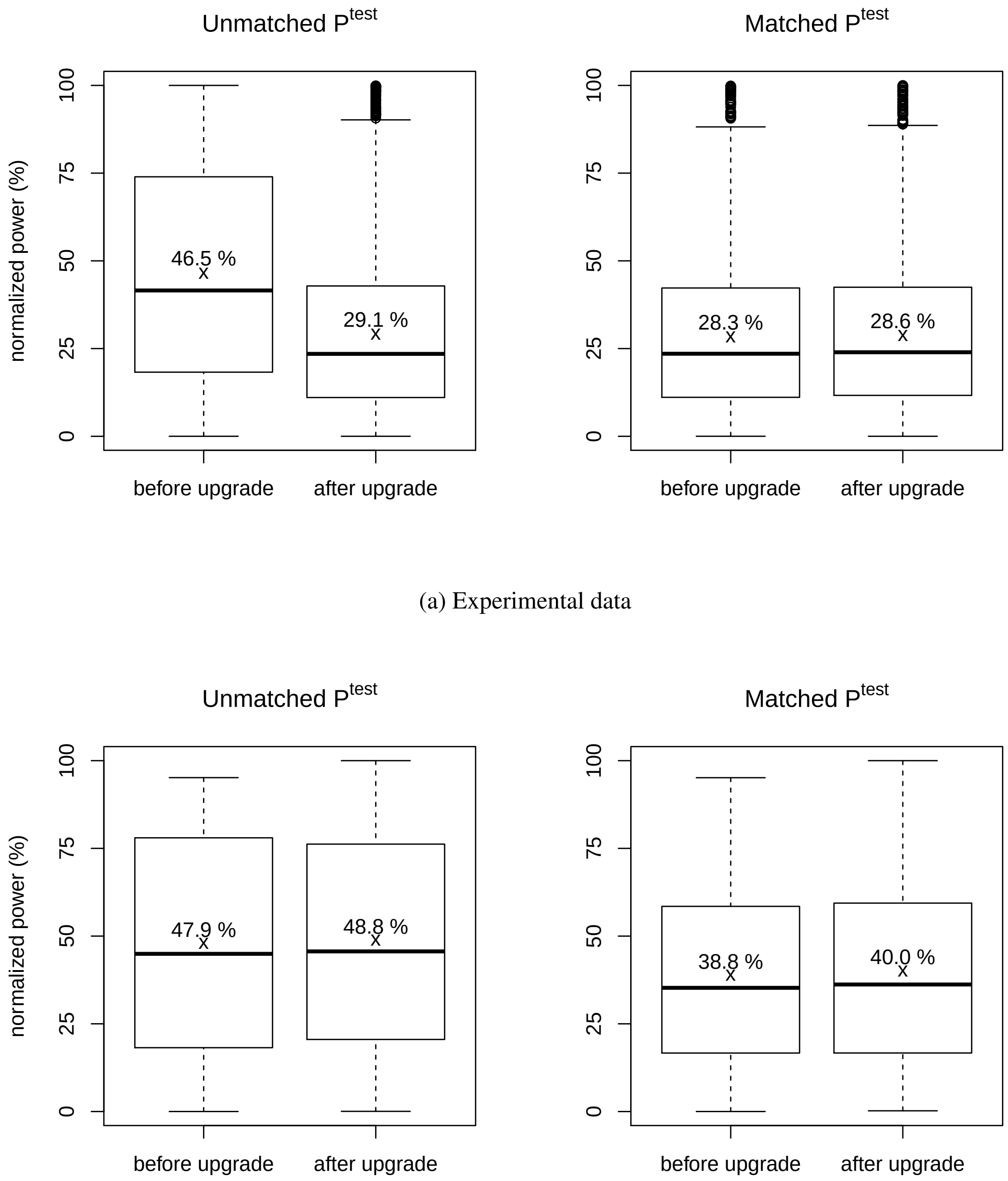
In Figure 2.6a, we present the boxplot of P^{test} data for both datasets under the unmatched conditions (i.e., original data) and the matched conditions (i.e., a subset of the original data). We noticed that unmatched data of experimental set show a higher mean power before the upgrade than after. This mean power pattern is, however, reversed on the matched data, as expected. The interpretation of the mean power pattern of the unmatched data is obvious: the difference in environmental covariates causes the wind turbine to produce, on average, more wind power in the before-upgrade period, so much so that the upgrade effect is overwhelmed and not detectable. Even though the unmatched data seemingly shows improvement in power production like mimicry data in Figure 2.6b, the imbalanced profile of weather conditions should be noticed and so the matching is required to stabilize their discrepancy. This analysis demonstrates the benefit of executing this matching procedure before comparing test power outputs and quantifying its net effect.

2.5 Sensitivity Analysis

Recall the mimicry pair is analyzed for the purpose of getting a sense of how well a proposed method can estimate a power production change, owing to turbine upgrade. While only 5% mimicked improvement is addressed when illustrating the methodology in Section 2.3 and 2.4, this section re-performs the matching on various mimic degrees. There are two reasons for this practice: (a) to see how sensitive the proposed method is in estimating a power production change at various magnitudes of the change (in Section 2.5.1), and (b) to compare the proposed matching method to the kernel plus method in Lee et al. (2015b) (in Section 2.5.2).

2.5.1 Sensitivity of Estimating Changes

In the mimicry case, the nominal power increase rate, denoted by r, is applied only to a partial range of wind power outputs corresponding to wind speed higher than 9 m/s. But when quantification is implemented, it is done to the power outputs under the entire spectrum of wind speed. Therefore, when it comes to verifying the estimation quality, we



(a) Experimental data

(b) Mimicry data

Figure 2.6: Boxplots of normalized test power values; x points, referred to by the label written at the right above in percentage, are the mean of each normalized P^{test}; the upgrade effect is revealed in the matched test powers while confounded in the unmatched.

Table 2.3: r = nominal power improvement rate; r' = effective power improvement rate; `UPG` and `DIFF*` estimates r' through the matching method and the kernel plus method, respectively.

r	2%	3%	4%	5%	6%	7%	8%	9%
r'	1.25%	1.87%	2.49%	3.11%	3.74%	4.36%	4.98%	5.60%
`UPG`	1.74%	2.21%	2.68%	3.16%	3.63%	4.11%	4.58%	5.05%
`UPG`/r'	1.4	1.2	1.1	1.0	1.0	0.9	0.9	0.9
`DIFF*`	1.97%	2.56%	3.15%	3.73%	4.30%	4.86%	5.42%	5.97%
`DIFF*`/r'	1.6	1.4	1.3	1.2	1.1	1.1	1.1	1.1

should compute the effective power increase rate, denoted by r', such that

$$r' := \frac{\sum_{j \in S_{\text{aft}}} P_j^{\text{test}}\{1 + r \cdot I(V_j^{\text{test}} > 9)\} - \sum_{j \in S_{\text{aft}}} P_j^{\text{test}}}{\sum_{j \in S_{\text{aft}}} P_j^{\text{test}}}.$$

As shown in Table 2.3, as r changes from 2% to 9%, r' changes from 1.25% to 5.6%. This range of power improvement is considered practical for detection purpose. If the improvement is smaller than 1%, it is going to be really hard for detection, and given the amount of noise in wind and power measurements, no known method can do an adequate job. On the other hand, when the improvement is greater than 6%, the level of improvement becomes a bit unrealistic due to technology limitation, and the detection job also becomes easier. It is possible that even the standard IEC binning method can detect this level of change. That is why we choose this specific range to test the sensitivity of our method.

The middle two rows in Table 2.3 compare `UPG` to r'. We notice that `UPG` considerably overestimates r' when it is small (smaller than 2%) with the overestimation as much as 40% for the smallest change at 1.25%. But the estimation quality of `UPG` gets stabilized as r' increases. In fact, for the last six cases, the difference between `UPG` and r' is within 10%. This result reflects the reality that the smaller degree of turbine upgrade is indeed difficult to estimate and demonstrates the merit of the proposed matching method.

2.5.2 Comparison between Matching and Kernel Plus Method

The best benchmark method for upgrade quantification is the kernel plus method presented in Lee et al. (2015b). In this subsection, we compare the covariate matching method with the kernel plus method.

The metric of turbine improvement used in Lee et al. (2015b) is labeled as DIFF, a percentage measuring the power production difference before and after the turbine upgrade and similar to UPG in spirit. But there is a subtle difference that needs to be addressed. The kernel plus method is applied to each turbine and get their respective DIFF value, namely $\text{DIFF}_{\text{test}}$ and $\text{DIFF}_{\text{ctrl}}$, whereas obtaining UPG already involves the use of control turbine as a baseline reference through the matching process. We believe that for a fair comparison, the metric from the kernel plus method, to be compared with UPG, should be DIFF* := $\text{DIFF}_{\text{test}}$ - $\text{DIFF}_{\text{ctrl}}$ that also adjusts the test turbine outcome using the control turbine as a baseline.

This adjusted metric DIFF* is then estimated for each r, or r', and compared to r' in the last two rows of Table 2.3. As we notice here, the kernel plus also considerably overestimates the small r' values but does a better job when r' gets bigger. But the degree of overestimation by the kernel plus method is severer, and the range of its estimation error greater than 10% is broader, than the covariate matching method. For this practical range of improvement rate, the covariate matching method outperforms the kernel plus method.

If applied to the experimental turbine pair, our analysis in Section 2.4.2 shows a 1.13% improvement. The DIFF* obtained by applying the kernel plus method to the same set of data is 1.48%. The results are expected as we know that the kernel plus tends to overestimate more, and it is less accurate for either method to estimate a small improvement, which in this case could be smaller than 1%.

Please note that the DIFF* values reported here are different from those reported in

Lee et al. (2015b). This discrepancy is due to data use difference: whereas Lee et al. (2015b) use 2-week-after-upgrade worth of data in their analysis, we used in this study 7-week-after-upgrade worth of data for the mimicry turbine pair and 5-week-after-upgrade worth of data for the experimental pair, as our covariate matching requires a longer duration to ensure a sufficient amount of data.

2.6 Remarks

This section discusses further a few issues arising in our research undertaking. Section 2.6.1 reviews in more details the priority order and interaction effects of environmental covariates as well as how the right order can benefit analyses. Section 2.6.2 discusses the issue of replacement while matching data records and affirms how the independence assumption of t-test is satisfied.

2.6.1 Priority Order and Interaction of Covariates

The priority order of environmental covariates used in the hierarchical subgrouping procedure in Section 2.3.1 is in the order of wind speed, wind direction, air density, wind shear and turbulence intensity.

The importance of wind speed V is obvious and it is universally agreed to be the most important factor affecting wind power production. Wind direction D also matters a great deal even though wind turbines have a yaw control mechanism that is supposedly to track the wind direction and point the turbine towards the direction from which the wind blows. Nonetheless, a score of studies showed that this tracking is not perfect, and consequently, including wind direction as one covariate can significantly reduce the prediction error of wind power (Lee et al., 2015a; Jeon and Taylor, 2012; Wan et al., 2010). Wind speed and wind direction are two most important factors affecting wind power output.

The effect of next tier of factors, namely air density A, wind shear W and turbulence intensity I, come more in the form of interacting with the two main effects, wind speed and wind direction. Lee et al. (2015a) illustrated, in Figure 4 of their paper, the existence of interaction effects between these second-tier factors and wind speed/direction.

We believe the nested structure of our hierarchical subgrouping helps handle the priority of main and interacting covariates. The variance-covariance matrix in the Mahalanobis distance (Section 2.3.3) also captures the interaction effects through the covariance terms and incorporates them in the calculation of the dissimilarity measure.

If a priority order is poorly pre-defined, the quality of matching may not be satisfactory compared to a well-defined order. To show numerical evidence of this argument, we conducted the matching on the mimicry set with a reversed order, $P^{\text{ctrl}}, I, W, A, D, V$, and their numerical diagnostics are shown in Table 2.4. Comparing this result to Table 2.1 (b), SDMs of D, A, W and P^{ctrl} with the reversed order are greater than those with the proper order. It should be noted that degrees of truncation in Table 2.4 are same as those in Table 2.1 for a fair comparison. However, as long as those SDMs are acceptable to perform an outcome analysis, the significance and quantification of turbine improvement does not change dramatically. The analysis using the reversed order leads to a UPG $= 3.33\%$ with p-value $< .0001$, which is similar to that with the right order (UPG $= 3.16\%$, while true value = 3.11%).

Still, although the outcome analysis appears to show certain degree of robust under acceptable SDMs, one might as well make use of the priority information, if known, since it helps find the acceptable matched set much more efficiently. If a priority order of covariates is unknown, it is recommended to perform some statistical analysis using, for example, random forests (Breiman, 2001), which can measure variable importance, before applying the matching.

Table 2.4: Numerical diagnostics when matching with a reversed priority order; P^{ctrl}, I, W, A, D, V; notice less decreased `SDM`s of D, A, W and P^{ctrl} than those of Table 2.1 (b), which implies poorly defined order may lead unsatisfactory quality of matching.

	V	D	A	W	I	P^{ctrl}
Unmatched	0.0605	0.1647	1.6060	0.2759	0.4141	0.0798
Matched	0.0022	0.0036	0.0377	0.0208	0.0055	0.0085

2.6.2 Matching with Replacement and Independence

Recall in Section 2.3.4 that we allow replacement when carrying out the matching procedure. Because of this, a data record in the non-treated dataset S_{bef} could possibly be paired with two or more data records in the treated dataset S_{aft}.

A potential problem of allowing replacement is that the replication of the same data records may cause a violation of independence of outcome variables. In order to settle this issue, information about frequency weights, such as the relative number of replications, may need to be taken into account (Stuart, 2010).

In our application, however, replacement does not seem to cause too much of a problem, for the following reasons: (a) such replication happens rather rarely by starting with the much larger set of non-treated period than the treated; (b) we analyze the differences between the paired power outputs. The dependency caused by replications, if any, is considerably weakened as the differences are taken out of the treated and non-treated outcomes, because the treated outcomes do not have replications and are thus independent. Consequently, *t*-test in Section 2.4.1 is applicable since differences $\{D_j; j \in S_{\text{aft}}\}$ are independently distributed.

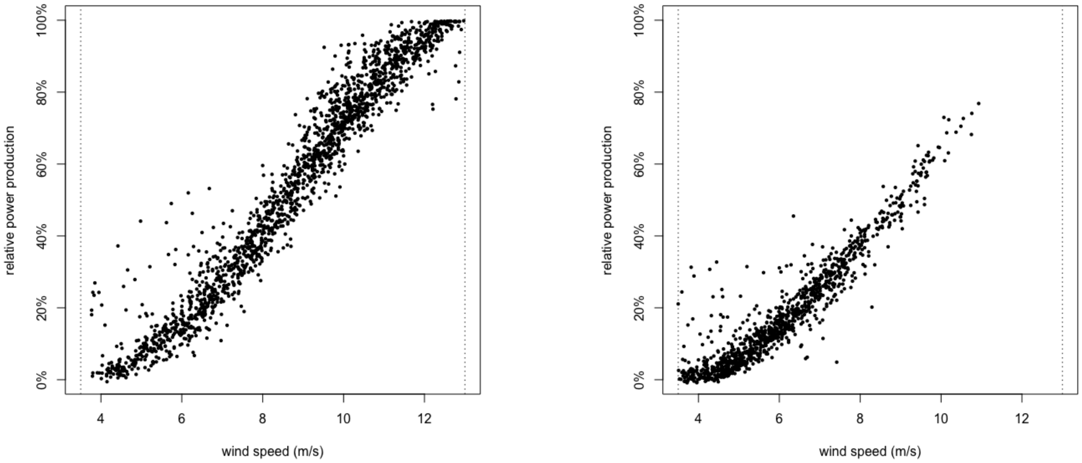
3. JOINT ESTIMATION OF MONOTONE CURVES VIA FUNCTIONAL PRINCIPAL COMPONENT ANALYSIS

3.1 Introduction

3.1.1 Research Problem and Motivating Example

There are many curve fitting techniques for function estimation; however, they are not always easily applied to real data for some practical restrictions, such as curve shape constraints, incomplete or noisy observations of each curve, and so on. Moreover, research on principal component analysis for functional data is usually studied when the changes of many curves are of interest; it is rarely done for monotone curves, though. For example, we introduce the 'wind power curve', coming from the industrial engineering fields, which originally motivates our study. The wind power curve explains the functional relationship between wind power output and wind speed input (Ackermann and Söder, 2005); see Figure 2.1. For effective wind farm management, operators are interested in estimating power curves to examine turbine performances.

There are several practical challenges in estimating wind power curves. Wind power curves are theoretically smooth and monotone increasing under the same conditions; a wind turbine produces higher power as wind speed increases. However, because of measurement errors or other environmental factors, which possibly affect the power production, wind power outputs have certain amounts of errors towards smooth power curve. Moreover, the input variable, wind speed, may have different ranges over curves and its observed values are sparse and irregular; wind blows disorderly. Figure 3.1 shows two examples of observed power curve trajectories to promote better understanding of these challenges. In this study, a question arises to estimate numerous monotone curves while their observed points are irregular and sparse on different ranges over curves.



(a)

(b)

Figure 3.1: Challenges in estimating wind power curves. (a) Irregular observations with noise. (b) Deficient observations to display a curve on an entire range during the certain period.

Likewise, not only monotone assumption but also some practical circumstances can matter in estimating curves. For the purpose of stable estimation and rich interpretation, we propose in this study functional principal components models to estimate monotone curves. By virtue of principal component analysis, we estimate numerous curves at once, rather than single curves separately. Moreover, the curve variations can be represented in a few key functions, a mean function and principal component functions, and individual characteristics of each curve can also be preserved through principal component scores.

In brief, our problem basically consists of two parts, smooth monotone function estimation and principal component analysis for functional data. Although many researchers have developed models for monotone function estimation with typical regularization techniques, and independently of monotonicity, principal component models for functional data, never before have principal component model for monotone curves been developed.

We propose the model that can do both simultaneously. In the following sections, we review the literature about monotonicity and functional principal component models.

3.1.2 Smooth Strictly Monotone Function Estimation

The monotonicity is often assumed on curves in various fields. First, the cumulative distribution function (CDF) is monotone increasing since the probability density function (PDF), the derivative of CDF, is always non-negative. As another example, the survival function, the probability function that a patient will survive beyond a specific time, is assumed to be monotone decreasing since function values are cumulative failures up to a specific time. The growth curves can also be a good example for monotone curves. In like manner, if any objects of interest have accumulative features over a continuum, the assumption of monotonicity is natural.

Many researchers have developed methods for estimating smooth monotone curves. Under the spline-based techniques, most of studies dealt with constrained coefficients estimation or constrained optimization techniques; see Ramsay (1988), Kelly and Rice (1990), Pya and Wood (2015). There are other schools of thought that analyze monotonicity with kernel-based viewpoint; Hall and Huang (2001), Hall and Müller (2003), and Mammen and Yu (2007) studied monotonicity in the nonlinear monotone regression framework.

We develop the basis spline-based model that does not require the constraint on basis coefficients, but preliminarily restricts the class of curves to liberate coefficient estimation from constraints. Specifically, we adopt the *class of monotone functions*, say $\mathcal{M}$, suggested by Ramsay (1998), that consists of functions m for which

1. $\log Dm$ is differentiable;

2. $D \log Dm = D^2 m / Dm$ is Lebesgue square integrable,

where D^r refers to a differential operator of order r. These conditions ensure that m is strictly monotone increasing and its first derivative is smooth and bounded almost everywhere. Ramsay (1998) also proves the theorem that represents the functions in this class, $m \in \mathcal{M}$, with a simple linear differential equation as

$$m = \beta_0 + \beta_1 D^{-1} \exp D^{-1} w, \tag{3.1}$$

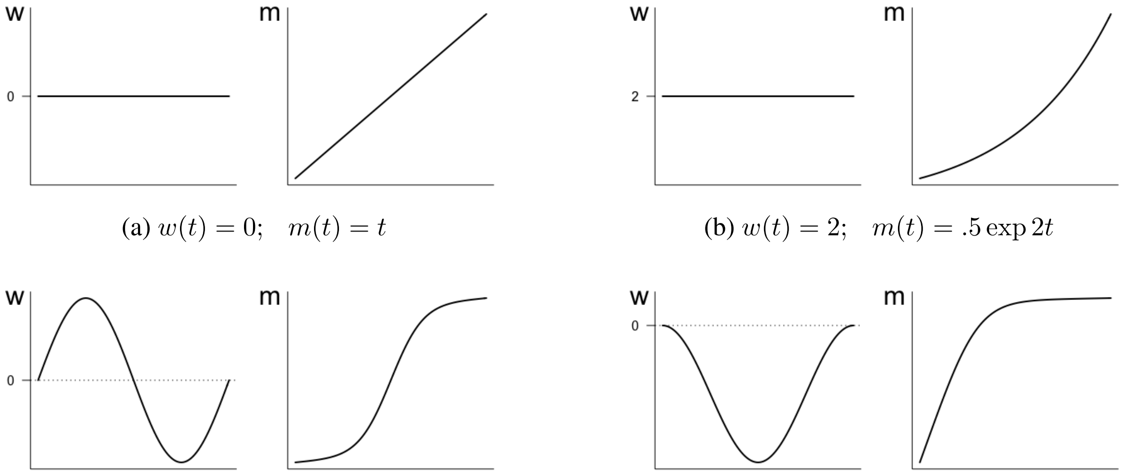
where β_0 and β_1 are arbitrary constants and the function w is a Lebesgue square integrable function such that $D^2 m / Dm$. The coefficient function w measures the 'relative curvature' of m, in that it assesses the size of the curvature $D^2 m$ relative to the slope Dm. This can also be written in the general form of integrals rather than differential equation as,

$$m(t) = \beta_0 + \beta_1 \int_{t_0}^{t} \exp \int_{t_0}^{s} w(u) \, \mathrm{d}u \, \mathrm{d}s, \tag{3.1$'$}$$

where t_0 is a lower limit of integration. See Ramsay (1998) and Ramsay (2006) for details about this monotone function representation.

Importantly, the relative curvature w in (3.1) can capture the particular shape of monotone curve m. Figure 3.2 illustrates four example curves of w and their corresponding m to show how monotone curves look like according to their relative curvatures. In the case of constant w as in (a) and (b), an explicit form of m can be obtained by simple calculus; zero $w(t) = 0$ leads a straight line $m(t) = t$, while non-zero constant $w(t) = c$ corresponds to an exponentially increasing curve, $m(t) = 1/c \, \exp ct$. The more sophisticated form of monotone curves can also be represented by a certain form of relative curvature curves as shown in (c) and (d); even though there are no explicit forms of m. Note that we set $\beta_0 = 1$ and $\beta_1 = 1$ in these figures to describe curves in a clear way.

Since a monotone curve m is now represented by an arbitrary function w in the class (3.1), Ramsay (1998) expands w to a set of smooth basis functions without any constraints. The



(a) $w(t) = 0; \quad m(t) = t$

(b) $w(t) = 2; \quad m(t) = .5 \exp 2t$

(c) $w(t) = 10 \sin 2\pi t$

(d) $w(t) = 5 \cos 2\pi t$ - 5

Figure 3.2: The relative curvature w according to the monotone curve m; See how monotone curves look like according to their corresponding relative curvatures.

basis coefficients are estimated by solving non-linear least square problems, and β's are obtained by linear regression.

Although the approach of Ramsay (1998) has an advantage of converting the problem of estimating a constrained function into that of estimating an unconstrained function, it has a limit in that it can only estimate a single monotone curve while we aim to estimate a collection of monotone curves. It could be a way to apply Ramsay (1998) to each curve separately, however, this may cause poor estimation; for example, when a curve has only few data points as in Figure 3.1 (b), the unique representation for every curves would be hardly estimated if it is estimated alone.

3.1.3 Functional Principal Component Analysis

Suppose an insufficiently-observed curve can borrow the information from entire data, the estimated curve would have a reasonable shape as other curves while preserving its own feature. For that end, we not only adopt Ramsay's monotone function class but also make use of functional principal component analysis (fPCA) to develop our model, which will be introduced in this section.

Principal component analysis (PCA) is broadly used to reduce dimension of data by a few number of important modes of variation from the mean. The traditional approach for PCA on functional data, as illustrated in Ramsay (2006), has a similar idea with the classical multivariate case but merely summation changes into integration. As multivariate PCA examines variance-covariance matrix and identifies eigenvectors, functional PCA examines variance-covariance function instead and identifies eigenfunctions as principal component functions. However, it is limited to the case that all curves are completely observed at an equally-spaced grid; see also Rao (1958), Besse and Ramsay (1986), Castro et al. (1986), Jones and Rice (1992). Although equally-spaced functional data can be augmented by projecting individually fitted curves on a fine grid, it does yet not make optimal use of the available information because it treats estimated curves as if they were observed; as James et al. (2000) addressed.

To overcome the drawbacks of traditional functional PCA, Staniswalis and Lee (1998), Besse et al. (1997) and Yao et al. (2005) proposed kernel-based approach for functional data on an irregular grid. On the other hand, James et al. (2000), Rice and Wu (2001), Zhou et al. (2008) and Guo et al. (2015) developed spline-based approaches for sparsely and unequally sampled curves. Both concern smoothness in estimating curves, kernel methods perform local smoothing by controlling a bandwidth while spline methods do smoothing in a global sense.

We adopt in this study spline model-based approaches to present fPCA. The reduce rank model suggested by James et al. (2000) can be briefly described as the followings; let

$y_i(t)$ be a measurement at time t from i^{th} curve,

$$\begin{aligned} y_i(t) &= \mu(t) + \boldsymbol{f(t)}^T \boldsymbol{\alpha}_i + \epsilon_i(t); \qquad (3.2) \\ &= \mu(t) + \sum_{k=1}^{K} f_k(t)\alpha_{ik} + \epsilon_i(t) \\ \text{subject to } \int f_k f_l &= 0 \; \forall k \neq l \text{ and } \int f_k^2 = 1 \text{ for } \forall k, \end{aligned}$$

where $\mu(t)$ is an overall mean function, $\boldsymbol{f(t)} = \{f_1(t), \ldots, f_K(t)\}^T$ is a vector of principal component functions, and $\epsilon_i(t)$ is a random measurement error. The random vector $\boldsymbol{\alpha}_i = (\alpha_{i1}, \ldots, \alpha_{iK})^T$ has mean zero and covariance Σ which is assumed to be diagonal for the simplicity. The constraints at the bottom line of (3.2) are necessary for the orthonormality of $\boldsymbol{f}$, that is, identifiable principal component functions.

The remainder of the paper is structured in the following order. In Section 3.2, we propose the functional principal component model for monotone curves. Section 3.3 presents a simulation study to see how well our model improves the traditional approaches. The application of proposed method to wind power curve data is demonstrated in Section 3.4.

3.2 Joint Estimation of a Collection of Monotone Curves

We suggest two approaches in this section. The first one simply performs existing two methods in a row; say ‘two-step’ approach. We briefly explain its procedure in Section 3.2.1 since concrete fitting algorithms merely follow their original paper; see Ramsay (1998) and James et al. (2000). On the other hand, Section 3.2.2 proposes the second approach which is the primary model performing the integrated inference of functional principal component analysis for monotone curves; rather than one at a time as two-step approach does.

The following Section 3.2.3 describe parameter estimation of the proposed model and

Section 3.2.4 consists of miscellaneous issues related to model selection.

3.2.1 Two-step Approach

One simple approach of fPCA for monotone curves is to fit curves as Ramsay (1998) suggested and succeedingly fit the reduced rank model followed by the estimated results.

Suppose we observe M monotone curves $\{y_i(t)\}_{i=1,\dots,M}$. Let $\boldsymbol{b(t)} = \{b_1(t), \dots, b_q(t)\}^T$ be a set of basis spline functions on the observed range of t. We assume each i-th curve can be represented by the class of monotone $\mathcal{M}$ as defined in (3.1). Fit the model as suggested by Ramsay (1998) for each curve separately to estimate β_{0i} and $w_i(t)$ for all $i \in \{1, \dots, M\}$; we here fix $\beta_{1i} = 1$ because of identifiability issue as discussed in Section 3.2.2. We then discretize the estimated $w_i(t)$'s on the sufficiently dense fine grid in the range of t, and consider them as if fully observed functional data. Based on these discretized data, fit the reduced rank model (3.2) as

$$w_i(t) = \mu(t) + \sum_{k=1}^{K} f_k(t)\alpha_{ik} + \zeta_i(t)$$

where $\int f_k f_l = 0 \, \forall k \neq l$, $\int f_k^2 = 1$ for $\forall k$ and $\zeta_i(t) \sim N(0, \xi^2)$. In addition, we penalize curves to be flexible in choosing the number of basis functions and orders; as Zhou et al. (2008) did.

Once all fitted with the significant PC functions, we can represent fitted monotone curves as

$$\hat{m}_i(t) = \hat{\beta}_{0i} + \hat{\beta}_{1i} \int_{c_0}^{t} \exp \int_{c_0}^{s} \hat{\mu}(u) + \sum_{k=1}^{K} \hat{f}_k(u)\hat{\alpha}_{ik} \, \mathrm{d}u \, \mathrm{d}s$$

where $\hat{\beta}_{0i}$ and $\hat{\beta}_{1i}$ are estimates from Ramsay's approach and $\hat{\mu}$, $\hat{f}_k$, and $\hat{\alpha}_{ik}$'s are from the reduced rank model. In terms of the relative curvature w, monotone curves share certain features by a mean function μ and PC functions f_k's, and retain individual characteristics via PC scores α_{ik}'s.

Although this approach is not in fact our primary suggestion, it is worth mentioning since it could improve single estimation of monotone curves m after all, by strengthening the estimation of each relative curvature w through functional PCA. The simulation study is implemented to see this improvement in Section 3.3

3.2.2 Reduced Rank Model in the class of Monontone Functions

While adopting the ideas of two approaches from Ramsay (1998) and James et al. (2000), we propose the model that integrates these two so estimates monotone curves jointly through functional principal component analysis approach. The main advantage of this model is to directly apply observed curve trajectories rather than to regard smoothed and projected values as if observed.

The model for functional data has originally an explicit function form, say 'functional model'. Each function is then expressed as a linear combination of basis splines, shortly B-spline, with corresponding basis coefficients. Also since we are commonly provided the curve trajectories as discrete points with noise, the functional model can be written as conventional multivariate data model by evaluating observed points on a basis space; we denote this expression as 'data model'. We in this section define our model in all available forms; functional model, basis function expansions, and data model.

Functional Model

Suppose M number of monotone increasing curves, $\{m_i(t) \in \mathcal{M}\}_{i=1,\ldots,M}$, are observed with noises, $\epsilon_i(t)$ such that $\epsilon_i(t) \sim N(0, \sigma^2)$ at any t for all $i = 1, \ldots, M$. Denote observed curves as $\{y_i(t)\}_{i=1,\ldots,M}$, and hence $y_i(t) = m_i(t) + \epsilon_i(t)$. For each i^{th} curve, we model its relative curvature, which is $w_i(t) = m_i''(t)/m_i'(t)$, in the linear form of a mean function, μ, and K number of principal component functions, $\boldsymbol{f} = \{f_1, \ldots, f_K\}^T$

as follows,

$$y_i(t) = \beta_{0i} + \int_{t_0}^{t} \exp \int_{t_0}^{s} \{\mu(u) + \boldsymbol{f(u)}^T \boldsymbol{\alpha}_i\} \, \mathrm{d}u \, \mathrm{d}s + \epsilon_i(t) \tag{3.3}$$

that are subject to

$$\int \boldsymbol{f(t)} \boldsymbol{f(t)}^T dt = \boldsymbol{I}_K; \tag{3.4}$$

$$\sum_{i=1}^{M} \alpha_{ik} = 0 \;\; \forall k; \quad \sum_{i=1}^{M} \alpha_{i1}^2 > \ldots > \sum_{i=1}^{M} \alpha_{iK}^2, \tag{3.5}$$

where β_{0i}'s are intercepts so that represent a starting value of each curve at an initial time point t_0. The orthonormality constraint on $\boldsymbol{f}$ (3.4) is for the sake of identifiability of principal component functions. We treat principal component scores $\boldsymbol{\alpha}_i = (\alpha_{i1} \cdot \ldots, \alpha_{iK})^T$, which represent the relative weights on $\boldsymbol{f}$, as if fixed effects rather than random effects because it is difficult to derive a closed form of conditional distribution of $\boldsymbol{\alpha}_i$; the model is complicated with two integrals and an exponential between them. We accordingly assume (3.5) for identifiable individual-level characteristics among α_i's as random effects are presumed to be mean zero and have ordered variance. See also Guo et al. (2015) who did this strategy as well.

It should be noted that the slope coefficient β_1 prior to integrated term in (3.1) is not defined in our model. If it were defined, there would be an identifiability issue; unequal parameters may represent the same curve. Let $\beta_1' = \beta_1 \exp(c_1)$ and $\boldsymbol{\theta}' = \boldsymbol{\theta} - \boldsymbol{c}_2$ for $c_1 \neq 0$ and $\boldsymbol{c}_2 \neq \boldsymbol{0}$, then

$$\begin{aligned} \beta_1' \int \exp \int \{\boldsymbol{b(t)}^T \boldsymbol{\theta}'\} &= \beta_1 \exp(c_1) \int \exp \int \{\boldsymbol{b(t)}^T (\boldsymbol{\theta} - \boldsymbol{c}_2)\} \\ &= \beta_1 \int \exp \int \{\boldsymbol{b(t)}^T \boldsymbol{\theta}\} \end{aligned}$$

when c_1 and $\boldsymbol{c}_2$ satisfy $\exp(c_1) \int \exp \int \{-\boldsymbol{b(t)}^T c_2\} = 1$.

Our model specifies the relative curvature w, introduced in Section 3.1.2, with mean and principal component functions as

$$w(t) = \frac{m''(t)}{m'(t)} = \mu(t) + \boldsymbol{f(t)}^T \boldsymbol{\alpha}.$$

Therefore, the model should be carefully interpreted because principal component functions correspond to the relative curvature w not the monotone curves m directly. Since the monotone curve m is configured through $\int \exp \int w$, positive w exponentially accumulates the increase of m, while negative w leads almost zero value in m so makes m increase slowly. Also, it could give an interesting insight about each curve to see a time point such that $w(t) = 0$, which is its inflection point; where the curvature vanishes so the curve changes from concave to convex or vice versa.

For monotone decreasing curves, one can simply change the sign of (3.3) as

$$y_i(t) = \beta_{0i} - \int_{t_0}^{t} \exp \int_{t_0}^{s} \{\mu(u) + \boldsymbol{f(u)}^T \boldsymbol{\alpha}_i\} \, \mathrm{d}u \, \mathrm{d}s + \epsilon_i(t). \qquad (3.3')$$

Thanks to this duality, we hereafter present the form (3.3) to any monotone curves for unity of model development.

Basis Function Expansions

A mean function, μ, and principal component functions, $\boldsymbol{f}$, can be transformed by some set of basis functions, say q-number of basis as $\boldsymbol{b(t)} = \{b_1(t), \ldots, b_q(t)\}^T$. At this process, we select orthonormal basis functions, which satisfy

$$\int \boldsymbol{b(t)} \boldsymbol{b(t)}^T dt = \boldsymbol{I}_q,$$

in order to easily induce the orthonormality of $\boldsymbol{f(t)}$ as in (3.4). Specifically, we represent $\mu(t)$ and $\boldsymbol{f(t)}$ as

$$\mu(t) = \boldsymbol{b(t)}^T \boldsymbol{\theta}_\mu; \;\; \boldsymbol{f(t)} = \boldsymbol{b(t)}^T \boldsymbol{\Theta}_f,$$

where $\boldsymbol{\theta}_\mu$ is a $q \times 1$ vector and $\boldsymbol{\Theta}_f = \{\boldsymbol{\theta}_{f_1}, \ldots, \boldsymbol{\theta}_{f_K}\}^T$ is a $q \times K$ matrix of basis coefficients. Under the basis expansion, the orthonormality of principal component functions can be achieved by orthonormal coefficient matrix $\boldsymbol{\Theta}_f$ as

$$\int \boldsymbol{f(t)} \boldsymbol{f(t)}^T dt = \boldsymbol{\Theta}_f^T \int \boldsymbol{b(t)} \boldsymbol{b(t)}^T dt \boldsymbol{\Theta}_f = \boldsymbol{\Theta}_f^T \boldsymbol{\Theta}_f = \boldsymbol{I}_K.$$

Hence, the model (3.3) can also be written as

$$y_i(t) = \beta_{0i} + \int_{t_0}^{t} \exp \int_{t_0}^{s} \{\boldsymbol{b(u)}^T \boldsymbol{\theta}_\mu + \boldsymbol{b(u)}^T \boldsymbol{\Theta}_f \boldsymbol{\alpha}_i\} \, \mathrm{d}u \, \mathrm{d}s + \epsilon_i(t); \qquad (3.6)$$

that are subject to

$$\boldsymbol{\Theta}_f^T \boldsymbol{\Theta}_f = \boldsymbol{I}_K$$
$$\sum_{i=1}^{M} \alpha_{ik} = 0 \;\; \forall k; \quad \sum_{i=1}^{M} \alpha_{i1}^2 > \ldots > \sum_{i=1}^{M} \alpha_{iK}^2.$$

The creation of orthonormal basis functions is deferred to Section A.1.

Data Model

Since we allow curves to be observed on an irregular and sparse grid, denote n_i as the number of points for an i^{th} curve, which may vary over curves. Let $\boldsymbol{Y}_i = \{y_i(t_1), \ldots, y_i(t_{n_i})\}^T$ be a vector of observations at the points $(t_1, \ldots, t_{n_i})$. Accordingly, the functional model on the basis (3.6) can be represented in the form of data observations as

$$\boldsymbol{Y}_i = \beta_{0i}\mathbf{1}_{n_i} + \boldsymbol{H}_i + \boldsymbol{\epsilon}_i \tag{3.7}$$

where $\mathbf{1}_{n_i}$ is a $n_i \times 1$ vector of ones, $\boldsymbol{H}_i = \{h_i(t_1), \ldots, h_i(t_{n_i})\}^T$ is a vector of evaluated values by the function $h_i(\cdot)$ such that

$$h_i(t) = \int_{t_0}^{t} \exp \int_{t_0}^{s} \{\boldsymbol{b}(\boldsymbol{u})^T \boldsymbol{\theta}_\mu + \boldsymbol{b}(\boldsymbol{u})^T \boldsymbol{\Theta}_f \boldsymbol{\alpha}_i\} \,\mathrm{d}u \,\mathrm{d}s \tag{3.8}$$

and $\boldsymbol{\epsilon}_i = \{\epsilon(t_1), \ldots, \epsilon(t_{n_i})\}^T$ is a vector of noises such that $\boldsymbol{\epsilon}_i \sim N(0, \sigma^2 I_{n_i})$.

Similary, the following represents the relative curvature as the data model,

$$\boldsymbol{W}_i = \boldsymbol{b}_i^T \boldsymbol{\theta}_\mu + \boldsymbol{b}_i^T \boldsymbol{\Theta}_f \boldsymbol{\alpha}_i$$

where $\boldsymbol{b}_i = \{\boldsymbol{b}(t_1), \ldots, \boldsymbol{b}(t_{n_i})\}$ is a $q \times n_i$ matrix of evaluated values at basis functions.

3.2.3 Fisher Scoring Algorithm for Penalized Likelihood Estimation

Penalized Likelihood

Since we assume the noises have normal distribution, $\boldsymbol{\epsilon}_i \sim N(0, \sigma^2 I_{n_i})$, the -2log-likelihood of M curves is given by

$$-2l = N \log \sigma^2 + \sum_{i=1}^{M} ||\boldsymbol{Y}_i - \beta_{0i}\mathbf{1}_{n_i} - \boldsymbol{H}_i||^2 / \sigma^2 \tag{3.9}$$

where $N = \sum_{i=1}^{M} n_i$ is the number of total observations, and $\boldsymbol{H}_i = \{h_i(t_1), \ldots, h_i(t_{n_i})\}^T$ as defined in (3.7).

We penalize the above likelihood (3.9) to control the smoothness of monotone curves. While it is common to penalize the curvature, which corresponds to the second derivative of curves, m'', we penalize the relative curvature, $w = m''/m'$. This penalization has an advantage of keeping the fitted function away from the boundary condition $m' = 0$, and therefore it always satisfies $m' > 0$ (Ramsay, 1998).

In the structure of $w_i(t) = \mu(t) + \sum_{k=1}^{K} f_k(t)\alpha_{ik}$, we make the penalty terms for μ and $f_1, \dots, f_K$ separately; therefore, the criterion we minimize is

$$F_{\lambda_\mu,\lambda_f}(\mu, \boldsymbol{f}, \boldsymbol{\alpha}) = -2l + \lambda_\mu \int \mu(t)^2 \,\mathrm{d}t + \lambda_f \sum_{k=1}^{K} \int f_k(t)^2 \,\mathrm{d}t, \tag{3.10}$$

where λ_μ and λ_f are smoothing parameters and $\boldsymbol{\alpha}$ denotes $\{\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_M\}^T$. Under the basis expression as $\mu(t) = \boldsymbol{b(t)}^T\boldsymbol{\theta}_\mu$ and $f_k(t) = \boldsymbol{b(t)}^T\boldsymbol{\theta}_{f_k}$, these penalization terms can be simplified by the following calculus,

$$\begin{aligned} \int \mu(t)^2 \,\mathrm{d}t &= \boldsymbol{\theta}_\mu^T \int \boldsymbol{b(t)}\boldsymbol{b(t)}^T \,\mathrm{d}t \,\boldsymbol{\theta}_\mu = \boldsymbol{\theta}_\mu^T\boldsymbol{\theta}_\mu \; ; \\ \sum_{k=1}^{K} \int f_k(t)^2 \,\mathrm{d}t &= \sum_{k=1}^{K} \boldsymbol{\theta}_{f_k}^T \int \boldsymbol{b(t)}\boldsymbol{b(t)}^T \,\mathrm{d}t \,\boldsymbol{\theta}_{f_k} = \sum_{k=1}^{K} \boldsymbol{\theta}_{f_k}^T\boldsymbol{\theta}_{f_k}. \end{aligned}$$

It tells that the orthonormal basis functions indeed makes the ridge penalties for coefficients. Additionally, the penalty term of f_k's can be further simplified to the constant value of K due to the orthonormality condition of $\boldsymbol{\Theta_f}$ coefficients such that

$$\boldsymbol{\Theta}_f^T\boldsymbol{\Theta}_f = \begin{bmatrix} \boldsymbol{\theta}_{f_1}^T\boldsymbol{\theta}_{f_1} & \boldsymbol{\theta}_{f_1}^T\boldsymbol{\theta}_{f_2} & \cdots & \boldsymbol{\theta}_{f_1}^T\boldsymbol{\theta}_{f_K} \\ \boldsymbol{\theta}_{f_2}^T\boldsymbol{\theta}_{f_1} & \boldsymbol{\theta}_{f_2}^T\boldsymbol{\theta}_{f_2} & \cdots & \boldsymbol{\theta}_{f_2}^T\boldsymbol{\theta}_{f_K} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\theta}_{f_K}^T\boldsymbol{\theta}_{f_1} & \boldsymbol{\theta}_{f_K}^T\boldsymbol{\theta}_{f_2} & \cdots & \boldsymbol{\theta}_{f_K}^T\boldsymbol{\theta}_{f_K} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \\ 0 & 0 & \cdots & 1 \end{bmatrix} = \boldsymbol{I}_K,$$

and so $\sum_{k=1}^{K} \boldsymbol{\theta}_{f_k}^T \boldsymbol{\theta}_{f_k} = K$. Therefore, the criterion (3.10) can be fully written as

$$F_{\lambda_\mu, \lambda_f}(\boldsymbol{\theta}_\mu, \boldsymbol{\Theta}_f, \boldsymbol{\alpha}) = N \log \sigma^2 + \sum_{i=1}^{M} ||\boldsymbol{Y}_i - \beta_{0i} \mathbf{1}_{n_i} - \boldsymbol{H}_i||^2/\sigma^2 + \lambda_\mu \boldsymbol{\theta}_\mu^T \boldsymbol{\theta}_\mu + \lambda_f K. \tag{3.11}$$

It should be remarked the penalty term corresponding to $\boldsymbol{f}$ has nothing to do with smoothing in the end. The presumed constraint $\boldsymbol{\theta}_f^T \boldsymbol{\theta}_f = 1$ leads some relevant amount of smoothness automatically. Although we originally intend to penalize the relative curvature w, we in fact penalize only the mean part of $w_i(t)$'s for $i \in \{1, \ldots, M\}$; that is, $\int \{\mu(t)\}^2 \mathrm{d}t$, which leads a ridge penalty on basis coefficients as aforementioned.

Fisher Scoring Algorithm

Non-linear maximum likelihood equations for $\boldsymbol{\theta}_\mu, \boldsymbol{\theta}_{f_1}, \ldots, \boldsymbol{\theta}_{f_K}$ and $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_M$ are obtained by taking partial differential to the criterion (3.11) for each parameter as follows,

$$0 = \frac{\partial F}{\partial \boldsymbol{\theta}_\mu} = -2 \sum_{i=1}^{M} \frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\theta}_\mu} \boldsymbol{r}_i / \sigma^2 + 2\lambda_\mu \boldsymbol{\theta}_\mu \tag{3.12a}$$

$$0 = \frac{\partial F}{\partial \boldsymbol{\theta}_{f_k}} = -2 \sum_{i=1}^{M} \frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\theta}_{f_k}} \boldsymbol{r}_i / \sigma^2 \qquad \forall k \in \{1, \ldots, K\} \tag{3.12b}$$

$$0 = \frac{\partial F}{\partial \boldsymbol{\alpha}_i} = -2 \frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\alpha}_i} \boldsymbol{r}_i / \sigma^2 \qquad \forall i \in \{1, \ldots, M\} \tag{3.12c}$$

where $\boldsymbol{r}_i = \boldsymbol{Y}_i - \beta_{0i} \mathbf{1}_{n_i} - \boldsymbol{H}_i$ is a $n_i \times 1$ vector of residuals, and each partial derivatives $\underset{q \times n_i}{\partial \boldsymbol{H}_i / \partial \boldsymbol{\theta}_\mu}, \underset{q \times n_i}{\partial \boldsymbol{H}_i / \partial \boldsymbol{\theta}_{f_k}}$ and $\underset{K \times n_i}{\partial \boldsymbol{H}_i / \partial \boldsymbol{\alpha}_i}$ are a matrix of evaluated values at $(t_1, \ldots, t_{n_i})$ by the following functions respectively,

$$\begin{aligned}
\frac{\partial h_i(t)}{\partial \boldsymbol{\theta}_\mu} &= \int_{t_0}^{t} \exp \int_{t_0}^{s} \boldsymbol{b}(\boldsymbol{u})\{\boldsymbol{b}(\boldsymbol{u})^T\boldsymbol{\theta}_\mu + \boldsymbol{b}(\boldsymbol{u})^T\boldsymbol{\Theta}_f\boldsymbol{\alpha}_i\}\,\mathrm{d}u\,\mathrm{d}s;\\
\frac{\partial h_i(t)}{\partial \boldsymbol{\theta}_{f_k}} &= \int_{t_0}^{t} \exp \int_{t_0}^{s} \alpha_{ik}\boldsymbol{b}(\boldsymbol{u})\{\boldsymbol{b}(\boldsymbol{u})^T\boldsymbol{\theta}_\mu + \boldsymbol{b}(\boldsymbol{u})^T\boldsymbol{\Theta}_f\boldsymbol{\alpha}_i\}\,\mathrm{d}u\,\mathrm{d}s = \alpha_{ik}\frac{\partial h_i(t)}{\partial \boldsymbol{\theta}_\mu};\\
\frac{\partial h_i(t)}{\partial \boldsymbol{\alpha}_i} &= \int_{t_0}^{t} \exp \int_{t_0}^{s} \boldsymbol{\Theta}_f^T\boldsymbol{b}(\boldsymbol{u})\{\boldsymbol{b}(\boldsymbol{u})^T\boldsymbol{\theta}_\mu + \boldsymbol{b}(\boldsymbol{u})^T\boldsymbol{\Theta}_f\boldsymbol{\alpha}_i\}\,\mathrm{d}u\,\mathrm{d}s = \boldsymbol{\Theta}_f^T\frac{\partial h_i(t)}{\partial \boldsymbol{\theta}_\mu}.
\end{aligned}$$

One can do the complex calculations, caused by integral and exponential functions, only once because the latter two $\partial \boldsymbol{H}_i/\partial \boldsymbol{\theta}_{f_k}$ and $\partial \boldsymbol{H}_i/\partial \boldsymbol{\alpha}_i$ are expressed by a product of the certain coefficients and $\partial \boldsymbol{H}_i/\partial \boldsymbol{\theta}_\mu$.

We provide the iterative algorithm below to solve these nonlinear equations through Fisher scoring procedure (Longford, 1987).

1. Initialize $\boldsymbol{\theta}_\mu^0, \boldsymbol{\Theta}_f^0, \boldsymbol{\alpha}_i^0$ and β_{0i}^0.

2. Update $\boldsymbol{\theta}_\mu^l$ as

$$\boldsymbol{\theta}_\mu^l \leftarrow \boldsymbol{\theta}_\mu^{l-1} + \Big[\sum_{i=1}^{M} \frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\theta}_\mu}\frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\theta}_\mu}^T + \lambda_\mu \boldsymbol{I}_q\Big]^{-1}\Big[\sum_{i=1}^{M}\frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\theta}_\mu}\boldsymbol{r}_i - \lambda_\mu\boldsymbol{\theta}_\mu\Big].$$

3. Update $\boldsymbol{\theta}_{f_k}^l$ for $\forall k \in \{1, \ldots, K\}$ as

$$\boldsymbol{\theta}_{f_k}^l \leftarrow \boldsymbol{\theta}_{f_k}^{l-1} + \Big[\sum_{i=1}^{M} \frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\theta}_{f_k}}\frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\theta}_{f_k}}^T + \lambda_f \boldsymbol{I}_q\Big]^{-1}\Big[\sum_{i=1}^{M}\frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\theta}_{f_k}}\boldsymbol{r}_i\Big]$$

   and re-update by orthonormalized ones through QR decomposing $\boldsymbol{\Theta}_f^l = \{\boldsymbol{\theta}_{f_1}^l, \ldots, \boldsymbol{\theta}_{f_k}^l\}^T$.

4. Update $\boldsymbol{\alpha}_i^l$ for $\forall i \in \{1, \ldots, M\}$ as

$$\boldsymbol{\alpha}_i^l \leftarrow \boldsymbol{\alpha}_i^{l-1} + \Big[\frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\alpha}_i}\frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\alpha}_i}^T\Big]^{-1}\Big[\frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\alpha}_i}\boldsymbol{r}_i\Big]$$

and rearrange such that $\sum_{i=1}^{M} \alpha_{ik} = 0$ for $\forall k$ and $\sum \alpha_{i1}^2 > \ldots > \sum \alpha_{iK}^2$.

5. Update β_{0i}^l for $i = 1, \ldots, M$ as

$$\beta_{0i}^l = \bar{\boldsymbol{Y}}_i - \bar{\boldsymbol{H}}_i.$$

6. Iterate step 2 to 5 until all converges.

Once all converged, an estimate of σ^2 is

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{i=1}^{M} ||\boldsymbol{Y}_i - \hat{\beta}_{0i} \mathbf{1}_{n_i} - \hat{\boldsymbol{H}}_i||^2,$$

where $\hat{\beta}_{0i}$ and $\hat{\boldsymbol{H}}_i$ are the converged estimates of the above iterative algorithm.

It should be noted that although there are no terms regarding to λ_f in (3.12b), we add a term, $\lambda_f \boldsymbol{I}_q$, in step 3 to avoid the problem caused by the computational singularity of the partial differentials of $h_i(t)$; see details about the computational singularity in Section A.2. Not only can this inclusion be regarded as a ridge correction to ensure non-singularity of matrix but also would it be in fact natural unless the term $\boldsymbol{\theta}_{f_k}^T \boldsymbol{\theta}_{f_k}$ were numerically simplified to 1. What this means, like the way that (3.11) leads a normal equation of $\boldsymbol{\theta}_\mu$ as (3.12a), a normal equation of $\boldsymbol{\theta}_{f_k}$ would be

$$0 = -2 \sum_{i=1}^{M} \frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\theta}_{f_k}} \boldsymbol{r}_i / \sigma^2 + 2 \lambda_f \boldsymbol{\theta}_f$$

if (3.11) were

$$F_{\lambda_\mu, \lambda_f}(\boldsymbol{\theta}_\mu, \boldsymbol{\Theta}_f, \boldsymbol{\alpha}) = N \log \sigma^2 + \sum_{i=1}^{M} ||\boldsymbol{Y}_i - \beta_{0i} \mathbf{1}_{n_i} - \boldsymbol{H}_i||^2 / \sigma^2 + \lambda_\mu \boldsymbol{\theta}_\mu^T \boldsymbol{\theta}_\mu + \lambda_f \boldsymbol{\theta}_{f_k}^T \boldsymbol{\theta}_{f_k},$$

and therefore, the Fisher information matrix would be

$$\sum_{i=1}^{M} \frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\theta}_{f_k}} \frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\theta}_{f_k}}^T + \lambda_f \boldsymbol{I}_q,$$

which is applied to the updated equation of step 3.

3.2.4 Model Selection

Specification of B-splines

For the reasonable smoothness of curves, it could be required to choose the optimal number and positions of knots of basis functions. However, since we adopt the penalizing approach for smoothness, specifying knots is no more important issue in our model estimation; see also Eilers and Marx (1996). A relatively large number of equidistant knots over the data range, typically 10-20 knots, is often sufficient.

Choice of Penalty Parameters, λ_μ and λ_f

The cross-validation is most commonly used to optimize the penalty parameters. Since the proposed model has two penalty parameters, λ_μ and λ_f, n-fold cross validation is preferable to reduce the computation time. Also, it is efficient to do a grid-search on a $\log$-scale for each penalty parameter. All examples of simulation and application in the following sections use 5-fold cross-validation.

Choice of the Number of Principal Components, K

In the multivariate case, the number of principal components has the upper limit as the number of variables. However, since 'variables' correspond to t in the case of functional data analysis, there is no upper limit for the number of principal component functions. Although the minimum between the number of observations and the number of basis functions algebraically could be the upper limit of PC functions, too many PC functions will blur the nature of principal component analysis; to describe the variation with a 'few' number of essential components. Moreover, it may as well consider PC functions, which merely describe little variation, as noise. Therefore, choosing the number of principal component functions K is important to compromises between complexity and parsimony in expressing functions and also to avoid overfitting.

One basic procedure of choosing K is to assess a quality of model fit in terms of prediction with the different number of PC functions. Since the cross-validation error is already obtained at any fixed K over a grid-space of λ_μ and λ_f in the model estimation, denoted by $\mathrm{CV}_k(\lambda_\mu, \lambda_f)$, the CV error at the optimal λ_μ and λ_f can be used to compare across different K's. In other words, one can choose the optimal K such that

$$K = \underset{k \in \{1, \ldots, n_k\}}{\arg\min} \, \mathrm{CV}_k^*$$

where $\mathrm{CV}_k^* = \min_{\lambda_\mu, \lambda_f} \mathrm{CV}_k(\lambda_\mu, \lambda_f)$ for a fixed k number of PC functions and n_k is typically 4 or more. In Section 3.3, we draw an experiment of choosing K to see how well CV errors tell the significant number of PC functions according to the data noise level.

Alternatively, we also suggest to choose the suitable K by dropping the components whose scores have the relatively small change in variance from the preceding component; this idea is similar to the view point of Cattell's scree test (Cattell, 1966). Specifically, one can fit the model with a sufficient number of PC functions, typically 4 or more, and plot the variances of PC scores in a decreasing order. From this plot, K can be chosen where the variance curve makes an elbow toward less steep decline. We later describe how to perform this procedure in a practical example in Section 3.4.

3.3 Simulation Study

Our simulation study illustrates the 'two-step' approach, described in Section 3.2.1, and primarily 'proposed' approach, in Section 3.2.2. We aim to see how our models improve monotone curve estimation through fPCA, by comparing to the monotone smoothing estimation of ‘Ramsay’, which is applied to each single curve (Ramsay, 1998).

The Section 3.3.1 presents the set-up details for generating data, and Section 3.3.2 clarifies the measurement, mean integrated square error, which will be used for comparative purposes. Lastly, we summarize the simulation result in Section 3.3.3.

3.3.1 Simulation Setup

We set up the relative curvature function, w, with a straight line mean function, μ, and two principal component functions, f_1, f_2, which are orthonormal, such that

$$\begin{aligned} \mu(t) &= 5 - 10t; \\ f_1(t) &= \sqrt{2} \sin 2\pi t; \\ f_2(t) &= \sqrt{2} \cos 2\pi t. \end{aligned}$$

Assuming $\beta_{0i} = 0$ for all curves, we generate $M = 50$ monotone curves from the model

$$y_i(t_j) = \int_0^{t_j} \exp \int_0^s \{\mu(u) + f_1(u)\alpha_{i1} + f_2(u)\alpha_{i2}\} \, \mathrm{d}u \, \mathrm{d}s + \epsilon_i(t_j),$$

where $j = 1, \ldots, n_i$ and t_j’s are randomly selected points in $[0, 1]$; we sample 50 to 100 number of points for each curve, and these points are uniformly distributed in $[0, 1]$. The principal component scores are once generated independently as $\alpha_{i1} \sim N(0, 0.5^2)$ and $\alpha_{i2} \sim N(0, 0.1^2)$, and fixed in the remaining every data generation. To see how each model works out at different noises, we add errors in three levels; $\epsilon_i(t_j) \sim N(0, 0.01^2), N(0, 0.05^2)$, and $N(0, 0.1^2)$.

Figure 3.3 shows the fundamental curves in our simulation; these provide more insights into the simulated curves. The orthonormal principal component functions, f_1 and f_2, are illustrated around the mean function of w, that is μ, in Figure 3.3 (a) and (b). Similarly, they are also illustrated towards the mean of y in Figure 3.3 (c) and (d) for some β_0 and α; these constant and scores are arbitrarily set to describe curves neatly. The first PC function f_1 represents the variation at the boundary, and also provides how much the curvature changes before and after an inflection point for each curve. On the other hand, the second PC function f_2 represent the variation at the center of the range, so any curves with relatively large variation in the middle might have large values of scores corresponding to f_2.

3.3.2 Mean Integrated Squared Errors

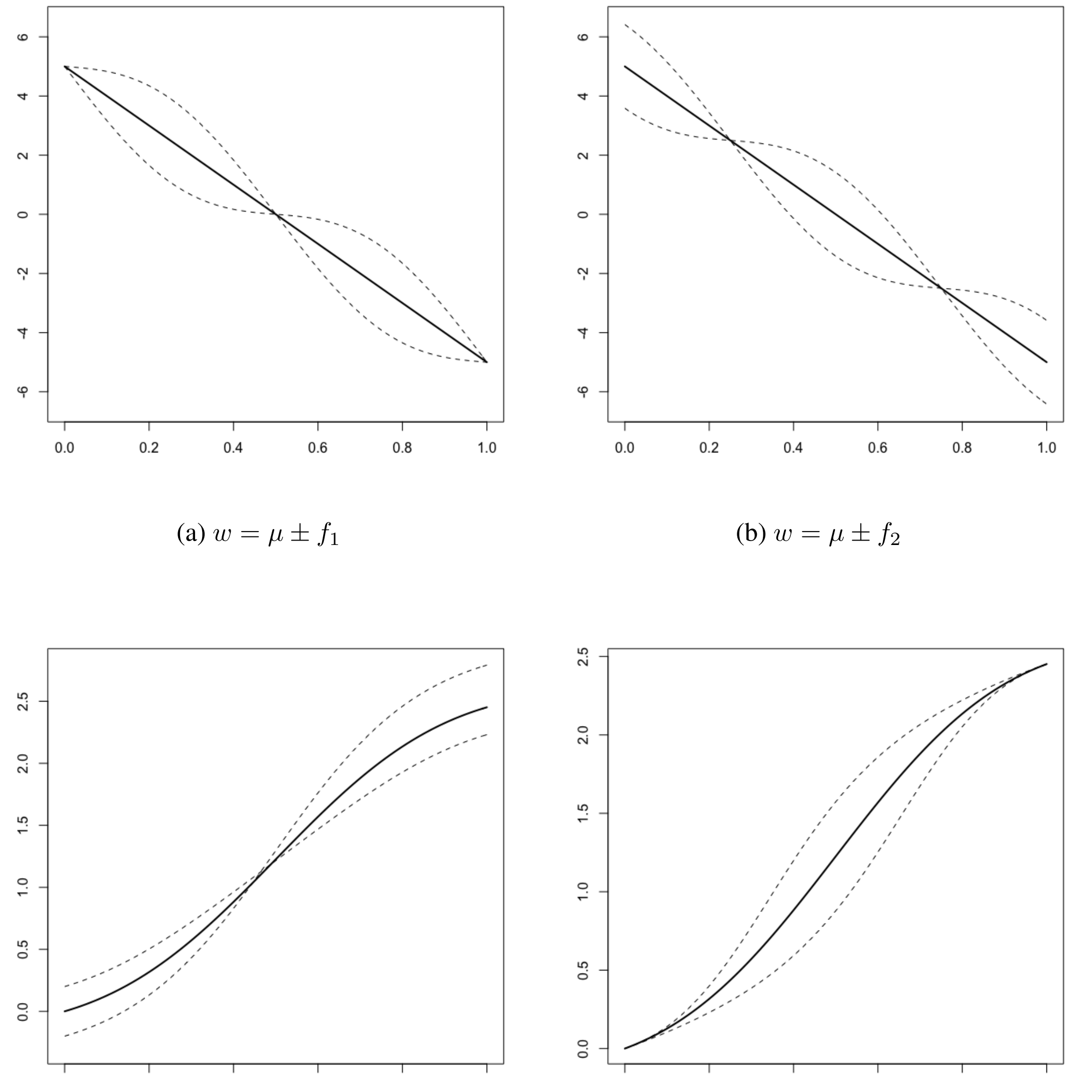
We assess the quality of estimators of w and m by comparing mean integrated squared errors (MISE) in an overall sense, defined as

$$\text{MISE}(\hat{w}) = \sum_{i=1}^{M} \int \{\hat{w}_i(t) - w_i(t)\}^2 \, \mathrm{d}t;$$
$$\text{MISE}(\hat{m}) = \sum_{i=1}^{M} \int \{\hat{m}_i(t) - m_i(t)\}^2 \, \mathrm{d}t,$$

where $\hat{w}_i$ and $\hat{m}_i$ are estimated curves.

For Ramsay's estimation, there are no terms toward principal component functions but a single vector represents each relative curvature, that is $w_i(t) = \boldsymbol{b(t)}^T \boldsymbol{\theta}_{\mu i}$. On the other hand, the two-step and proposed approaches have a functional principal component model for w as $w_i(t) = \boldsymbol{b(t)}^T (\boldsymbol{\theta}_\mu + \boldsymbol{\Theta}_f \boldsymbol{\alpha}_i)$. The form of function m_i is accordingly determined by w_i for each case.

The process of integration, in computing MISE with regard to w, can be ignored by



(a) $w = \mu \pm f_1$

(b) $w = \mu \pm f_2$

(c) $m = \beta_0 + D^{-1} \exp D^{-1}(\mu \pm \alpha f_1)$

(d) $m = \beta_0 + D^{-1} \exp D^{-1}(\mu \pm \alpha f_2)$

Figure 3.3: Simulated curves in terms of w (top) and m (bottom) where $\mu(t) = 5 - 10\,t$, $f_1(t) = \sqrt{2}\sin 2\pi t$ and $f_2(t) = \sqrt{2}\cos 2\pi t$.

virtue of orthonormal basis functions; in the case of Ramsay's estimation,

$$\begin{aligned}\text{MISE}(\hat{w}) &= \frac{1}{M}\sum_{i=1}^{M}\int \{\boldsymbol{b(t)}^T(\hat{\boldsymbol{\theta}}_{\mu i} - \boldsymbol{\theta}_{\mu i})\}^2\,\mathrm{d}t \\ &= \frac{1}{M}\sum_{i=1}^{M}(\hat{\boldsymbol{\theta}}_{\mu i} - \boldsymbol{\theta}_{\mu i})^T\int \boldsymbol{b(t)}^T\boldsymbol{b(t)}\,\mathrm{d}t\;(\hat{\boldsymbol{\theta}}_{\mu i} - \boldsymbol{\theta}_{\mu i}) \\ &= \sum_{i=1}^{M}(\hat{\boldsymbol{\theta}}_{\mu i} - \boldsymbol{\theta}_{\mu i})^T(\hat{\boldsymbol{\theta}}_{\mu i} - \boldsymbol{\theta}_{\mu i}),\end{aligned}$$

where $\hat{\boldsymbol{\theta}}_{\mu i}$ are estimates. Similarly, in the case of two-step or proposed estimation,

$$\begin{aligned}\text{MISE}(\hat{w}) &= \frac{1}{M}\sum_{i=1}^{M}\int \{\boldsymbol{b(t)}^T(\hat{\boldsymbol{\theta}}_{\mu} + \hat{\boldsymbol{\Theta}}_f\hat{\boldsymbol{\alpha}}_i - \boldsymbol{\theta}_{\mu} - \boldsymbol{\Theta}_f\boldsymbol{\alpha}_i)\}^2\,\mathrm{d}t \\ &= \frac{1}{M}\sum_{i=1}^{M}(\hat{\boldsymbol{\theta}}_{\mu} + \hat{\boldsymbol{\Theta}}_f\hat{\boldsymbol{\alpha}}_i - \boldsymbol{\theta}_{\mu} - \boldsymbol{\Theta}_f\boldsymbol{\alpha}_i)^T\int \boldsymbol{b(t)}^T\boldsymbol{b(t)}\,\mathrm{d}t\;(\hat{\boldsymbol{\theta}}_{\mu} + \hat{\boldsymbol{\Theta}}_f\hat{\boldsymbol{\alpha}}_i - \boldsymbol{\theta}_{\mu} - \boldsymbol{\Theta}_f\boldsymbol{\alpha}_i) \\ &= \frac{1}{M}\sum_{i=1}^{M}(\hat{\boldsymbol{\theta}}_{\mu} + \hat{\boldsymbol{\Theta}}_f\hat{\boldsymbol{\alpha}}_i - \boldsymbol{\theta}_{\mu} - \boldsymbol{\Theta}_f\boldsymbol{\alpha}_i)^T(\hat{\boldsymbol{\theta}}_{\mu} + \hat{\boldsymbol{\Theta}}_f\hat{\boldsymbol{\alpha}}_i - \boldsymbol{\theta}_{\mu} - \boldsymbol{\Theta}_f\boldsymbol{\alpha}_i),\end{aligned}$$

where $\hat{\boldsymbol{\theta}}_{\mu}, \hat{\boldsymbol{\Theta}}_f$ and $\hat{\boldsymbol{\alpha}}_i$ are estimates. Therefore, only estimates and true parameters matter to obtain MISE($\hat{w}$).

However, computing MISE($\hat{m}$) has no choice but to perform the integration approximately over the range of basis functions; because of complexity of monotone function class $\mathcal{M}$ as (3.1). It can be shortly represented as,

$$\text{MISE}(\hat{m}) = \sum_{i=1}^{M}\int \{\hat{\beta}_{0i} + \hat{h}_i(t) - \beta_{0i} - h_i(t)\}^2\,\mathrm{d}t$$

where $\hat{h}_i(t)$ is the function of (3.8) with estimated coefficients.

Table 3.1: Simulation study on the choice of the number of principal component functions K; For the data generated by $K = 2$, the procedure of comparing CV errors mostly chooses $K = 2$ correctly; As the noise level σ gets larger, the chance of choosing $K = 3$ increases.

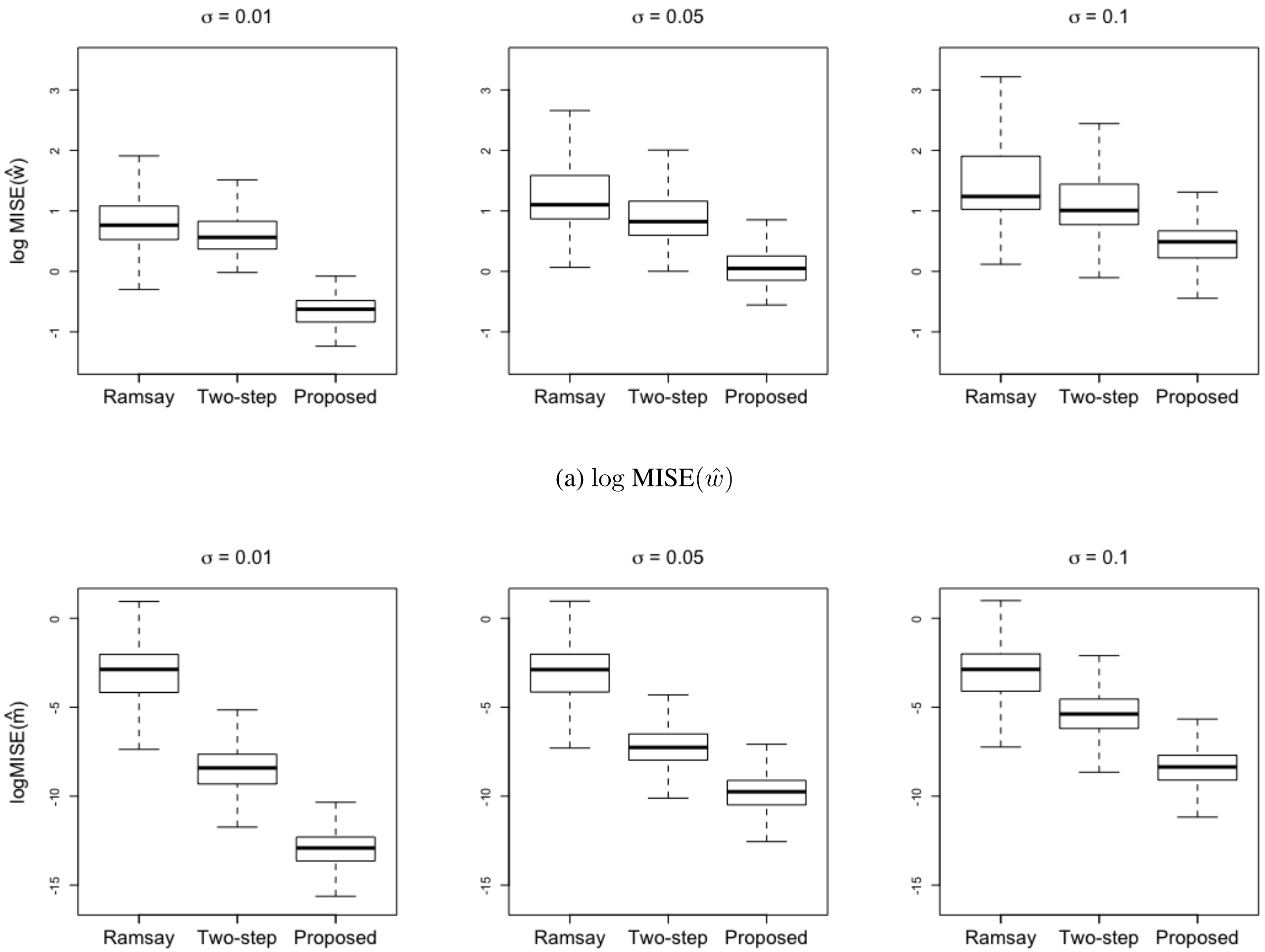
	$\sigma = 0.01$			$\sigma = 0.05$			$\sigma = 0.1$		
optimal K	1	2	3	1	2	3	1	2	3
occurrence	0%	100%	0%	0%	99%	1%	0%	94%	6%

3.3.3 Result Summary

We generate the simulation data 100 times and estimate coefficients for each run with $K = 1, 2, 3$. We first aim to obtain the optimal number of principal component functions, and then build MISEs to compare across three approaches.

We perform the procedure of comparing CV errors to choose the optimal K on each simulation data. As expected, almost all of them are optimal with $K = 2$, which is true, at any noise level as Table 3.1 shows. It is however remarkable that the cases of choosing $K = 3$ as optimal occur more often as the noise level gets larger. Since the noise on observed trajectories may confuse the amount of variance, which is in fact caused by principal component functions, this incidence is natural.

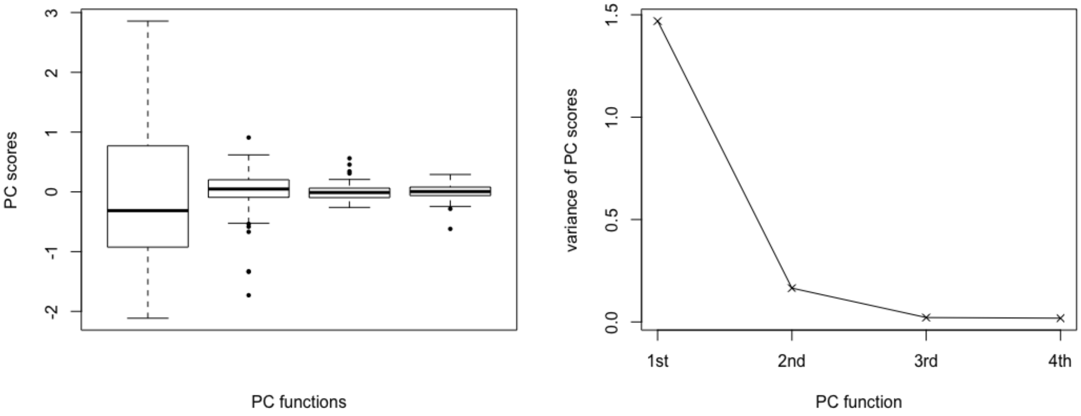
Also, Figure 3.4 shows MISE distributions of $\hat{w}$ and $\hat{m}$ in a log-scale for three approaches at different noise levels; Ramsay's smooth monotone function estimation, two-step approach applying reduced rank model followed by Ramsay's method, and the proposed approach. Overall, the two-step approach improves Ramsay's single estimation, but the proposed method does even more; the centers of the proposed method's boxplots are always below others. In terms of MISE($\hat{w}$), the amount of improvement is particularly higher with the proposed than the two-step approach at the low noise level; not only the center but also the range of MISE distribution is distinctively smaller for the proposed. Also, while all of MISEs increase as the noise level increases, our approach always out-



(a) $\log$ MISE$(\hat{w})$

(b) $\log$ MISE$(\hat{m})$

Figure 3.4: Distribution of $\log$-scale MISE for comparing Ramsay's, two-step, and proposed approaches based on 100 simulation runs for three levels of noise; $\sigma = 0.01, 0.05, 0.1$



(a) Distribution of PC scores

(b) Variance of PC scores

Figure 3.5: Exploring principal component scores; two PC functions $K = 2$ are sufficient to explain the overall variation of data.

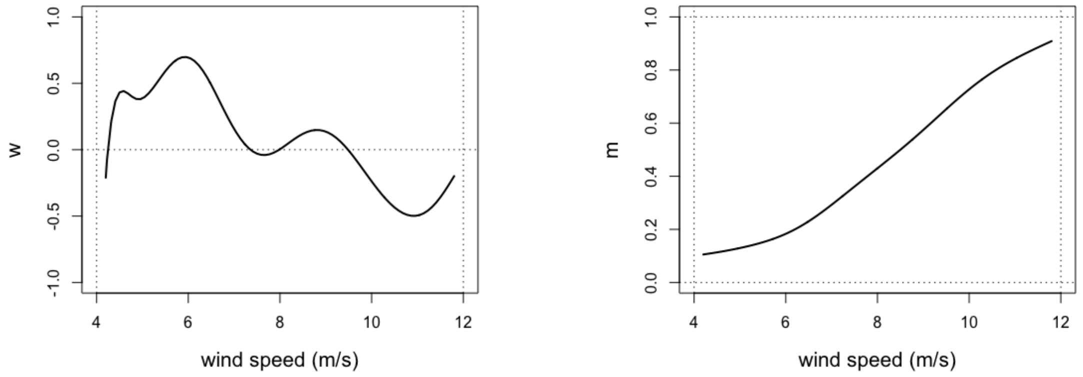
performs others in every level.

3.4 Application: Wind Power Curve Data

In this section, the proposed model is applied to wind power curve data which motivates our study as introduced in Section 2.1.

In the dataset, wind power productions and wind speeds are recorded every 10-minutes for about a year and a half. We assume that one curve is created by records during a week; therefore, the total number of curves is $M = 74$. We only consider the range of wind speeds, from 4 to 12 m/s, where most wind curves are strictly increasing; as shown in Figure 2.1.

To determine the number of significant number of principal component functions, we first examine the distribution of principal component scores obtained with the sufficiently many components; we start with $K = 4$ at this point. As shown in Figure 3.5, the variance of scores corresponding to the second and succeeding principal functions are apparently



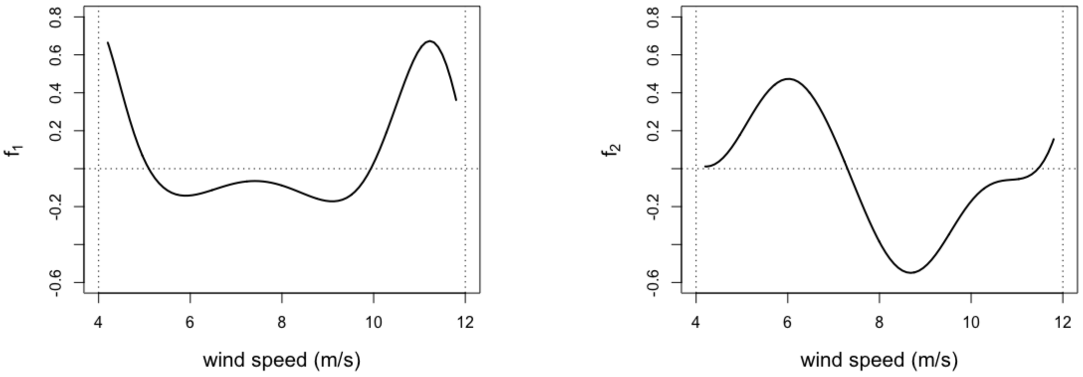
(a) $\mu(t)$ (b) $m(t) = \int \exp \int \mu(t)$

Figure 3.6: (a) The mean of relative curvatures; (b) The monotone curve with regard to the mean of relative curvatures

in agreement. Therefore, we regard $K = 2$ principal component functions are sufficient to describe the variation of wind power curves.

With two PC functions, the mean function of relative curvatures $\mu(t)$ and its corresponding monotone curve are obtained as illustrated in Figure 3.6. It is remarkable that our optimization cares about the smoothness of monotone curves not that of relative curvatures, so the mean of relative curvatures $\mu(t)$ are rather rough. However, $\mu(t)$ provides an outline of curvature change; a wind curve increases convexly ($m'' > 0$) before $7 \sim 9$ m/s while concavely ($m'' < 0$) after then. Note that the tuning parameters λ_μ and λ_f are determined based on 5-fold cross validation, and the basis coefficients converge mostly within 10 iterations.

Figure 3.7 shows PC functions estimated from the wind power curves. We here interpret the interesting features of them. The first PC function explains how different the curvature at the boundary is from that in the middle. The second PC function, however, explains how different the curvatures are between before and after an inflection point.

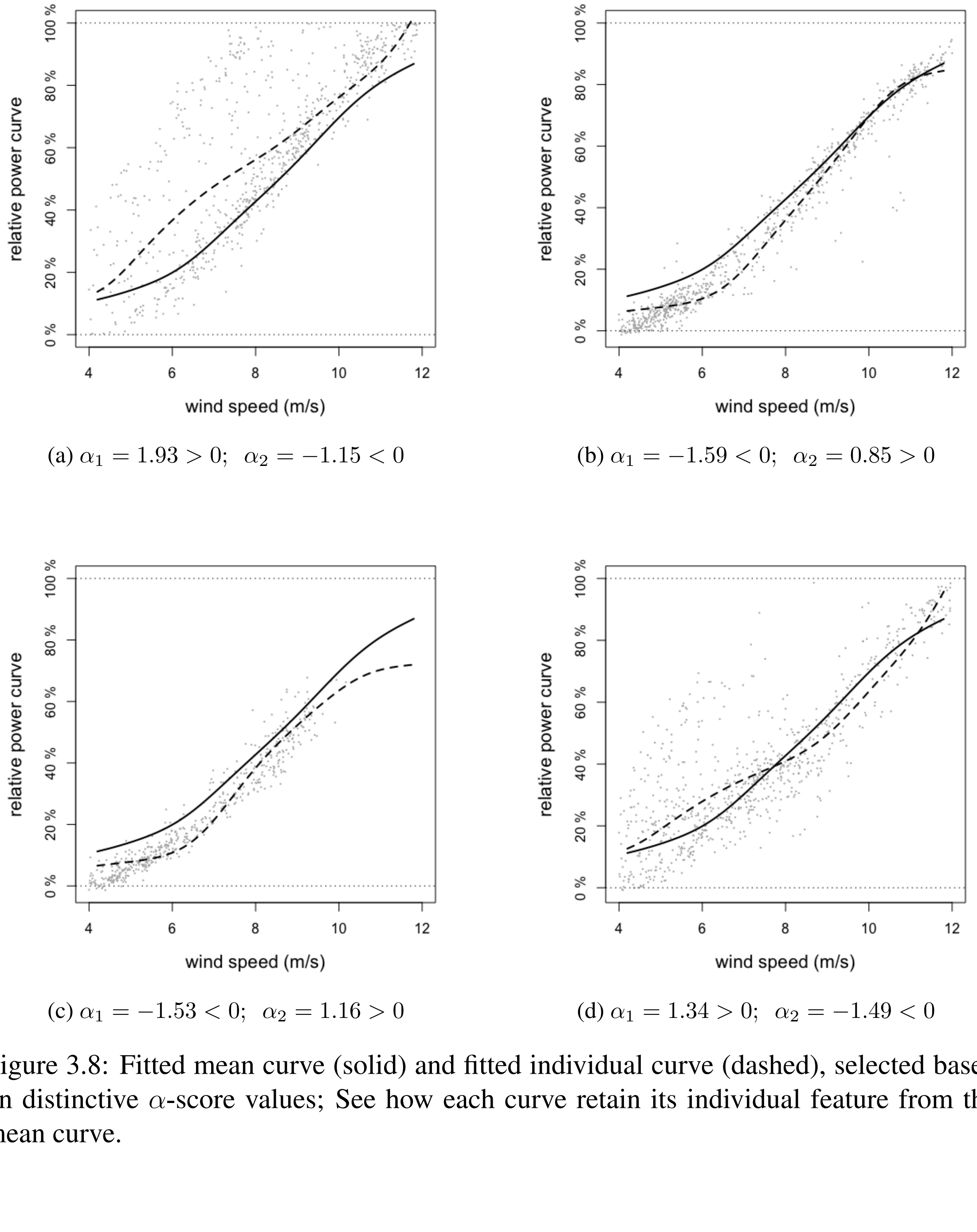


(a) 1^{st} PC function

(b) 2^{nd} PC function

Figure 3.7: Principal component functions; 1^{st} PC function determines how much a power curve accelerates at the low and high wind speed range; 2^{nd} PC function determines how large a curvature changes from concave to convex or vice versa.

For example, the positive greater the first score is, the faster the power curve increase at the starting point and slows down at the end; as shown in Figure 3.8 (a). It is because, the first PC function goes far away from the mean of w in a positive direction at the low wind speed, so $\int \exp \int$ (positives) will be accumulated. Whereas, it goes in a negative direction at the higher wind speed, so $\int \exp \int$ (negatives) will be accumulated; which are almost zeros. Figure 3.8 (b) illustrates the case of negative first score, whose curve stays flat at the beginning but gains momentum in the middle. In that manner, if the second score is a high positive value, its curve is convex before the inflection point, which commonly locates around $7 \sim 9$ m/s, while concave after the inflection point; see Figure 3.8 (c). However, in case of a negative score as Figure 3.8 (d) shows, the curve changes its curvature from concave to convex.



(a) $\alpha_1 = 1.93 > 0;\ \ \alpha_2 = -1.15 < 0$

(b) $\alpha_1 = -1.59 < 0;\ \ \alpha_2 = 0.85 > 0$

(c) $\alpha_1 = -1.53 < 0;\ \ \alpha_2 = 1.16 > 0$

(d) $\alpha_1 = 1.34 > 0;\ \ \alpha_2 = -1.49 < 0$

Figure 3.8: Fitted mean curve (solid) and fitted individual curve (dashed), selected based on distinctive α-score values; See how each curve retain its individual feature from the mean curve.

4. STATISTICAL MODELING ON SPATIO-TEMPORAL BINARY DATA FOR DESCRIBING INFECTIOUS DISEASE SPREADING PATTERN

4.1 Preamble

In this chapter, we propose two modeling approaches for spatio-temporal binary data that aim to describe an infectious disease spreading pattern. The first model in Section 4.2 focuses on spatial dependence among spaces at the same time. Additional to a model estimation, predicting next joint status is another primary goal. On the other hand, the second model introduced in Section 4.3 considers a binary time sequence of each space as Markov process. It concerns that observed binary data may have measurement errors, and therefore, their hidden states are modeled instead. Both models apply to ALS patients data.

4.2 Autologistic Network Model with Absorbing States

4.2.1 Introduction

In this study, we consider spatio-temporal binary data that assess whether a disease outbreaks; $Y(s,t) = 1$ if a location s is infected at time t, and 0 otherwise. We assume numerous locations are interconnected and impinge on each other; a spreading factor could be an epidemic on a human body, a pathogen in a farmland, or a virus in a computer network, for example. We propose an autologistic network model in order to describe how an infectious disease spreads out over space and time.

There are three motivations for the proposed model. First, no prior information about spatial neighborhood is given. Unlike common spatial data, closeness between locations might not provide a clue to spatial dependence, rather some unknown structures may explain it. For example, human body parts are complicatedly connected by neural network and the same is true for a computer system. Secondly, a disease of interest is unrecoverable. If a location is once infected $(= 1)$, it never returns to a normal status $(= 0)$ but reaches an absorbing state. Lastly, spatial dependence might be different depending on the previous status. Specifically, the locations, infected a long time ago, may have different impacts from the locations which are infected just now.

Many researchers have developed autologistic regressive models that can describe spatial dependence. Hughes et al. (2011), however, assumed closeness of locations determines a neighborhood structure among locations, and their model took into account spatial but atemporal correlation. To incorporate the feature of absorbing states, Kaiser et al. (2014) formulated a model by specifying sufficient support conditions for joint distribution of locations, however, a certain structure of neighborhood was yet required. Agaskar and Lu (2013) considered a network association to ease the requirement of pre-specified neighborhood. However, their model neither involved absorbing states nor a spatial association at the same time.

Predicting a spreading pattern of a disease or a virus over space and time is another primary goal in addition to description of spatial dependence. Zhu et al. (2005) developed a model for spatial-temporal binary data by formulating joint distribution to predict a future status as well as spatial correlation. However, absorbing states were not the case for their model and a pre-specified neighbor structure on a lattice was also required to fit the model. We in this paper formulate a joint distribution of locations, which is appropriate to a proposed model, according to Hammersley-Clifford theorem. Then, we make use of transition probabilities of joint responses, involving the case of absorbing states, in order to forecast a near future.

The remainder of this section is structured in the following order. In Section 4.2.2, we propose an autologistic network model with absorbing states. Section 4.2.3 discusses a maximum pseudo likelihood estimator that is based on an iterative expected-maximization algorithm. In Section 4.2.4, We provide a joint distribution of locations to make a prediction. Section 4.2.5 presents a simulation study to see how well our model estimation does inference.

4.2.2 Statistical Model

Suppose we observe a binary random variable that $Y_i(s_j, t)$ is 1 if a location s_j is infected by a certain disease or virus at time t for a subject i, and 0 if normal. Denote $\boldsymbol{Y}_{it}$ as a random vector of all locations; that is, $\{Y_i(s_1, t), \ldots, Y_i(s_M, t)\}^T$, in which M is the number of locations. The independent variables including an intercept and time t as well as other covariates, which potentially have influence on $Y_i(s_j, t)$, are denoted by $\boldsymbol{X}_i$.

We denote $p_i(s_j, t)$ as a conditional probability of infection at a location s_i and time t, given independent variables and binary responses of other locations at previous and current times, defined by

$$p_i(s_j, t) = P\{Y_i(s_j, t) = 1 | \boldsymbol{X}_i, Y_i(s_k, t-1), Y_i(s_k, t) \text{ for } \forall k \neq j\}.$$

Note that already infected locations, $\{s_j : Y_i(s_j, t-1) = 1\}$, do not need to be modeled since we assume a disease is unrecoverable so their probabilities of being infected at time t are always 1; state ‘infected’ is absorbing. For that reason, we model the conditional probability of infection for responses in the *active set*, a set of current responses that are previously not infected, which is denoted as

$$\mathcal{A}_{ijt} = \{Y_i(s_j, t) : Y_i(s_j, t-1) = 0\}. \tag{4.1}$$

For a fixed i and t, we propose a model for spatio-temporal binary responses $Y_i(s_j, t)$

in an active set $\mathcal{A}_{ijt}$ as

$$\text{logit}\{p_i(s_j,t)\} = \boldsymbol{X}_i^T\boldsymbol{\beta} + \sum_{s_k \in \mathcal{P}_i^0(s_j,t)} \eta_{0jk}\{Y_i(s_k,t) - \kappa_i\} + \sum_{s_k \in \mathcal{P}_i^1(s_j,t)} \eta_{1jk}\{Y_i(s_k,t) - \kappa_i\} \quad (4.2)$$

$$\text{subject to } \eta_{0jk} = \eta_{0kj} > 0 \text{ and } \eta_{1jk} = \eta_{1kj} > 0 \text{ for all } j \neq k$$

where logit$(p) = \log\{p/(1-p)\}$,

$$\kappa_i = \exp(\boldsymbol{X}_i^T\boldsymbol{\beta})/\{1 + \exp(\boldsymbol{X}_i^T\boldsymbol{\beta})\},$$

$\mathcal{P}_i^0(s_j,t)$ is a set of other locations that are previously normal and $\mathcal{P}_i^1(s_j,t)$ is a complement set of $\mathcal{P}_i^0(s_j,t)$ that contains previously infected locations. This partition can also be written in a set representation as follows,

$$\mathcal{P}_i^0(s_j,t) = \{s_k : k \neq j; Y_i(s_j, t-1) = 0\};$$
$$\mathcal{P}_i^1(s_j,t) = \{s_k : k \neq j; Y_i(s_j, t-1) = 1\},$$

for j in $\{1,\ldots,M\}$.

The reason for centering the last two terms by κ_i is to make the binary autocovariates' effect of 0's and 1's unbiased. If uncentered, $p_i(s_j,t)$ will be biased toward 1-valued autocovariates and so there will be nothing to do with 0-valued autocovariates as a role of explanatory variables; therefore, $p_i(s_j,t)$ will never decreases. We define a centering constant κ_i that makes a marginal expectation of $p_i(s_j,t)$ equal to an expectation under an *independence model*, as

$$\text{logit}\{p_i(s_j,t)\} = \boldsymbol{X}_i^T\boldsymbol{\beta}. \quad (4.3)$$

If $\boldsymbol{\beta} = 0$, κ_i is 0.5, so that 0 and 1 values are equitably distinguished by -0.5 and 0.5, respectively. See Caragea and Kaiser (2009) for details about centering.

By partitioning the locations into $\mathcal{P}^0$ (previously normal) and $\mathcal{P}^1$ (previously infected), our model (4.2) allows normal and infected locations having different impacts via η_{0jk} and η_{1jk}, respectively. It can also learn a network association among locations from data by allowing all possible connections between two locations to be estimated. Hence, the parameter η_{0jk} indicates an impact of the location s_k on s_j when s_k is previously healthy. Similarly, η_{1jk} is an impact of the location s_k on s_j when s_k is previously infected.

We restrict symmetricity and positivity on the coefficients of autocovariate, which are η-type estimates. First, it is natural to presume symmetricity for spatial dependence since η indeed represents a correlation between locations at the same time t. Secondly, we do not allow negative η estimates which have no meanings. It does not make sense that healthy locations make another to be infected or infected locations make another to stay healthy.

We illustrate a simple case here to promote better understanding on η coefficients. Suppose a subject i is observed with two normal and one infected locations at time $t-1$ such that $Y_i(s_1, t-1) = 0, Y_i(s_2, t-1) = 0, Y_i(s_3, t-1) = 1$, and (s_2, s_3) have a significant influence on s_1. The effect of s_2 is represented by a coefficient η_{012} because a location s_2 is not yet infected but normal at $t-1$. Accordingly, if s_2 is currently infected, i.e. $Y_i(s_2, t) = 1$, logit$\{p_i(s_1, t)\}$ will increase as much as $\eta_{012}(1-\kappa_i)$. Otherwise, logit$\{p_i(s_1, t)\}$ will decrease as much as $\eta_{012}(0-\kappa_i)$. This also implies that strongly linked locations are more likely to be infected or stay healthy simultaneously. On the other hand, s_1 will always be ill-affected by the location s_3, which is in absorbing state; logit$\{p_i(s_1, t)\}$ will increase as much as $\eta_{113} \cdot 1$.

The model (4.2) can also be written in a vector and matrix form as

$$\text{logit}\, p_i(s_j, t) = \boldsymbol{X}_i^T\boldsymbol{\beta} + \{\boldsymbol{\delta}^0_{ijt}(\boldsymbol{Y}_{it} - \kappa_i\mathbf{1})\}^T\boldsymbol{\eta}_{0j} + \{\boldsymbol{\delta}^1_{ijt}(Y_{it} - \kappa_i\mathbf{1})\}^T\boldsymbol{\eta}_{1j}, \tag{4.2$'$}$$

with the coefficients $\boldsymbol{\eta}_{0j}$ and $\boldsymbol{\eta}_{1j}$ denoted as a vector of length $(M-1)$ such that

$$\begin{aligned}\boldsymbol{\eta}_{0j} &= \{\eta_{0j1}, \ldots, \eta_{0j(j-1)}, \eta_{0j(j+1)}, \ldots, \eta_{0jM}\}^T;\\ \boldsymbol{\eta}_{1j} &= \{\eta_{1j1}, \ldots, \eta_{1j(j-1)}, \eta_{1j(j+1)}, \ldots, \eta_{1jM}\}^T,\end{aligned}$$

and more simply they can also be denoted by a vector of length $M(M-1)$ such that $\boldsymbol{\eta}_0 = \{\boldsymbol{\eta}_{01}, \boldsymbol{\eta}_{02}, \ldots, \boldsymbol{\eta}_{0M}\}^T$ and $\boldsymbol{\eta}_1 = \{\boldsymbol{\eta}_{11}, \boldsymbol{\eta}_{12}, \ldots, \boldsymbol{\eta}_{1M}\}^T$. Also, $\boldsymbol{\delta}^0_{ijt}$ implies a $(M-1) \times M$ matrix whose k^{th} row indicates whether s_k is in $\mathcal{P}^0_i(s_j, t)$ for all $k \neq j$, and $\boldsymbol{\delta}^1_{ijt}$ similarly indicates whether it is in $\mathcal{P}^1_i(s_j, t)$. For example, suppose $M = 4$ and $\mathbf{Y}_{i(t-1)} = (0, 0, 1, 1)^T$, then

$$\begin{aligned}\boldsymbol{\delta}^0_{i1t} &= \begin{pmatrix} . & 1 & . & . \\ . & . & 0 & . \\ . & . & . & 0 \end{pmatrix}; \quad \boldsymbol{\delta}^1_{i1t} = \begin{pmatrix} . & 0 & . & . \\ . & . & 1 & . \\ . & . & . & 1 \end{pmatrix};\\ \boldsymbol{\delta}^0_{i2t} &= \begin{pmatrix} 1 & . & . & . \\ . & . & 0 & . \\ . & . & . & 0 \end{pmatrix}; \quad \boldsymbol{\delta}^1_{i2t} = \begin{pmatrix} 0 & . & . & . \\ . & . & 1 & . \\ . & . & . & 1 \end{pmatrix},\end{aligned}$$

where '.' denotes numerically zero as well for void points while '0' meaningfully indicates its partition whether $\mathcal{P}^0$ or $\mathcal{P}^1$ based on the previous status. Note that $\boldsymbol{\delta}_{i3t}$ and $\boldsymbol{\delta}_{i4t}$ are not defined since $Y_i(s_3, t-1) = 1$ is absorbing.

4.2.3 Estimation

To estimate parameters, we approximate a joint likelihood by the product of full conditional likelihoods, that is the pseudo likelihood, for simplicity in computation. We also apply an l_1 regularization for sparsity of estimates to select the best subset of autocovariates for better interpretation of the model. Then, a penalized pseudo likelihood is maximized by an iterative Expected-Maximization (EM) algorithm. Note that a maximum pseudo likelihood estimator almost surely converges to a maximum likelihood estimator; see Besag (1975).

Let N be the number of subjects. We assume M number of locations are fully observed at every time points t, however, there could be censored observations in terms of time, which means the number of observing times can vary over subjects, say n_i for each subject i.

The conditional log-likelihood for the binary response is given by

$$l_{ijt}(\boldsymbol{\beta}, \boldsymbol{\eta}_0, \boldsymbol{\eta}_1) = Y_i(s_j, t) \cdot \text{logit}\{p_i(s_j, t)\} - \log[1 + \exp(\text{logit}\{p_i(s_j, t)\})] \tag{4.4}$$

for i, j, and t such that $Y_i(s_j, t) \in \mathcal{A}_{ijt}$, where $p_i(s_j, t)$ is modeled by (4.2). Accordingly, we maximize the l_1-penalized pseudo log-likelihood,

$$F_\lambda(\boldsymbol{\beta}, \boldsymbol{\eta}_0, \boldsymbol{\eta}_1) = \sum_{i=1}^{N} \sum_{t=1}^{n_i} \sum_{j=1}^{M} l_{ijt}(\boldsymbol{\beta}, \boldsymbol{\eta}_0, \boldsymbol{\eta}_1) I_{\mathcal{A}_{ijt}} - \lambda \sum_{j<k} (|\eta_{0jk}| + |\eta_{1jk}|), \tag{4.5}$$

where $I_{\mathcal{A}_{ijt}}$ is an indicator function whether $Y_i(s_j, t) \in \mathcal{A}_{ijt}$ and λ is a tuning parameter for a regularization. It should be noted we only penalize the parameters which stand for spatial dependence; $\boldsymbol{\eta}_0$ and $\boldsymbol{\eta}_1$, not $\boldsymbol{\beta}$. The sparsity on $\boldsymbol{\eta}_0$ and $\boldsymbol{\eta}_1$ is necessary, not only because of large number of parameters, $M(M-1)$ elements in each $\boldsymbol{\eta}$, but also in order to reflect the reality that a specific location has possibly no association with another.

The ordinary maximum likelihood estimation of a logistic regression such as Newton's method cannot apply straightforwardly because the parameter κ in (4.2) is a nonlinear function of $\boldsymbol{\beta}$. Instead, we first estimate parameters which are linear and then update κ iteratively until all converges as the following steps,

1. Fit the independent model (4.3); set $\hat{\boldsymbol{\beta}}^{(0)}$ by an initial;
2. Given $\hat{\kappa}_i^{(l-1)} = \text{logit}^{-1}\{\boldsymbol{X}_i^T \hat{\boldsymbol{\beta}}^{(l-1)}\}$, maximize (4.5) to estimate $\hat{\boldsymbol{\beta}}^{(l)}, \hat{\boldsymbol{\eta}}_0^{(l)}, \hat{\boldsymbol{\eta}}_1^{(l)}$;
3. Update $\hat{\kappa}_i^{(l)} = \text{logit}^{-1}\{\boldsymbol{X}_i^T \hat{\boldsymbol{\beta}}^{(l)}\}$, and iterate step 2 until all converges.

The standard logistic regression method, described by Hastie and Pregibon (1992), and the ordinary way of generalized linear model with regularization, in Friedman et al. (2010), are used in step 1 and step 2, respectively. Once iteration stops with converged coefficients, we perform a procedure of bias correction suggested by Tang et al. (2016).

The optimal tuning parameter λ can be chosen through cross-validation at step 2 during the first iteration and then fixed for the remaining iterations. Alternatively, one can manually choose the λ to meet the desired amount of sparsity or setup different sparsity by λ_0 and λ_1 on $\boldsymbol{\eta}_0$ and $\boldsymbol{\eta}_1$, respectively.

4.2.4 Prediction

Besides describing spatial network association, our study ultimately aims to predict an infectious disease progression over space and time. Specifically, when responses at $t-1$, $\boldsymbol{Y}_{i(t-1)} = \{Y_i(s_1, t-1), \dots, Y_i(s_M, t-1)\}$, are given, it is of our interest to predict the next responses, that is, $\boldsymbol{Y}_{it} = \{Y_i(s_1, t), \dots, Y_i(s_M, t)\}$. To that end, we in this section formulate the conditional distribution of joint responses, i.e. $P\{\boldsymbol{Y}_{it} | \boldsymbol{Y}_{i(t-1)}\}$.

As the probability of infection is modeled only for the previously healthy responses in $\mathcal{A}_{ijt}$, the joint distribution of $\boldsymbol{Y}_{it}$ conditional on $\boldsymbol{Y}_{i(t-1)}$ should also be obtained on a relevant support. Suppose that a set $\mathcal{S}$ includes all possible outcomes of M locations.

Since each state is either 0 or 1, there are 2^M elements in $\mathcal{S}$. We now define the support of a random vector $\boldsymbol{Y}_{it}$ conditional on its previous status $\boldsymbol{Y}_{i(t-1)}$ as

$$\mathcal{S}_{it} = \{\boldsymbol{Y}_{it} \in \mathcal{S} | Y_i(s_j, t) = 1 \text{ for } s_j \text{ s.t. } Y_i(s_j, t-1) = 1\}. \tag{4.6}$$

Note that the elements, which are against absorbing states, are excluded out of $\mathcal{S}$. That is, any joint outcomes with $Y_i(s_j, t) = 0$ and $Y_i(s_j, t-1) = 1$ are not in $\mathcal{S}_{it}$. Therefore, if k locations are infected at $t-1$, the support set $\mathcal{S}_{it}$ will have 2^{M-k} candidate outcomes which can be potentially observed at the next time t. For example, if the last location among $M = 4$ is infected alone but remaining three are normal at $t-1$, i.e. $\boldsymbol{Y}_{i(t-1)} = (0, 0, 0, 1)^T$, then $\mathcal{S}_{it}$ consists of the following $7 (= 2^3)$ outcomes such that $(0, 0, 0, 1)^T$, $(0, 0, 1, 1)^T$, $(0, 1, 0, 1)^T$, $(1, 0, 0, 1)^T$, $(0, 1, 1, 1)^T$, $(1, 0, 1, 1)^T$, $(1, 1, 0, 1)^T$, and $(1, 1, 1, 1)^T$.

We consequently formulate the joint distribution of $\boldsymbol{Y}_{it} \in \mathcal{S}_{it}$, conditioned on the previous responses $\boldsymbol{Y}_{i(t-1)}$ and the coefficients from (4.2) as well as subject-specific covariates $\boldsymbol{X}_i$, as

$$\begin{aligned} &P\{\boldsymbol{Y}_{it} | \boldsymbol{Y}_{i(t-1)}; \boldsymbol{X}_i, \boldsymbol{\beta}, \boldsymbol{\eta}_0, \boldsymbol{\eta}_1\} \\ &\quad \propto \sum_{j=1}^{M} Y_i(s_j, t) \boldsymbol{X}_i^T \boldsymbol{\beta} - \sum_{j=1}^{M} \Big\{ Y_i(s_j, t) \sum_{s_k \in \mathcal{P}_i^0(s_j, t)} \eta_{0jk} \kappa_i \Big\} - \sum_{j=1}^{M} \Big\{ Y_i(s_j, t) \sum_{s_k \in \mathcal{P}_i^1(s_j, t)} \eta_{1jk} \kappa_i \Big\} \\ &\qquad + \frac{1}{2} \sum_{j=1}^{M} \Big\{ Y_i(s_j, t) \sum_{s_k \in \mathcal{P}_i^0(s_j, t)} \eta_{0jk} Y_i(s_k, t) \Big\} + \frac{1}{2} \sum_{j=1}^{M} \Big\{ Y_i(s_j, t) \sum_{s_k \in \mathcal{P}_i^1(s_j, t)} \eta_{1jk} Y_i(s_k, t) \Big\} \end{aligned} \tag{4.7}$$

with a normalization by the sum of right sides over all i and t in $\mathcal{S}_{it}$. Since the entries of $\boldsymbol{X}_i^T \boldsymbol{\beta} - \sum_{\mathcal{P}^0} \eta_{0jk} \kappa_i - \sum_{\mathcal{P}^1} \eta_{1jk} \kappa_i$ do not overlap with $\boldsymbol{Y}_{it} = \{Y_i(s_1, t), \ldots, Y_i(s_M, t)\}$, they contribute to the joint distribution completely. Whereas, the contribution of the terms $\sum_{\mathcal{P}^0} \eta_{0jk} Y_i(s_k, t)$ and $\sum_{\mathcal{P}^1} \eta_{1jk} Y_i(s_k, t)$ is halved by means of Hammersley-Clifford theorem (Hammersley and Clifford, 1971). Note that since we do not consider more than

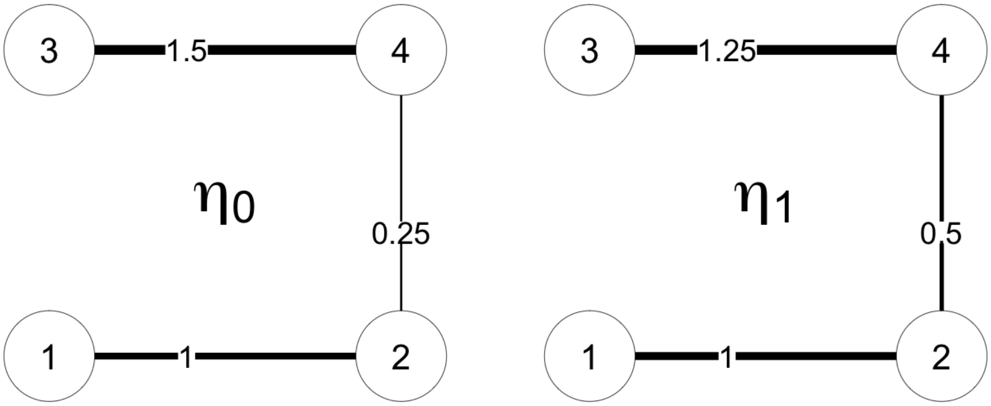


Figure 4.1: True parameters on a graph; Nonzero values of $\eta_{jk}(=\eta_{kj})$ are labeled on corresponding edges while zeros are not drawn; The width of edges represents the strength of conditional dependence between nodes.

pairwise dependencies, the joint distribution is derived by the second order. See also the appendix of Hughes et al. (2011).

4.2.5 Simulation Study

In simulation study, we have an experiment how well our model make an inference about true parameters, particularly $\boldsymbol{\eta}_0$ and $\boldsymbol{\eta}_1$. Suppose there are four nodes (s_1, s_2, s_3, s_4) that possibly have an association to each other with absorbing states. The length of $\boldsymbol{\eta}_0$ and $\boldsymbol{\eta}_1$ is accordingly $4 \times (4-1) = 12$ each.

We begin assigning values to $\boldsymbol{\eta}_0$ and $\boldsymbol{\eta}_1$ that are moderately sparse and symmetric as illustrated in Figure 4.1. We assume there are no explanatory variables $\boldsymbol{X}$ for simplicity. Note that $\beta = 0$ leads a fair centering on binary data, which means autocovariates of 0 and 1 are transformed to -0.5 and 0.5, respectively; $\kappa = 0.5$.

The pre-specified parameters determine the transition probabilities through (4.7). Since we have four locations, there are $2^4 (= 16)$ number of joint outcomes. The first row of each cell in Table 4.1 displays a true transition probability from $t-1$ to t taking into account the absorbing states. At this point, we can provide interesting insights between $\boldsymbol{\eta}$'s and their corresponding transition probabilities. For example, because of strong association between s_3 and s_4 in $\boldsymbol{\eta}_0$, transition probabilities from $(1,0,0,0)$ to $(1,0,0,0)$, $(1,0,1,1)$ or $(1,1,0,0)$, that is the probabilities that s_3 and s_4 are in the same status, are relatively higher than others except the completely infected case; their probabilities are $0.136, 0.120,$ and 0.198, respectively.

We generate a base status from the Bernoulli distribution with a probability 0.25, which is intended to make one location among four is infected at $t = 0$ in an average sense; that means,

$$Y(s_j, 0) = 1 \text{ with probability } 0.25$$

for any $j \in \{1, 2, 3, 4\}$. Then the next status is generated based on the transition probabilities, and this generation is repeated until all locations are infected.

One simulation data consist of 500 initial status, and coefficients are estimated as described in Section 4.2.3. We run this simulation process for 100 times to obtain empirical confidence interval of each entry in transition probability matrix. The second row of each cell in Table 4.1 indicates 95% confidence interval. It is remarkable that all of true probabilities fall into their confidence interval, so we see that our estimation makes a reasonable inference.

4.2.6 Application: ALS Patients Data

Amyotrophic lateral sclerosis (ALS), also known as Lou Gehrig's disease, is a neurological disease that typically first affects a particular group of muscles to make them lose their physical functionality and then spreads to other locations as it progresses. As the disease progresses, the brain of a patient gradually loses the ability to signal and control muscle movement that leads to muscle weakness, impaired physical functionality and finally death. Although such spread mechanism is critical to the development of clinical therapies, it remains a large unknown problem.

Table 4.1: Each cell consists of true transition probability and 95% confidence interval from 100 simulations; Non-absorbing cases are not taken into account so they have zero probabilities, denoted by '.'

$t-1\backslash t$	0000	0001	0010	0011	0100	0101	0110	0111
0000	0.112	0.047	0.053	0.099	0.06	0.032	0.028	0.068
	(0.096,0.125)	(0.037,0.056)	(0.041,0.063)	(0.085,0.113)	(0.052,0.075)	(0.025,0.039)	(0.024,0.037)	(0.056,0.079)
0001	.	0.094	.	0.198	.	0.064	.	0.136
	.	(0.075,0.111)	.	(0.17,0.226)	.	(0.05,0.078)	.	(0.112,0.157)
0010	.	.	0.106	0.198	.	.	0.057	0.136
	.	.	(0.083,0.126)	(0.17,0.226)	.	.	(0.047,0.074)	(0.112,0.157)
0011	.	.	.	0.292	.	.	.	0.201
	.	.	.	(0.255,0.327)	.	.	.	(0.166,0.236)
0100	.	.	.	.	0.12	0.064	0.057	0.136
	.	.	.	.	(0.104,0.149)	(0.05,0.078)	(0.047,0.074)	(0.112,0.157)
0101	.	.	.	.	.	0.121	.	0.256
	.	.	.	.	.	(0.095,0.144)	.	(0.221,0.294)
0110	.	.	.	.	.	.	0.111	0.266
	.	.	.	.	.	.	(0.09,0.145)	(0.22,0.302)
0111	.	.	.	.	.	.	.	0.378
	.	.	.	.	.	.	.	(0.326,0.43)
1000	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.
1001	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.
1010	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.
1011	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.
1100	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.
1101	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.
1110	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.
1111	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.

Table 4.1 continued

$t-1\backslash t$	1000	1001	1010	1011	1100	1101	1110	1111
0000	0.068 (0.056,0.079)	0.028 (0.024,0.037)	0.032 (0.025,0.039)	0.06 (0.052,0.075)	0.099 (0.085,0.113)	0.053 (0.041,0.063)	0.047 (0.037,0.056)	0.112 (0.096,0.125)
0001	. .	0.057 (0.047,0.074)	. .	0.12 (0.104,0.149)	. .	0.106 (0.083,0.126)	. .	0.225 (0.192,0.25)
0010	. .	. .	0.064 (0.05,0.078)	0.12 (0.104,0.149)	. .	. .	0.094 (0.075,0.111)	0.225 (0.192,0.25)
0011	. .	. .	. .	0.177 (0.154,0.217)	. .	. .	. .	0.331 (0.294,0.365)
0100	. .	. .	. .	. .	0.198 (0.17,0.226)	0.106 (0.083,0.126)	0.094 (0.075,0.111)	0.225 (0.192,0.25)
0101	. .	. .	. .	. .	. .	0.2 (0.164,0.229)	. .	0.423 (0.382,0.477)
0110	. .	. .	. .	. .	. .	. .	0.183 (0.144,0.219)	0.439 (0.382,0.48)
0111	. .	. .	. .	. .	. .	. .	. .	0.622 (0.57,0.674)
1000	0.136 (0.112,0.157)	0.057 (0.047,0.074)	0.064 (0.05,0.078)	0.12 (0.104,0.149)	0.198 (0.17,0.226)	0.106 (0.083,0.126)	0.094 (0.075,0.111)	0.225 (0.192,0.25)
1001	. .	0.112 (0.092,0.145)	. .	0.237 (0.199,0.288)	. .	0.209 (0.164,0.24)	. .	0.442 (0.382,0.487)
1010	. .	. .	0.128 (0.097,0.154)	0.239 (0.205,0.289)	. .	. .	0.186 (0.147,0.219)	0.447 (0.384,0.494)
1011	. .	. .	. .	0.349 (0.296,0.413)	. .	. .	. .	0.651 (0.587,0.704)
1100	. .	. .	. .	. .	0.318 (0.279,0.359)	0.17 (0.136,0.2)	0.15 (0.123,0.179)	0.361 (0.323,0.403)
1101	. .	. .	. .	. .	. .	0.321 (0.26,0.356)	. .	0.679 (0.644,0.74)
1110	. .	. .	. .	. .	. .	. .	0.294 (0.234,0.353)	0.706 (0.647,0.766)
1111	. .	. .	. .	. .	. .	. .	. .	1 (1,1)

It is challenging to explore the association among muscles since the ALS disease pattern in reality does not merely depend on the closeness of muscles, however it is complicatedly associated with neurological features. For example, when one's right wrist muscle loses its function, the left wrist muscle, although far away from the right one, is more likely to be impaired next, rather than the right elbow, which is physically closer to the right wrist.

Another challenging is to reflect the feature of absorbing states of ALS disease; muscles can never recover its function once infected. Not only should the data be recorded reflecting this feature, but a model should also be developed not contradicting this fact. With a similar motivation, Kaiser et al. (2014) developed a Markov random field model to involve absorbing states of plant disease, however, we cannot directly apply this model because it assumes the neighborhood structure of observations based on their adjacency.

Therefore, the proposed model, which is assumption-free about the neighboring structure of muscles and can take into account absorbing feature, is applied to ALS patients data in this section.

We include $N = 926$ ALS patients who visit a clinic and examine their muscle strength at $M = 16$ body locations multiple times with an interval of two months during an year; that is, $t = 0, 2, 4, 6, 8, 10, 12$. The measured muscles are the right and left side of wrist extension, elbow extension, elbow flexors, shoulder flexors, ankle dorsiflexion, knee flexion, knee extension and hip flexors, as illustrated in Figure 4.2. Each patient's demographical characteristics, such as age, sex, and weight, and clinical records, such as clinical visit number, symptom onset site, and symptom duration, are also recorded. There are censored observations for some patients because of missed visits or deaths. We include the independent variables $\boldsymbol{X}_i$ as the visiting times (t), the symptom onset site (a binary variable whether it is on bulbar or others) and the symptom duration when patients entered the study.

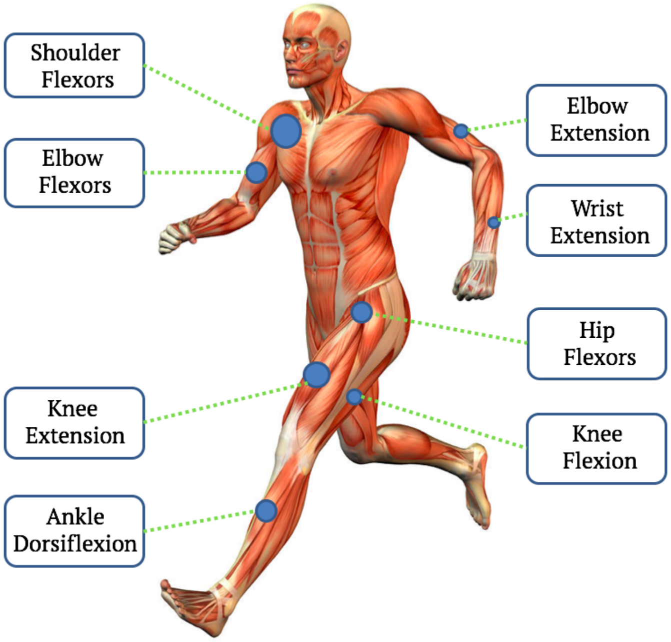


Figure 4.2: Measured muscles on a human body map; The right and left sides of eight muscles, totally sixteen number of muscles, are examined.

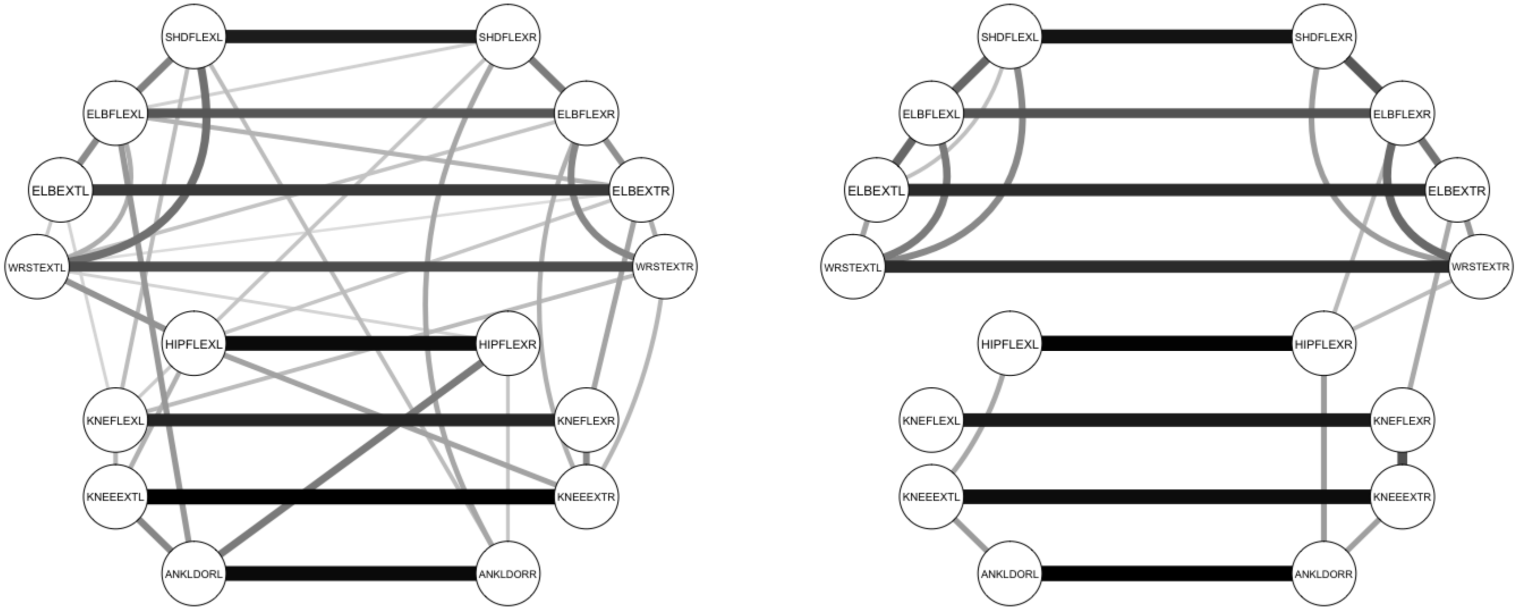
We utilize muscle strengths to generate a local binary variable independently on whether each muscle is infected or not. The regression equations, provided by NIMS (1996) and Bohannon (1997), define the normal strength of healthy people based on their gender, age, height, and weight. By fitting these equations using our patients' information, we can compute the expected muscle strength when assuming that their muscles are yet healthy. The computed strength is then used as a benchmark to determine whether the observed strength is infected or not. As such, we assess each muscle status as impaired $(= 1)$ if its measured strength is 40% less than the computed/expected normal strength, or healthy $(= 0)$ otherwise. Note that a muscle, which has been marked as infected, is always regarded as infected after then. Such binary decisions will be assumed to be the true latent binary process for simplicity in this application task.

The tuning parameter for regularization, λ, is obtained by 10-fold cross validation. We also break the iteration when every updated estimate falls within 1% difference from the old estimates; they converge in 5 iterations.

Figure 4.3 illustrates a vector of estimated $\boldsymbol{\eta}_0$ and $\boldsymbol{\eta}_1$ in a graphical way. Most interestingly, horizontal connections between right and left side of each muscle, both previously healthy and infected, are mostly stronger than other connections. This implies, no matter what status a muscle was at the past visit, that muscle is likely to keep its status as same as its opposite side. Also, the estimates of $\boldsymbol{\eta}_1$ are sparser than those of $\boldsymbol{\eta}_0$, under the same amount of regularization, i.e. at the same value of λ. This means, the network that muscles affect others in a way of keeping healthiness together or being newly infected at the same time is more complicated than the network that already infected muscles make others also infected. This could also be interpreted as newly infected muscles have diversified impacts on others, even vertically between upper and lower body parts, while the muscles infected far in the past are mostly associated only with physically neighboring muscles or their opposite sides.

To predict the ALS disease spreading pattern on a human body, suppose a hypothetical male patient enter a study when only one muscle has been infected by ALS since 21.6 months ago, and his symptom onset site is off the bulbar muscle. Although the prediction could be done by sampling from the joint distribution obtained from 4.2.4, we do this by taking the most probable status as the next status according to that distribution. In this process, we ignore the cases that a probability of staying as before without progression is the highest. Also, instead of providing the transition probability matrix which is a huge matrix as $2^{16} \times 2^{16}$, we keep predicting one time ahead future status until all body parts are infected to see the full path of disease progression over space and time.

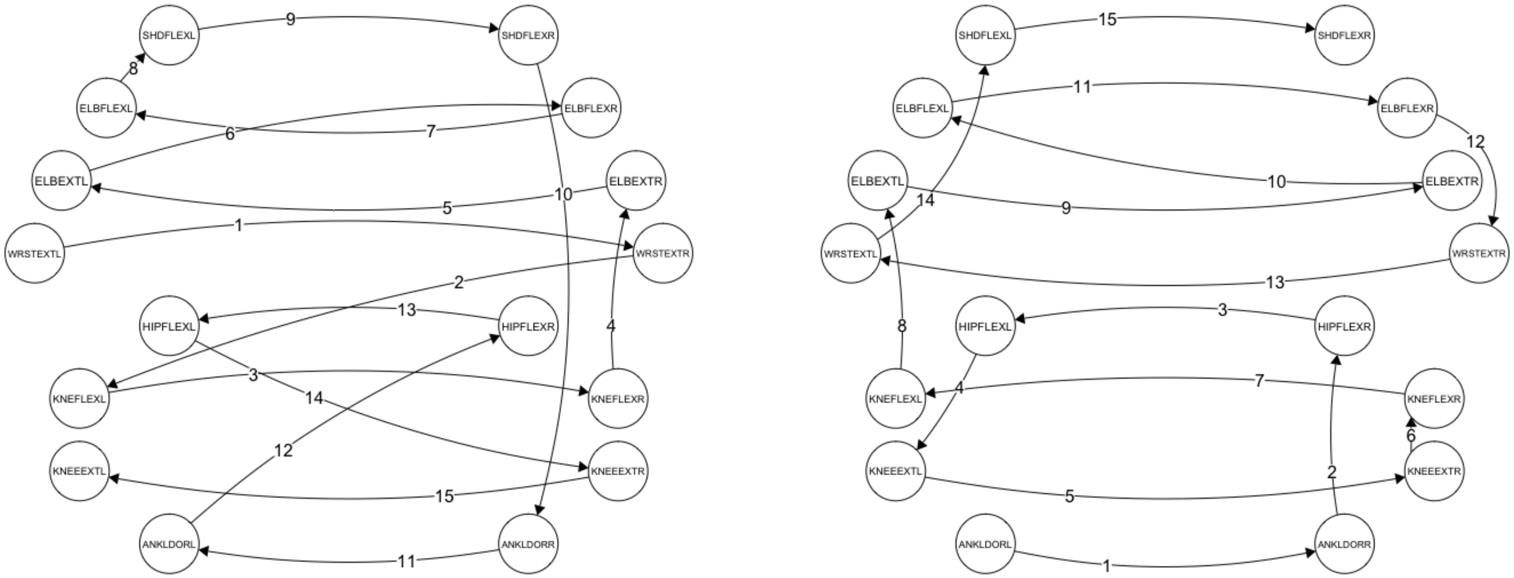
Figure 4.4 illustrates predicted full paths of disease progression for the hypothetical patient. In case that the first infect muscle is a left wrist extension, the disease spreads to a right wrist extension first, and the infection on knee muscles follows after. This



(a) $\boldsymbol{\eta}_0$

(b) $\boldsymbol{\eta}_1$

Figure 4.3: Estimates on a graph; The color depth and width of edges represent strength of conditional association.



(a) Predicted path given a left wrist is infected at $t = 0$

(b) Predicted path given a left ankle is infected at $t = 0$

Figure 4.4: Prediction of ALS disease progression over time; Labels at each edge denote time t; Each predicted pathway is the most probable one based on the joint distribution.

predicted pattern is in agreement with our estimates as Figure 4.3 implies in that the most of spreading directions are between left and right side. Also, the knee muscles, which are highly linked to upper muscles, are predicted to be first infected among the lower body parts, and so are the elbow muscles among the upper body parts. The overall spreading path for the case that a left ankle flexor is initially infected is quite similar; it shows right-left paths and elbow-knee links.

4.3 Binary Hidden Markov Model on Flexible Spatial Network

4.3.1 Introduction

There is another challenging of reflecting the feature of absorbing states of ALS disease; observed binary states may not be in agreement with absorbing states while their underlying states are absorbing in an infected state. Figure 4.5 illustrates example sequences of elbow extensor and flexor muscle strength measurements in a relative unit to the normal strength. Although this patient is suffering from ALS disease, his or her muscle strengths are not measured in a decreasing order over time but it seems to take a favorable turn at some moments; this is not true but could happen because of measurement errors or a patient's overall condition regardless of the disease at that visit. Hence, we propose to incorporate hidden Markov model with 'true' binary states that are absorbing in an infected state while 'observed' binary states are not necessarily. An autologistic regressive model with absorbing will be developed for true hidden binary states in this study.

The remainder of this section is structured in the following order. In Section 4.3.2, we describe the proposed hidden Markov model to explore the ALS disease spreading pattern as well as true hidden states with a feature of absorbing. Section 4.3.3 shows the results obtained by applying the model to the provided ALS patients data.

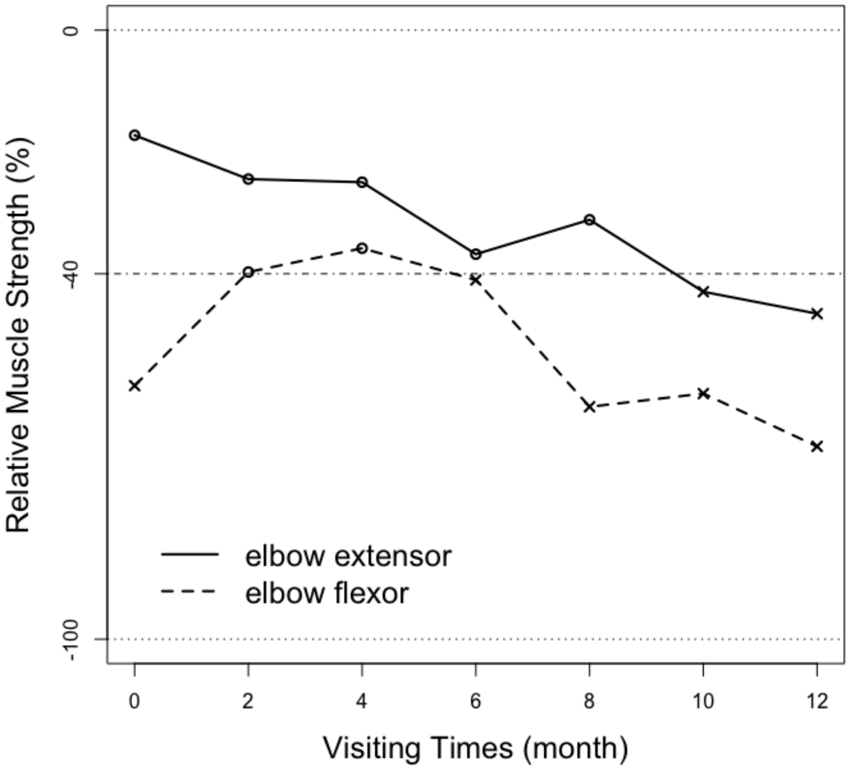


Figure 4.5: Two selected sequences of 'observed' relative muscle strength; × denotes 'infected' $Y_i(s_j, t) = 1$ if strength is 40% less than normal; ○ denotes 'healthy' $Y_i(s_j, t) = 0$ otherwise; The binary states of elbow extensor are in agreement with absorbing while those of elbow flexor are against absorbing.

4.3.2 Statistical Model and Estimation

Suppose $S_i(s_j, t)$ is a true binary state of a subject i indicating

$$S_i(s_j, t) = \begin{cases} 1 & \text{if a location } s_j \text{ is infected at time } t \\ 0 & \text{otherwise} \end{cases}$$

for $i \in \{1, \ldots, N\}$, $j \in \{1, \ldots, M\}$, and $t \in \{1, \ldots, n_i\}$. True states are hidden and absorbing in state 1 (infected); once infected, it never goes back to state 0 (healthy). In reality, we however observe $Y_i(s_j, t)$, which is also binary but may not be absorbing because of measurement errors, such that

$$Y_i(s_j, t) = \begin{cases} 1 & \text{if a location } s_j \text{ is `observed' as infected at time } t \\ 0 & \text{otherwise.} \end{cases}$$

We propose a hidden Markov model (HMM) for true state sequences occurring on a number of locations over time. Like the standard HMM, there are two type of parameters to estimate; emission probabilities and transition probabilities. Emission probabilities govern the distribution of observed states given hidden states, while transition probabilities control the progression of hidden states from time $t - 1$ to t. We assume emission probabilities are always common to all locations, whereas, transition probabilities are different in location and time. Figure 4.6 briefly introduces the structure of our model.

Accordingly, our model is divided into two stages; Stage I illustrates the estimation of emission probabilities and Stage II describes the estimation of transition probabilities through autologistic regression model. We then jointly update hidden states for each location by applying the Viterbi algorithm (Viterbi, 1967). Each of these two stages and updating procedure works when the others are conditional. For example, true states are

$$\begin{tikzcd}
{} \arrow[r] & S_i(s_j,t-1) \arrow[r, "\boldsymbol{A}_{ijt}"] \arrow[d, "\boldsymbol{E}"] & S_i(s_j,t) \arrow[r, "\boldsymbol{A}_{ij(t+1)}"] \arrow[d, "\boldsymbol{E}"] & S_i(s_j,t+1) \arrow[r] \arrow[d, "\boldsymbol{E}"] & {} \\
& Y_i(s_j,t-1) & Y_i(s_j,t) & Y_i(s_j,t+1) &
\end{tikzcd}$$

Figure 4.6: Structure of hidden Markov model; $\boldsymbol{E}$ = emission probability from hidden to observed states; $\boldsymbol{A}_{ijt}$ = transition probability of s_j from $t-1$ to t for a subject i

.

determined by emission and transition probabilities, while emission probabilities and transition probabilities can be estimated under the condition true states are provided. As such, we iterate them until all estimates and updates are converged.

Stage I: Emission Probabilities

Emission probabilities describe the conditional distribution of observations when hidden states are given. For $\delta, \gamma \in \{0, 1\}$, denote the probability that a hidden state δ emits a state γ as

$$e_\delta(\gamma) = P\{Y_i(s_j,t) = \gamma | S_i(s_j,t) = \delta\}, \tag{4.8}$$

such that $\sum_{\gamma=0}^{1} e_\delta(\gamma) = 1$ for a fixed δ. Since hidden and observed states are both binary in our study, we can construct the 2×2 emission matrix $\boldsymbol{E}$

$$\boldsymbol{E} = \begin{pmatrix} 1 - e_0(1) & e_0(1) \\ e_1(0) & 1 - e_1(0) \end{pmatrix},$$

where $e_0(1)$ and $e_1(0)$ are in fact misclassification probabilities; they can also be described as false positive and false negative, respectively.

The estimation of these two probabilities is done simply because we assume the stationary emission probability for all i, j and t. Given true states $S_i(s_j,t)$ for all i, j and t, we then compute the empirical frequency of misclassified observations such that

$$\hat{e}_0(1) = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{t=1}^{n_i} I\{Y_i(s_j,t)=1|S_i(s_j,t)=0\}/\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{t=1}^{n_i} I\{S_i(s_j,t)=0\};$$
$$\hat{e}_1(0) = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{t=1}^{n_i} I\{Y_i(s_j,t)=0|S_i(s_j,t)=1\}/\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{t=1}^{n_i} I\{S_i(s_j,t)=1\},$$

where $I(\mathcal{G})$ is an indicator function of a set $\mathcal{G}$.

Stage II: Transition Probabilities

Transition probabilities control the change of true hidden states from time $t-1$ to t; $P\{S_i(s_j,t)|S_i(s_j,t-1)\}$. Since we consider binary data that are absorbing in the 'infected' state, our only interest is in the change from 'healthy' to 'infected'; from 0 to 1. In other words, the transition probability from 1 to 0 is zero in any case because once infected location will not change its state but always stay at 1. Therefore, the transition probability matrix from $t-1$ to t for a location s_j of a subject i is given by

$$\boldsymbol{A}_{ijt} = \begin{pmatrix} 1-p_i(s_j,t) & p_i(s_j,t) \\ 0 & 1 \end{pmatrix},$$

where each row and column corresponds to binary states at $t-1$ and t, respectively. Also, $p_i(s_j,t)$ is defined as

$$p_i(s_j,t) = P\{S_i(s_j,t)=1|S(s_j,t-1)=0, S(s_k,t-1)\ \forall k\neq j\},$$

which indicates the conditional probability of s_j being newly infected at t given the previous states of other locations. Note that transition probabilities from 0 to 1 are spatio-time-varying for each subject, which means they are non-stationary in location and time.

We propose the autologistic regressive model for $p_i(s_j, t)$ as below

$$\text{logit}\, p_i(s_j,t) = \boldsymbol{X}_i^T\boldsymbol{\beta} + \sum_{s_k \in \mathcal{N}_{ijt}^0} \eta_{0jk} S_i^*(s_k, t-1) + \sum_{s_k \in \mathcal{N}_{ijt}^1} \eta_{1jk} S_i^*(s_k, t-1), \tag{4.9}$$

where $\boldsymbol{X}_i$ are subject-specific covariates that possibly affect the state transition, and $S_i^*(s_j, t)$ denotes a centered state, $S_i^*(s_j, t) = S_i(s_j, t) - 0.5$, so that the autocovariate effects of 0 and 1 are fairly reflected through -0.5 and 0.5, respectively. We also separate the effect of autocovariates based on their previous states by defining neighborhood such that

$$\mathcal{N}_{ijt}^0 = \{s_k | S_i(s_k, t-1) = 0 \text{ and } s_k \in \mathcal{N}_j\};$$
$$\mathcal{N}_{ijt}^1 = \{s_k | S_i(s_k, t-1) = 1 \text{ and } s_k \in \mathcal{N}_j\},$$

where $\mathcal{N}_j$ consists of locations that are neighbors of s_j; denoted by $\mathcal{N}_j = \{s_k | s_k \sim s_j\}$. It is remarkable that $\mathcal{N}_j$ not only could be a complete network among locations such that $\mathcal{N}_j = \{s_k | s_k \neq s_j\}$ but also could be specified based on any prior knowledge about spatial dependence structure. Accordingly, the coefficient η_{0jk} implies the effect of previously healthy neighbor s_k on s_j, and similarly, η_{1jk} implies the effect of previously infected neighbor s_k on s_j. Not only does the suggested autologistic model make any pairwise locations connectable within the neighborhood, but it also allows the directed impacts, which means we do not restrict $\eta_{0jk} = \eta_{0kj}$ nor $\eta_{1jk} = \eta_{1kj}$ for any $j \neq k$.

For computation efficiency in estimation, we maximize the product of conditional densities f_{ijt}, which is defined on the support $\mathcal{A}_{ijt} = \{S_i(s_j, t) | S_i(s_j, t-1) = 0\}$, as

$$f_{ijt}\{S_i(s_j,t) | S_i(s_k, t-1)\ \forall k \neq j\} = p_i(s_j,t)^{S_i(s_j,t)}\{1 - p_i(s_j,t)\}^{1-S_i(s_j,t)},$$

where $p_i(s_j, t)$ is modeled as (4.9), also known as the pseudo likelihood. Note that the maximum pseudo likelihood estimator almost surely converges to the maximum likelihood estimator; see Besag (1975). Additionally, we do this estimation under a sparsity condition on the coefficients so as to take benefits in model interpretation; specifically, it can choose the simpler model by selecting the best subset of autocovarates. Hence, we maximize the l_1-penalized pseudo log-likelihood such that

$$F_\lambda(\boldsymbol{\beta}, \boldsymbol{\eta}_0, \boldsymbol{\eta}_1) = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{t=1}^{n_i}\Big\{S_i(s_j,t)\cdot\text{logit}\, p_i(s_j,t) - \log[1+\exp\{\text{logit}\, p_i(s_j,t)\}]\Big\} I_{\mathcal{A}_{ijt}} - \lambda \sum_{j\neq k}(|\eta_{0jk}| + |\eta_{1jk}|)$$

where $I_{\mathcal{A}_{ijt}}$ denotes an indicator function whether $S_i(s_j,t) \in \mathcal{A}_{ijt}$ and both of $\boldsymbol{\eta}_0$ and $\boldsymbol{\eta}_1$ are a $M(M-1)$ vector of $\{\eta_{0jk}\}_{j\neq k}$ and $\{\eta_{1jk}\}_{j\neq k}$, respectively. Also, λ is a tuning parameter determining the amount of penalization, and its optimal value can be chosen by K-fold cross-validation.

Updating Hidden States

There are many techniques regarding the probability of one or more hidden states; for example, the forward algorithm, the forward-backward algorithm, and the Viterbi algorithm. See also Rabiner (1989) for details of these techniques. We apply the Viterbi algorithm (Viterbi, 1967), which is commonly used to find the most likely sequence of true hidden states given the model parameters and a sequence of observed states, $\{Y_i(s_j,1),\ldots,Y_i(s_j,n_i)\}$. This algorithm mainly differs from others in that a joint probability of hidden states is of interest rather than a probability of a single hidden state.

Let us briefly introduce the Viterbi algorithm in this subsection. We apply this algorithm for a fixed subject i and location s_j, so hereafter S_t denotes $S_i(s_j,t)$ for the simplicity of notation. Likewise, $Y_t = Y_i(s_j,t)$ and $\boldsymbol{A}_t = \boldsymbol{A}_{ijt}$.

Given transition probability matrix $\boldsymbol{A}_t$ and emission probability matrix $\boldsymbol{E}$, the probability of most probable path having $\delta \in \{0, 1\}$ as a true state at t is defined by

$$v_\delta(t) = \max_{S_1,\dots,S_t} P\{S_1, \dots, S_{t-1}, S_t = \delta, Y_1, \dots, Y_t | \boldsymbol{A}_t, \boldsymbol{E}, \pi_0\}, \tag{4.10}$$

which can be recursively calculated as

$$v_\delta(t) = \max_{\gamma \in \{0,1\}} v_\gamma(t-1)\, a_{\gamma\delta}\, e_\delta(Y_t),$$

where $a_{\gamma\delta}$ is a transition probability from γ to δ, determined by $\boldsymbol{A}_t$, and $e_\delta(\cdot)$ is an emission probability as defined in (4.8). The probability of state 1 being the initial state, $\pi_0 = P(S_0 = 1)$ could be fairly set as 0.5 or estimated through the model without autocovariates such that

$$\text{logit}\, \pi_0 = \boldsymbol{X}^T \beta.$$

Coming back to the original notation, we jointly update true states $\{S_i(s_j, 1), \dots, S_i(s_j, n_i)\}$ as the most likely sequence generating $\{Y_i(s_j, 1), \dots, Y_i(s_j, n_i)\}$ for each i and j, based on (4.10) computed by the last time point n_i.

It is remarkable that the updating procedure we perform here differs from other common applications of Viterbi algorithm, in that the prior probability π_0 and transition probabilities are not consistent over subject and location.

4.3.3 Application: ALS Patients Data

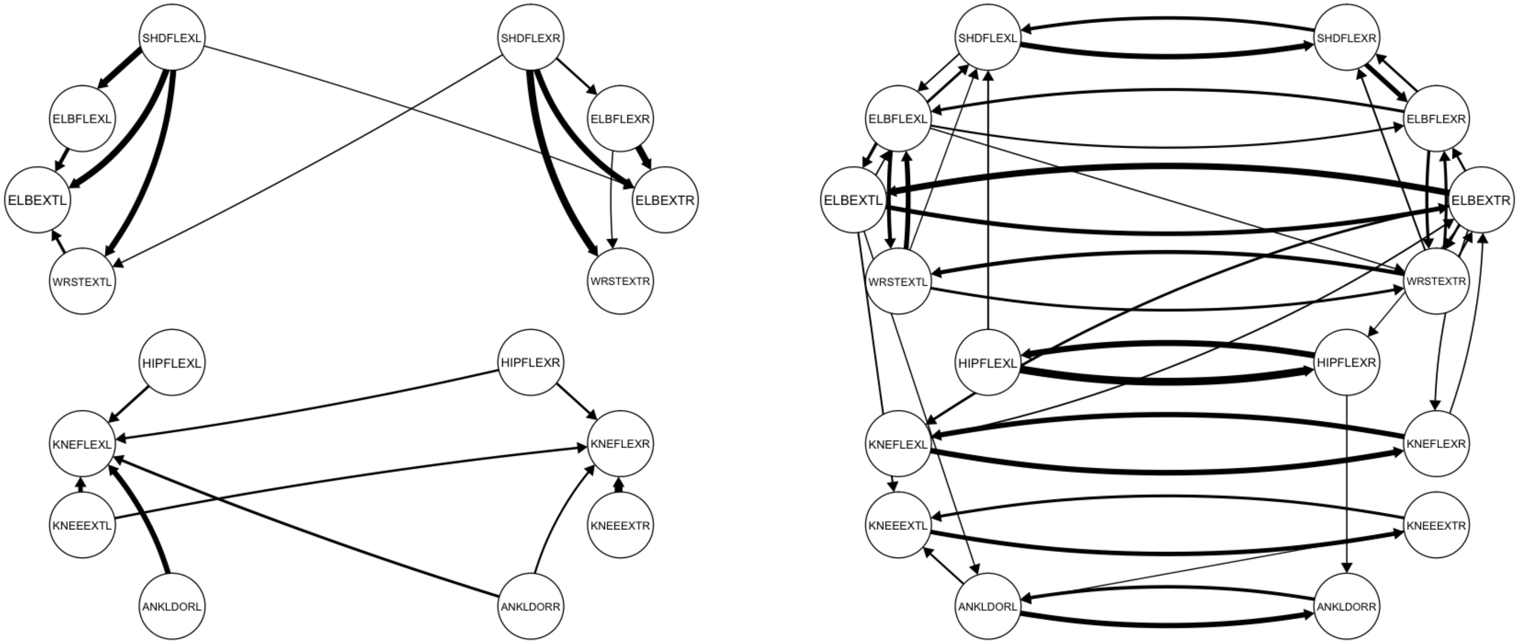
To initiate the iteration, we first make the modified data, which will be considered as if true binary data, by converting 0's followed by a state 1 in the observed data into 1's. Assuming the true states are known, we estimate the probabilities of misclassification, $e_0(1)$ and $e_1(0)$, and the coefficients of the autologistic model, $\boldsymbol{\beta}, \boldsymbol{\eta}_0$, and $\boldsymbol{\eta}_1$, where $\boldsymbol{X}_i$ consists of the visiting times (t), the symptom onset site (a binary variable whether it is on bulbar or others) and the symptom duration when patients entered the study. Then, a sequence of hidden states is updated through Viterbi algorithm for each muscle location of each subject.

We do not assume a specific neighborhood structure but allow any muscle can be a neighbor to another; in other words, it is a complete network $\mathcal{N}_j = \{s_k | s_k \neq s_j \text{ for } k = 1, \ldots, 16\}$. Note that we therefore have a $16 \times 15 = 240$ length of $\boldsymbol{\eta}_0$ and $\boldsymbol{\eta}_1$ each. The tuning parameter λ could be fixed to an optimized value in the first iteration of course, but we select a certain value leading the reasonable amount of sparsity in $\boldsymbol{\eta}$ coefficients to improve the interpretability. The iterations are broken when every estimate has only little change as much as less than 5% relative difference.

The misclassification probabilities are finally estimated as $\hat{e}_0(1) = 0.0447$ and $\hat{e}_1(0) = 0.0302$. From this, we see the false positive rate is slightly greater than the false negative rate with the dichotomization by 40% cut-off.

Let us also make interesting interpretations of autocovariate estimates obtained in stage II: transition probabilities. Figure 4.7 illustrates a vector of estimated $\boldsymbol{\eta}_0$ and $\boldsymbol{\eta}_1$ in a graphical way. According to $\boldsymbol{\eta}_0$ estimates which tell the impacts of healthy muscles, if shoulder muscles were healthy in the past, elbow and wrist muscles of the same side would keep healthy as well. Recall a positive η_0 indicates the negative impact on the probability of infection because $S = 0$ is transformed to the negative term $S^* = -0.5$ by centering in the model (4.9). Similarly, knee muscles tend to stay healthy if hip and ankle muscles were healthy previously while this type of impact is stronger in the upper body parts than the lower.

The estimates of $\boldsymbol{\eta}_1$ also provide interesting features of ALS disease spreading, in that they represent the impacts of infected muscles to the state of previously healthy muscles. Most impressively, horizontal impacts between right and left side are strong in every muscle; for example, if the left hip flexor was infected at $t - 1$, the right hip flexor would be



(a) $\boldsymbol{\eta}_0$

(b) $\boldsymbol{\eta}_1$

Figure 4.7: Estimates of spatial dependence among muscles; the edge width indicates the strength of conditional influence.

$$\begin{array}{c|ccccccccccccc}
t & 1 & & 2 & & 3 & & 4 & & 5 & & 6 & & 7 \\
\hline
S & \mathbf{0} & \to & \mathbf{0} & \to & \mathbf{0} & \to & \mathbf{1} & \to & \mathbf{1} & \to & \mathbf{1} & \to & \mathbf{1} \\
 & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
Y & \mathbf{1} & & \mathbf{0} & & \mathbf{0} & & \mathbf{1} & & \mathbf{1} & & \mathbf{0} & & \mathbf{1}
\end{array}$$

Figure 4.8: A sequence of observed and hidden states for a selected subject and muscle

.

the most likely infected at t. Also, the fact that the denser connection within the upper body parts compared to the lower implies the upper body muscles such as shoulder, elbow, and wrist muscles are more interdependently to be influenced by each other than the lower body muscles are. Moreover, the impacts from upper to lower, although relatively fainter, imply the disease spreads vertically through elbow and knee on the whole.

When comparing $\boldsymbol{\eta}_0$ with $\boldsymbol{\eta}_1$, it is also remarkable that $\boldsymbol{\eta}_0$ is nearly a complementary set of $\boldsymbol{\eta}_1$ and vice versa. In other words, if s_k has a strong influence on s_j with a high value of η_{0jk}, then η_{1jk} is less likely to be large; for example, the impact of shoulder muscles stands out when they were healthy while little impact is there when they were infected. This could also be concluded that the shoulder muscles particularly do not likely to spread the disease to neighbors but help others maintain the healthy state.

At last, we are able to check hidden binary states updated through the Viterbi algorithm. Figure 4.8 illustrates a time sequence of hidden and observed states for a selected subject and muscle. Understandably, the observed sequence is refined to reveal absorbing features in the hidden sequence; the observed state at $t = 1$, i.e. $Y_i(s_j, 1) = 1$, turns out to be false positive with $S_i(s_j, 1) = 0$, whereas $Y_i(s_j, 6) = 0$, followed by two 1's, turns out to be in fact $S_i(s_j, 6) = 1$.

5. SUMMARY AND DISCUSSION

Chapter 2: Covariate Matching Methods for Testing and Quantifying Wind Turbine Upgrades

We are interested in statistical inference about the upgrade effect on wind turbine performance. It is a challenging issue because the upgrade effect on wind power production could be biased and confounded by unmanageable environmental conditions. Some of these conditions are measured on a wind farm, while others are unknown or not measured. We propose a covariate matching method, allowing for a fair and direct comparison of power outcomes without establishing the power curve model.

Compared to the current studies on wind power analysis, our matching method entertains several advantages: (a) it does not compare the estimated power outputs from the fitted power curve models but compares the observed power outputs directly; (b) by using the control turbine power output as benchmark, our method takes into account both measured and unmeasured environmental conditions; (c) when future technology innovation allows additional environmental covariates to be measured, their inclusion in our matching method is straightforward and it does not complicate the subsequent analysis steps. By testing on both experimental data and simulated data, the proposed matching method appears to be sensitive to detecting small to moderate changes, resulting from upgrades on a wind turbine.

Chapter 3: Joint Estimation of Monotone Curves via Functional Principal Component Analysis

The proposed functional principal component analysis is to illustrate the modes of variation of monotone curves that are irregularly and sparely observed. The advantages of our model are: (1) transforming the problem of fitting monotone-constrained functions to the problem of fitting unconstrained functions; (2) estimating individual functions by borrowing strength of mean information of all functions; (3) being able to estimate incompletely observed functional data, such as irregular or sparse data; and (4) recovering the relative curvature of curves, which can describe the curve dynamics, such as inflection points.

The useful applications of our model would be (1) cumulative distribution function estimation, (2) survival function estimation, (3) growth curve estimation, as well as (4) any monotone increasing or decreasing curves that are accumulated over a continuum. For the sake of interpretable relative curvature, which is even directly estimated in our model, the dynamic of curvature can also be flexibly studied to any monotone curves. Moreover, our approach can be applied to functional tests or regressions with purposes of comparing the characteristics of curves or detecting factors causing different features in a collection of curves.

Since a quantile rather than a mean could be of interest in the same situation, it would be an appealing study to describe the mode of variation of quantiles of monotone curves for the future work.

Chapter 4: Statistical Modeling on Spatio-temporal Binary Data for Describing Infectious Disease Spreading Pattern

The autologistic network model with absorbing states is first proposed to describe disease spreading pattern from spatio-temporal binary data. It learns the network association among numerous locations from data rather than pre-specifying the neighborhood information. The absorbing feature of disease is fulfilled by defining the active set of responses which contribute to the likelihood. Also, we allow previously normal and infected locations having different impacts, so it can draw the sophisticated association among spaces, such as how the previous status affect the current dependencies. For the purpose of predicting the disease progression, we formulate the joint distribution of multiple locations given initial status.

The second model is developed with a similar goal; however, it has a different view on binary data that might have measurement errors so underlying binary processes are modeled instead. The autologistic model regressing the current responses to the previous ones is incorporated into the stage of estimating transition probability, which is set to reflect absorbing states of binary process. Unlike the first model, it can estimate the asymmetric but directional impacts. Moreover, the procedure of updating hidden states can benefit researchers to understand subjects' underlying conditions and false positive rate.

Although we apply the both models on a complete network structure of locations, they are in fact flexible to utilize any prior information about neighborhood, if provided, such as neural network association affecting disease infection. This could be done quickly by fitting on a subset of autocovariates who are in a pre-specified neighborhood.

Future research can focus on ordered categorical data or a mixture of continuous and discrete measure to retain more information, rather than dichotomized data. For that end, an absorbing feature could be achieved under the presumption of monotone decreasing responses.

REFERENCES

Thomas Ackermann and Lennart Söder. Wind power in power systems: an introduction. *Wind Power in Power Systems*, pages 25–51, 2005.

Ameya Agaskar and Yue M Lu. Alarm: A logistic auto-regressive model for binary processes on networks. In *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, pages 305–308. IEEE, 2013.

Axel Albers. Relative and integral wind turbine power performance evaluation. In *Proceedings of the 2012 European Wind Energy Conference & Exhibition*, London, UK, 2004. November 22-25.

Julian Besag. Statistical analysis of non-lattice data. *The statistician*, pages 179–195, 1975.

Philippe Besse and James O Ramsay. Principal components analysis of sampled functions. *Psychometrika*, 51(2):285–311, 1986.

Philippe C Besse, Hervé Cardot, and Frédéric Ferraty. Simultaneous non-parametric regressions of unbalanced longitudinal data. *Computational Statistics & Data Analysis*, 24(3):255–270, 1997.

Richard W Bohannon. Reference values for extremity muscle strength obtained by handheld dynamometry from adults aged 20 to 79 years. *Archives of physical medicine and rehabilitation*, 78(1):26–32, 1997.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Eunshin Byon, Lewis Ntaimo, Chanan Singh, and Yu Ding. Wind energy facility reliability and maintenance. In *Handbook of Wind Power Systems: Optimization, Modeling, Simulation and Economic Aspects*, pages 639 – 672. Springer-Verlag, Berlin, 2013.

Petruţa C Caragea and Mark S Kaiser. Autologistic models with interpretable parameters.

Journal of agricultural, biological, and environmental statistics, 14(3):281–300, 2009.

P E Castro, W H Lawton, and E A Sylvestre. Principal modes of variation for processes with continuous sample curves. *Technometrics*, 28(4):329–337, 1986.

Raymond B Cattell. The scree test for the number of factors. *Multivariate behavioral research*, 1(2):245–276, 1966.

Luca Delle Monache, F Anthony Eckel, Daran L Rife, Badrinath Nagarajan, and Keith Searight. Probabilistic weather prediction with an analog ensemble. *Monthly Weather Review*, 141(10):3498–3516, 2013.

DOE. Windexchange: US installed wind capacity 2015. Technical report, 2015. Available at http://apps2.eere.energy.gov/wind/windexchange/wind_installed_capacity.asp.

Paul H C Eilers and Brian D Marx. Flexible smoothing with b-splines and penalties. *Statistical science*, pages 89–102, 1996.

Jerome Friedman, Trevor J Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

Mengmeng Guo, Lan Zhou, Jianhua Z Huang, and Wolfgang Karl Härdle. Functional data analysis of generalized regression quantiles. *Statistics and Computing*, 25(2):189–202, 2015.

Peter Hall and Li-Shan Huang. Nonparametric kernel regression subject to monotonicity constraints. *Annals of Statistics*, pages 624–647, 2001.

Peter Hall and Hans-Georg Müller. Order-preserving nonparametric regression, with applications to conditional distribution and quantile function estimation. *Journal of the American Statistical Association*, 98(463):598–608, 2003.

John M Hammersley and Peter Clifford. Markov fields on finite graphs and lattices. 1971.

Trevor J Hastie and Daryl Pregibon. Statistical models in s, chapter generalized linear models. *Wadsworth & Brooks/Cole*, 51, 1992.

John Hughes, Murali Haran, and Petruţa C Caragea. Autologistic models for binary data on a lattice. *Environmetrics*, 22(7):857–871, 2011.

Stefano M Iacus, Gary King, and Giuseppe Porro. Causal inference without balance checking: coarsened exact matching. *Political Analysis*, 20(1):1–24, 2012.

IEC. 61400–12–1:2005 wind turbines part 12-1: Power performance measurements of electricity producing wind turbines. *International Electrotechnical Commission*, 2005.

Gareth M James, Trevor J Hastie, and Catherine A Sugar. Principal component models for sparse functional data. *Biometrika*, pages 587–602, 2000.

S Rao Jammalamadaka and Ashis Sengupta. *Topics in Circular Statistics*, volume 5. World Scientific, 2001.

Jooyoung Jeon and James W Taylor. Using conditional kernel density estimation for wind power density forecasting. *Journal of the American Statistical Association*, 107(497): 66–79, 2012.

M C Jones and John A Rice. Displaying the important features of large collections of similar curves. *The American Statistician*, 46(2):140–145, 1992.

Mark S Kaiser, Karl T Pazdernik, Amy B Lock, and Forrest W Nutter. Modeling the spread of plant disease using a sequence of binary random fields with absorbing states. *Spatial Statistics*, 9:38–50, 2014.

Colleen Kelly and John Rice. Monotone smoothing with application to dose-response curves and the assessment of synergism. *Biometrics*, pages 1071–1085, 1990.

Mohammed G Khalfallah and Aboelyazied M Koliub. Suggestions for improving wind turbines power curves. *Desalination*, 209:221–229, 2007.

Andrew Kusiak, Haiyang Zheng, and Zhe Song. Wind farm power prediction: a data-mining approach. *Wind Energy*, 12(3):275–293, 2009.

Giwhyun Lee, Yu Ding, Marc G Genton, and Le Xie. Power curve estimation with multivariate environmental factors for inland and offshore wind farms. *Journal of the Amer-*

ican Statistical Association, 110(509):56–67, 2015a.

Giwhyun Lee, Yu Ding, Le Xie, and Marc G Genton. A kernel plus method for quantifying wind turbine performance upgrades. *Wind Energy*, 18(7):1207–1219, 2015b.

Nicholas Longford. A fast scoring algorithm for maximum likelihood estimation in unbalanced mixed models with nested random effects. *ETS Research Report Series*, 1987 (1), 1987.

Prasanta C Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55, 1936.

Enno Mammen and Kyusang Yu. Additive isotone regression. *Lecture Notes-Monograph Series*, pages 179–195, 2007.

Torben S Nielsen, Henrik A Nielsen, and Henrik Madsen. Prediction of wind power using time-varying coefficient functions. In *Proceedings of the 15'th IFAC World Congress on Automatic Control*, Barcelona, Spain, 2002. July 21-26.

Database of the National Isometric Muscle Strength NIMS. Muscular weakness assessment: use of normal isometric strength data. *Archives of Physical Medicine and Rehabilitation*, 77(12):1251–1255, 1996.

Lisa A Osadciw, Yanjun Yan, Xiang Ye, Glen Benson, and Eric White. Wind turbine diagnostics based on power curve using particle swarm optimization. In Lingfeng Wang, Chanan Singh, and Andrew Kusiak, editors, *Wind Power Systems: Applications of Computational Intelligence*, pages 151–165. Springer-Verlag, Berlin, 2010.

Stig Øye. The effect of vortex generators on the performance of the ELKRAFT 1000 kw turbine. In *9th IEA Symposium on Aerodynamics of Wind Turbines, Stockholm, ISSN*, pages 0590–8809, 1995.

Pierre Pinson, Henrik A Nielsen, Henrik Madsen, and Torben S Nielsen. Local linear regression with adaptive orthogonal fitting for the wind power application. *Statistics and Computing*, 18(1):59–71, 2008.

Natalya Pya and Simon N Wood. Shape constrained additive models. *Statistics and Computing*, 25(3):543–559, 2015.

Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

James O Ramsay. Monotone regression splines in action. *Statistical science*, pages 425–441, 1988.

James O Ramsay. Estimating smooth monotone functions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(2):365–375, 1998.

James O Ramsay. *Functional data analysis*. Wiley Online Library, 2006.

C Radhakrishna Rao. Some statistical methods for comparison of growth curves. *Biometrics*, 14(1):1–17, 1958.

John A Rice and Colin O Wu. Nonparametric mixed effects models for unequally sampled noisy curves. *Biometrics*, 57(1):253–259, 2001.

Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

Paul R Rosenbaum and Donald B Rubin. Constructing a control group using multivariate matched sampling methods that incorporate the propensity score. *The American Statistician*, 39(1):33–38, 1985.

Donald B Rubin. Matching to remove bias in observational studies. *Biometrics*, 29(1): 159–183, 1973.

Donald B Rubin. Using propensity scores to help design observational studies: application to the tobacco litigation. *Health Services and Outcomes Research Methodology*, 2:169–188, 2001.

Ismael Sanchez. Short-term prediction of wind energy production. *International Journal of Forecasting*, 22(1):43–56, 2006.

Joan G Staniswalis and J Jack Lee. Nonparametric regression analysis of longitudinal data. *Journal of the American Statistical Association*, 93(444):1403–1418, 1998.

Elizabeth A Stuart. Matching methods for causal inference: a review and a look forward. *Statistical Science*, 25(1):1, 2010.

Lu Tang, Ling Zhou, and Peter X-K Song. Method of divide-and-combine in regularised generalised linear models for big data. *arXiv preprint arXiv:1611.06208*, 2016.

Onder Uluyol, Girija Parthasarathy, Wendy Foslien, and Kyusung Kim. Power curve analytic for wind turbine performance monitoring and prognostics. In *Proceedings of Annual Conference of the Prognostics and Health Management Society*, volume 2, pages 049:1–8, Montreal, Canada, 2011.

Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269, 1967.

Yih-Huei Wan, Erik Ela, and Kirsten Orwig. Development of an equivalent wind plant power curve. Technical Report NREL/CP-550-48146, 2010. Available at http://www.nrel.gov/docs/fy10osti/48146.pdf.

Lin Wang, Xinzi Tang, and Xiongwei Liu. Blade design optimisation for fixed-pitch fixed-speed wind turbines. *ISRN Renewable Energy*, Article ID 682859, 2012.

Yanjun Yan, Lisa A Osadciw, Glen Benson, and Eric White. Inverse data transformation for change detection in wind turbine diagnostics. In *Proceedings of the 22nd IEEE Canadian Conference on Electrical and Computer Engineering*, pages 944–949, St. John's, Newfoundland, Canada, 2009.

Fang Yao, Hans-Georg Müller, and Jane-Ling Wang. Functional data analysis for sparse longitudinal data. *Journal of the American Statistical Association*, 100(470):577–590, 2005.

Zhong Zhao. Using matching to estimate treatment effects: data requirements, matching metrics, and monte carlo evidence. *Review of Economics and Statistics*, 86(1):91–107, 2004.

Lan Zhou, Jianhua Z Huang, and Raymond J Carroll. Joint modelling of paired sparse functional data using principal components. *Biometrika*, 95(3):601, 2008.

Jun Zhu, Hsin-Cheng Huang, and Jungpin Wu. Modeling spatial-temporal binary data using markov random fields. *Journal of Agricultural, Biological, and Environmental Statistics*, 10(2):212–225, 2005.

APPENDIX A

REMARKS ON ORTHONORMAL BASIS

A.1 Creation of Orthonormal Basis Function

We follow the computation of creating orthonormal basis function from Zhou et al. (2008). They provided details about the transformation from arbitrary basis functions to orthonormal. Here we explain briefly their techniques.

Let $\tilde{\boldsymbol{b}}(\boldsymbol{t}) = \{\tilde{b}_1(t), \ldots, \tilde{b}_q(t)\}^T$ be an initially chosen general B-spline basis; this is not necessarily orthonormal at this point. A transformation matrix $\boldsymbol{T}$ such that $\boldsymbol{b}(\boldsymbol{t}) = \boldsymbol{T}\tilde{\boldsymbol{b}}(\boldsymbol{t})$ can be constructed as follows. Write $\tilde{\boldsymbol{b}} = \{\tilde{\boldsymbol{b}}(t_1), \ldots, \tilde{\boldsymbol{b}}(t_g)\}^T$ for the equally-spaced and sufficiently dense grid, $(t_1, \ldots, t_g)$. Let $\tilde{\boldsymbol{b}} = \boldsymbol{Q}\boldsymbol{R}$ be the QR decomposition of $\tilde{\boldsymbol{b}}$, where $\boldsymbol{Q}$ has orthonormal columns and $\boldsymbol{R}$ is an upper triangular matrix. Then, $\boldsymbol{T} = (g/L)^{1/2}\boldsymbol{R}^{-T}$ will be a desirable transformation matrix since

$$\frac{L}{g}\boldsymbol{b}^T\boldsymbol{b} = \frac{L}{g}T\tilde{\boldsymbol{b}}^T\tilde{\boldsymbol{b}}T^T = \frac{L}{g}\boldsymbol{T}\boldsymbol{R}^T\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{R}\boldsymbol{T}^T = \boldsymbol{I},$$

where L is a length of range of t.

Figure A.1 illustrates an example of $q = 10$ basis functions whose order is 5. See how the orthonormal basis looks like, which is transformed from the ordinal B-spline.

A.2 Computational Singularity coming from Orthonormal Basis

The partial differentials in normal equations (3.12a, 3.12b, 3.12c) form the majority part of an information matrix in Fisher scoring algorithm. We here address the importance of penalizing not just for smoothing but also for the computational stability.

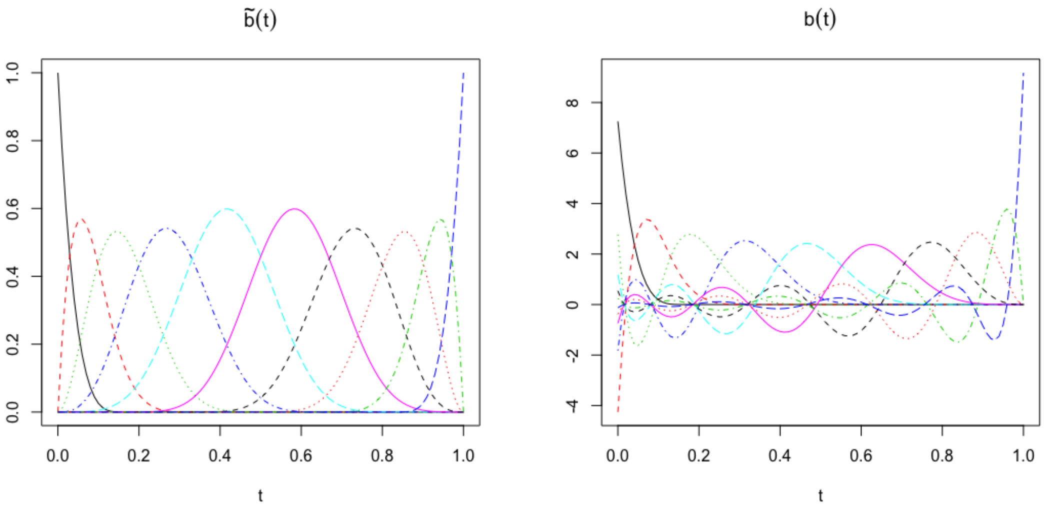


Figure A.1: Illustration of B-spline functions where $q = 10$ with an order 5; (Left) B-splines; (Right) Orthonormalized B-splines

Consider the partial differential of $h_i(t)$ with respect to $\boldsymbol{\theta}_\mu$, that is

$$\begin{aligned}\frac{\partial H_i(t)}{\partial \boldsymbol{\theta_\mu}} &= \int \boldsymbol{B(t)} \exp\{\boldsymbol{B(t)}^T\boldsymbol{\theta_\mu} + \boldsymbol{B(t)}^T\boldsymbol{\theta_f}\boldsymbol{\alpha_i}\}\,\mathrm{d}t; \\ &= \int \boldsymbol{B(t)} \exp\{W(t)\}\,\mathrm{d}t, \qquad\qquad \text{(A.1)}\end{aligned}$$

where $\boldsymbol{B(t)} = \int \boldsymbol{b(t)}\,\mathrm{d}t$ is a q-vector of integrated basis functions, and $W(t) = \int w(t)\,\mathrm{d}t = \boldsymbol{B(t)}^T\boldsymbol{\theta_\mu}+\boldsymbol{B(t)}^T\boldsymbol{\theta_f}\boldsymbol{\alpha_i}$ is an integrated function of relative curvature w. Then, $\partial h_i(t)/\partial\boldsymbol{\theta_\mu}$ is a vector of functions coming from integral of exponential (> 0), multiplied by integrated basis function, $\boldsymbol{B(t)}$. Since the $\boldsymbol{B(t)}$ looks like the left panel of Figure A.2, the function elements in (A.1) have therefore different shapes to each other; some are increasing fast and some are always near zero. For this reason, if these functions are evaluated at observed

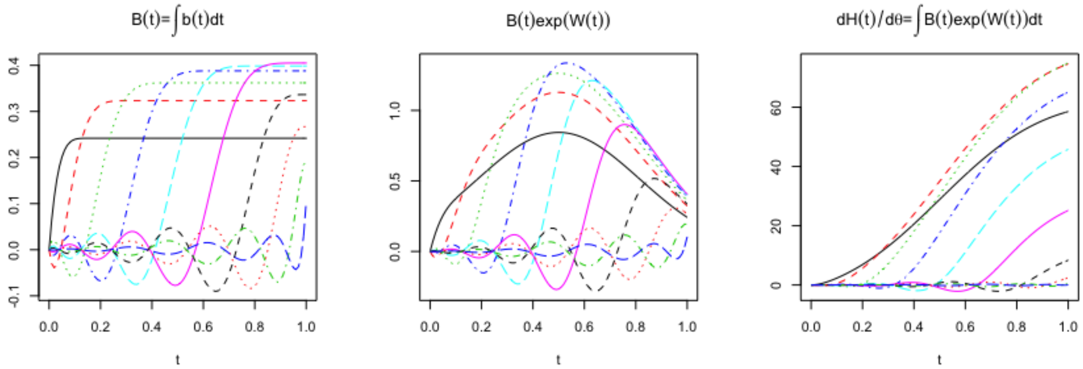


Figure A.2: Illustration of (left) the integrated basis function $\boldsymbol{B(t)}$, (center) intermediate function in calculation, and (right) the partial differential of $H(t)$ with regard to $w(t)$ of Figure A.3

points to get the form of data model, a cross-product matrix

$$\frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\theta}_\mu} \frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\theta}_\mu}^T \tag{A.2}$$

will computationally singular because of possibility of relatively near zero values at the diagonal.

To help understandings, we illustrate the form of functions derived at each computation step. Suppose there is a curve of $w(t)$, which is a straight line, as illustrated in Figure A.3. Accordingly, the form of partial differential of $h(t)$ can be drawn by multiplying $\boldsymbol{B(t)}$ (the left panel of Figure A.2) and $W(t)$ (the center panel of Figure A.3); see the center and right panel of Figure A.2. As aforementioned, functions of $\partial h_i(t)/\partial \boldsymbol{\theta_\mu}$ have different forms, therefore a cross product matrix of evaluated values (A.2) has values, for this example, as shown in Table A.1.

Making a long story short, the penalty parameters λ_μ and λ_f play a role not only as a tuner of smoothing amount but also as a controller for computational stability in terms of a ridge correction. Hence, if there is an issue with non-existence of inverse matrix due to

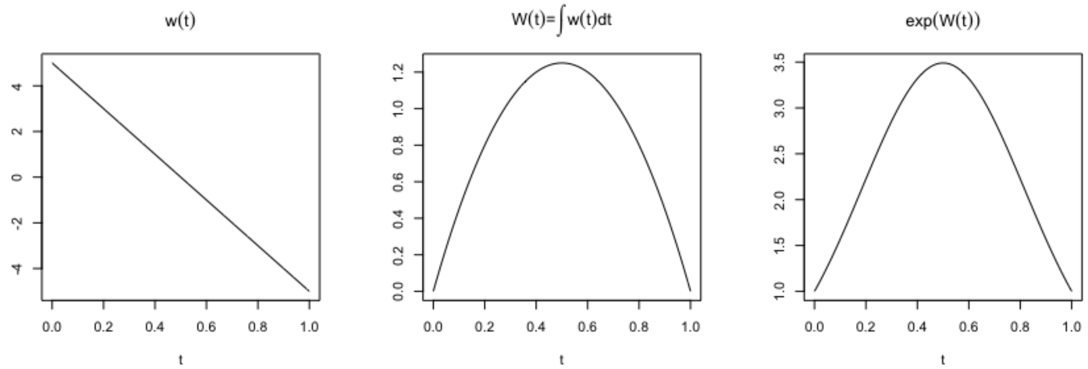


Figure A.3: Example curve of $w(t)$, its integrated function, and exponential of integrated one

Table A.1: Example of a cross product matrix (A.2) according to $\boldsymbol{b}$ of Figure A.1 and w of Figure A.3; See large values at upper-left block and distinctively gets smaller toward lower-right block; Relatively too small values at the diagonal may cause the singularity; Regularization makes this problem overcome in a sense of the ridge correction.

	bspl5.1	bspl5.2	bspl5.3	bspl5.4	bspl5.5	bspl5.6	bspl5.7	bspl5.8	bspl5.9	bspl5.10
bspl5.1	123813.68	154995.62	148755.14	117527.56	68236.01	26109.53	3875.89	522.57	-54.09	36.31
bspl5.2	154995.62	194311.94	187005.93	148283.47	86438.78	33183.68	4943.97	663.89	-67.48	45.55
bspl5.3	148755.14	187005.93	181096.78	144939.39	85369.10	33047.57	4969.21	660.28	-63.89	43.82
bspl5.4	117527.56	148283.47	144939.39	118140.88	71347.62	28169.83	4324.97	561.59	-48.24	34.47
bspl5.5	68236.01	86438.78	85369.10	71347.62	45340.78	18968.78	3025.46	397.59	-35.91	25.25
bspl5.6	26109.53	33183.68	33047.57	28169.83	18968.78	9039.44	1698.93	187.24	-2.23	5.50
bspl5.7	3875.89	4943.97	4969.21	4324.97	3025.46	1698.93	558.72	47.87	7.38	-2.06
bspl5.8	522.57	663.89	660.28	561.59	397.59	187.24	47.87	39.03	-9.00	4.30
bspl5.9	-54.09	-67.48	-63.89	-48.24	-35.91	-2.23	7.38	-9.00	6.41	-2.85
bspl5.10	36.31	45.55	43.82	34.47	25.25	5.50	-2.06	4.30	-2.85	1.37

computationally singularity, setting λ's at suitable values will be a key technique to make algorithm converge.