BIVARIATE B-SPLINES AND ITS APPLICATIONS IN SPATIAL DATA ANALYSIS

A Dissertation

by

HUIJUN PAN

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

DOCTOR OF PHILOSOPHY

August 2011

Major Subject: Statistics

BIVARIATE B-SPLINES AND ITS APPLICATIONS IN SPATIAL DATA ANALYSIS

A Dissertation

by

HUIJUN PAN

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

DOCTOR OF PHILOSOPHY

Approved by:

Co-Chairs of Committee,	Jianhua Huang
	Lan Zhou
Committee Members,	Daren B.H. Cline
	Ximing Wu
Head of Department,	Simon J. Sheather

August 2011

Major Subject: Statistics

ABSTRACT

Bivariate B-splines and its Applications in Spatial Data Analysis. (August 2011)

Huijun Pan, B.S., University of Science and Technology of China;

M.S., Texas A&M University

Co-Chairs of Advisory Committee, Dr. Jianhua Huang
Dr. Lan Zhou

In the field of spatial statistics, it is often desirable to generate a smooth surface for a region over which only noisy observations of the surface are available at some locations, or even across time. Kriging and kernel estimations are two of the most popular methods. However, these two methods become problematic when the domain is not regular, such as when it is rectangular or convex. Bivariate B-splines developed by mathematicians provide a useful nonparametric tool in bivariate surface modeling. They inherit several appealing properties of univariate B-splines and are applicable in various modeling problems. More importantly, bivariate B-splines have advantages over kriging and kernel estimation when dealing with complicated domains. The purpose of this dissertation is to develop a nonparametric surface fitting method by using bivariate B-splines that can handle complex spatial domains.

The dissertation consists of four parts. The first part of this dissertation explains the challenges of smoothing over complicated domains and reviews existing methods. The second part introduces bivariate B-splines and explains its properties and implementation techniques. The third and fourth parts discuss application of the bivariate B-splines in two nonparametric spatial surface fitting problems. In particular, the third part develops a penalized B-splines method to reconstruct a smooth surface

from noisy observations. A numerical algorithm is derived, implemented, and applied to simulated and real data. The fourth part develops a reduced rank mixed-effects model for functional principal components analysis of sparsely observed spatial data. A numerical algorithm is used to implement the method and tested on simulated and real data.

DEDICATION

My parents for their unconditional love and support

ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude to my co-advisors Dr. Jianhua Huang and Dr. Lan Zhou. It was fortunate and a great honor to be a student of these two prominent researchers and teachers. Dr. Huang and Dr. Zhou's enthusiasm towards research was contagious and motivational for me, and their generous contributions of time and ideas made this work possible. With their patience and tolerance, Dr. Huang and Dr. Zhou not only instructed me on my research, but also taught me to improve in other aspects such as how to communicate with others, how to be persistent on one's goals and dreams and so on. I am thankful for the excellent examples they have provided as successful researchers, teachers and mentors.

I would like to thank Dr. Daren B.H. Cline and Dr. Ximing Wu for serving as my committee members and for the invaluable advice on my work. Also thanks to Dr. Yu Ding for his willingness to be a substitute at my dissertation defense. I want to emphasize my thanks to Dr. Cline who advised me on my master's project and Ph.D dissertation and taught me two courses. I really learned a lot from him.

I want to thank my family and friends for all their love and support. In particular, I want to thank my parents, who raised me, supported me and taught me with an attitude of working hard; I want to thank my classmate, roommate and best friend Xuan Wang who teaches me to be an honest and independent person; and I want to thank Jia You who provided me with the Texas temperature data to make my dissertation complete.

Finally, I want to thank all my friends at Texas A&M who made these five years of my stay enjoyable.

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

Suppose a target variable, for example temperature or ozone concentration, has values over a two-dimensional domain. It is a common problem in spatial data analysis to derive the value of the the target variable at any location in the domain based on discrete observations disturbed by noises.

This problem is reasonable and feasible to solve because of the existence of spatial variation and spatial dependence. Spatial variation means that the target variable has different values at different locations; while spatial dependence means that the variable value at one location is correlated with that at some other locations.

There are different ways to describe these two properties. One is a stochastic view which treats the value of a target variable at each location as a random variable and use the covariance function between these random variables or a variogram to represent the correlation; another is to use a deterministic surface function to describe the variations and connections among values at different locations. This work takes the latter point of view. To be more specific, we are interested in estimating a smooth function $f(x, y)$ over some bounded domain $\Omega \subseteq R^2$ given observations $\{z_i\}_{i=1}^n$ at a collection of discrete points $\{v_i = (x_i, y_i)\}_{i=1}^n$ in the domain.

Two issues make this problem challenging. One is the data structure and the other is the domain shape. Data structure means how the data is generated. Some datasets are dense in locations while others are sparse. Some are measured at one

---

The style of this dissertation follows Journal of Statistical Planning & Inference.

moment while others are measured across time and therefore have time-specific variations. Domain shape indicates, for example, whether the domain has holes inside or it is just simply connected or it is a polygon. These characteristics affect the efficiency of estimation methods and can even paralyze some methods.

Some literatures analyze spatial data with complicated structures on regular domains but the methods may fail when applied to complicated domains. Others propose techniques to handle complicated domains but for simple data structures and models, such as least square regression model on dense observations with identical independent white noise, and these techniques have issues in interpretation or may not be adopted to complicate models accounting for complicated data structures.

To conquer the challenges, we introduce a collection of bivariate splines to handle complicated domain and propose several models for different data structures by using these splines. This collection of bivariate splines have several advantages. First, it spans a reasonably rich function space to approximate the true functions over complicated domains. Second, it can be easily incorporated into many widely used functional data models tailored for different data structure. Third, the computation costs for splines evaluation and model calibration and model selection are acceptable. The splines we introduce are called bivariate B-splines. They are well-developed by mathematicians but seldom used in statistical community. Only two papers have used it so far (Guillas and Lai, 2010; Lai and Wang, 2010).

We apply this collection of splines in two different scenarios. For each scenario, we give statistical model based on the bivariate B-splines, along with discussions on the estimation procedure, model selection and other statistical issues. The first scenario is when the observations are dense in location and are a snapshot of the surface. In other words, the observations have independently and identically distributed noises. For example, we estimate the temperature surface over a county for a specific day from

the observations for that day at all stations in the county. A penalized smoothing model is used for this case. Second, we deal with the case that the data is sparse in location but we have several snapshots of the surface. Following the temperature example, now we have records of the temperature for several days but with limited data on each day. We assume that there is no evolution or correlation over days. But the deviation of the observations from true values is day-specific. Therefore the measurement errors are no longer identically distributed over days. We cannot get a stable estimation of the temperature surface for each day in this case, but we can estimate the mean surface of all days by excluding the day-specific variation. A reduced rank mixed-effects model is proposed.

The structure of this work is as follows. Chapter II introduces bivariate B-splines and shows its mathematical properties. Chapter III and Chapter IV cover the two scenarios discussed above respectively.

CHAPTER II

BIVARIATE B-SPLINES

2.1 Introduction

The design of splines plays an important role in spline-based nonparametric statistical methodology. For smoothing over complicated spatial domains, we expect that the splines have at least the following four properties. First, the splines are bivariate functions and well defined on complicated domains in the sense that their application will not introduce extra spatial correlations over the holes in the domains. Second, the splines span a considerably rich function space that contains or is close to the target true function. Third, the computational cost of the analysis procedure when applying the splines is affordable. Fourth, the splines can be used for different modeling tasks. Bivariate B-splines, piecewise polynomials defined on triangle division of the domain, satisfy these four criteria. Some properties of bivariate B-splines are similar to those of univariate B-splines and the similarity is where the name comes from. This chapter introduces bivariate B-splines and shows their mathematical properties relevant to our applications in Chapters III and IV. Bivariate B-splines are well developed by mathematicians and Lai and Schumaker (2007) provides a comprehensive reference.

2.2 Barycentric coordinates

Given a non-degenerate a triangle A with vertices counter-clock numbered as $< v_1, v_2, v_3 >$, any point $v \in R^2$ can be written as

$$v = b_1 v_1 + b_2 v_2 + b_3 v_3, \text{ with } b_1 + b_2 + b_3 = 1. \tag{2.1}$$

The coefficients (b_1, b_2, b_3) are called the barycentric coordinates of point v with respect to the triangle A, denoted as $b_v = (b_1, b_2, b_3)$. Here the constraint $b_1+b_2+b_3 = 1$ guarantees the unique representation of point v. Although all points in R^2 could be represented as (2.1), we only consider the points inside the triangle A in this work.

So far we have two representation systems for points in R^2: cartesian coordinates and barycentric coordinates. There is one-to-one mapping between these two systems. Assume the cartesian coordinates of v_1, v_2 and v_3 are $v_1 = (x_1, y_1), v_2 = (x_2, y_2), v_3 = (x_3, y_3)$, and $v = (x, y)$. Then

$$x = b_1x_1 + b_2x_2 + b_3x_3, \qquad y = b_1y_1 + b_2y_2 + b_3y_3,$$

which converts the barycentric coordinates to the cartesian coordinates. Reversely, by solving the equation system

$$\begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 1 \\ \mathrm{x} \\ \mathrm{y} \end{bmatrix}, \tag{2.2}$$

we get

$$\begin{aligned} b_1 &= \frac{(x_2y_3 - y_2x_3) - x(y_3 - y_2) + y(x_3 - x_2)}{2 \cdot Area(T)}, \\ b_2 &= \frac{(x_3y_1 - y_3x_1) - x(y_1 - y_3) + y(x_1 - x_3)}{2 \cdot Area(T)}, \\ b_3 &= \frac{(x_1y_2 - y_1x_2) - x(y_2 - y_1) + y(x_2 - x_1)}{2 \cdot Area(T)}. \end{aligned} \tag{2.3}$$

Apparently, (b_1, b_2, b_3) are linear in (x, y).

The constraint $b_1 + b_2 + b_3 = 1$ on barycentric coordinates makes them identifiable. There exist other constraints that make the coordinates unique. We adopt this one because it gives barycentric coordinates an interesting geometric interpretation. When point v is located inside or on the edges of the triangle T, we connect the point

v with v_1, v_2 and v_3 to generate three triangles $\{A_1, A_2, A_3\}$ as Figure 1, then the barycentric coordinates are

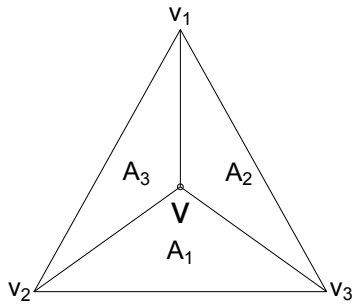

Figure 1: Triangle with one point inside.

$$b_i = \frac{\text{Area of } A_i}{\text{Area of } A}, \qquad i = 1, 2, 3. \tag{2.4}$$

In spatial analysis, we could choose any two orthogonal directions and any point as original point to build up the cartesian coordinates. Therefore, all statistical analysis tools are desirable to be invariance to linear transformation of cartesian coordinates. Now we prove that barycentric coordinates have this property.

Theorem 2.2.1. Barycentric coordinates are invariant to linear transformation of cartesian coordinates.

Proof. Assume that point $v = (x, y) \in R^2$ has barycentric coordinates $b_v = (b_1, b_2, b_3)$ with respect to a triangle $A =< v_1, v_2, v_3 >$ with $v_i = (x_i, y_i)$. Then, (b_1, b_2, b_3) satisfies (2.2), which in turn can be written as

$$b_1 + b_2 + b_3 = 1 \tag{2.5}$$

and

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \end{bmatrix}. \tag{2.6}$$

Applying the linear transformation

$$\begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} + \begin{bmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.7}$$

to all points (x, y) and (x_i, y_i) appeared on two sides of (2.6), we obtain

$$\begin{bmatrix} x_1^* & x_2^* & x_3^* \\ y_1^* & y_2^* & y_3^* \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} x^* \\ y^* \end{bmatrix}. \tag{2.8}$$

Combining (2.5) and (2.8), we know that (b_1, b_2, b_3) is barycentric coordinates of (x^*, y^*) with respect to the transformed triangle $A^* =< v_1^*, v_2^*, v_3^* >$, where $v_i^* = (x_i^*, y_i^*), i = 1, \ldots, 3$. □

Theorem 2.2.1 shows that the barycentric coordinates are invariant to linear transformations. In following sections, bivariate B-splines and all model equations are built on barycentric coordinates and therefore they are unchanged with respect to linear transformations.

2.3 Bivariate B-splines

After introducing barycentric coordinates, we define bivariate B-splines in this section. Given a triangle A and a point $v \in A$ with coordinates $b_v = (b_1, b_2, b_3)$. We define functions for pre-fixed non-negative integer d

$$B_{d,ijk}(v) := \frac{d!}{i!j!k!} b_1^i b_2^j b_3^k, \qquad i + j + k = d,\ b_1 + b_2 + b_3 = 1,\ v \in A. \tag{2.9}$$

Here i, j, k are nonegative integers. We call the set of polynomials

$$B_d := \{B_{d,ijk}\}_{i+j+k=d} \tag{2.10}$$

bivariate B-splines with respect to triangle A. Figure 2 gives an example of B_d for $d = 2$.

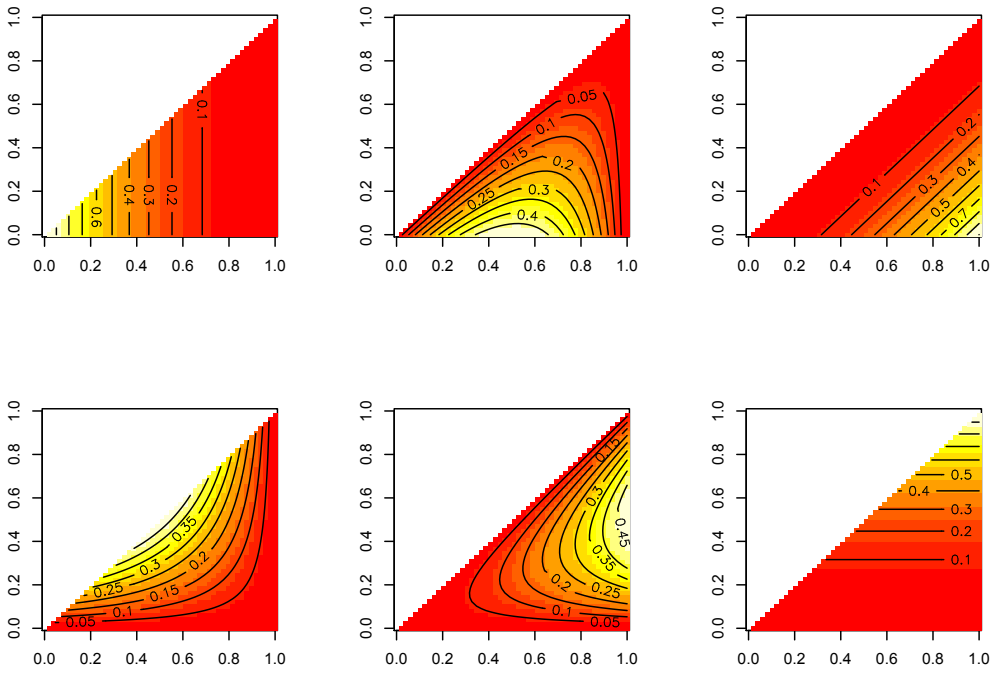

Figure 2: Complete collection of bivariate B-splines when domain is a triangle and $d = 2$. When $d = 2$, B_d contains six functions shown above. The functions values vary between 0 and 1.

We have known that b_1, b_2, b_3 are all linear polynomials in x and y. It follows that $B_{d,ijk}(v)$ is a polynomial in x and y with degree d. Let $P_d(A)$ be the space of polynomials defined on the triangle A with degree d, then $B_{d,ijk} \in P_d(A)$. In fact, B_d does not only belong to the space $P_d(A)$ but can also form its basis as stated in the following theorem.

Theorem 2.3.1. The set of bivariate B-splines B_d forms a basis for space $P_d(A)$ and

1) $\sum_{i+j+k=d} B_{d,ijk}(v) = 1, \text{ for all } v \in A.$

2) $0 \leq B_{d,ijk}(v) \leq 1, \text{ for all } v \in A.$

3) $B_{d,ijk}$ has a unique maximum at the point $\xi_{ijk} = (iv_1 + jv_2 + kv_3)/d \in A.$

Proof. 1) and 2) are straightforward according to $\sum_{i+j+k=d} \frac{d!}{i!j!k!} b_1^i b_2^j b_3^k = 1.$

Now we prove 3) and the claim that B_d is a basis for P_d. Rewrite bivariate B-splines in (2.9) by absorbing the constrain $b_1 + b_2 + b_3 = 1$, we have

$$B_{d,ijk}(v) = \frac{d!}{i!j!k!} b_1^i b_2^j (1 - b_1 - b_2)^k.$$

Then take derivative with respect to b_1 and b_2 and set these derivatives to zero, we have

$$i - (i + k)b_1 - ib_2 = 0 \tag{2.11}$$

and

$$j - (j + k)b_2 - jb_1 = 0. \tag{2.12}$$

Solving these equations we get $b_1 = i/d$, $b_2 = j/d$, $b_3 = k/d$, so the point $\xi_{ijk} = (iv_1 + jv_2 + kv_3)/d$ is an extreme value point. By checking the sign of the second derivative, we know that $B_{d,ijk}$ reaches maximum at ξ_{ijk} □

Theorem 2.3.1 tells us that B_d is a basis for P_d. It follows that for any function $s \in P_d(A)$, there exist coefficients $\{c_{ijk}\}$ such that

$$s(v) = \sum_{i+j+k=d} c_{ijk} B_{d,ijk}(v), \tag{2.13}$$

or in vector form

$$s(v) = B_d^T(v)\mathbf{c}, \tag{2.14}$$

where

$$B_d(v) = \{B_{d,d,0,0}(v), B_{d,d-1,1,0}(v), B_{d,d-1,0,1}(v), ..., B_{d,0,0,d}(v)\}^T \tag{2.15}$$

and

$$\mathbf{c} = \{\mathrm{c_{d,0,0}}, \mathrm{c_{d-1,1,0}}, \mathrm{c_{d-1,0,1}}, \mathrm{c_{d-2,2,0}}, ..., \mathrm{c_{0,0,d}}\}^{\mathrm{T}}. \tag{2.16}$$

In (2.16), c_{ijk} is corresponding to the lth element in vector $\mathbf{c}$ where

$$l = \sum_{m=0}^{d-i} m + 1 - j. \tag{2.17}$$

Note that the ordering of the elements of c is not important. Our choice is just one possibility; using a different choice won't influence our method.

2.4 Directional derivatives

This section gives the expressions of the directional derivatives of bivariate B-splines. Directional derivative of a multivariate smooth function f at point v with respect to direction w is generally defined as

$$D_w f(v) := \frac{\partial}{\partial t} f(v + tw)|_{t=0} = \lim_{t \to 0} \frac{f(v + tw) - f(v)}{t}. \tag{2.18}$$

Assume that the direction w has barycentric coordinates (w_1, w_2, w_3). Define

$$b_v = (b_1, b_2, b_3) \tag{2.19}$$

and

$$b_{v+tw} = (b_1 + tw_1, b_2 + tw_2, b_3 + tw_3). \tag{2.20}$$

Plugging (2.19) and (2.20) back into (2.18), we immediately get

$$\begin{aligned} &D_w B_{d,ijk}(v) \\ &\quad = d[w_1 B_{d-1,(i-1)jk}(v) + w_2 B_{d-1,i(j-1)k}(v) + w_3 B_{d-1,ij(k-1)}(v)]. \end{aligned} \tag{2.21}$$

We can calculate higher order derivatives $D_w^n B_{d,ijk}(v)$ by iteratively applying (2.21).

After knowing the directional derivatives of basis functions, we can easily derive the directional derivatives of a function $s \in P_d(A)$ as follow.

Theorem 2.4.1. Assume we have a bivariate function $s(v)$ with the basis expansion

$$s(v) = \sum_{i+j+k=d} c_{ijk} B_{d,ijk}(v).$$

For all (i, j, k) that $i + j + k = d$, we define

$$c_{ijk}^{(0)}(w) = c_{ijk} \tag{2.22}$$

and

$$c_{ijk}^{(m)}(w) := w_1 c_{i+1,j,k}^{(m-1)} + w_2 c_{i,j+1,k}^{(m-1)} + w_3 c_{i,j,k+1}^{(m-1)}, \qquad \text{for } m = 1, ..., d. \tag{2.23}$$

Then,

$$D_w^m s(v) = \frac{d!}{(d-m)!} \sum_{i+j+k=d-m} c_{ijk}^{(m)}(w) B_{d-m,ijk}(v). \tag{2.24}$$

Proof. When $m = 1$, we need to show that

$$D_w s(v) = d \sum_{i+j+k=d-1} c_{ijk}^{(1)}(w) B_{d-1,ijk}(v). \tag{2.25}$$

Once we prove the result for $m = 1$, the result for all other m's can be proved easily by repeatedly applying (2.25) .

According to definition (2.22)

$$c_{ijk}^{(1)} = w_1 c_{i+1,j,k} + w_2 c_{i,j+1,k} + w_3 c_{i,j,k+1}. \tag{2.26}$$

Applying (2.21) and (2.26), we have

$$\begin{aligned} D_w s(v) &= \sum_{i+j+k=d} c_{ijk} D_w B_{d,ijk}(v) \\ &= \sum_{i+j+k=d} c_{ijk} d[w_1 B_{d-1,(i-1)jk}(v) + w_2 B_{d-1,i(j-1)k}(v) \\ &\quad + w_3 B_{d-1,ij(k-1)}(v)] \\ &= d \sum_{i+j+k=d} c^{(1)}_{ijk}(w) B_{d-1,ijk}(v). \end{aligned}$$

□

This is (2.25). Thus the proof is complete.

2.5 Smoothness conditions

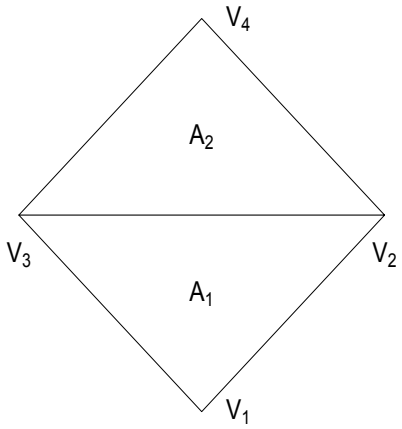

Figure 3: First example of two triangles sharing an edge.

Now assume that we have two triangles $A_1 :=< v_1, v_2, v_3 >$ and $A_2 :=< v_4, v_3, v_2 >$ sharing a common edge $e =< v_2, v_3 >$ with the bivariate B-splines $\{B^{(1)}_{d,ijk}\}_{i+j+k=d}$ defined on A_1 and $\{B^{(2)}_{d,ijk}\}_{i+j+k=d}$ on A_2. See Figure 3. If we have a polynomial $p_1(v)$

defined on A_1 and a polynomial $p_2(\tilde{v})$ on A_2, then according to Theorem 2.3.1, we have coefficients $\{c^{(1)}_{ijk}\}_{i+j+k=d}$ and $\{c^{(2)}_{ijk}\}_{i+j+k=d}$ such that $p_1(v) = \sum_{i+j+k=d} c^{(1)}_{ijk} B^{(1)}_{d,ijk}(v)$ and $p_2(\tilde{v}) = \sum_{i+j+k=d} c^{(2)}_{ijk} B^{(2)}_{d,ijk}(\tilde{v})$.

In some situations, we want these two polynomials to connect smoothly over the common edge e. We call this issue as smoothness constraint. Latter sections will explain why we are interested in such issue and give more detailed description of the scenarios under which it rises. Here we skip its application but just find the conditions under which the smoothness constrains are satisfied.

We say that the two bivariate functions defined on two adjacent triangles connect smoothly on the common edge with order r if they have same rth and less order derivatives along any direction on the common edge. We have following theorem to give the conditions under which two bivariate functions are smoothly connected.

Theorem 2.5.1. Suppose there are two triangles A_1 and A_2 sharing edge e. w is any direction unparallel to common edge e and $D^n_w p(v)$ is nth order derivative in direction w at point v. Then

$$D^l_w p_1(v) = D^l_w p_2(v), \qquad \text{all } v \in e \text{ and } l = 0, .., r \tag{2.27}$$

if and only if

$$c^{(1)}_{ljk} = \sum_{\nu+\mu+\kappa=l} c^{(2)}_{\nu,k+\mu,j+\kappa} B^{(2)}_{l,\nu\mu\kappa}(v_4), \qquad j+k = d-l,\ l = 0, ..., r. \tag{2.28}$$

Proof. In the following proof, point v is always on common edge e. The barycentric coordinates (b_1, b_2, b_3) of $v \in e$ with respect to triangle A_1 could be written as $(0, b_2, 1-b_2)$. Then the barycentric coordinates of v with respect to triangle A_2 are $(0, 1-b_2, b_2)$. Therefore functions $p_1(v)$ and $p_2(v)$ are reduced to univariate functions

$$p_1(v) = \sum_{j+k=d} c^{(1)}_{0jk} \frac{d!}{j!k!} b_2^j (1-b_2)^k \tag{2.29}$$

and

$$p_2(v) = \sum_{j+k=d} c^{(2)}_{0jk} \frac{d!}{j!k!}(1-b_2)^j (b_2)^k. \tag{2.30}$$

In this case, p_1 and p_2 join continuously on edge e if and only if $c^{(1)}_{0jk} = c^{(2)}_{0kj}$ which is the form of (2.28) when $r = 0$.

Now we show the result for $r > 0$. We choose two unparallel directions $w_1 = v_3 - v_2$ and $w_2 = v_4 - v_2$. Derivatives along other directions are linear combinations of the derivative along these two. Thus, if we can prove the equality along these two directions, the equalities along other directions are affirmed automatically. It is easy to prove that $D^l_w p_1(v) = D^l_w p_2(v)$ along these two directions by comparing (2.29) and (2.30). □

(2.28) gives us the conditions for smoothness constrains. One thing to notice is that the smoothness conditions (2.28) are linear in coefficients $\{c^{(1)}_{ijk}, c^{(2)}_{ijk}\}$. This linearity will be used in lemma 2.7.1.

2.6 Triangulation

Consider a collection of N triangles $\Delta := \{A_1, ..., A_N\}$ on the domain Ω.

Definition 2.6.1. A collection Δ of triangles that covers a domain Ω is called a triangulation of the domain, given that if a pair of triangles in Δ intersect, then their intersection is either a common vertex or a common edge.

For computer implementation, we need to represent the structure of triangulations in a convenient and concise way. There are several approaches in the literature and we pick one here based on our personal preference. First label all vertices in the triangulation in any order as long as there are no duplicate labels and then construct two lists. One is the vertex list with the ith row storing the cartesian coordinates (x_i, y_i) for the ith vertex. The other list is the triangle list with ith row, a triple

(l_{i1}, l_{i2}, l_{i3}), recording that the ith triangle is comprised of the l_{i1}th, l_{i2}th and l_{i3}th points in the vertex list.

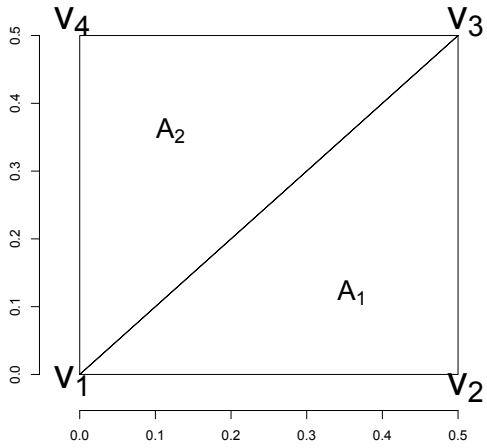

Figure 4: Second example of two triangles sharing an edge.

Table 1: Vertex list for triangulation in Figure 4.

v_i	x_i	y_i
1	0	0
2	.5	0
3	.5	.5
4	0	.5

Table 2: Triangle list for triangulation in Figure 4.

vertice 1	vertice 2	vertice 3
1	2	3
1	3	4

As an example, Tables 1 and 2 gives the two-list representation of the of triangular giving in Figure 4. Here, the first row of the triangle list records that the first triangle A_1 that consists of vertices $< v_1, v_2, v_3 >$ while the second triangle A_2 consists of

$< v_1, v_3, v_4 >$. The locations of vertices v_1, v_2, v_3 and v_4 are listed in vertex list. The representation is not unique; for example, the first row of the triangle list could be $< 2, 3, 1 >$ instead. But using different representations does not change the structure of the corresponding triangulation.

2.7 Smoothness conditions

We extend the definition of bivariate B-splines to a triangulation of a spatial domain and derive some useful representations of smoothness conditions.

2.7.1 Smoothness restrictions

Given a triangulation Δ, we can define bivariate B-splines $B_{d,i}$, as introduced in Section 2.1, on each triangle $A_i \in \Delta$. The subscript i in $B_{d,i}$ indicates that this collection of splines is corresponding to triangle A_i and therefore is differernt from the meaning of i in (2.9). Since each $B_{d,i}$ spans the polynomial space $P_d(A_i)$, the whole collection of bivariate B-splines $\{B_{d,i}\}_{i=1}^{N}$ span a spline space

$$S_d(\Delta) = \{s : s|_{A_i} \in P_d(A_i), i = 1, 2, ..., N\}. \tag{2.31}$$

Then for any $s \in S_d$, there exists a vector $\mathbf{c}_\mathrm{i}$ as in (2.16) such that

$$s|_{A_i} = B_{d,i}^T \mathbf{c}_\mathrm{i}.$$

Then

$$s = \mathbf{B}_\mathrm{d}^\mathrm{T} \mathbf{c},$$

where $\mathbf{B}_\mathrm{d} = (\mathrm{B}_{\mathrm{d},1}^\mathrm{T}, \mathrm{B}_{\mathrm{d},2}^\mathrm{T}, ..., \mathrm{B}_{\mathrm{d,N}}^\mathrm{T})^\mathrm{T}$ and $\mathbf{c} = (\mathbf{c}_1^\mathrm{T}, \mathbf{c}_2^\mathrm{T}, ..., \mathbf{c}_\mathrm{N}^\mathrm{T})^\mathrm{T}$.

The least square approximation of a target function f in the space S_d is defined as the solution of the optimization problem

$$\mathrm{Min}_s ||s - f||^2 = \mathrm{Min}_\mathbf{c} ||\mathbf{B}_\mathrm{d}^\mathrm{T} \mathbf{c} - \mathrm{f}||^2. \tag{2.32}$$

Although $s \in S_d$ is a polynomial when restricted to one triangle, it is not continuous everywhere in the domain Ω. There may be jumps across the common edges of neighboring triangles. This property is not desirable. Recall that (2.28) gives us the conditions for smoothly joining two polynomials defined on two neighboring triangles. It is easy to extend these conditions to the whole triangulation if we need the smoothness across every common edge. Then we have following lemma.

Lemma 2.7.1. Given a triangulation Δ and a function s that $s|_{A_i}(v) = B_{d,i}^T(v)\mathbf{c}_\mathrm{i}$, for integer $r = 0, 1, ..., d$, s is rth order differentiable everywhere in domain Ω if and only if there exists a matrix H such that $H\mathbf{c} = \mathbf{0}$. Matrix H depends on d, r and the structure of triangulation.

The matrix H, by controlling the values of coefficients, enforces smoothness over every shared edge in the triangulation. We can construct H by applying (2.28) on every two triangles that share an edge.

2.7.2 Construction of the smoothness matrix

Equation (2.28) provides the basis for constructing smooth matrix H. We explain the detailed steps of the construction here through an example. Consider the triangulation in Figure 4 with $d = 2$ and $r = 0$. There are $(d+1)(d+2)/2 = 6$ bivariate B-splines defined on each triangle with corresponding coefficients $\{c_{ijk}^{(1)}\}$ and $\{c_{ijk}^{(2)}\}$. Following (2.28) we have that

when $l = 0$,

$$c_{0jk}^{(1)} = \quad c_{0kj}^{(2)} B_{000}^{(2)}(v_4) = c_{0kj}^{(2)}, \qquad \text{for any non-negative intergers } (k, j) \text{ that } k + j = d. \tag{2.33}$$

when $l = 1$,

$$c_{1jk}^{(1)} = \quad c_{1kj}^{(2)} B_{1,100}^{(2)}(v_4) + c_{0,k+1,j}^{(2)} B_{1,010}^{(2)}(v_4) + c_{0,k,j+1}^{(2)} B_{1,001}^{(2)}(v_4), \qquad \text{for any non-negative intergers } (k, j) \text{ that } k + j = d - 1. \tag{2.34}$$

when $l = 2$,

$$\begin{aligned} c_{200}^{(1)} = \quad & c_{200}^{(2)} B_{2,200}^{(2)}(v_4) + c_{020}^{(2)} B_{2,020}^{(2)}(v_4) + c_{002}^{(2)} B_{2,002}^{(2)}(v_4) \\ & + c_{110}^{(2)} B_{2,110}^{(2)}(v_4) + c_{101}^{(2)} B_{2,101}^{(2)}(v_4) + c_{011}^{(2)} B_{2,011}^{(2)}(v_4). \end{aligned} \tag{2.35}$$

$$\text{for any non-negative intergers } (k, j) \text{ that } k + j = d - 1.$$

Denote a vector $\mathbf{c} = (\mathrm{c}_{\mathrm{d},0,0}^{(1)}, \mathrm{c}_{\mathrm{d}-1,1,0}^{(1)}, ..., \mathrm{c}_{0,0,\mathrm{d}}^{(1)}, \mathrm{c}_{\mathrm{d},0,0}^{(2)}, \mathrm{c}_{\mathrm{d}-1,1,0}^{(2)}, ..., \mathrm{c}_{0,0,\mathrm{d}}^{(2)})^{\mathrm{T}}$ as the union of $\{c_{ijk}^{(1)}\}$ and $\{c_{ijk}^{(2)}\}$ listed in order as (2.16). Evaluating bivariate B-splines defined on triangle A_1 at point v_4, the smoothness matrix is as follows with the first three rows corresponding to (2.33), fourth and fifth row to (2.34) and sixth row to (2.35).

$$H = \left[\begin{array}{cccccccccccc} -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 2 & 1 & -2 & -2 & 1 \end{array}\right]. \tag{2.36}$$

The triangulation in Figure 4 is symmetric around the sharing edge and both triangles are sosceles triangles, so the elements in matrix H are coincidently 0 , 1 and -1. We consider another irregular triangulation in Figure 5. The vertex list and the triangle list are given in Tables 3 and 4, the smoothness matrix H is given in (2.37).

Table 3: Vertex list for triangulation in Figure 5.

v_i	x_i	y_i
1	0	0
2	.8	0
3	1	1
4	-0.75	1

Table 4: Triangle list for triangulation in Figure 5.

vertice 1	vertice 2	vertice 3
1	2	3
1	3	4

$$H = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & .471 & 0 & -.471 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & .471 & 0 & -.471 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & .941 & .221 & -.941 & -.443 & .221 \end{bmatrix} \tag{2.37}$$

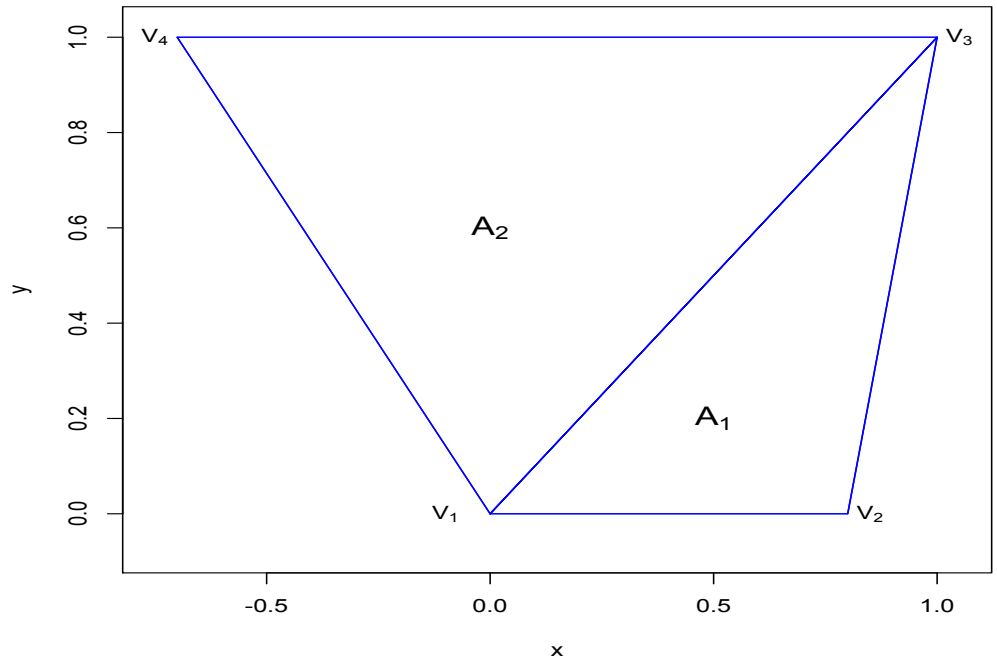

Figure 5: Third example of two triangles sharing a edge.

CHAPTER III

PENALIZED SPATIAL SMOOTHING

3.1 Introduction

Estimating a smooth surface over a two-dimensional domains is a common problem in spatial data analysis. Assume we have observations z_i at locations $(x_i, y_i) \in R^2$ that

$$z_i = f(x_i, y_i) + \epsilon_i, \qquad \epsilon_i \sim N(0, 1), i = 1, 2, ..., n.$$

Our problem is estimating a bivariate function f given the data $\{(x_i, y_i, z_i)\}$ for some unknown function f. For example in Figure 6, red dots represent the locations where soil organic matters were measured. Then GIS software generates continuous color maps of the density of organic matters over the whole area by interpolating the unobserved from surrounding observed data.

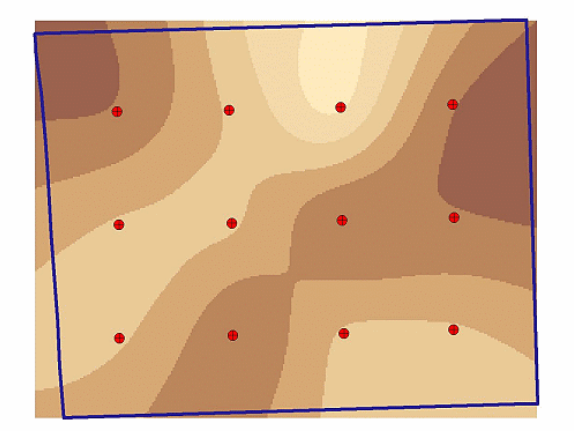

Figure 6: Heatplots of estimated soil organic matter surface based on GIS data. Red points show the locations of observations.

Wavelet-based methods, kernel smoothing, kriging and spline smoothing are four commonly used techniques for smoothing surface estimation. All these methods work

well for rectangular domains with measurements on grid, as Figure 6. However, they become inefficient or even problematic for complex domains. Wavelet-based methods (Horgan, 1999) cannot be applied when domains are not rectangular. Kernel smoothing (Wand and Jones, 1995) and kriging (Cressie, 1993) assume that the similarity between two locations depending on their Euclidean distance and therefore estimate function values as weighted sums of observations around with weights depending on the Euclidean distances. However, the usage of the Euclidean distance causes problems when estimating the surface. We explain the disadvantage of the Euclidean distance in the following example.

Figures 7 and 8 are examples of complicated domains studied in literatures. Consider four outlined points in Figure 7. The similarity between the points outlined by square will not be equal to that of the points outlined by triangles although the two pairs of points have the same the Euclidean distances. In Figure 8, The hole in the southern part of the island is an airport and the one in the north-eastern end is an oil refineries and a water purification plant. These two holes are not part of the domain and the spatial correlations between residency on opposite sides of each hole is broken down.

To fix the problem brought by the Euclidean distance, Wang and Ranalli (2007) proposed low-rank thin-plate splines defined as functions of the geodesic distance instead of the Euclidean distance, while Eilers (2006) employed the Schwarz-Christoffel transform to convert the complex domains to regular domains. We can also apply the kernel method and kriging on the geodesic distance. These modifications can be viewed as mapping the original domains and functions to the new domains and functions, then optimal smoothers can be found. But since the smoothness penalties are defined on converted functions, what they measure on original functions are no longer clear.

Figure 7: Horseshoe-shaped domain.

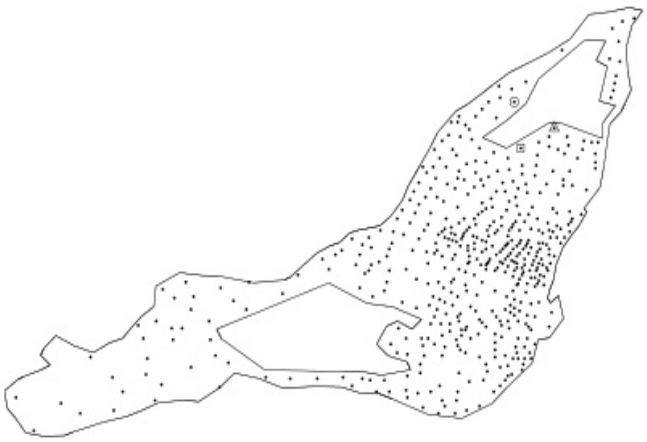

Figure 8: Island of Montreal with dots representing the centroid of units where census data are summarized.

Based on discussions above, we prefer to work on original domains with the Euclidean distances. We will focus on spline-based methods because we can define locally supported splines with supports inside the domains. In this way, we will overcome the challenges of holes and concave boundaries. The rest of this chapter is structured as follow. In Section 3.2, we list and review some existing spline-based smoothing methods focusing on complex two-dimensional domains. Then we introduce the splines and corresponding models in Section 3.3 and discuss future work in Section 3.4.

3.2 Literature review

Spline-based smoothing generally minimizes an objective function within a function space spanned by a collection of splines. The objective function is the squared distance between observed data and a spline function, adding a roughness penalty (Greens and Silverman, 1994). Let Ω be the domain, We have n observations z_i at location (x_i, y_i). Two kinds of objective functions are widely used so far. They are

$$PLS_1(f) = \sum_{i=1}^{n} \Big(z_i - f(x_i, y_i) \Big)^2 + \lambda \int_\Omega (f_{xx} + f_{yy})^2 \, dxdy, \tag{3.1}$$

and

$$PLS_2(f) = \sum_{i=1}^{n} \Big(z_i - f(x_i, y_i) \Big)^2 + \lambda \int_\Omega (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2) \, dxdy, \tag{3.2}$$

where λ in both functions is the smoothing parameter that trades off the goodness-of-fit of the data and the smoothness of the fitted function.

Traditional spline-based work on smoothing over a two-dimensional domain use either tensor-product splines or thin-plate splines (Duchon, 1977). Tensor-product splines are tensor products of univariate splines defined on grids. They cannot exclude interior holes or gaps and therefore will smooth over these areas. This behavior will introduce extra dependence between locations and produce bias. Thin-plate splines, defined on the Euclidean distance, will have the same problem as kernel smoothing and kriging.

In the statistical literature, two collections of splines are studied when considering complex domain. One is the soap smoother and the other is the finite element splines from the applied mathematics. Now we start to review these methods.

3.2.1 Finite element smoother

The finite element technique is a sophisticated method in applied mathematics. Finite elements have three components: partition, nodes and basis. We partition the

domain, identify locations of nodes and then define a group of basis, called nodal basis, on each piece of the partition according to the nodes. The basis spans a continuous, piecewise polynomial function space, in which we approximate the real function. We call this space the finite element space S^0. Hansen et al. (1998) first introduced this technique to the statistical community to deal with complex domains. Ramsay (2002) found minimizer of PLS_1 subject to certain constraint in the finite element space. Koenker and Mizera (2004) proposed a new penalty called total variation penalty that calculated the wiggleness of continuous but not smooth bivariate function and then found the minimizer in the finite element space.

Denote $H^m(\Omega)$ as the space of all continuous functions on a domain Ω whose mth-order partial derivatives are all square integrable and whose partial derivatives of order less then m are all continuous. One big problem of the finite element technique is that the functions it defines are surely continuous everywhere but not guaranteed to be differentiable. Therefore, PLS_1 and PLS_2 are not well defined. Hansen et al. (1998) and Koenker and Mizera (2004) only focused on continuous but not smooth functions, so they had different scopes from us. Ramsay (2002) imposed normal derivatives zero boundary condition to make the minimizer of PLS_1 in $S^0 \cap H^2$ equivalent to solution of a system of partial differential equations that only involve first order derivatives. In this way, the problem can be easily solved by finite element method. But the constraints of normal derivatives zero on boundary is arbitrary and may not be satisfied in many cases.

3.2.2 Soap film smoother

Wood et al. (2008) followed the variational method of deriving thin-plate spline to find the closed form for the minimizer of PLS_1 which is a linear combination of Green's functions and functions in the null space of the smoothness penalty. However,

the Green's functions can only be solved when boundary values are given. He proposed that minimizer of PSL_1 in (3.1) subject to known boundary conditions must satisfy the following differentiable equations

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = \rho, \tag{3.3}$$

$$\text{where} \quad \frac{\partial^2 \rho}{\partial x^2} + \frac{\partial^2 \rho}{\partial y^2} = 0 \tag{3.4}$$

except at data points $\{(x_k, y_k)\}$ which are pre-selected, and $\rho = 0$ on the boundary. For each data point (x_k, y_k), there is a function $g_k(x, y)$ that satisfies (3.3) and (3.4) except at (x_k, y_k). Then a linear combination of all $\{g_k\}$, called the soap film smoother $f = \sum_{k=1}^{n} \gamma_k g_k(x, y)$, is a solution to (3.3) and (3.4) except at all knots, therefore f is the minimizer to PSL_1.

The process of building the soap film smoother requires function values on the boundary are known. When function values are unknown on the boundary, Wood et al. (2008) proposed an approach to approximate the minimizer by representing function values on the boundary as a weighted sum of cyclic basis $\{\alpha_j\}$ with unknown basis coefficients. So when studying unknown boundary minimizing problem, Wood et al. (2008) used two collections of basis that one captures function variation within domain and the other represents function values on the boundary. Finally, the minimizer without boundary conditions is of the form

$$f(x, y) = \sum_{j=1}^{J} \alpha_j a_j(x, y) + \sum_{k=1}^{n} \gamma_k g_k(x, y). \tag{3.5}$$

Each data point determines one basis function in $\{g_k(x, y)\}_{k=1}^{n}$ and J is the number of cyclic basis we use to express the boundary. Coefficients $\{\alpha_j\}_{j=1}^{J}$ and $\{\gamma_k\}_{k=1}^{n}$ are estimated by minimizing PLS_1. We call $f(x, y)$ given in (3.5) a soap film smoother. When the size of the dataset is large, it is computationally heavy and not worthwhile to use the complete basis $\{g_k(x, y)\}_{k=1}^{n}$. Wood et al. (2008) provided a low rank

approximation by replacing data points with small number of pre-selected knots when constructing basis $\{g_k\}$.

3.2.3 Bivariate B-spline smoother

The bivariate B-splines, also defined on each piece of the partition as nodal basis, is a special case of the finite element. The difference between bivariate B-splines and nodal basis is that the smoothness constraints are enforced to make the smoothers built on bivariate B-splines to have the second and higher continuous derivatives. Then these derivatives, and therefore the penalty matrix, are well defined. The theory of bivariate B-splines is popular in mathematics but not familiar to statistical community. A comprehensive reference is Lai and Schumaker (2007). Guillas and Lai (2010) is the pioneer in using bivariate B-splines in statistical problems. They applied these splines in spatial functional regression problem. Later on, Lai and Wang (2010) adopted bivariate B-splines in penalized spatial regression problem and built up a theory on the convergency rate. Although their paper showed theoretically the convergency on penalized bivariate regression splines, the implement is unclear. This work studies the same penalized regression problem but gives details on computational techniques. In addition, their paper used pre-determined smoothing parameter. We using GCV to choose the smoothing parameter.

3.3 Proposed method

Our proposed method is a penalized smoothing model by applying bivariate B-splines.

3.3.1 Penalized smoothing model

Chapter II has introduced bivariate B-splines and its mathematical properties. We know that a collection of bivariate B-splines defined on each triangle of a trian-

gulation define a space S_d,

$$S_d(\Delta) = \{s : s|_{A_i} \in P_d(A_i), i = 1, 2, ..., N\}. \tag{3.6}$$

When we add smoothness matrix H discussed in Section 2.6, we have a new space $S_d^r(\Delta)$ that

$$\begin{aligned} S_d^r(\Omega) &= \{s \in C^r(\Omega) | s|_{A_i} \in P^d(A_i)\} \\ &= \{s \in C^r(\Omega) | s = Bc, Hc = 0\}. \end{aligned}$$

Minimizing either PLS_1 or PLS_2 in space S_d^r, the problem is now converted to

$$\underset{c}{Min} \left\{ \|z - Bc\|^2 + \lambda c^T P c \right\} \qquad \text{subject to } Hc = 0, \tag{3.7}$$

where $P = \int_\Omega (B_{xx} + B_{yy})^2 \, dx \, dy$ for PLS_1 and $P = \int_\Omega (B_{xx}^2 + 2B_{xy}^2 + B_{yy}^2) \, dx \, dy$ for PLS_2.

Minimizing (3.7) is not straightforward because of the constraint $Hc = 0$. We can release the constraints via QR decomposition on the transpose of matrix H and then problem (3.7) is converted to a conventional penalized regression problem without any restriction. More specifically, we assume

$$H^T = QR = [Q_1 \, Q_2] \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}, \tag{3.8}$$

where Q is an orthogonal matrix and R is an upper triangle matrix. The submatrix Q_1 is the first r columns of Q where r is the rank of matrix H. Then we find $\tilde{c}$ that $c = Q_2\tilde{c}$. It is guaranteed that $Hc = 0$. The problem (3.7) is changed to

$$\underset{\tilde{c}}{Min} \left\{ \|z - BQ_2\tilde{c}\|^2 + \lambda (Q_2\tilde{c})^T P (Q_2\tilde{c}) \right\}. \tag{3.9}$$

Given the smoothing parameter λ, the estimated coefficient vector $\hat{c}$ is

$$\hat{c} = Q_2\hat{\tilde{c}} = Q_2 Q_2^T (B^T B + \lambda P)^{-1} Q_2 Q_2^T B^T z. \tag{3.10}$$

3.3.2 Smoothing parameter

Recall that our spatial fitting problem is

$$\underset{\tilde{c}}{Min}\left\{\|z - BQ_2\tilde{c}\|^2 + \lambda(Q_2\tilde{c})^T P(Q_2\tilde{c})\right\}. \tag{3.11}$$

Given the smoothing parameter λ, this minimizing problem can be solved easily. The problem now is choosing proper λ. This parameter controls the trade-off between data-fitting and model parsimony. Large λ enforces smoother fitted function while small one gives model more flexibility and smaller fitting errors. Cross Validation (CV) is a well-known technique to choose smoothing parameter by minimizing prediction errors. Moreover, CV has a closed form for linear least square problems and is generalized to the so-called generalized cross-validation (GCV) (Craven and Wahba, 1979; Wahba, 1990). The closed form of GCV is (Wood et al., 2008; Ramsay, 2002).

$$G(\lambda) = \frac{n||z - A(\lambda)z||^2}{[tr(I - \gamma A(\lambda))]^2}, \tag{3.12}$$

here $A(\lambda)$ in equation (3.12) is the hat matrix depending on λ with the form

$$A(\lambda) = BQ_2Q_2^T(B^TB + \lambda P)^{-1}Q_2Q_2^TB^T \tag{3.13}$$

and $\gamma \geq 1$ is the scaling parameter.

Obviously, numerical searching algorithm should be adopted to find the optimal λ . We follow the methods in (Wood, 2004) which searched for the optimal λ with the Newton updates of $\eta = \log(\lambda)$. In detail, we first replace λ with e^η. Then according to the Newton method, the ith update is

$$\eta^{(i+1)} = \eta^{(i)} - M^{-1}m, \tag{3.14}$$

where m and M are the values of the first and second derivatives of GCV score with respect to η evaluated at $\eta^{(i)}$. Once the algorithm is converged, we find the optimal η and therefore $\lambda = e^\eta$. The transformation from λ to η guarantees that the chosen λ at convergence step is non-negative.

3.4 Computational issue

So far, we have discussed how to theoretically solve the spatial smoothing problem by using bivariate B-splines. However, in practice, there are some computational issues left for discussion. One issue is how to choose a good triangulation and how to implement it in programming software such as R. Another issue is the computational cost of the estimation procedure.

3.4.1 Appropriate triangulation

In univariate spline smoothing problems, the number of knots and the locations of knots have remarkable effects on the convergence rate of the splines built on the knots. Similar situations happen to bivariate B-splines whose convergence rate is affected by the number and shapes of triangles in the triangulation and how these triangles are neighbored with each other, e.g., sharing edges or only vertices. Hansen (1998) proposed an adaptive method to find the optimal triangulation in the framework of model selection. This dissertation uses penalty to control model complexity so we do not discuss data-driven triangulation design.

We introduce some notations before going to the details. For one triangle A, define $|A|$ as the largest diameter of circle containing A, and ρ_A the smallest radius of the circle contained in A. The ratio $\beta_A = \rho_A/|A|$ measures the shape of triangle. When the triangle is equilateral triangle, the ratio reaches its maximum while when the triangle is long and lean, the ratio becomes small. Then, consider triangulation Δ. Let $|\Delta| = max\{|A_i| \mid A_i \in \Delta\}$ and $\rho_\Delta = min\{\rho_{A_i} \mid A_i \in \Delta\}$. Then $\beta_\Delta = \rho_\Delta/|\Delta|$. In general, the larger the ratio β is and the more triangles we have, the better the fitting is in sense of smaller in-sample fitting errors. We have built a R package that generates triangulation automatically (users can specify the maximum $|\Delta|$). Moreover, how the triangles in the triangulation are connected with each other affects the performance

of the splines defined on them. We call this the shape of the triangulation. See Lai and Schumaker (2007) for more details on different shapes of triangulations.

However, the increase in the number of triangles will increase the number of splines, or equivalently the dimension of the space S_d^r, tremendously. So we prefer a reasonable number of triangles to comprise between the goodness of data-fitting and computation. Lai and Wang (2010) studied the convergence rate of bivariate B-splines in penalized regression problems. The main results are quoted here. Denote the true function as f and the estimated function as f_λ when the penalty parameter is λ.

Assume following conditions are satisfied:

- The true function f belongs to the Sobolev space $W^{l+1,\infty}(\Omega) = \{u \in L^p(\Omega) : D^\alpha u \in L^p(\Omega),\ \forall |\alpha| \leq l+1\}$ for an integer $l \geq 1$.

- The noise ϵ satisfies that $\lim_{\eta\to\infty} E[\epsilon^2 I(\epsilon > \eta)] = 0$. The standard deviation $\sigma(x)$ is continuous on Ω and $0 < c_\sigma \leq \inf_{x\in\Omega} \sigma(x) \leq C_\sigma < \infty$ for some constants c_σ and C_σ.

- The constants F_1 and F_2 defined as follow $F_2/F_1 = O(1)$. Here F_1 and F_2 are positive constants that are independent of function f that satisfies,
$$F_1||f||_{\infty,\Omega} \leq \left\{ \sum_{x_i\in\Omega, i=1,\ldots,n} f(x_i)^2 \right\}^{1/2} \leq F_2||f||_{\infty,\Omega}.$$

- The number of triangulation N and the sample size n satisfies that $N \asymp n^{1/(l+2)}$.

- The penalized parameter λ satisfies $\lambda = o(n^{\frac{l}{2(l+2)}})$.

Then the convergence rate is

$$\begin{aligned} ||f_\lambda - m||_{\infty,\Omega} \ &= \\ &O_p\left\{ \tfrac{\lambda}{n|\Omega|^3}|m|_{2,\infty,\Omega} + \left(1 + \tfrac{\lambda}{n|\Delta|^5}\right)\tfrac{F_2}{F_1}|\Delta|^{l+1}|m|_{l+1,\infty,\Omega} \right\}. \end{aligned} \tag{3.15}$$

3.4.2 Computation

Both the parameter estimation (3.10) and the GCV evaluation (3.12) involve the inversion of the matrix $(B^TB + \lambda P)$ where B is the evaluation matrix ob bivariate B-splines and P is the penalty matrix. When we have N triangles, $(B^TB + \lambda P)$ is a $N(d+1)(d+2)/2 \times N(d+1)(d+2)/2$ matrix. This matrix could be very large and the computation of its inverse becomes infeasible when the domain is irregular with a curvy boundary and requires a lot of small triangles to mimic the boundary, such as the second triangulation example in Section 3.2. However, thanks to the locality of bivariate B-splines, matrix B^TB and P is block diagonal with each block of dimension $(d+1)(d+2)/2 \times (d+1)(d+2)/2$. Thus, in practice, we can calculate the inversion of each small block to derive the inversion of the whole matrix. We can also define sparse matrices B, P and H by only storing the non-zero elements. These representations save a lot of memory space and therefore speed up the computing.

3.5 Simulation

This section compares the bivariate B-splines smoothing and soap file smoother on simulated datasets to show the strength of the bivariate B-splines methods.

3.5.1 Complicated domains

First simulation is designed on the horseshoe-shaped domain as shown in Figure 9. This is a widely studied domain in the literatures on the topic of smoothing over difficult domains (Wood et al., 2008; Ramsay, 2002).

The surface function is

$$f(x, y) = 8sin(xy) \tag{3.16}$$

as shown on the top left panel of Figure 10. We uniformly sampled $n = 300$ locations in the domain and then estimated the surface based on the evaluations of the true

function at these locations plus white noises from $N(0, \sigma^2)$ with $\sigma = .1, 1$ and 2.

Three estimation methods: i) soap film smoother (soap), ii) thin-plate regression splines (tprs) and iii) bivariate B-splines (BBS) were applied and compared in terms of out-of-sample mean squared errors (MSE). The bivariate B-splines method used triangulation as displayed in Figure 9 with $d = 3$ and $r = 1$. Out-of-sample mean squared errors were derived based on intensive 70×70 grid by taking the mean of all squared prediction errors at grid points which are inside the domain. All smoothing parameters were chosen by the GCV. We repeated the simulation procedure for each method at each level of noise for 400 times and then compared the distribution of MSE. The result of one replicate of estimation is shown in Figure 10 and the summary of all 400 replicates are in Figure 11.

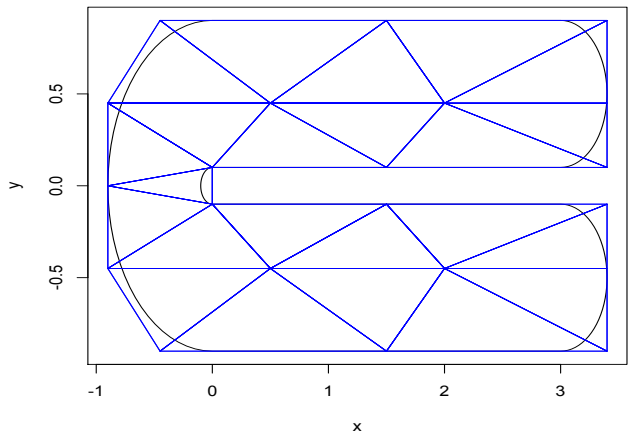

Figure 9: Triangulation on horseshoe-shaped domain.

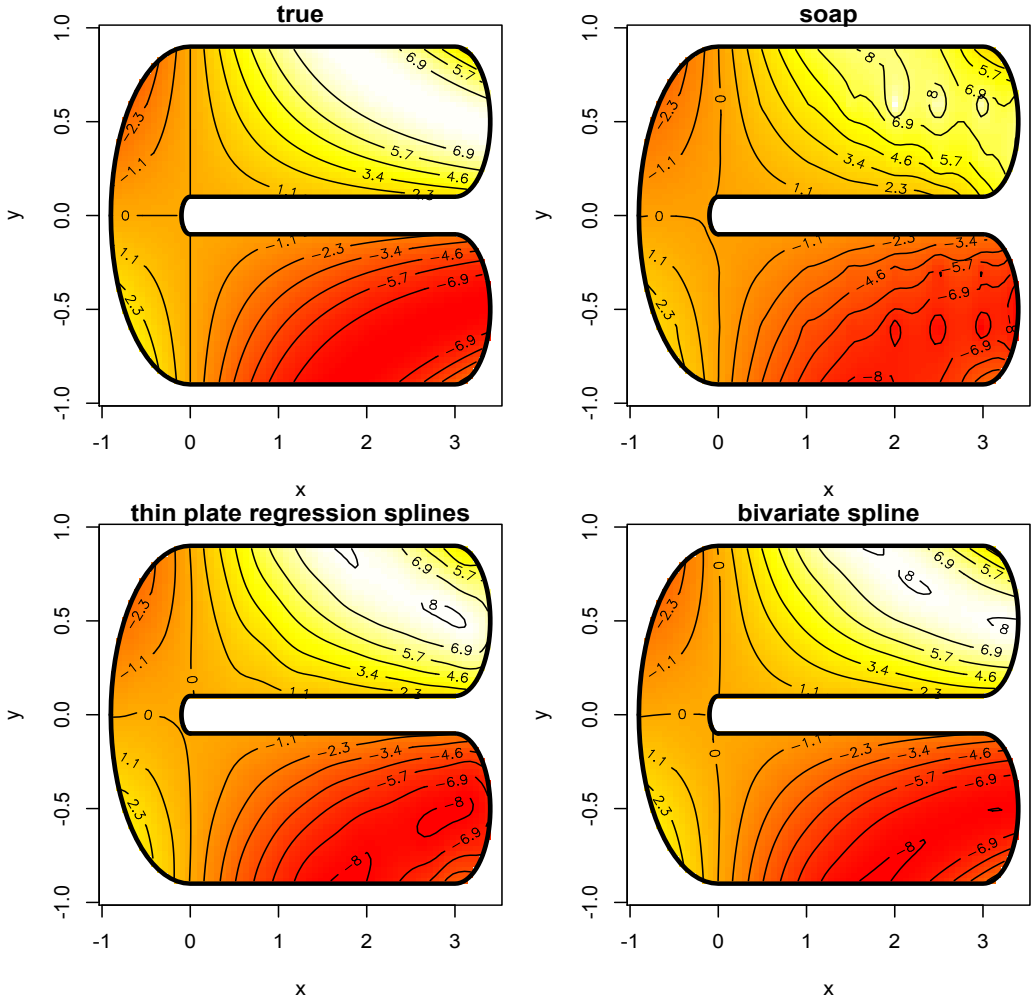

Figure 10: Heatplots with contour lines of the true surface and estimated surfaces over horseshoe-shaped domain at $\sigma = .1$. Top left panel is the true surfaces and others are estimated surface. Top right panel is from the soap film smoother with 32 knots and 39 cyclic splines on boundary. Bottom left panel is from the thin-plate regression splines with k=100 splines. Bottom right panel is from the bivariate B-splines smoothing with triangulation shown in Figure 9 and d=3, r=1.

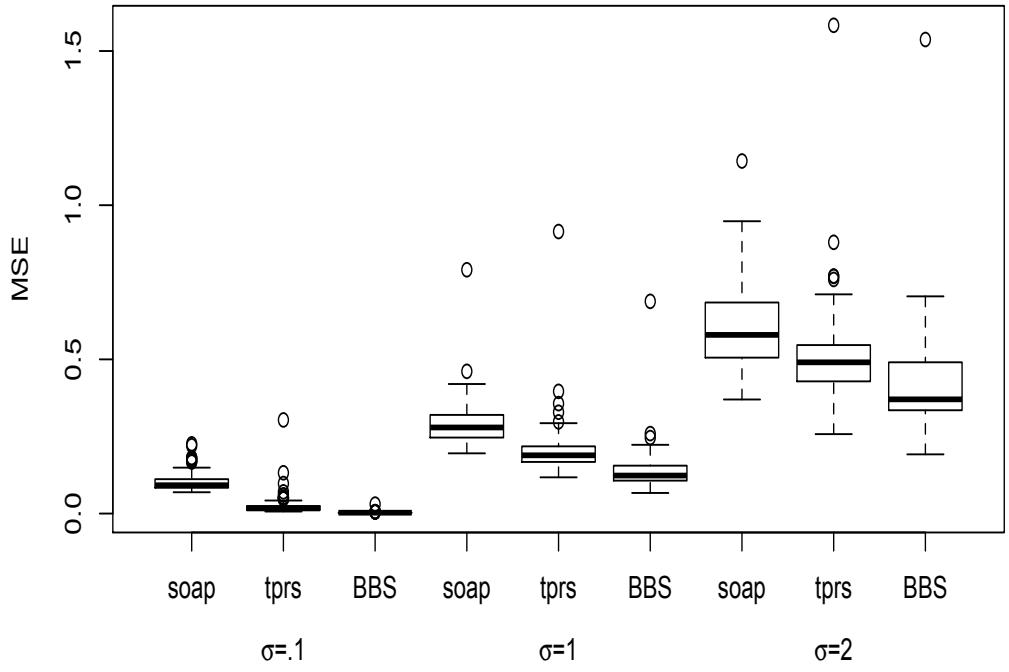

Figure 11: Comparison of out-of-sample MSE for three methods. We use soap file smoother (soap), thin-plate splines (tprs) and bivariate B-splines (BBS) at three levels of random errors $\sigma = .1, 1$ and 2 in the example of smoothing over the horseshoe-shaped domain.

Figure 10 shows that compared with the bivariate B-splines, the soap film smoother gives bigger prediction errors where function has large absolute values. Results in Figure 11 confirms that the bivariate B-splines outperforms the soap film smoother at each level.

3.5.2 A simple domain

Section 3.5.1 has shown that both bivariate B-splines and soap film smoother can be applied on complicated domains and the bivariate B-splines provide smaller out-of-sample MSE than the soap film smoother. In this section, we study the smoothing problem over a simple rectangular domain as shown in Figure 12 to see how the bivariate B-splines perform when we could solve the problem by simply using the thin-plate regression splines.

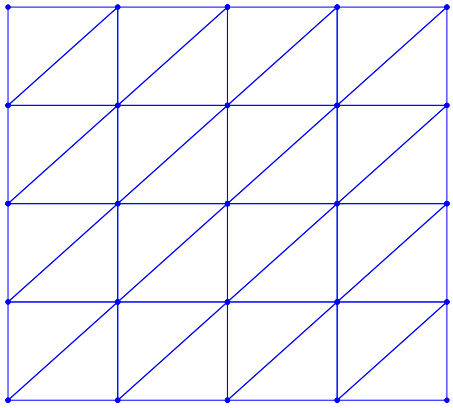

Figure 12: Triangulation on a rectangular domain.

The true function is

$$f(x,y) = \frac{.75}{\pi 0.3 \cdot 0.4} \exp\left(-\frac{(x-.2)^2}{0.3^2} - \frac{(y-.3)^2}{0.4^2} \right) + \frac{.45}{\pi 0.3 \cdot 0.4} \exp\left(-\frac{(x-.7)^2}{0.3^2} - \frac{(y-.8)^2}{0.4^2} \right). \tag{3.17}$$

This function, studied in Wood et al. (2008) and shown on the upper left panel of Figure 13, defines a smooth surface over the rectangular domain $[0,1] \times [0,1]$ with two peaks at points $(0.2, 0.3)$ and $(0.7, 0.9)$.

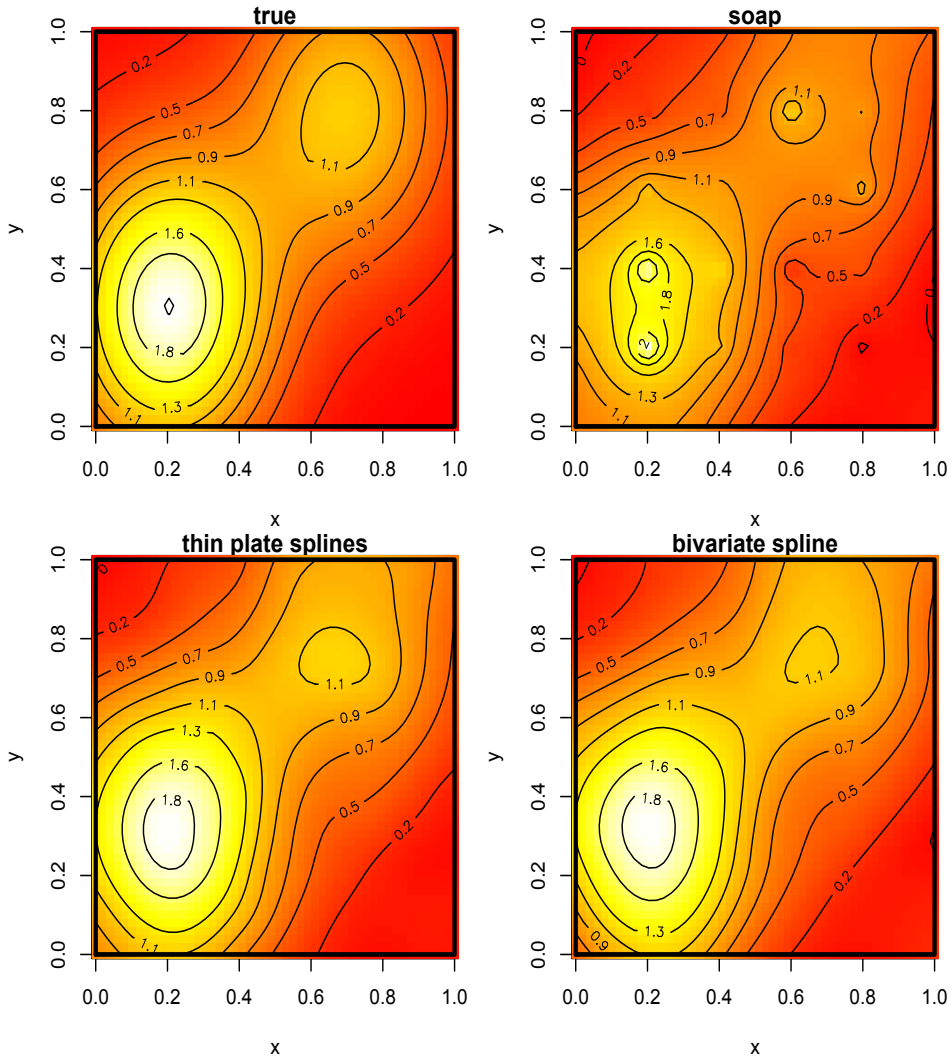

Figure 13: Heatplots with contour lines of true surface and estimated surfaces over simple domain at $\sigma = .2$. The top left panel is true surface and others are estimated surface; the top right panel is from the soap film smoother with 16 knots and 39 cyclic splines on the boundary; the bottom left panel is from the penalized thin-plate regression splines with k=100 ; the bottom right panel is from the penalized bivariate B-splines.

We uniformly sampled 500 locations in the domain. Our simulated observations were true function values at those 500 locations plus identical independent random noises from $N(0, \sigma^2)$ with σ of three levels: .01, .05, and .2. We fitted the soap film smoother, thin plate splines and bivariate B-splines for each noise level. The soap film smoother used 100 knots and 39 cyclic splines for the boundary as described in Wood et al. (2008). Bivariate B-splines used the triangulation shown in Figure 12 with $d = 3, r = 1$ and so the true degree of freedom was 69. We repeated these three estimation procedures 400 times to compare the out-of-sample MSE. All smoothing parameters were chosen by GCV. Figure 13 is one replicate of the fitting result at $\sigma = 0.2$ and the summary of all 400 replicates are shown in Figure 14 and Table 5.

The simulation results show that the thin-plate regression splines method has the smallest MSE at each level. The bivariate B-splines and soap film smoother lose some accuracy on simple domains. But the differences between MSEs of thin-plate regression splines and that of the soap film smoother or bivariate B-splines are small compared with the average MSE of the thin-plate splines. In addition, the bivariate B-splines method always gives smaller average MSE than soap film smoothing methods for all three noise levels.

When we compare estimated surface by the soap film smoother in two simulation examples (Top right panel in Figure 10 and Figure 14), we find the common phenomenon of some wiggles on contour lines. The locations of the wiggle are exactly where knots are placed. It indicates that the soap film smoother produces a surface that is not smooth and therefore increase the prediction errors around knots. This problem results from the procedure of deriving the soap film smoother.

Table 5: Summary of $\text{MSE}^{0.2}$ in simulation on simple domain.

	$\sigma = .2$			$\sigma = .05$			$\sigma = .01$		
	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
Soap	.285	.384	.323	.211	.276	.228	.182	.215	.191
Tprs	.272	.344	.305	.182	.241	.201	.114	.169	.129
BBS	.272	.345	.309	.193	.228	.210	.149	.168	.154

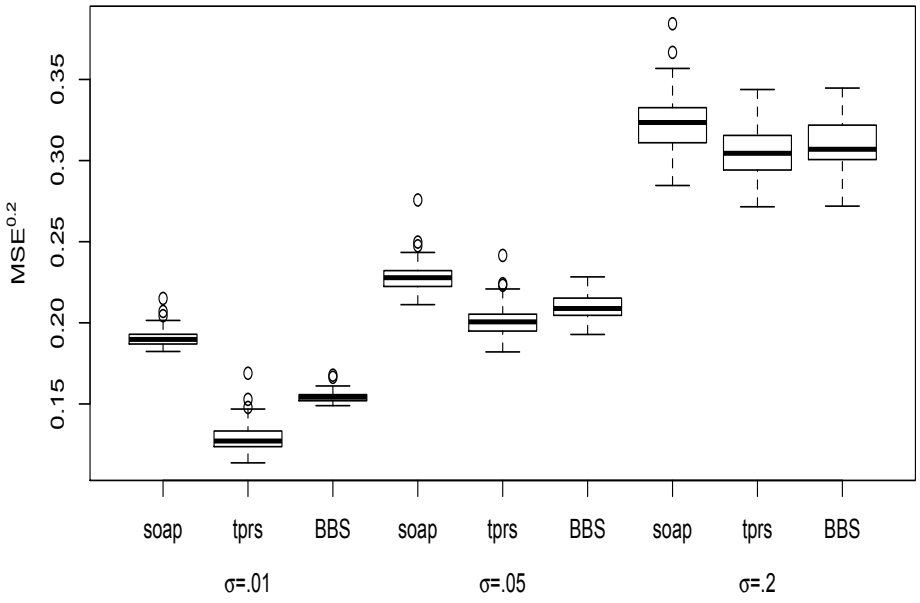

Figure 14: Comparison of out-of-sample MSE for the three methods. We use soap file smoother (soap), thin-plate splines (tprs) and bivariate B-splines (BBS) at three levels of random errors $\sigma = .01, .05$ and $.2$ in the example of smoothing over the rectangular domain.

3.6 Texas temperature surface analysis

In this section, we apply bivariate B-splines in a real example of constructing the temperature surface over texas based on data collected by the International Research Institute for Climate and Society (IRICS).

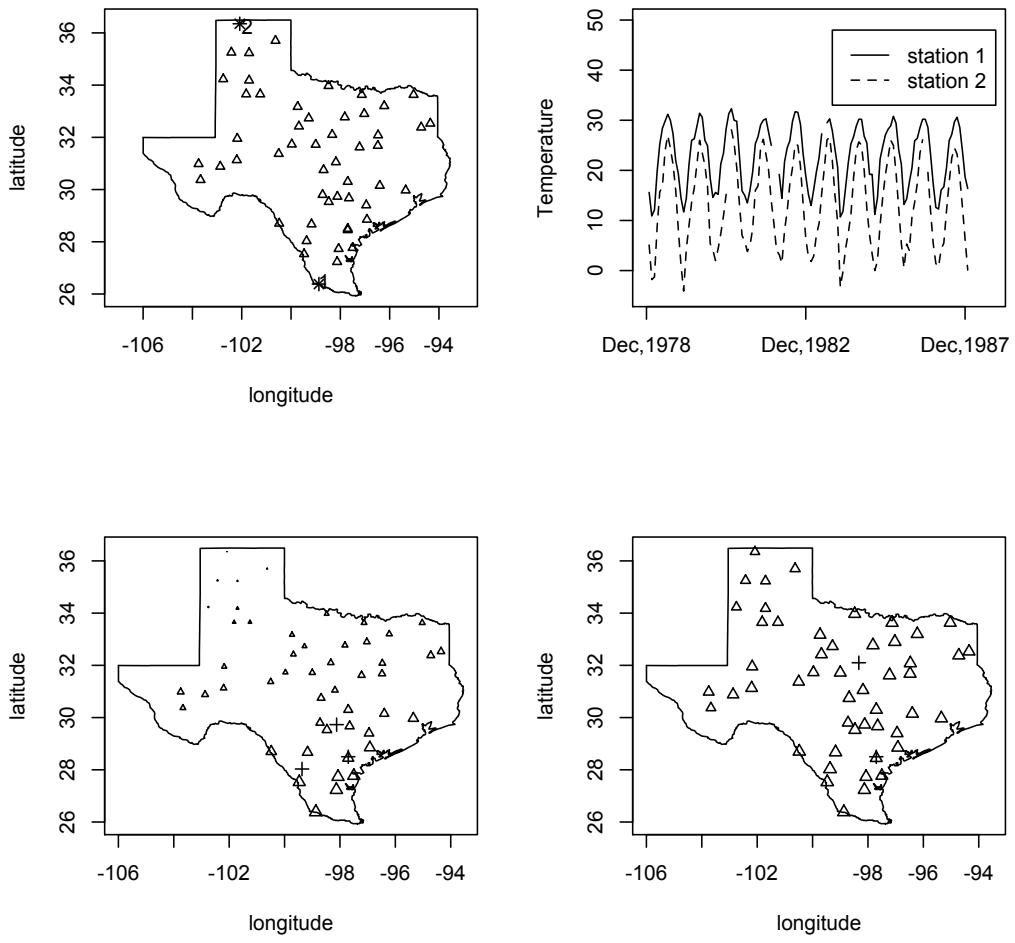

Figure 15: Stations in Texas and monthly temperatures. Source: IRICS. Top left panel shows the locations of all stations. Top right panel shows the temperatures over time for two stations highlighted on top left panel. The discontinuous parts indicate missing data at those times. Bottom two panels show that stations having observations on December and August in 1987. The larger the triangular size is, the higher the temperature is.

The dataset consists of monthly average temperatures at 52 stations in the state of Texas from Jan, 1867 to Dec, 1995. Figure 15 gives a summary of the temperatures. In some time periods, most stations have no record. For example, only 15 stations have records of the temperatures in 1989 and 11 stations in 1990. But almost all stations have records between 1930 and 1987.

We designed the following triangulation in Figure 16 that covers the irregular spatial domain such that there are observations in each triangle and set $d = 3, r = 1$. We can reconstruct one temperature surface for each month based on the data for all stations in that month. In order to check the accuracy of surface reconstruction, we randomly sampled 4 stations (10% of the total stations) out for testing and use the rest station data for modeling. Squared prediction errors are recorded. We repeated this training and testing procedure 100 times for four representative months (December, 1987; August, 1987; December, 1986; August, 1986), two of which are in summer and the others two are in winter. Figure 17 shows the contour plots of the estimated surfaces.

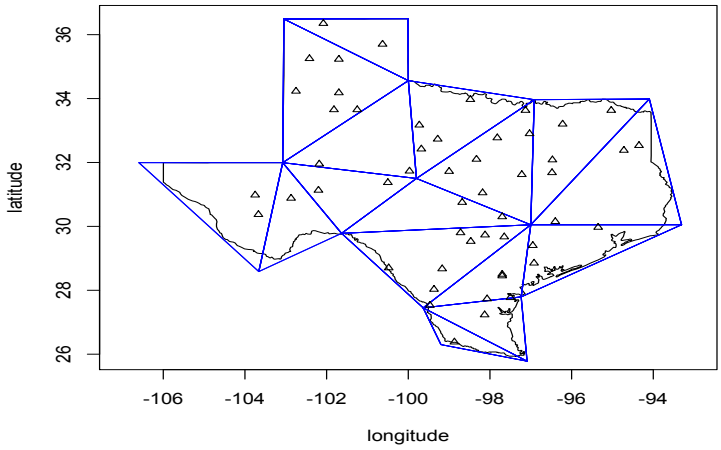

Figure 16: Triangulation on Texas and station locations.

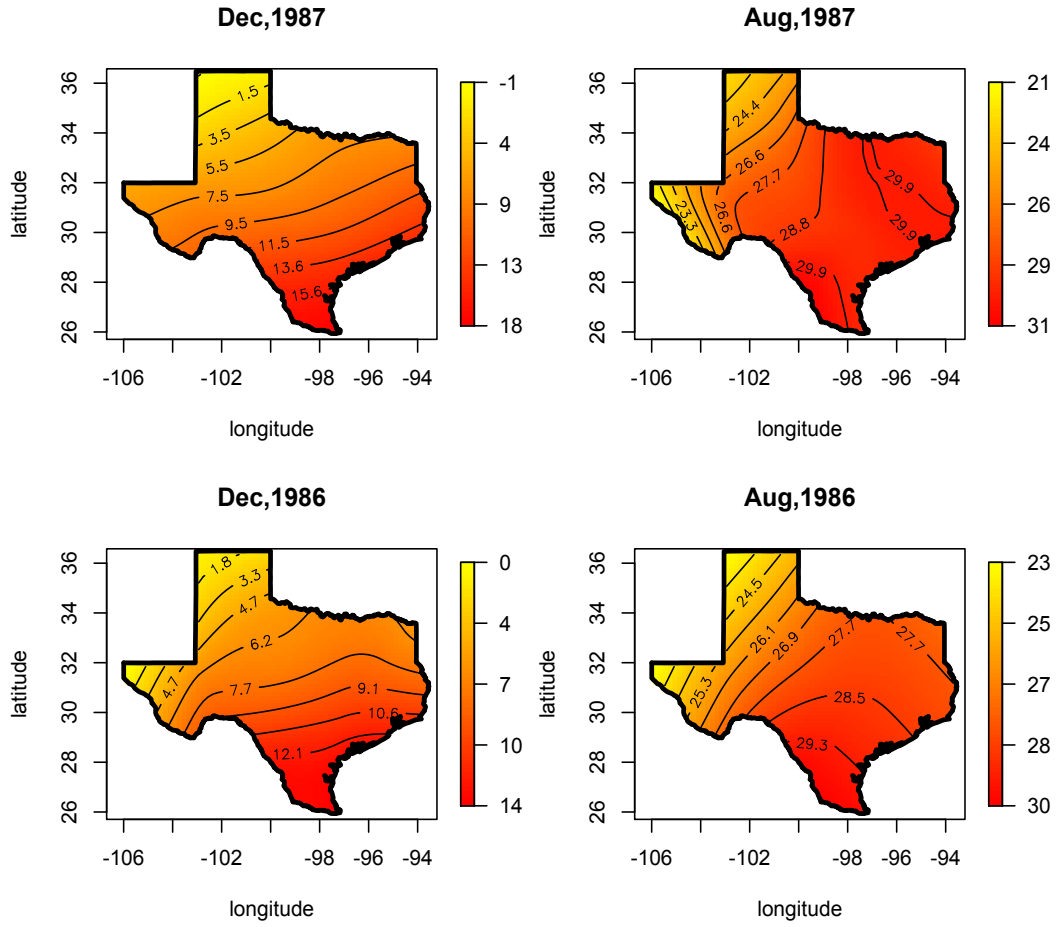

Figure 17: Contour plots of estimated temperature surface for four selected months.

Figure 18 presents the boxplots of squared prediction errors for each selected month. From the boxplots, we observed that the mean squared prediction errors is small relative to the variation of the true monthly temperatures. Some extreme values in the plots come from locations where the geological characteristics are different from the surrounding area. For example, the highest outlier in the each of the four boxplots comes from corresponding stations indexed 1 and 2 in Figure 19. Station 1 is close to Amistad National Recreation Area and Station 2 is in Sabine National Forest. It is reasonable to infer that the geological changes in these areas cause the significant

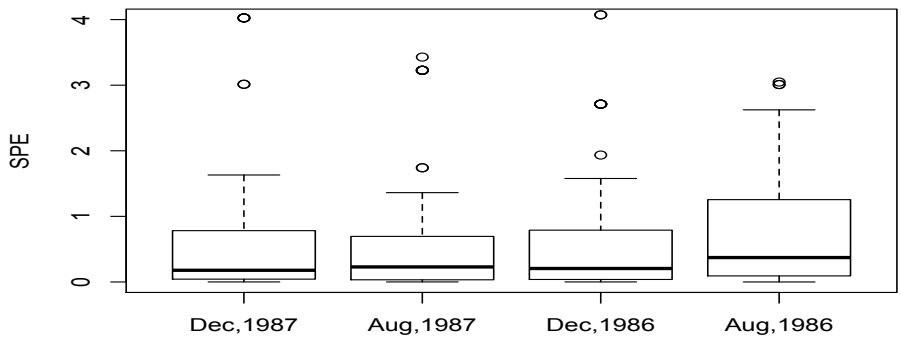

Figure 18: Boxplots of the squared prediction errors for four selected months.

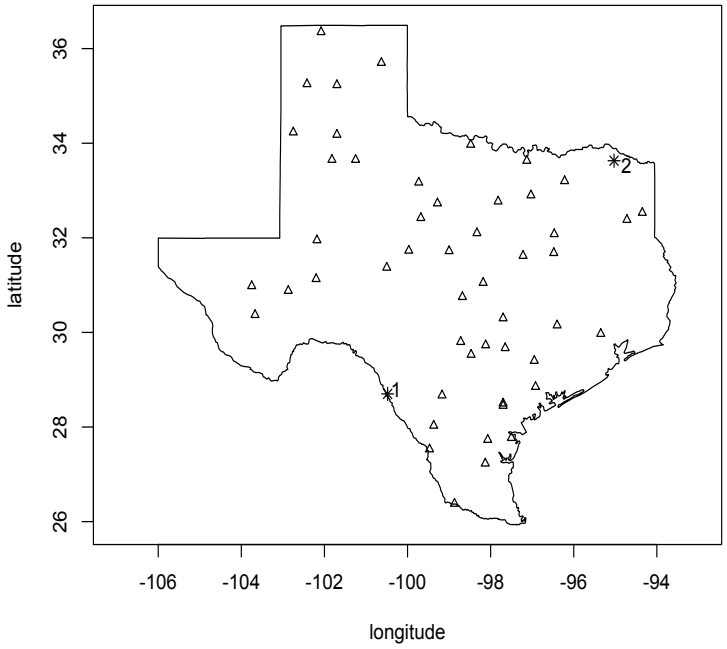

Figure 19: Stations with large prediction errors.

differences in temperatures and therefore the spatial smoothing based prediction fails.

CHAPTER IV

FUNCTIONAL PRINCIPAL COMPONENT ANALYSIS FOR SPARSE SPATIAL DATA ANALYSIS

Spatial surfaces are often observed across time or in different situations. Each surface, called one subject, are measured at an irregular and sparse set of locations which may change completely over subjects. We use a reduced rank mixed-effects model to identify the mean surface and subject-specific variations; penalized bivariate B-splines are used to model the mean surface and functional principal components. This method works well when the locations of measurements vary over subjects and are sparse for each subject.

4.1 Introduction

In spatial data analysis, one general problem is to estimate the target continuous surface over the whole domain based on discrete observations of the surface. However the number and locations of observations affect the estimates accuracy. Figure 20 displays two estimations of the same soil organic matter surface from different numbers of GIS observations using the same method. It shows that the sparse dataset may generate significantly different surface, unlike from the dense datasets.

In real world, spatial data sparse in location is a general case due to time limit or technique limit or others. One example is the dataset of monthly average temperatures in Texas quoted form the International Research Institute for Climate and Society. The dataset contains monthly mean temperatures for 52 stations in Texas from January, 1967 to December, 1995. But for each month, on average only 20 stations have records. Theoretically the temperature distribution over Texas for one

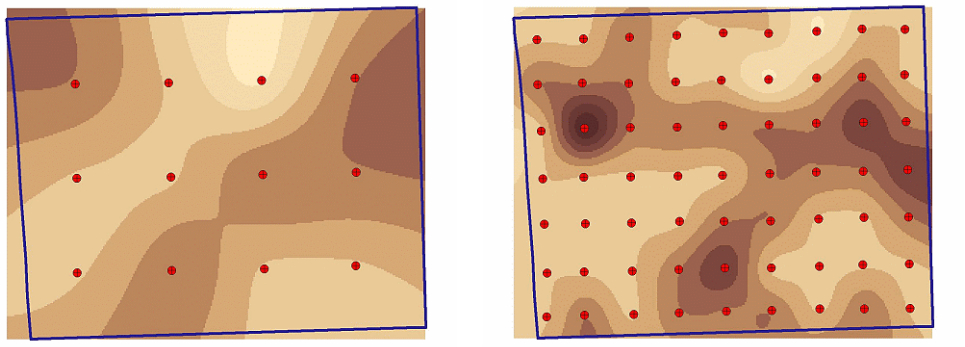

Figure 20: Estimation of soil organic matter surface from two different GIS datasets [Estimation of soil organic matter surface from two different GIS datasets.]

specific month could be estimated from the observations for that month. However, the sparsity of observations makes the estimate inconsistent and unstable.

To fix the problem above, additional data is in general collected to give more information on the shape of the surface. To be more specific, assume that we have a complicated domain Ω and target surface $\mu(v), v \in \Omega$. We observe the surface for n days to get independent n subjects. For subject i, we have m_i observations $z_{ij}, j = 1, ..., m_i$ at locations $v_{ij} = (x_{ij}, y_{ij})$. Each of the observations is disturbed by random noise which is correlated with other noises within the same subject but independent from those of other subjects. Estimating mean surface $\mu(v)$ and the mode of subject-specific variation from the mean given the whole dataset (z_{ij}, v_{ij}) is the problem we are interested in.

4.2 Existing methods

To derive mean temperature surface, one straightforward method is fitting data for one subject i. But this method is inaccurate and inconsistent since the observations are sparse. In addition, it does not utilize the whole dataset.

Another method is pulling all data from different time together and treat them

with identical independent errors. Then the whole dataset is dense in locations and we can fit a penalized least square model to it. However, the aviations of subjects from mean function is not consistent over subjects. Some subjects may have huge aviations while others have small. These subject-specific differences are due to the condition of measurements or other factors. Therefore, this estimation method of fitting one least square model to the whole dataset ignores the variability over subjects and brings extra errors.

An alternative method is to choose a collection of appropriate bivariate splines, then project everyday's data separately to the splines and finally apply classic principal component analysis on the estimated spline coefficients or the resulted "data" by evaluating the estimated surface at a fine grid. There are two drawbacks for this approach. First, fitting a surface to one-day data, as mentioned above, may produce inaccurate result due to the sparsity of data. Second, this approach uses the same weight on each day's observations and therefore does not take into account the subject-specific variations. In this dissertation, we propose a reduced rank mixed-effects model to solve the problem.

In longitudinal data analysis, estimating the mean curve from curves measured at sparse time points were well studied (James et al., 2000; Peng and Paul, 2009). However, modeling surface over two-dimensional domain with observations sparse in location were paid few attentions in spatial data analysis, even though it is a common situation. This dissertation tries to fill in the gap. We extend and adjust the model proposed in James et al. (2000) to the settings of spatial data analysis.

In the following part of this section, we will first introduce a mixed-effects model (MEM) which accounts for both the systematical mean surface and daily variation. Then we will propose a reduced rank mixed-effects model (RMEM) based on the mixed-effects model to deal with the issue of data sparsity.

4.3 Reduced rank mixed-effects model

4.3.1 Mixed-effects model

Observed surface function is

$$f_i(v) = \mu(v) + \epsilon_i(v), \qquad v \in \Omega, i = 1, ..., n. \tag{4.1}$$

The simplest assumption is that $\{\epsilon_i,\ i = 1, ..., n\}$ is independently and identically from normal distribution. Then ordinary least square regression will give an estimation of the mean function. But in our sparse data analysis, the deviation of observation from the evaluation of mean function is not identically distributed. It can be decomposed into two parts: subject-specific variation and white noise. To take into account of this variance structure, we propose a mixed-effects model:

$$f_i(v) = \mu(v) + h_i(v) + \epsilon_i(v), \tag{4.2}$$

where $v \in R^2$ is location, $\mu(v)$ is mean surface and $h_i(v)$ represents ith subject-specific variation from the mean. Random error $\epsilon(v)$ is so far assumed to have normal distribution with constant variance. More complicated error structure may be studied in future.

This is a functional version of mixed-effects model in ANOVA analysis. There are several nonparametric methods to estimate the surface functions $\mu(v)$ and $\{h_i(v)\}$. Classical methods includes kriging or kernel smoothing. As we discussed in Chapter III, bivariate B-splines outperforms kriging and kernel functions when the domain is complicated. So we use bivariate B-splines here.

We use bivariate B-splines $\mathbf{b}(\mathrm{v}) = \{\mathrm{b}_1(\mathrm{v}), \mathrm{b}_2(\mathrm{v}), ..., \mathrm{b}_\mathrm{K}(\mathrm{v})\}^\mathrm{T}$ to represent functions $\mu(x, y)$ and $h_i(x, y)$. Then we have

$$f_i(v) = \mathbf{b}^\mathrm{T}(\mathrm{v})\theta_\mu + \mathbf{b}^\mathrm{T}(\mathrm{v})\gamma_\mathrm{i} + \epsilon_\mathrm{i}(\mathrm{v}), \qquad \mathrm{i} = 1, 2, ..., \mathrm{N}. \tag{4.3}$$

$$\gamma_i \sim N(0, D), \qquad \epsilon_i \sim N(0, \sigma^2 I).$$

θ_μ is the basis coefficient vector for mean function and γ_i is the coefficient vector for function h_i. Let B_i be the evaluation of basis $\mathbf{b_i}$ at $\mathbf{v_i}$, then the corresponding data model to the mixed-effects model (4.3) is

$$\mathbf{z}_\mathrm{i} = \mathrm{B_i}\theta_\mu + \mathrm{B_i}\gamma_\mathrm{i} + \epsilon_\mathrm{i}, \tag{4.4}$$

$$\gamma_i \sim N(0, \Gamma), \qquad \epsilon_i \sim N(0, \sigma^2 I).$$

This is a mixed-effects model (MEM). We can use Restricted Maximum Likelihood (REML) to estimate parameters θ_μ and Γ.

4.3.2 Reduced rank mixed-effects model

In model (4.4), Γ is a $K \times K$ matrix of $K(K+1)/2$ parameters and θ_μ of K parameters. The total number of parameters in the mixed-effects model is large compared with the sparse dataset. In this situation, the estimation would be significantly unstable. So we reduce model rank by using function principal component to describe the mode of daily variation and get a model called reduced rank mixed-effects model.

$$f_i(v) = \mu(v) + \mathbf{g}^\mathrm{T}(\mathrm{v})\alpha_\mathrm{i} + \epsilon_\mathrm{i}(\mathrm{v}), \qquad \mathrm{i} = 1, ..., \mathrm{n}, \tag{4.5}$$

$$\epsilon_i(v) \sim N(0, \sigma^2), \qquad \alpha_i \sim N(0, D).$$

Here $\mathbf{g}(\mathrm{v}) = (\mathrm{g_1^T(v)}, \mathrm{g_2^T(v)}, ..., \mathrm{g_J^T(v)})^\mathrm{T}$ are J ($J \ll K$) mutual independent bivariate functions that are orthonormal in sense that

$$\int_\Omega g_i^T(v) g_j(v) dv = \delta_{ij}. \tag{4.6}$$

These functional principal components represent the trends in daily difference. The principal component score α_i is a J dimensional random vector. By using these principal component functions, we reduce the rank of model from K in (4.3) to J

in (4.6). Again, using basis set $\mathbf{b}(\mathrm{v})$ to represent functions $\mu(v)$ and $\{g_j(v)\}_j$ in the model, we have $\mu(v) = B^T\theta_\mu, \mathrm{and} g_j(v) = B^T\theta_j$. Collecting all $\theta_j's$ in a matrix $\Theta = (\theta_1, \theta_2, ..., \theta_J)$, we have the model in matrix form

$$\mathbf{z}_\mathrm{i} = \mathrm{B}^\mathrm{T}\theta_\mu + \mathrm{B}^\mathrm{T}\Theta\alpha_\mathrm{i} + \epsilon_\mathrm{i}, \qquad \mathrm{i} = 1, ..., \mathrm{n}, \tag{4.7}$$

$$\epsilon_i(v) \sim N(0, \sigma^2), \qquad \alpha_i \sim N(0, D).$$

Both Θ and α_i are unknown, so the model is unidentifiable. Consider (4.6), we add restrictions that

$$\Theta^T\Theta = I, \qquad \int_\Omega B^T(v)B(v)dv = I. \tag{4.8}$$

to make the model identifiable. Our basis must be orthonormalized before using it. Now the RMEM is

$$\mathbf{z}_\mathrm{i} = \mathrm{B}^\mathrm{T}\theta_\mu + \mathrm{B}^\mathrm{T}\Theta\alpha_\mathrm{i} + \epsilon_\mathrm{i}, \qquad \mathrm{i} = 1, ..., \mathrm{n}, \tag{4.9}$$

$$\epsilon_i(v) \sim N(0, \sigma^2), \qquad \alpha_i \sim N(0, D),$$
$$\int_\Omega B^T(v)B(v)dv = I, \qquad \Theta^T\Theta = I.$$

Parameter set is $M = \{\theta_\mu, \Theta, D, \sigma^2\}$. Because D is a $J \times J$ matrix where $J \ll K$, the number of parameters for RMEM is much less than that of MEM and this rank reduction makes the estimation procedure stable when the size of the dataset is small.

4.3.3 Penalized likelihood

Model (4.9) is still a mixed-effects model and the log-likelihood, given data $\mathbf{z} = (\mathrm{z}_1^\mathrm{T}, ..., \mathrm{z}_\mathrm{n}^\mathrm{T})^\mathrm{T}$ is

$$\begin{aligned} -2logL(M|\mathbf{z}) &= \sum_{i=1}^{n}\Big(m_i\log(2\pi) + \log|\sigma^2 I + B_i\Theta D\Theta^T B_i^T| \\ &\quad +(z_i - B_i\theta_\mu)^T(\sigma^2 I + B_i\Theta D\Theta^T B_i^T)^{-1}(z_i - B_i\theta_\mu)\Big). \end{aligned}$$

According to the discussion in Chapter III, the roughness penalties for functions $f = B\theta_\mu$ and $\{g_i = B\theta_i\}$ are

$$\begin{aligned} SP(f) &= \theta_\mu^T P \theta_\mu, \\ SP(g_i) &= \theta_j^T P \theta_j. \end{aligned}$$

Therefore the penalized log-likelihood is

$$\begin{aligned} -2logL(M|\mathbf{z}) &= \sum_{i=1}^{n} \Big(m_i \log(2\pi) + \log|\sigma^2 I + B_i \Theta D \Theta^T B_i^T| \\ &\quad + (z_i - B_i\theta_\mu)^T (\sigma^2 I + B_i \Theta D \Theta^T B_i^T)^{-1} (z_i - B_i \theta_\mu) \Big) \\ &\quad + \lambda_\mu \theta_\mu^T P \theta_\mu + \lambda_f \sum_{j=1}^{K} \theta_j^T P \theta_j. \end{aligned}$$

4.4 Model estimation

Theoretically, we can use MLE or REML to estimate unknown parameters. In maximizing likelihood estimation, we take the first-order partial derivative of the $-2logL$ with respect to each parameters and then set the partial derivatives to zero and solve the equations. There is no close form for the estimates of parameters. We need some numerical methods to find the optimal set M that minimize $-2logL$. We could use the Newton-Raphson or simplex or other methods. More details are given seen in Peng and Paul (2009). However, because the domain for each parameter is a bounded space and some parameters are of high dimension, the searching procedure is in practice difficult. We propose an EM algorithm here to estimate the parameters in set M.

4.4.1 Penalized joint likelihood

In this dissertation, we treat $\{\alpha_i\}_{i=1}^{N}$ as missing data instead of random variables and use the EM algorithm to estimate parameters in set M. The penalized joint log-likelihood is

$$\begin{aligned} PSL \quad & (M|\alpha, \mathbf{z}) \propto \\ & \textstyle\sum_{i=1}^{n} \Big(m_i \log\sigma^2 + \frac{1}{\sigma^2}(z_i - B_i\theta_\mu - B_i\Theta\alpha_i)^T(z_i - B_i\theta_\mu - B_i\Theta\alpha_i) \\ & + \log|D| + \alpha_i^T D^{-1}\alpha_i \Big) + \lambda_\mu \theta_\mu^T P \theta_\mu + \lambda_f \textstyle\sum_{j=1}^{J} \theta_j^T P \theta_j. \end{aligned} \tag{4.10}$$

According to EM algorithm, we need to calculate conditional marginal likelihood $E_{\alpha|\mathbf{z}}\big[PSL(M|\alpha, \mathbf{z})\big]$ and then optimize it with respect to all parameters in M.

4.4.2 EM steps

The α_i appears in $PSL(M|\alpha, \mathbf{z})$ in the form of α_i and $\alpha_i\alpha_i^T$, so we just need to derive the conditional expectations $E[\alpha_i|\mathbf{z_i}]$ and $E[\alpha_i\alpha_i^T|\mathbf{z_i}]$ to replace corresponding α_i and $\alpha_i\alpha_i^T$ in (4.10).

Both α_i and z_i are normally distributed, so their joint distribution is normal

$$\begin{pmatrix} \alpha_i \\ z_i \end{pmatrix} = N\left(\begin{pmatrix} 0 \\ B_i\theta_\mu \end{pmatrix}, \begin{pmatrix} D & \alpha_i^T\Theta^T B_i^T \\ B_i\Theta\alpha_i & \sigma^2 I + B_i\Theta D\Theta^T B_i^T \end{pmatrix} \right). \tag{4.11}$$

We could derive $E[\alpha_i|\mathbf{z_i}]$ and $E[\alpha_i\alpha_i^T|\mathbf{z_i}]$ easily and therefore $E_{\alpha|\mathbf{z}}\big[PSL(M)\big]$. This is the E-step in EM algorithm. Then maximize $E_{\alpha|\mathbf{z}}\big[PSL(M)\big]$ with respect to each parameter in M-step. The detailed EM algorithm is as follow.

1. Given current estimates for all parameters, predict α_i and $\alpha_i\alpha_i^T$.

   As we know

According to the properties of multivariate normal distribution, we could derive the conditional distribution from (4.11) that

$$\begin{aligned} E(\alpha_i|z_i) &= D\Theta^T B_i^T(\sigma^2 I + B_i\Theta D\Theta^T B_i^T)^{-1}(z_i - B_i\theta_u) \\ &= (D + \Theta^T B_i^T B_i\Theta/\sigma^2)^{-1}\Theta^T B_i^T(z_i - B_i\theta_u). \\ Var(\alpha_i|z_i) &= D - D\Theta^T B_i^T(\sigma^2 I + B_i\Theta D\Theta^T B_i^T)^{-1}B_i\Theta D \\ &= (D + \Theta^T B_i^T B_i\Theta/\sigma^2)^{-1}. \end{aligned}$$

Thus, we know that

$$\begin{aligned} \hat{\alpha}_i &= E(\alpha_i|z_i, \hat{\theta}_u, \hat{\Theta}, \hat{\sigma}^2, \hat{D}) \\ &= (\hat{\sigma}^2\hat{D}^{-1} + \hat{\Theta}^T B_i^T B_i\hat{\Theta})^{-1}\hat{\Theta}^T B_i^T(\mathbf{z_i} - \mathbf{B_i}\hat{\theta}_\mathbf{u}). \\ \widehat{\alpha_i\alpha_i^T} &= E(\alpha_i\alpha_i^T|z_i, \hat{\theta}_u, \Theta, \hat{\sigma}^2, \hat{D}) = \hat{\alpha}_i\hat{\alpha}_i^T + (\hat{D}^{-1} + \hat{\Theta}^T B_i^T B_i\hat{\Theta}/\hat{\sigma}^2)^{-1}. \end{aligned}$$

2. Given current estimates for α_i, θ_u and Θ, we estimate σ^2 and D

$$\begin{aligned} \hat{\sigma}^2 &= \tfrac{1}{\sum n_i}\sum_{i=1}^N E[\epsilon_i^T\epsilon_i|z_i] \\ &= \tfrac{1}{\sum n_i}\sum_{i=1}^N E[(z_i - B_i\hat{\theta}_u - B_i\hat{\Theta}\alpha_i)^T(z_i - B_i\hat{\theta}_u - B_i\hat{\Theta}\alpha_i)|z_i] \\ &= \tfrac{1}{\sum n_i}\sum_{i=1}^N \Big((z_i - B_i\hat{\theta} - B_i\hat{\Theta}\hat{\alpha}_i)^T(z_i - B_i\hat{\theta} - B_i\hat{\Theta}\hat{\alpha}_i) \\ &\quad + trace[B_i\hat{\Theta}(\hat{D}^{-1} + \hat{\Theta}^T B_i^T B_i\hat{\Theta}/\hat{\sigma}^2)^{-1}\hat{\Theta}^T B_i^T]\Big). \\ \hat{D}_{jj} &= \tfrac{1}{N}\sum_{i=1}^N E[\alpha_{ij}^2|z_i] = \tfrac{1}{N}\sum_{i=1}^N (E^2[\alpha_{ij}|z_i] + Var(\alpha_{ij}|z_i)) \\ &= \tfrac{1}{N}\sum_{i=1}^N \Big(\hat{\alpha}_{ij} + (\hat{D}^{-1} + \hat{\Theta}^T B_i^T B_i\hat{\Theta}/\hat{\sigma}^2)^{-1}_{jj}\Big), \qquad j = 1, ..., J. \end{aligned}$$

3. Given current estimates for σ^2, D and α_i, we estimate Θ and θ_u by minimizing

$$\begin{aligned} \sum_{i=1}^N &\Big[(z_i - B_i\hat{\theta}_u - B_i\hat{\Theta}\hat{\alpha}_i)^T(z_i - B_i\hat{\theta}_u - B_i\hat{\Theta}\hat{\alpha}_i)\Big] \\ &+\lambda_u\sigma^2\theta_u^T K\theta_u + \lambda_f\sigma^2\textstyle\sum_{j=1}^J \theta_j^T K\theta_j \end{aligned} \tag{4.12}$$

Therefore

$$\hat{\theta}_u = \left(\sum_{i=1}^{N} B_i^T B_i + \lambda_u \sigma^2 K \right)^{-1} \sum_{i=1}^{N} B_i^T (z_i - B_i \hat{\Theta} \hat{\alpha}_i).$$

For matrix Θ we have two approaches to estimate it. One is to use iterative procedure to update each column of Θ iteratively till converge while the other one is to re-formalize the model to derive a closed form for the MLE of Θ.

- In the first approach, the ith column $\hat{\theta}_j$ in matrix Θ is

$$\begin{aligned} \hat{\theta}_j = & \left(\sum_{i=1}^{N} \widehat{\alpha_{ij}^2} B_i^T B_i + \lambda_f \sigma^2 K \right)^{-1} \\ & \sum_{i=1}^{N} B_i^T \left(\widehat{\alpha}_{ij} (z_i - B_i \hat{\theta}_u) - \sum_{k \neq j} \widehat{\alpha_{ij} \alpha_{ik}} B_i \hat{\theta}_k \right). \end{aligned} \tag{4.13}$$

  Repeat this procedure for each column of Θ till there is no significant change in the estimates of Θ.

- The second approach derives the closed-form for Θ through rewriting the model. We define a matrix $\tilde{B}_i = (B_i \alpha_{i1}, ..., B_i \alpha_{iJ}), \tilde{K} = K \otimes I_J$ and $\tilde{\Theta} = (\theta_1^T, ..., \theta_J^T)^T$. Then $z_i = B_i \theta_u + \tilde{B}_i \tilde{\Theta} + \epsilon_i$. So straightforwardly

$$\hat{\tilde{\Theta}} = \left(\sum_{i=1}^{N} \tilde{B}_i^T \tilde{B}_i + \lambda_u \sigma^2 \tilde{K} \right)^{-1} \sum_{i=1}^{N} \tilde{B}_i^T (z_i - B_i \hat{\theta}_u).$$

  Compared with the first approach, the second one gives more accurate estimation since it is in closed form but it might cost more computation time because larger matrix operations are involved.

4. Repeat above three steps till convergency.

5. Orthogonalize matrix Θ by setting Θ to be the first J eigenvectors of $\hat{\Theta}^T \hat{D} \hat{\Theta}$.

Above are the iteration part of EM algorithm. The problem left is providing initial values. First, We use z_i regress on B_i to get an initial estimate of θ_u. Second,

for each subject, we use the fitting error from the above simple regression to regress on B_i again to get estimation of $\Theta\alpha_i$. Third, we collect all $\{\Theta\alpha_i\}$ and decompose them to Θ and α_i's. Finally, we calculate the empirical variance of α_i as initial estimate of $\hat{D}$ and the residual mean square error as the initial estimate of σ^2.

4.5 Model selection

Two parts should be considered for model selection. One part is triangulation and penalty parameters while the other is the selection of the number of significant principal surfaces.

4.5.1 Triangulation

Both the triangulation and penalty parameters determine the complexity of the model. Hansen et al. (1998) did not use smoothness penalty term but use the Rao statistic to choose an optimal triangulation. In contrast, Koenker and Mizera (2004) used an arbitrary triangulation and then used a penalty term to restrict the complexity of the model. We adopt the philosophy of roughness penalty as Koenker and Mizera (2004). Penalties on the roughness of mean function and that of principal functions are introduced in (4.9). Cross-validation, generalized cross-validation, AIC/BIC are generally used criteria to choose penalty parameters. However, it is not easy to derive a form for generalized cross-validation or define AIC/BIC. So we use crossing-validation. Simplex method is used to find the penalty parameters to minimize $-2\log L$ in (4.10) calculated from cross-validation.

4.5.2 Number of principal surfaces

Identifying the number of principal component surfaces is important in sparse data analysis. Extra principal component functions take into account of the variation in mean function. Lack of principal component functions cannot capture all

subject-specific variation and leave the white noise related. In this case, all estimation procedure based on uncorrelated errors will fail.

Several methods are proposed for selecting the number of principal surfaces. We follow the idea of classic principal component analysis which is to find components corresponding to significant large eigenvalues. We estimate variance matrix D for each case $J = 1, 2, ...$ to get a group of estimates $\hat{D}_1, \hat{D}_2, ...$ with decreasing diagonal elements in each estimated matrix. Then we choose the J such that

$$\text{In matrix } \hat{D}_J, \qquad \sum_{j=1}^{J-1} \hat{D}_{jj} < c_1 \sum_{j=1}^{J} \hat{D}_{jj}.$$

while

$$\text{In matrix } \hat{D}_{J+1}, \qquad \sum_{j=1}^{J} \hat{D}_{jj} > c_2 \sum_{j=1}^{J+1} \hat{D}_{jj}.$$

The values of c_1 and c_2 are determined case-by-case. This idea is also used in James et al. (2000) and Zhou et al. (2008).

4.6 Simulation

This section sets up some simulations to illustrate our estimation procedure. We choose a rectangular domain with a hole inside as shown in Figure 21. Test functions are designed as follow, in which two principal component functions are orthonormal.

$$z_i(x, y) = \mu(x, y) + f_1(x, y) + f_2(x, y) + \epsilon_i,$$

where

$$\begin{aligned}
\mu(x, y) &= 100\left(1 + \frac{x}{100} + \exp(-\frac{(x-60)^2}{500})\right), \\
f_1(x, y) &= \log(x + y + 5)/3.363, \\
f_2(x, y) &= 2.23(\sqrt{x + y} - .708\log(x + y + 5)).
\end{aligned}$$

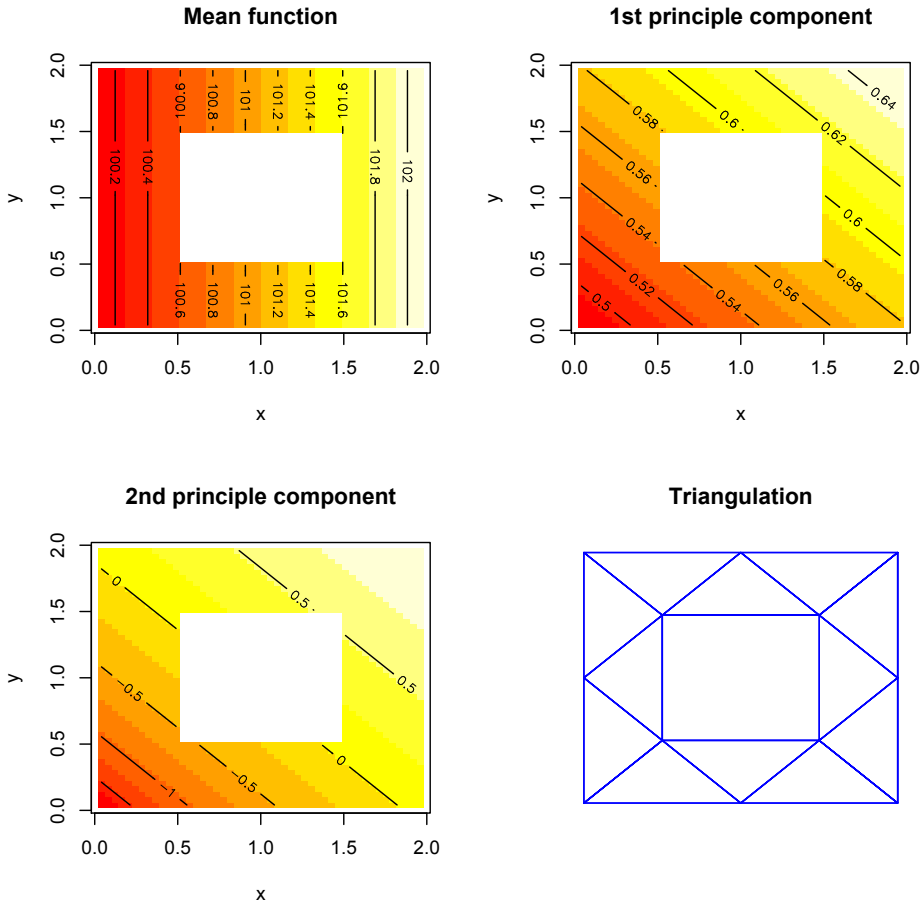

Figure 21: Heatplots of pre-selected functions for reduced rank mixed-effects models. Top left penal shows the mean function; top right panel shows the first principal component function; bottom left panel is the second principal component function; bottom right panel is the triangulation we used in the estimation.

We set up six cases to test the performance of our model and estimation procedure. In detail, we chose the number of subjects as 100 or 50, and the number of observations for each subject is uniformly sampled from range of 11 to 20. The locations of observations are also uniformly sampled from the domain. We considered two levels of D_α and two levels of σ, which are $D_\alpha = (36, 4)$ or $D_\alpha = (6, 4)$ while $\sigma = .1$ or 1.

We simulated each case for 400 times. Number of principal components is preselected as 2. In each replicate, we get fitted value $\hat{\mathbf{z}}_\mathrm{i}$ and then calculate mean squared errors between $\hat{\mathbf{z}}_\mathrm{i}$ and true value $\mathbf{z}_\mathrm{true}$ defined as

$$\mathrm{MSE} = \left[\sum_{i=1}^{n} (\hat{\mathbf{z}}_\mathrm{i} - \mathbf{z}_\mathrm{true})^\mathrm{T} (\hat{\mathbf{z}}_\mathrm{i} - \mathbf{z}_\mathrm{true}) \right] / \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{m_i}, \tag{4.14}$$

where m_i is number of observations each time. In the meantime, we obtained the estimated mean function $\hat{\mu}(x, y)$ and principal component functions $\hat{f}_1(x, y)$ and $\hat{f}_2(x, y)$ and calculated integrated squared error defined as

$$\mathrm{ISE(f)} = \int_\Omega \left(f(x, y) - \hat{f}(x, y) \right)^2 dxdy. \tag{4.15}$$

The simulation results are reported in Table 6 show that our procedure gives consistent estimation of the parameters in the model and the mean function and principal component functions. Even in the last two cases where the diagonal elements of variance matrix D are close to each other, our estimation procedure could identify these two components correctly. Figures 22 and 23 show the fitted surfaces in six scenarios mentioned in Table 6.

Table 6: Table of simulation results for reduced reank mixed-effects model. "nsubj" is number of subjects while "nobs" is total number of observations; the "z" column is fifth root of MSE defined in (4.14); the MISE for each smooth function is defined in (4.15); the MSE for all parameters are shown in last three columns.

			MSE$^{0.2}$)				MSE$^{0.2}$)		
nsubj	nobs	$(D_{\alpha 1}, D_{\alpha 2}, \sigma)$	z	μ	f_1	f_2	D_{11} (MSE)	D_{22} (MSE)	σ(MSE)
50	770	(6,4,1)	.685	.569	.442	.508	1.500	1.029	8.702e-4
50	770	(6,4,0.1)	.299	.554	.450	.468	1.207	0.813	2.549e-5
50	770	(36, 4, 1)	.691	.658	.219	.396	55.810	0.824	8.869e-4
50	770	(36,4,0.1)	.304	.703	.197	.274	53.108	0.770	8.008e-6
100	1500	(36, 4, 1)	.681	.600	.201	.363	28.455	0.502	4.403e-4
100	1500	(36, 4, 0.1)	.288	.590	.192	.264	24.114	0.299	1.090e-5

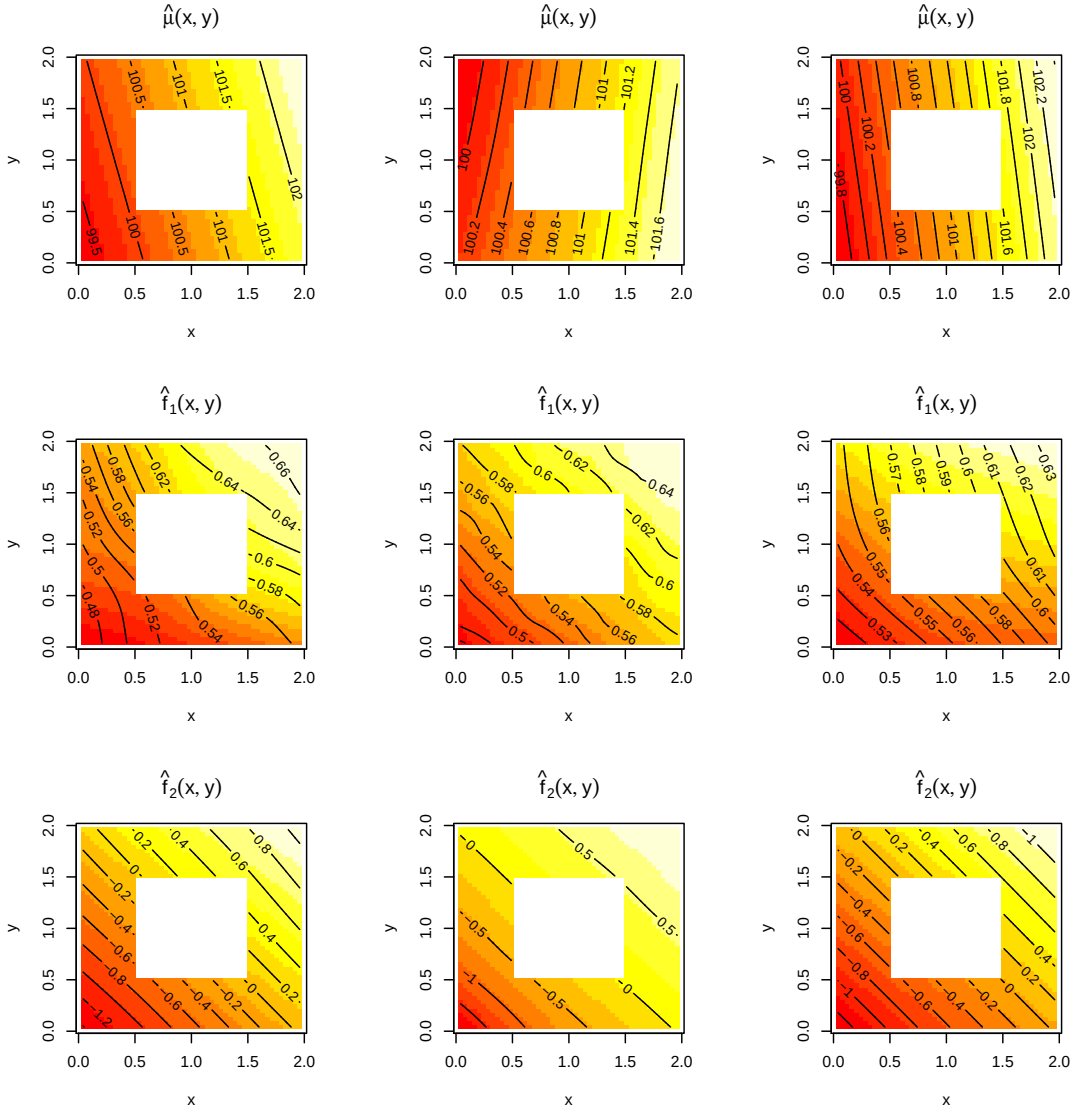

Figure 22: Heatplots of fitted functions for one simulated dataset for the first three settings in Table 6 from left to right.

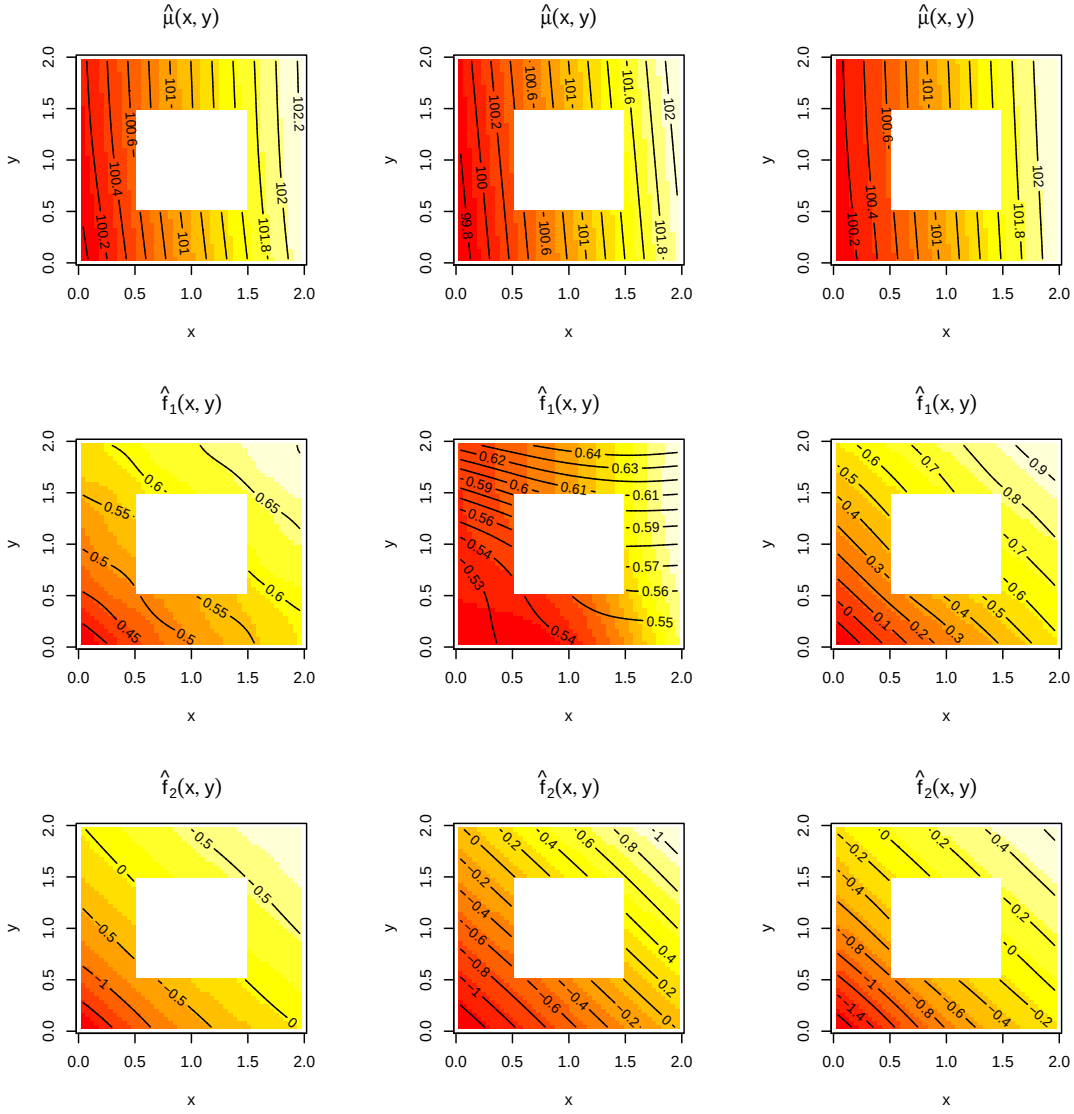

Figure 23: Heatplots of fitted functions for one simulated dataset for last three settings in Table 6 from left to right.

4.7 Texas temperature analysis

In this section, we analyze the temperature variation over space and time in Texas by applying our model to monthly average temperatures recorded by International Research Institute for Climate and Society (IRICS) at 52 weather stations spread out in Texas.

We first analyze all December temperature over 10 years. A sequence of models with different numbers of principal component functions are considered. The diagonal elements of estimated covariance matrix D in decreasing order is given in Table 7.

Table 7: The diagonal elements of estimated covariance matrix $\hat{D}$ when we choose three principal components and four.

Number of principal comp.	3			
Principal comp.	1	2	3	
$\hat{D}$	186.55	14.04	11.57	
Number of principal comp.	4			
Principal comp.	1	2	3	4
$\hat{D}$	187.14	16.63	12.47	1.82

We decide to use three principal components for the model based on the results in Table 7. Figures 24 and 25 show the estimated functions and point-wise bootstrap standard error for all estimations. 100 resamples are taken in the bootstrap. Mean function has a strong latitude trend. The first principal component function is close to a constant function and therefore indicates year-specific aviation from the 100-year average. The second and third principal component functions show geological variations. All bootstrap standard deviations are comparatively small except on boundary. The large standard deviation at the boundary is due to the sparsity of observations. Figure 26 shows the time series plot of principle scores.

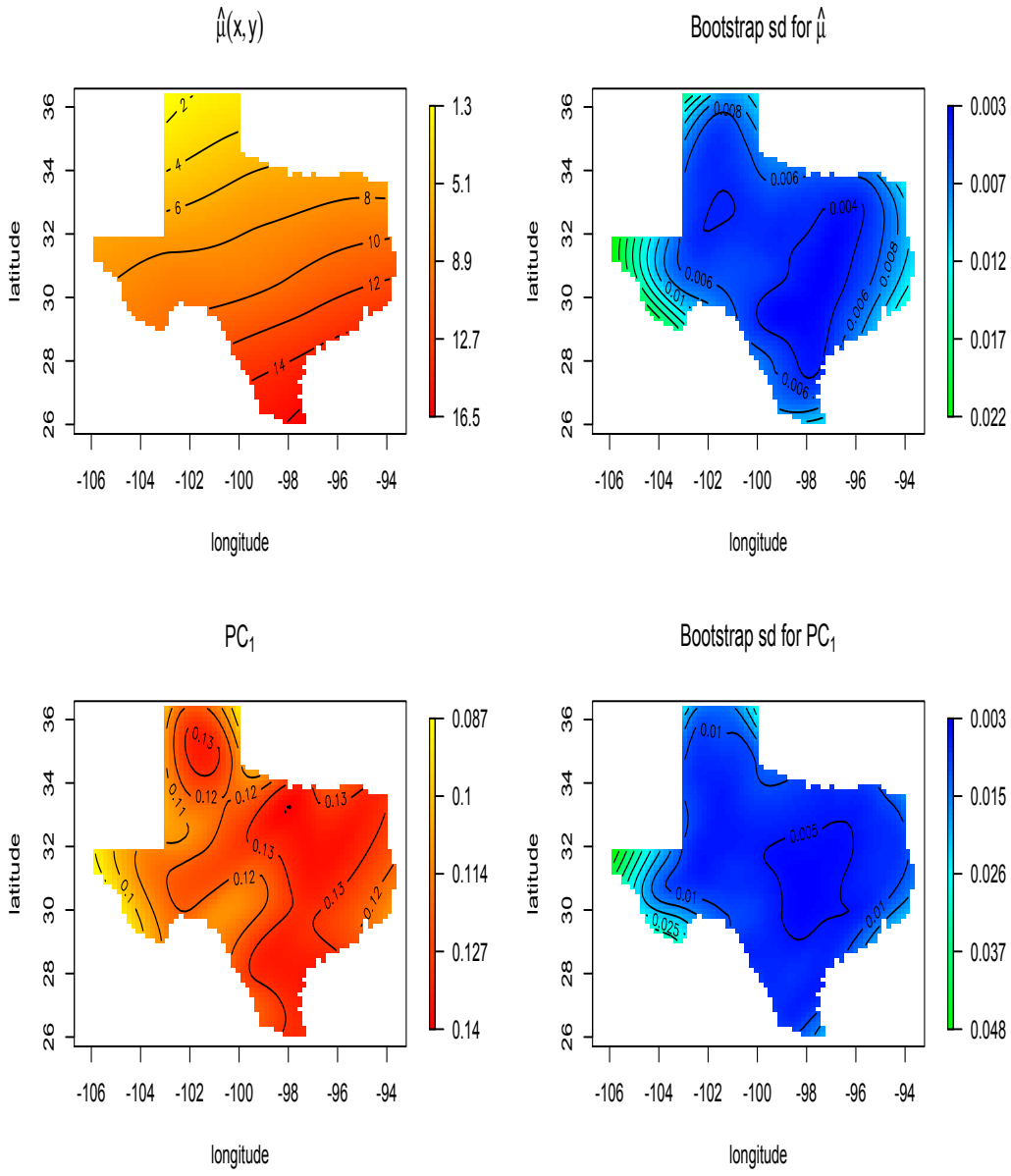

Figure 24: Estimated mean function and first principal component function.

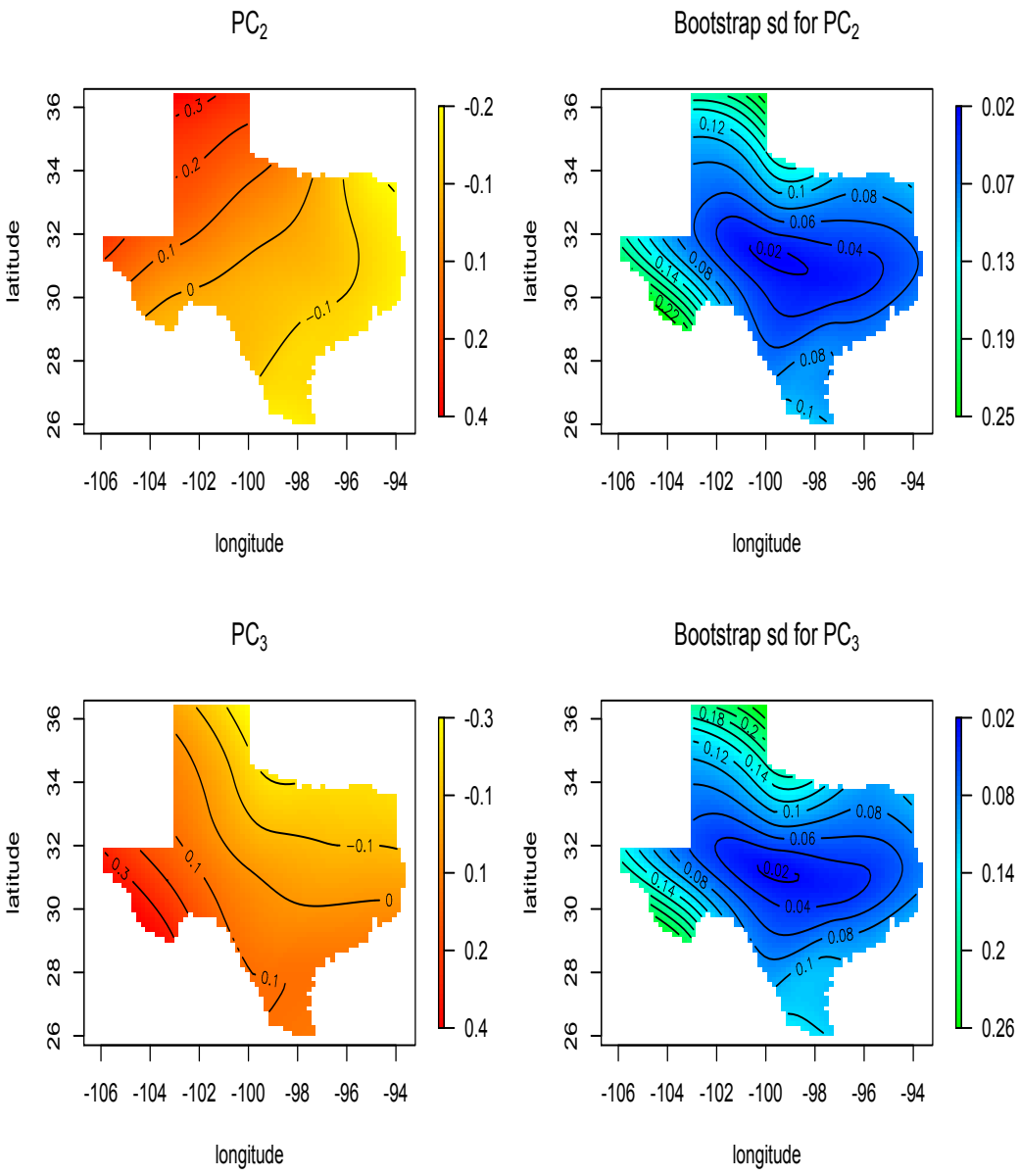

Figure 25: Estimated second and third principal component functions.

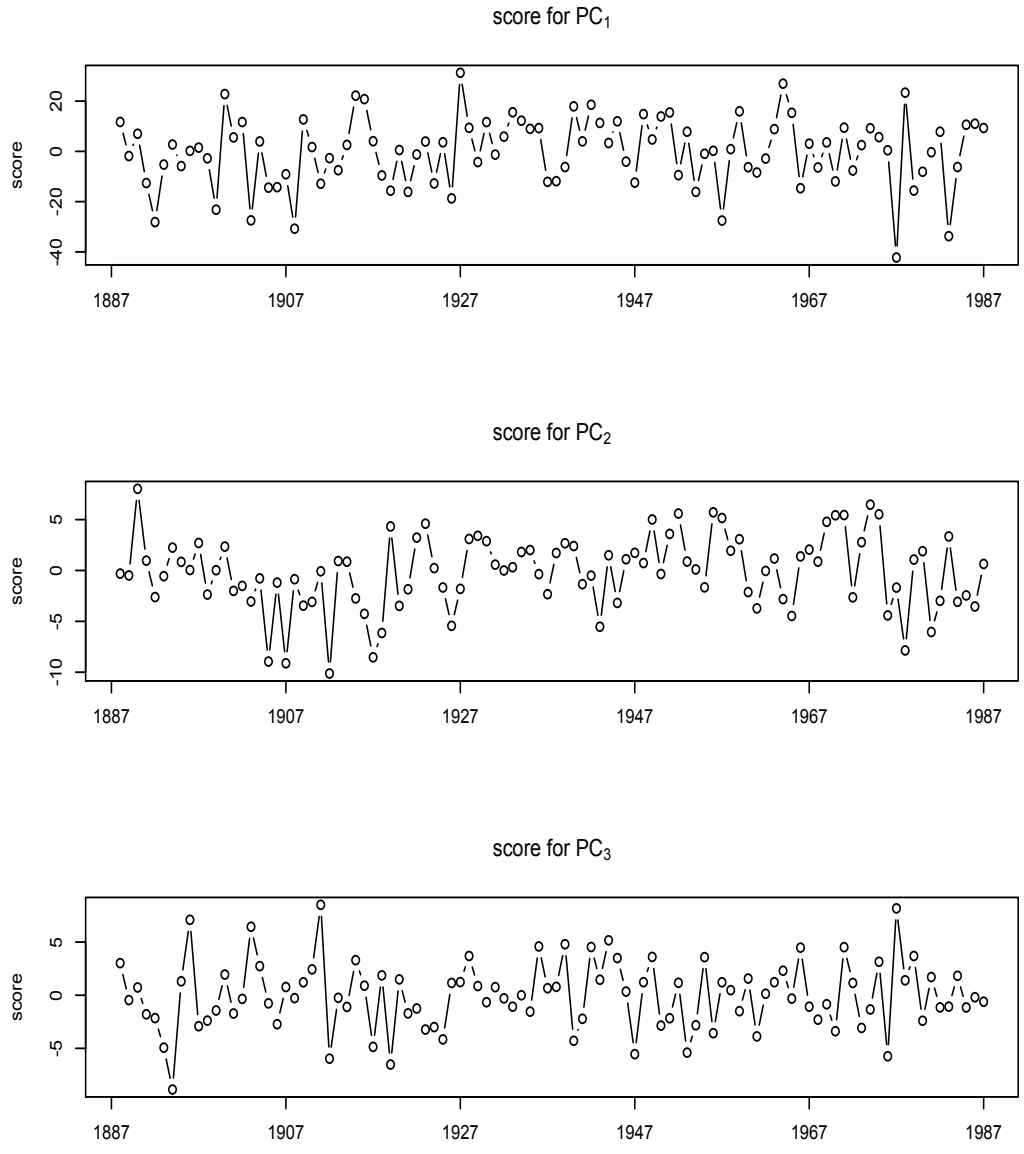

Figure 26: Time series plots for the principal component scores.

CHAPTER V

CONCLUSION AND FUTURE WORK

This work introduced bivariate B-splines and explained its usage in two spatial problems. These splines, as polynomial functions of barycentric coordinates over triangle partition of domain, solve the challenge risen when spatial domain is irregular. It is easy to evaluate and could be widely incorporated to different models. We apply these splines to estimate smooth surfaces over complicated two-dimensional domain in two different scenarios and proposed two models: the penalized spatial smoothing model and the reduced rank mixed-effects model. Simulation results are promising which demonstrate that bivariate B-splines are powerful and flexible and the models are appropriate.

In the future, we will work on the dynamic factor model which takes into account of the dynamic effects over time. We can let α_i follow a Vector Auto regressive model instead of identically and independently sampled from a normal distribution as in the reduced rank mixed-effects model. The dynamic factor model is

$$\begin{cases} z_i &= B_i\theta_\mu + B_i\Theta\alpha_i + \epsilon_i, \qquad \epsilon_i \sim N(0, \sigma^2 I) \\ \alpha_i &= \sum_{r=1}^{R} \psi_r \alpha_{i-r} + \eta_i, \qquad \eta_i \sim N(0, D) \\ E(\epsilon_i \eta_{i'}) &= 0, \qquad i, i' = 1, .., n \\ \Theta^T\Theta &= I \end{cases} \tag{5.1}$$

with parameter set $M = \left\{\sigma^2, D, \Theta, \theta_\mu, \{\psi_r\}_{r=1}^R\right\}$.

We will also continue to explore the triangulation design for the purpose of improving modeling accuracy and speeding up the computations.

REFERENCES

Craven, P., Wahba, G., 1979. Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation. Numer. Math. 31, 377–403.

Cressie, N., 1993. Statistics for Spatial Data. Wiley, New York.

Duchon, J., 1977. Splines minimizing rotation-invariant semi-norms in sobolev spaces. Lect. Notes Math. 571, 85–100.

Eilers, P., 2006. P-spline smoothing on difficult domains, in: Workshop on Statistical Modelling of Complex Systems, Munich.

Greens, P., Silverman, B., 1994. Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach. Chapman and Hall, London.

Guillas, S., Lai, M.J., 2010. Bivariate splines for spatial functional regression models. J. Nonparametr. Stat. 22, 477–497.

Hansen, M., Kooperberg, C., Sardy, S., 1998. Triogram models. J. Am. Statist. Assoc. 93, 101–119.

Horgan, G., 1999. Using wavelets for data smoothing: a simulation study. J. App. Statist. 26, 923–932.

James, G., Hastie, T., Sugar, C., 2000. Principal component models for sparse functional data. Biometrika 87, 587–602.

Koenker, R., Mizera, I., 2004. Penalized triograms: total variation regularization for bivariate smoothing. J. R. Statist. Soc. B 66, 145–163.

Lai, M.J., Schumaker, L.L., 2007. Spline Functions on Triangulations. Cambridge University Press, Oxford.

Lai, M.J., Wang, L., 2010. Bivariate penalized splines for regression. Submitted.

Peng, J., Paul, D., 2009. A geometric approach to maximum likelihood estimation of the functional principal component from sparse longitudinal data. J. Comput. Graph. Statist. 18, 995–1015.

Ramsay, T., 2002. Spline smoothing over difficult regions. J. R. Statist. Soc. B 64, 307–319.

Wahba, G., 1990. Spline Models for Observational Data. SIAM, Philadelphia.

Wand, M., Jones, M., 1995. Kernel Smoothing. Chapman and Hall, London.

Wang, H., Ranalli, M., 2007. Low-rank smoothing splines on complicated domains. Biometrics 63, 209–217.

Wood, S., 2004. Stable and efficient multiple smoothing parameter estimation for generalized additive models. J. Am. Statist. Assoc. 99, 673–686.

Wood, S., Bravington, M., Hedley, S., 2008. Soap film smoothing. J. R. Statist. Soc. B 70, 931–955.

Zhou, L., Huang, J., Carroll, R., 2008. Joint modelling of paired sparse functional data using principal components. Biometrika 95, 601–619.

VITA

Name:	Huijun Pan
Address:	Department of Statistics
	Texas A&M University
	3143 TAMU
	College Station, TX 77843-3143
Email Address:	panhuij@gmail.com
Education:	B.S., Mathematics, University of Science and Technology of China, 2006
	M.S., Statistics, Texas A&M University, 2008