creative commons
COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

Nonparametric logistic regression based on sparse triangulation over a compact domain

Seoyeon Kim

Department of Statistics

The Graduate School of Sungshin Women's University

Nonparametric logistic regression based on sparse triangulation over a compact domain

A Master's Thesis
Submitted to the
Graduate School of Sungshin Women's University

in partial fulfillment of the requirements
for the degree of
Master of Statistics

Seoyeon Kim

November, 2024

This is to certify that we have examined the
Master's Thesis of
Seoyeon Kim
Submitted to Department of Statistics

Approved as to style and content:

Thesis Advisor Kwan-Young Bak

Committee Chairman Seongoh Park

Committee Member Heewon Park

Committee Member Dongha Kim

The Graduate School of Sungshin Women's University

Abstract

Based on the investigation of logistic regression models utilizing sparse triangulation within a compact domain in $\mathbb{R}^2$, this study addresses the limited research extending the triogram model to logistic regression. A primary challenge arises from the potential instability induced by a large number of vertices, hindering the effective modeling of complex relationships. To mitigate this challenge, we propose introducing sparsity to boundary vertices of the triangulation based on the Ramer-Douglas-Peucker algorithm and employing the K-means algorithm for adaptive vertex initialization. A second order coordinate-wise descent algorithm is adopted to implement the proposed method. Validation of the proposed algorithm's stability and performance assessment are conducted using synthetic and handwritten digit data (LeCun et al., 1989). Results demonstrate the advantages of our method over existing methodologies, particularly when dealing with non-rectangular data domains.

Keywords : barycentric coordinates, coordinate descent algorithm, logistic regression, RDP algorithm, triangulation

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Multiple regression is an important cornerstone of supervised learning in statistics with countless applications. It is used to identify the relationship between multiple predictors and a response variable. The basic idea extends to the generalized linear model in which the predictor is related to the response variable via a link function when the conditional distribution of the response belongs to an exponential family with some regularity conditions; see Nelder and Wedderburn (1972). A straightforward approach in the (generalized) linear model is to use a linear predictor, which is a linear combination of predictor variables. However, this approach is often too restrictive in many practical applications, especially when the predictors are related to the mean of the response via a complicated relationship. Nonparametric regression methods have the advantage of uncovering complex relationships between the predictors and response. Examples of nonparametric methods include local polynomial regression, kernel regression, basis expansion methodology such as spline and wavelet regression, and so on. One may refer to Tsybakov (2008); Hastie et al. (2009); Wasserman (2006) for an overview of nonparametric regression.

Many nonparametric estimation methods are known to enjoy good theoretical properties at least in the asymptotic sense. However, when examining performance in finite samples, considerations of the domain can lead to significant differences. Even in problems of estimating one-dimensional functions, extensive research has been conducted on methodologies aimed at addressing the impact of domain shape on estimation accuracy. Examples include estimating functions on positive domains (Geenens, 2021; Wright and Zabin, 1994), boundary effects (Müller, 1991), and estimation across the entire real line

(Bak et al., 2021). Especially when dealing with multidimensional spaces, nonparametric methods typically require large sample sizes, making the influence of domain shape on estimation even more evident. Therefore, the development of techniques for smoothing and spatial regression applied to datasets distributed across domains with intricate geometries is a significant research topic in nonparametric estimation. To explore related issues and recent research findings, one can refer to Ferraccioli et al. (2021); Ramsay (2002); Sangalli et al. (2013); Wang and Ranalli (2007); Scott-Hayward et al. (2014), as well as the references cited therein.

Different shapes of domains impacting estimation accuracy is also observed within the generalized linear model framework. However, research into the development of estimation methodologies that reflect this phenomenon is very limited. Within the nonparametric approach to generalized linear models, the standard approach involves considering a tensor product space. In a popular approach using the regression spline model, this corresponds to constructing a tensor product spline basis for estimation. For example, Stone (1994) considered the use of polynomial splines and their tensor products in multivariate function estimation and showed that it leads to desirable statistical properties. However, a possible drawback of the tensor product spline method is that it implicitly assumes the shape of the domain. Specifically, tensor product splines assume that predictors are observed on a rectangular domain. In cases where the shape of the domain is irregular and complex, the supports of tensor product basis functions may not partition the domain appropriately. As a remedy for this, a nonparametric regression method based on triangulation, which efficiently partitions the domain using triangles, has been developed. Barycentric coordinates functions defined with respect to the resulting triangles form a basis for a space of piecewise polynomial functions over triangulation. For details concerning the triangulation and the barycentric coordinates basis functions, one may refer to Mark Hansen and Sardy (1998); Lai and Schumaker (2007); Jhong et al. (2022) and the references cited therein.

Striking a good balance between bias and variance is a fundamental issue in nonparametric estimation. In the triogram regression model, this comes down to choosing the optimal number and location of the vertices of the triangulation. Ideally, vertices should be densely placed in regions with high local fluctuation in the regression function, while regions with smooth variations should have fewer vertices. If an appropriate triangular partition can be obtained, the estimator can capture the local trends in the data without compromising the overall smoothness. To this end, Mark Hansen and Sardy (1998) considered stepwise selection of vertices with the use of the Rao (score) statistic for addition and the Wald statistic for deletion. Koenker and Mizera (2004) used total variation-type penalty in the quantile regression framework. In a similar vein, Jhong et al. (2022) introduced a sparsity-inducing roughness penalty in the mean regression problem and studied the asymptotic properties of the related estimators.

In this study, we investigate the logistic regression model based on sparse triangulation of the compact domain in $\mathbb{R}^2$. Despite the promising possibility, there is very little research extending the triogram model to logistic regression. One practical reason is that a large number of vertices can compromise the stability of the algorithm, making it challenging to model complex relationships effectively. To address this issue, we introduce sparsity to the boundary vertices of the triangulation based on the Ramer-Douglas-Peucker (RDP) algorithm (Douglas and Peucker, 1973; Ramer, 1972), and employ the K-means algorithm to initialize the interior vertices in a data-adaptive way. Additionally, we adopt the coordinate descent algorithm to enhance the stability of the implementation strategy. We validate the stability of the proposed algorithm and assess the performance of the estimates through the application of synthetic data and handwritten digit data (LeCun et al., 1989). The results illustrate that our method offers advantages compared to existing methodologies when the data is observed in a non-rectangular domain.

The rest of the paper is organized as follows. Section 2 reviews the basics of the trian-

gulation and the associated barycentric coordinates basis functions, and defines the logistic regression estimator. Section 3 describes the implementation scheme including the proposed triangulation process and coordinate descent algorithm. A numerical study including simulation and analysis of the digit data is presented in Section 4. Section 5 summarizes the findings of this study and presents discussion about possible generalizations of the results.

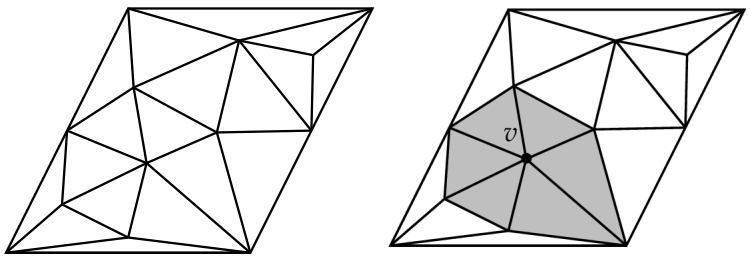
Chapter 2

Background and problem set-up

2.1 Preliminaries

This section introduces the concepts of triangulation and tent spline basis, and defines the notations used throughout the paper. For details concerning the triangulation and the corresponding spline space, one may refer to Stone (1994), Mark Hansen and Sardy (1998), Koenker and Mizera (2004), Jhong et al. (2022), and Lai and Schumaker (2007).

Let Ω be a compact region in $\mathbb{R}^2$. Let T be a triangle that is the convex hull of three points not located in one line. A collection $\triangle = \{T_1, \cdots, T_g\}$ of triangles in the plane with disjointe interiors is called a triangulation of $\Omega = \bigcup_{T\in\triangle} T$; see Figure 1(a).



(a) Triangulation $\triangle$ (b) star(v)

Figure 1: The left panel presents a triangulation, the process of dividing a given polygon into triangles. The right panel shows the star(v), a set of triangles that share a vertex v.

When dividing the data domain using triangulation, we define splines using barycentric coordinates since we are not familiar with defining splines on triangles. Barycentric coordinates represent a given point x as a weighted combination of the three vertices of a triangle and can be expressed as a ratio of the triangle's area. The sum of the ratios is always

equal to 1, and using these coordinates, any point within the triangle can be represented.

The signedArea (SA) of a triangle depends on the arrangement of its vertices. It is negative for clockwise vertex arrangements and positive for counterclockwise arrangements. The barycentric coordinate vector of $x = (x_1, x_2)$ with respect to the triangle determined by v_1, v_2, v_3 is defined as $b^{123}(x) = (b_1^{123}(x), b_2^{123}(x), b_3^{123}(x)) \in \mathbb{R}^3$, and is calculated using Cramer's rule Strang (2012) as follows

$$b_1^{123}(x) = \frac{\text{SA}(x, v_2, v_3)}{\text{SA}(v_1, v_2, v_3)},\ b_2^{123}(x) = \frac{\text{SA}(v_1, v_2, v_3)}{\text{SA}(v_1, x, v_3)},\ b_3^{123}(x) = \frac{\text{SA}(v_1, v_2, x)}{\text{SA}(v_1, v_2, v_3)}.$$

See the Appendix. The SA is a linear function of x, and the barycentric coordinates, which are expressed as ratios of areas, are also linear with respect to x; see Figure 2.

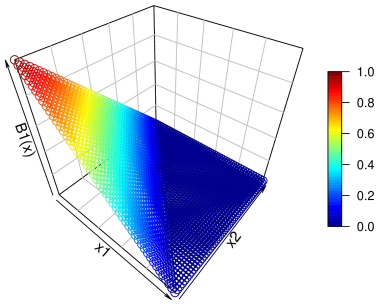
We consider the space of continuous linear splines over a given triangulation $\triangle$. The linear tent spline basis functions $\{B_j\}_{j=1}^J$ can be defined in terms of the barycentric coordinates functions of the triangles over $\triangle$ with the dimension J be the number of vertices. Specifically, given as vertex set $\{v_1, \cdots, v_J\}$ in the triangulation $\triangle$, basis functions are defined as

$$B_j(x) = \begin{cases} b_j^{T_x}(x) & \text{if} \quad x \in \text{star}(v_j) \\ 0 & \text{otherwise} \end{cases}$$

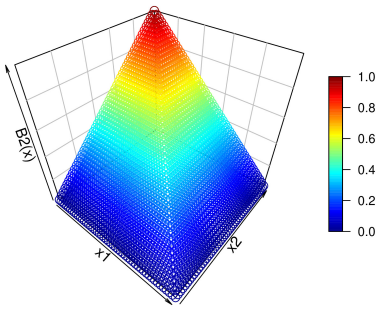
for $j = 1, \cdots, J$, where T_x is the triangle containing x and star(v_j) is the set of all triangles that share the vertex v_j. Here, $b_j^{T_x}(\cdot)$ is the barycentric coordinates function with respect to triangle T_x. The barycentric coordinates function is illustrated in Figure 2, 3.

Upon obtaining the basis functions, any continuous piecewise linear function is expressed as

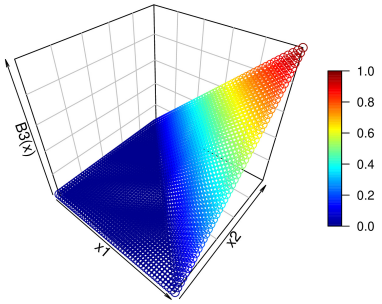
$$s_b(x) = \sum_{j=1}^{J} b_j B_j(x) \quad \text{for} \quad b \in \mathbb{R}^J. \tag{2.1}$$



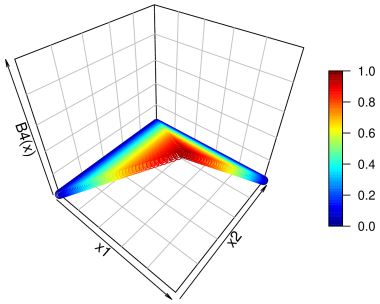
(a) $B_1(x)$



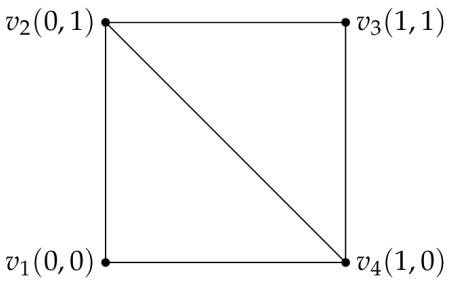
(b) $B_2(x)$



(c) $B_3(x)$



(d) $B_4(x)$



(e) Domain

Figure 2: The tent spline basis functions B_1, B_2, B_3, B_4 for triangular domain of (e) are linear to x.

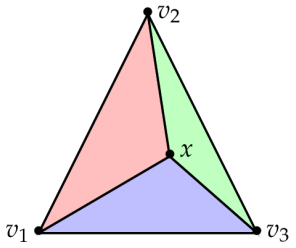


Figure 3: The values $b_1^{123}(x)$, $b_2^{123}(x)$, $b_3^{123}(x)$ are the relative areas of the green, the blue and the red triangle with respect to the area of the triangle determined by $\{v_1, v_2, v_3\}$.

The $\{B_j\}_{j=1}^J$ are linearly independent since $B_j(v_k) = 1$ for $j = k$ and $B_j(v_k) = 0$ otherwise for vertices $\{v_k\}_{k=1}^J$. By defining a basis through tent splines defined by barycentric coordinates, an effective fitting of nonparametric regression model is possible for the given arbitrary triangulation of the data domain. This can significantly improve estimation accuracy, especially when the shape of the domain is complex and irregular, and when the sample size is small, as will be illustrated in the numerical study of Section 4.

2.2 Model and estimator

Logistic regression model is introduced to deal with the binary classification problem in which the response variable Y takes on a binary value of 0 or 1. Given the predictors $X = x$, the conditional distribution of $Y|X = x$ is assumed to follow the Bernoulli distribution with probability $p(x)$. The regression function is defined as

$$\mathbb{E}[Y|X = x] = p(x) \quad \text{for} \quad x \in \Omega \subset \mathbb{R}^2.$$

The probability function p is modeled by a set of tent spline basis functions $\{B_j\}_{j=1}^J$.

For $b = (b_1, \cdots, b_J) \in \mathbb{R}^J$, we denote

$$p_b(x) = \sigma\left(\sum_{j=1}^{J} b_j B_j(x)\right),$$

where

$$\sigma(z) = \frac{1}{1 + \exp^{-z}} \text{ for } z \in \mathbb{R}$$

denotes the logistic function.

Suppose that we are given a set of data $\{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \Omega, y_i \in \{0, 1\}$. We define the likelihood function as

$$L(b) = \prod_{i=1}^{n} p_b(x_i)^{y_i}(1 - p_b(x_i))^{1-y_i}.$$

The log-likelihood function is given by

$$\ell(b) = \sum_{i=1}^{n} \left[y_i \log p_b(x_i) + (1 - y_i) \log(1 - p_b(x_i))\right],$$

where

$$\log p_b(x_i) = \log\left(1 + e^{-b^T B(x_i)}\right)$$

and

$$\log(1 - p_b(x_i)) = \log\left(\frac{e^{-b^T B(x_i)}}{1 + e^{-b^T B(x_i)}}\right).$$

Here, $B(x_i) = (B_1(x_i), \cdots, B_J(x_i)) \in \mathbb{R}^J$.

It follows that

$$\begin{aligned}\ell(b) &= \sum_{i=1}^{n} \left[y_i b^T B(x_i) - b^T B(x_i) - \log\left(1 + e^{-b^T B(x_i)}\right)\right] \\ &= \sum_{i=1}^{n} \left[y_i b^T B(x_i) - \log\left(1 + e^{b^T B(x_i)}\right)\right]. \end{aligned} \tag{2.2}$$

The maximum likelihood estimator is defined as

$$\hat{\beta} = \text{argmax}_{b \in \mathbb{R}^J} \ell(b).$$

The Sparse Triogram Probability Estimator (STriPE) of p is given by

$$\hat{p} = p_{\hat{\beta}}.$$

Chapter 3

Implementation scheme

3.1 Coordinate descent algorithm

This section summarizes the algorithm for fitting logistic regression based on a given triangulation. We first consider the standard Newton-Raphson algorithm for the logistic regression model. Let $\ell : \Omega \to \mathbb{R}$ be an objective function to minimize. A maximization problem can be reformulated as a minimization problem by taking the negative of the objective function. Thus, the negative log-likelihood $S(b)$ is defined as follows

$$S(b) = -\ell(b) = -\sum_{i=1}^{n} \left[y_i b^T B(x_i) - \log\left(1 + e^{b^T B(x_i)}\right)\right].$$

The gradient vector is given by

$$\nabla S(b) = \sum_{i=1}^{n} \left[\sigma\left(b^T B(x_i)\right) - y_i\right] B(x_i). \tag{3.1}$$

The Hessian matrix is given by

$$\nabla^2 S(b) = \sum_{i=1}^{n} \sigma\left(b^T B(x_i)\right)\left(1 - \sigma(b^T B(x_i))\right) B(x_i)B(x_i)^T. \tag{3.2}$$

The iterative update formula of the pure Newton-Raphson algorithm is given by

$$\tilde{\beta}_{k+1} = \tilde{\beta}_k - \left(\nabla^2 S(\tilde{\beta}_k)\right)^{-1} \nabla S(\tilde{\beta}_k) \quad \text{for} \quad k = 0, 1, \cdots .$$

The algorithm summarized above is outlined in Mark Hansen and Sardy (1998). How-

ever, this algorithm may exhibit some instability in practical applications. As the dimension of the spline space increases, the area of the triangle narrows, leading to convergence issues in the algorithm. The poor condition number of the Gram matrix has led to a decrease in numerical stability. In response to this issue, we consider the coordinate descent algorithm along with the initialization strategy to be described in the next subsection. We use the Taylor second-order approximation of the univariate objective function and employ it to obtain an update formula.

We now consider the coordinate-wise optimization algorithm. Since initial value of coefficients be $\tilde{\beta} = (\tilde{\beta}_0, \cdots, \tilde{\beta}_0)$, we have the univariate objective function for the jth coefficient as follows,

$$S_j(b_j) = S(\tilde{\beta}_0, \cdots, \tilde{\beta}_{j-1}, b_j, \tilde{\beta}_{j+1}, \cdots, \tilde{\beta}_p).$$

The univariate objective function is approximated

$$q_j(b_j) = -\ell(\tilde{\beta}) + S_j^{'}(\tilde{\beta}_j)(b_j - \tilde{\beta}_j) + \frac{1}{2} S_j^{''}(\tilde{\beta}_j)(b_j - \tilde{\beta}_j)^2.$$

To obtain the closed-form solution for the minimizer, we differentiate the above expression as follows,

$$q_j^{'}(b_j) = S_j^{'}(\tilde{\beta}_j) + S_j^{''}(\tilde{\beta}_j)(b_j - \tilde{\beta}).$$

Setting the equation equal to zero, we derive the minimizer

$$b_j = \tilde{\beta}_j - \frac{S_j^{'}(\tilde{\beta}_j)}{S_j^{''}(\tilde{\beta}_j)}.$$

The jth element of $\nabla S(\tilde{\beta})$ and the (j,j)th element of $\nabla^2 S(\tilde{\beta})$ represent $S_j^{'}(\tilde{\beta}_j)$ and $S_j^{''}(\tilde{\beta}_j)$

respectively. Therefore, we obtain the following update formula

$$\tilde{\beta}_j \leftarrow \tilde{\beta}_j - \eta \frac{\nabla S(\tilde{\beta})_j}{\nabla^2 S(\tilde{\beta})_{jj}}, \tag{3.3}$$

where η is an appropriately chosen step size.

3.2 Initial triangulation with Ramer–Douglas –Peucker and K-means algorithm

The RDP algorithm (Douglas and Peucker, 1973; Ramer, 1972), an effective method for polyline simplification, plays a crucial role in the reduction of vertices in a given linear path or polygon while preserving its overall shape. RDP algorithm operates on the principle of recursive division of the line, starting with an ordered set of points or lines and a specified distance threshold, ε, greater than zero. At the outset, the algorithm encompasses all points between the initial and final points of the curve, designating these terminal points as retained. It then identifies the points most distant from the line segment defined by these terminal points. This point, being the furthest on the curve from the approximating line segment, is critical for assessing the necessity of vertex retention. If this point's distance from the line segment is less than ε, it implies that the curve can be simplified without significantly deviating from the original shape by discarding any points not marked for retention.

The numerical instability of the triogram method arises primarily when the number of vertices is large. There are several approaches to mitigate this issue and improve estimation accuracy at the same time. Koenker and Mizera (2004) advocated for ℓ_2 regularization, while Jhong et al. (2022) introduced a total variation type penalty to induce sparsity. Mark Hansen and Sardy (1998) proposed using Rao-Wald statistics for stepwise selection of vertices. However, these approaches primarily address regularization and adaptation at interior vertices, rather than resolving the instability stemming from an increase in bound-

ary vertices. Therefore, particularly in logistic regression where algorithms tend to exhibit instability, having a large number of vertices does not ensure sufficient stability.

As a remedy, we adopt the RDP algorithm to impose sparsity on the boundary vertices. Previous studies have utilized the convex hull algorithm proposed by Eddy (1977) for initial triangulation; see, for example, Jhong et al. (2022) and Toussaint and Avis (1982). This algorithm is employed to find the minimum convex polygon for a given set of points, which often results in the generation of a large number of somewhat redundant boundary vertices. This dense representation of the boundary can cause instability of the optimization algorithm. If a large number of vertices are selected using the convex hull algorithm, adjusting the number of internal vertices does not significantly improve numerical performance. This is where the RDP algorithm has an advantage because it allows for a sparse representation of the boundary.

Since we are dealing with two-dimensional data, researchers can visually assess whether sparse triangulation is suitable for the given data. It seems sufficient to evaluate the adequacy by comparing sparse representations with the visualization of the data's shape and convex hull. Although the sparsity parameter ε for the RDP algorithm can be tuned based on standard validation methods, we find that the choice of ε does not have a significant impact on practical performance as long as it stays within a reasonable range. We recommend choosing ε to be a value in $\{0.01, 0.05, 0.1\}$ with some validation techniques if required.

3.3 Selection of interior centroids with K-means algorithm

We adopt the K-means algorithm (MacQueen, 1967) to determine the number and location of interior vertices in a data-adaptive way. The K-means algorithm groups data into clusters and enables the use of each group's center as interior vertices in the triangulation

process. This ensures that triangulation can be formed in areas with a large amount of data and significant variation.

In the triangulation process, K is the tuning parameter of the K-means algorithm, which represents the number of interior vertices. The choice of K has a significant impact on the performance of the proposed method. We choose optimal K via the following Bayesian information criterion (BIC) to ensure that it is an appropriate value based on the data:

$$\text{BIC} = J \log(n) - 2\ell(b),$$

where J represents total number of vertices and $\ell(b)$ represents (2.2). Here, J is determined by sum of K and the number of boundary vertices.

Through numerical experiments, we confirmed that the triangulation strategy, combined with the coordinate descent algorithm presented in Section 3.1, significantly enhances numerical stability and estimation accuracy. The proposed algorithm was implemented using `R` software, employing the `grDevices`, `stats`, and `RDP` packages. The overall implementation algorithm is summarized in Algorithm 1 below.

Algorithm 1 Implementation algorithm for STriPE

Input: x : predictors $\in \mathbb{R}^{n\times 2}$,
ε : threshold,
K : number of interior vertices,
J : total number of vertices,
η : step size,
δ : tolerance,
max_iter : maximum iterations,
Function: Use function: `grDevices.chull()`,
`stats.kmeans()`,
`RDP.RamerDouglasPeucker()`
Initial triangulation:
Compute the boundary vertices:
$vertex\_chull$ = `chull`(x)
$vertex\_simplified$ = `RamerDouglasPeucker`($vertex\_chull[,1], vertex\_chull[,2], \varepsilon$)
Choose K using BIC statistic
Compute the interior vertices:
$interior\_centroids$ = `kmeans`(x, K)
Combine boundary vertices and interior centroids
Compute the design matrix G $\in \mathbb{R}^{n\times J}$
Coefficient initialization: $\tilde{\beta} = \tilde{\beta}_{\text{old}} = (1, \cdots, 1) \in \mathbb{R}^J$
while diff $> \delta$ and iteration $< max\_iter$ **do**
  $\tilde{\beta}_{\text{old}} = \tilde{\beta}$
  for $j = 1$ *to* J **do**
    Compute the gradient $\nabla S(\tilde{\beta}_j)$ using (3.1)
    Compute the Hessian $\nabla^2 S(\tilde{\beta}_j)$ using (3.2)
    Update $\tilde{\beta}_j \leftarrow \tilde{\beta}_j - \eta \frac{\nabla S(\tilde{\beta})_j}{\nabla^2 S(\tilde{\beta})_{jj}}$ using (3.3)
  end for
  diff $= \|\tilde{\beta} - \tilde{\beta}_{\text{old}}\|$
end while
Output: $\tilde{\beta}$

Chapter 4

Numerical studies

4.1 Simulation study

This section illustrates the advantages of the proposed method based on simulation studies. We consider three probability functions defined on non-rectangular domains. Each function is defined as a logistic transformation of a linear combination of basis functions defined on a triangulation obtained by adding 1, 3, and 2 interior vertices to pre-specified boundary vertices, respectively. The contour plots of the example functions and the domain areas can be seen in Figure 4.

We randomly generated $x_1, \ldots, x_n$ in this domain and applied the proposed triangulation strategy. The ε parameter of the RDP algorithm is adjusted to 0.05 for all three examples. In Figure 5, three plots in the top row represent the triangulation obtained from the standard convex hull algorithm. It is seen that each example has a total of $21, 24$, and 14 vertices, respectively. Although the number of interior vertices is determined as $K = 1, 3$, and 2, respectively, for three examples, a dense basis representation is obtained because some of the boundary vertices are redundant, which causes a detrimental effect on the stability of the algorithm. On the other hand, the plots in the bottom row depict the result of applying the RDP algorithm, yielding a sparse representation through $7, 10$, and 7 vertices with the regions closely resembling the true function's domain. This significantly stabilizes the optimization algorithm.

We consider sample sizes of $n = 100, 200, 300, 400$ and 500. Through 50 replicates, we record the mean squared error (MSE) obtained by making predictions on randomly

selected points. The MSE is defined as

$$\text{MSE}(\hat{p}) = \frac{1}{1000}\sum_{s=1}^{1000}(p(t_s) - \hat{p}(t_s))^2,$$

where t_s represents randomly selected points, independent of the training data, in Ω. To illustrate the performance, we compared the proposed method with the thin plate regression (Wood, 2003), cubic spline (De Boor, 1978) and kernel logistic regression (Zhu and Hastie, 2005). The selection of knots for implementing the cubic spline (CS) method was based on the quantile values of observations, while the hyperparameters of the kernel logistic regression (KLR) method were determined by the cross-validation. The average MSE values and standard errors for the STriPE method, thin plate spline (TPS) method, CS method, and KLR are summarized in Table 1, with the smallest MSE in boldface. Numerical results confirm that the proposed method exhibits superior performance. While the difference in MSE values between the proposed method and other methods tends to decrease as the sample size increases, we still observe that our method outperforms others, with particularly significant differences occurring in small samples.

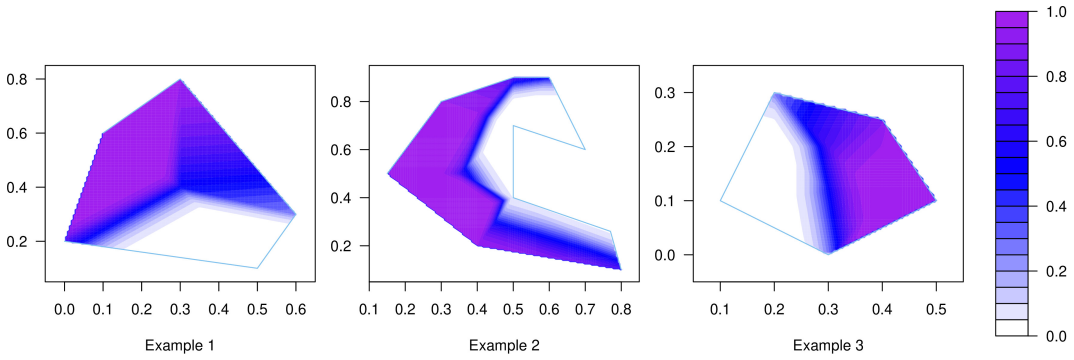


Figure 4: The plots represent the contour plots of three example functions and the domain of those functions.

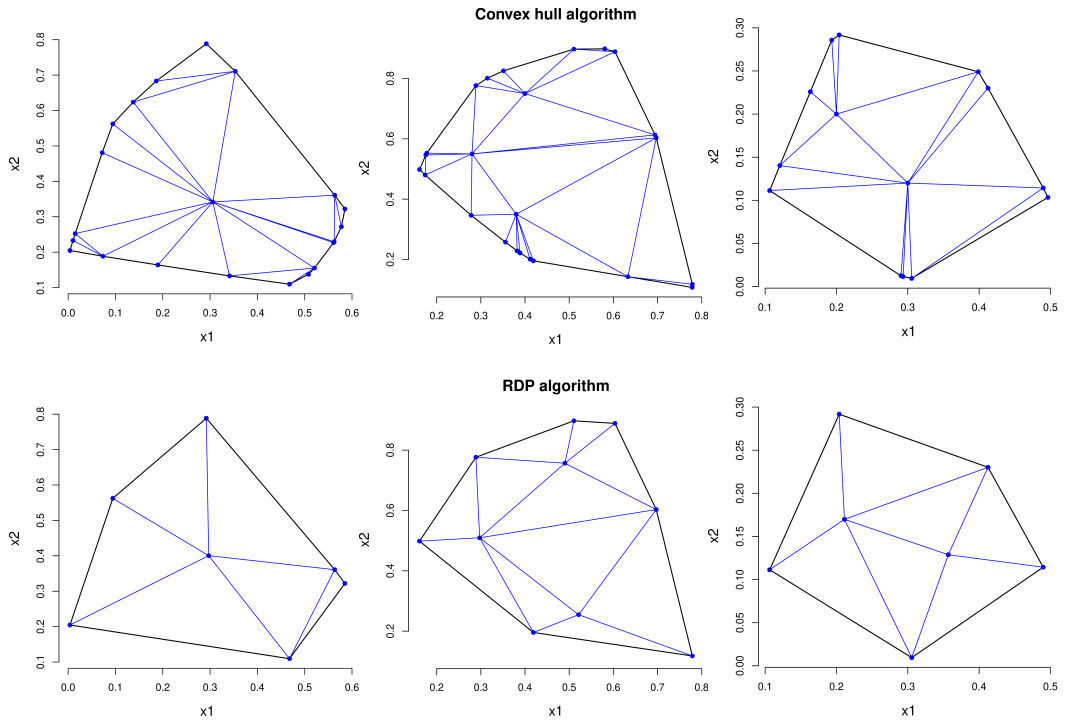
Example 1				
Sample Size	STriPE(se)	TPS(se)	CS(se)	KLR(se)
$n = 100$	$\mathbf{0.0160(0.0013)}$	$0.0345(0.0048)$	$0.0355(0.0053)$	$0.0425(0.0059)$
$n = 200$	$\mathbf{0.0061(0.0004)}$	$0.0305(0.0011)$	$0.0116(0.0012)$	$0.0281(0.0028)$
$n = 300$	$\mathbf{0.0051(0.0004)}$	$0.0073(0.0004)$	$0.0069(0.0004)$	$0.0221(0.0015)$
$n = 400$	$\mathbf{0.0036(0.0002)}$	$0.0057(0.0002)$	$0.0056(0.0002)$	$0.0198(0.0011)$
$n = 500$	$\mathbf{0.0032(0.0002)}$	$0.0046(0.0002)$	$0.0043(0.0002)$	$0.0177(0.0014)$
Example 2				
Sample Size	STriPE(se)	TPS(se)	CS(se)	KLR(se)
$n = 100$	$\mathbf{0.0247(0.0014)}$	$0.0377(0.0043)$	$0.0400(0.0050)$	$0.0916(0.0020)$
$n = 200$	$\mathbf{0.0129(0.0005)}$	$0.0160(0.0009)$	$0.0150(0.0007)$	$0.0890(0.0012)$
$n = 300$	$\mathbf{0.0112(0.0004)}$	$0.0121(0.0005)$	$0.0124(0.0005)$	$0.0900(0.0011)$
$n = 400$	$\mathbf{0.0090(0.0003)}$	$0.0098(0.0003)$	$0.0099(0.0003)$	$0.0877(0.0009)$
$n = 500$	$\mathbf{0.0087(0.0002)}$	$0.0092(0.0004)$	$0.0091(0.0003)$	$0.0891(0.0010)$
Example 3				
Sample Size	STriPE(se)	TPS(se)	CS(se)	KLR(se)
$n = 100$	$\mathbf{0.0122(0.0011)}$	$0.0287(0.0045)$	$0.0271(0.0043)$	$0.0180(0.0007)$
$n = 200$	$\mathbf{0.0055(0.0004)}$	$0.0062(0.0006)$	$0.0067(0.0009)$	$0.0154(0.0003)$
$n = 300$	$\mathbf{0.0047(0.0003)}$	$0.0047(0.0004)$	$0.0050(0.0005)$	$0.0146(0.0002)$
$n = 400$	$\mathbf{0.0030(0.0002)}$	$0.0033(0.0003)$	$0.0034(0.0003)$	$0.0143(0.0001)$
$n = 500$	$\mathbf{0.0026(0.0001)}$	$\mathbf{0.0026(0.0001)}$	$0.0026(0.0002)$	$0.0141(0.0001)$

Table 1: The average MSE values and the standard errors (in parentheses) of STriPE, TPS, CS, and KLR with 50 replicated simulations. In bold, best row-wise.

4.2 Digit data analysis

We apply our method to analyze the handwritten zip code database presented in LeCun et al. (1989). We use features "intensity" and "symmetry" for this analysis. Intensity represents the count of black pixels in the images, and symmetry represents how closely a character resembles its specular image. We choose digits 7 and 8. Out of a total of 1187 training observations, we exclude outliers, resulting in the use of 1184 training observations. Test observations consist of 313. The class label y is set to 0 for digit 7 and 1 for digit 8. Intensity is considered as the predictor variable x_1, and symmetry is used as the predictor variable x_2.

In the left panel of Figure 6, we observe that the data is not distributed in a rectangular shape. We obtained an initial triangulation using the proposed initialization strategy.



(a) Example 1 (b) Example 2 (c) Example 3

Figure 5: Plot of the example 1, 2, and 3 via initial triangulation based on the vertices determined by the convex hull algorithm (top) and the RDP algorithm (bottom).

The user-defined parameter ε for the RDP algorithm was tuned to 0.01, and the number of centroids in the K-means algorithm was tuned to 3. This approach determines 10 boundary vertices and 3 interior vertices. In the left panel of Figure 6, the red dots and blue dots represent Digit 8 and Digit 7, respectively, while the black diamonds represent the initial vertices. We can observe that the three interior vertices are positioned at the boundary of the data, indicating their importance in forming the decision boundary. The right panel of Figure 6 visualizes the triangulation obtained using the given initial vertices.

For $x = (x_1, x_2) \in \Omega$, we compute $\hat{p}(x)$. If $\hat{p}(x)$ is greater than 0.5, we predict y as 1 and otherwise, we predict y as 0. The decision boundary where $\hat{p}(x)$ becomes 0.5 is depicted as a black line in both plots in Figure 7. The calculated in-sample accuracy from

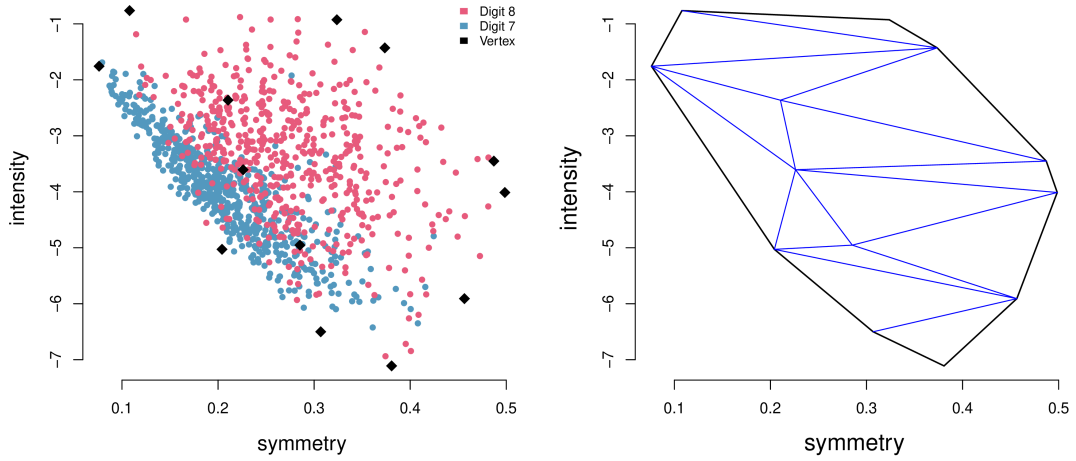


Figure 6: Left panel presents a plot of the training data and the initial vertices. The red dots and blue dots represent Digit 8 and Digit 7, respectively, while the black diamonds represent the initial vertices. The right plot shows the initial triangulation determined by the initial vertices.

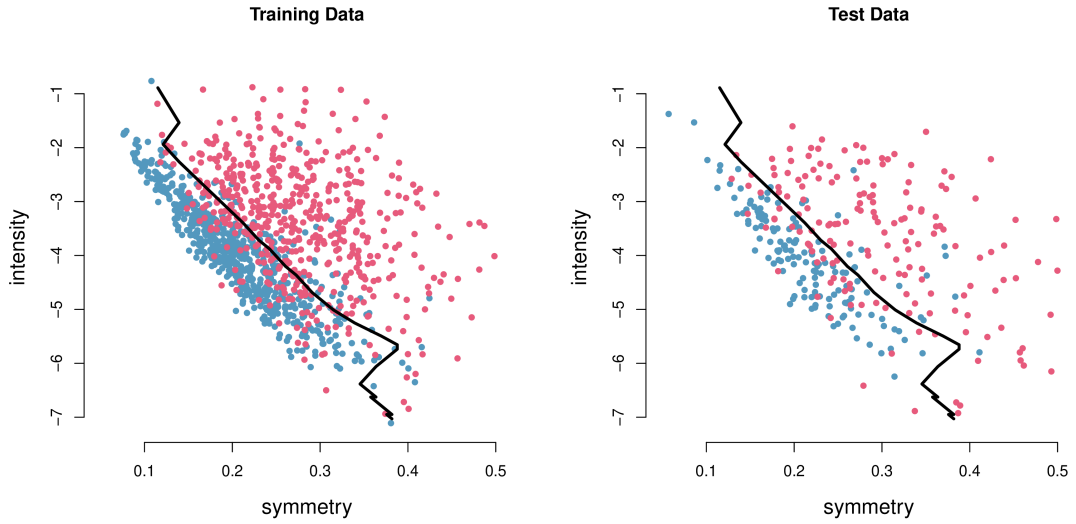


Figure 7: Plot of the training data (left) and test data (right) along with the decision boundary (black solid line).

the training data is 0.8472, and the out-of-sample accuracy computed from the test data is 0.8233. For comparison, fitting a logistic model using thin plate spline, cubic spline and kernel logistic regression resulted in an out-of-sample accuracy of $0.81, 0.8066, 0.7866$, respectively. Figure 7 visualizes the training and test data along with the decision boundary. The result implies that there is a tendency for intensity to decrease as symmetry increases for both digit 7 and digit 8. Symmetry is a factor that determines the classification of digit data. Digit 8 is greater than digit 7 in terms of the symmetry, and the region of digit 8 is formed on the right side of the decision boundary.

Chapter 5

Conclusion

This paper proposed a multivariate nonparametric logistic regression method based on the sparse triangulation within a compact domain in $\mathbb{R}^2$. Sparsity on the boundary vertices of the triangulation was imposed by applying the RDP algorithm to the initial vertices obtained by the convex hull algorithm. The complexity of estimation methods was controlled by the ε of the RDP algorithm and the number of centroids K in the K-means algorithm used to obtain the data-dependent interior vertices. This strategy combined with the coordinate descent algorithm helps stabilize the convergence property of the implementation algorithm. The performance of the proposed method was investigated using the synthetic and handwritten digit data. Results illustrate that the proposed method outperforms existing methods when the data is observed on non-rectangular domains.

The results of this paper are expected to provide a foundation for further research. They can be generalized and extended in a few ways. First, we can extend the method to the case of p-dimensional predictors where $p > 2$. To our knowledge, there has been no research applying spline methodology based on barycentric coordinates to nonparametric function estimation problems for $p > 2$. It is expected that by efficiently partitioning the domain using simplices and the associated spline basis, one can significantly improve the efficiency of nonparametric function estimation methods.

Second, we can consider combining the proposed methodology with sparsity-inducing penalization. In Jhong et al. (2022), a method for automatically selecting the number of vertices in triangulation-based regression problems using a total-variation type penalty was proposed. Building on this, we can develop a penalization method within the generalized

linear model framework for choosing the number of vertices in a data-adaptive way after placing a sufficient number of vertices to ensure the flexibility of the model.

Appendix

For a triangle $\triangle^{123}$ formed by vertices v_1, v_2, v_3, as shown in Figure 8, a simple method to calculate the area of the triangle is as follows.

$$\begin{aligned}\triangle^{123} &= (v_{21} - v_{11})(v_{22} - v_{32}) - \frac{1}{2}(v_{21} - v_{11})(v_{22} - v_{12}) \\ &\quad - \frac{1}{2}(v_{22} - v_{32})(v_{21} - v_{31}) - \frac{1}{2}(v_{31} - v_{11})(v_{12} - v_{32}) \\ &= \frac{1}{2}\{v_{11}(v_{32} - v_{22}) + v_{21}(v_{12} - v_{32}) + v_{31}(v_{22} - v_{12})\},\end{aligned}$$

where $v_1 = (v_{11}, v_{12}), v_2 = (v_{21}, v_{22}), v_3 = (v_{31}, v_{32})$.

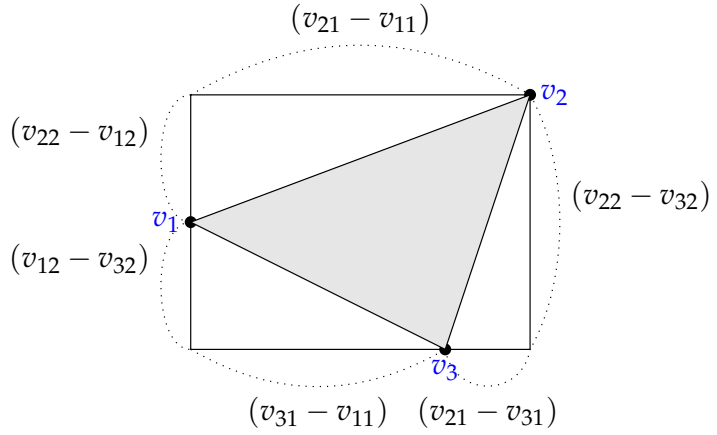


Figure 8: The gray triangle $\triangle^{123}$ formed by vertices v_1, v_2 and v_3 in $\mathbb{R}^2$.

Another method to calculate the area of a triangle is by using Cramer's rule. The equation $Vb = x$, consisting of a matrix formed by the vertices, a solution vector composed of the barycentric coordinates, and a vector x representing the point at which the barycentric

coordinates are calculated, is as follows

$$\begin{bmatrix} 1 & 1 & 1 \\ v_{11} & v_{21} & v_{31} \\ v_{12} & v_{22} & v_{32} \end{bmatrix} \begin{bmatrix} b_1^{123} \\ b_2^{123} \\ b_3^{123} \end{bmatrix} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix},$$

where $b_1^{123} + b_2^{123} + b_3^{123} = 1$.

Applying Cramer's rule,

$$\det(V) = \begin{vmatrix} 1 & 1 & 1 \\ v_{11} & v_{21} & v_{31} \\ v_{12} & v_{22} & v_{32} \end{vmatrix}, \ \det(V_1) = \begin{vmatrix} 1 & 1 & 1 \\ x_1 & v_{21} & v_{31} \\ x_2 & v_{22} & v_{32} \end{vmatrix},$$

$$\det(V_2) = \begin{vmatrix} 1 & 1 & 1 \\ v_{11} & x_1 & v_{31} \\ v_{12} & x_2 & v_{32} \end{vmatrix}, \ \det(V_3) = \begin{vmatrix} 1 & 1 & 1 \\ v_{11} & v_{21} & x_1 \\ v_{12} & v_{22} & x_2 \end{vmatrix}.$$

Depending on the arrangement of the vertices, the area of the triangle has a sign, so it can be expressed as $\det(V) = \mathrm{SA}(v_1, v_2, v_3),\ \det(V_1) = \mathrm{SA}(x, v_2, v_3)$, $\det(V_2) = \mathrm{SA}(v_1, x, v_3)$, and $\det(V_3) = \mathrm{SA}(v_1, v_2, x)$. It can be confirmed that the results obtained using Cramer's rule and the simple area calculation for the triangle are the same. Since the SA is linear with respect to x, the barycentric coordinates, which are expressed as ratios of the SA, are also linear with respect to x.

References

Bak, K.-Y., Jhong, J.-H., Lee, J., Shin, J.-K., and Koo, J.-Y. (2021). Penalized logspline density estimation using total variation penalty. *Computational statistics & data analysis*, 153:107060.

De Boor, C. (1978). *A practical guide to splines*, volume 27. springer-verlag New York.

Douglas, D. H. and Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization*, 10(2):112–122.

Eddy, W. F. (1977). A new convex hull algorithm for planar sets. *ACM Transactions on Mathematical Software (TOMS)*, 3(4):398–403.

Ferraccioli, F., Arnone, E., Finos, L., Ramsay, J. O., and Sangalli, L. M. (2021). Nonparametric density estimation over complicated domains. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 83(2):346–368.

Geenens, G. (2021). Mellin–meijer kernel density estimation on r+. *Annals of the Institute of Statistical Mathematics*, 73(5):953–977.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Jhong, J.-H., Bak, K.-Y., and Koo, J.-Y. (2022). Penalized polygram regression. *Journal of the Korean Statistical Society*, 51(4):1161–1192.

Koenker, R. and Mizera, I. (2004). Penalized triograms: Total variation regularization for bivariate smoothing. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(1):145–163.

Lai, M.-J. and Schumaker, L. L. (2007). *Spline functions on triangulations*. Cambridge University Press.

Lai, M.-J. and Wang, L. (2013). Bivariate penalized splines for regression. *Statistica sinica*, pages 1399–1417.

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., and Jackel, L. (1989). Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

Mark Hansen, C. K. and Sardy, S. (1998). Triogram models. *Journal of the American Statistical Association*, 93(441):101–119.

Müller, H.-G. (1991). Smooth optimum kernel estimators near endpoints. *Biometrika*, 78(3):521–530.

Nelder, J. A. and Wedderburn, R. W. (1972). Generalized linear models. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 135(3):370–384.

Ramer, U. (1972). An iterative procedure for the polygonal approximation of plane curves. *Computer graphics and image processing*, 1(3):244–256.

Ramsay, T. (2002). Spline smoothing over difficult regions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 64(2):307–319.

Sangalli, L. M., Ramsay, J. O., and Ramsay, T. O. (2013). Spatial spline regression models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 75(4):681–703.

Scott-Hayward, L. A. S., MacKenzie, M. L., Donovan, C. R., Walker, C., and Ashe, E. (2014). Complex region spatial smoother (cress). *Journal of Computational and Graphical Statistics*, 23(2):340–360.

Stone, C. J. (1994). The use of polynomial splines and their tensor products in multivariate function estimation. *The Annals of Statistics*, 22(1):118–171.

Strang, G. (2012). *Linear algebra and its applications*.

Toussaint, G. T. and Avis, D. (1982). On a convex hull algorithm for polygons and its application to triangulation problems. *Pattern Recognition*, 15(1):23–29.

Tsybakov, A. B. (2008). *Introduction to Nonparametric Estimation*. Springer Publishing Company, Incorporated, 1st edition.

Wang, H. and Ranalli, M. G. (2007). Low-rank smoothing splines on complicated domains. *Biometrics*, 63(1):209–217.

Wang, R., Ramos, D., and Fierrez, J. (2012). Improving radial triangulation-based forensic palmprint recognition according to point pattern comparison by relaxation. In *2012 5th IAPR International Conference on Biometrics (ICB)*, pages 427–432. IEEE.

Wasserman, L. (2006). *All of nonparametric statistics*. Springer Science & Business Media.

Wood, S. N. (2003). Thin plate regression splines. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 65(1):95–114.

Wright, G. A. and Zabin, S. M. (1994). Nonparametric density estimation for classes of positive random variables. *IEEE transactions on information theory*, 40(5):1513–1535.

Zhu, J. and Hastie, T. (2005). Kernel logistic regression and the import vector machine. *Journal of Computational and Graphical Statistics*, 14(1):185–205.

국문 초록

컴팩트 도메인의 희소 삼각분할 기반 비모수 로지스틱 회귀 모형

성신여자대학교
대학원
통계학과
김서연

본 연구는 $\mathbb{R}^2$의 컴팩트 도메인의 희소 삼각분할을 기반으로 한 로지스틱 회귀 모형을 연구하여 삼각화 모델을 로지스틱 회귀 모형으로 확장하고 있습니다. 다수의 꼭짓점으로 인해 유발되는 잠재적 불안정성으로 설명변수와 반응변수 간의 복잡한 관계를 효과적으로 모델링하는 데 어려움이 있습니다. 이 문제를 해결하기 위해, RDP(Ramer-Douglas-Peucker) 알고리즘을 기반으로 삼각분할의 바깥쪽 경계 꼭짓점에 희소성을 도입하고, K-means 알고리즘을 이용하여 데이터에 맞게 적응적으로 삼각분할의 내부 꼭짓점을 선택하는 초기 삼각화 전략을 제안합니다. 제안된 방법을 구현하기 위해 2차 좌표별 하강 알고리즘을 채택하였습니다. 합성 데이터와 손글씨 숫자 데이터 LeCun et al. (1989)를 사용하여 제안된 알고리즘의 안정성 검증 및 성능을 평가하였습니다. 결과는 직사각형 형태가 아닌 데이터 도메인 또는 복잡한 데이터 도메인을 다룰 때 기존 방법들에 비해 제안한 방법론의 장점을 입증하였습니다.

핵심용어 : 무게중심 좌표, 좌표 하강 알고리즘, 로지스틱 회귀, RDP 알고리즘, 삼각화