TR-No. 19-07, Hiroshima Statistical Research Group, 1–28

Estimation for Spatial Effects by Using the Fused Lasso

Mineaki Ohishi[1]*, Keisuke Fukui[2], Kensuke Okamura[3], Yoshimichi Itoh[3] and Hirokazu Yanagihara[1,3]

[1]Department of Mathematics, Graduate School of Science, Hiroshima University
1-3-1 Kagamiyama, Higashi-Hiroshima, Hiroshima 739-8526, Japan
[2]Department of Medical Statistics, Research and Development Center, Osaka Medical College
2-7 Daigaku-machi, Takatsuki, Osaka, 569-8686, Japan
[3]Tokyo Kantei Co., Ltd.
Meguro Nishiguchi Bldg., 8F, 2-24-15 Kami-Osaki, Shinagawa, Tokyo, 141-0021, Japan

Abstract

When we deal with data that depends on spaces, it is important to clarify effects from spaces called spatial effects. In this paper, we consider estimation for spatial effects. Our idea is that we split the space subjected to analysis into smaller spaces and estimate spatial effects with respect to those smaller spaces, that is to say, we evaluate spatial effects discretely. Since split small spaces have adjacent relationships, we take the join of adjacent spaces into account in the estimation by using the fused Lasso. Then, if spatial effects of adjacent spaces are equal, the corresponding spaces are joined. Because the estimation method can perform clustering by joining adjacent spaces, we can expect that it offers additional value as secondary use. For the purposes of efficient and accurate estimation even if a large sample data, we provide an update equation of the coordinate descent algorithm in closed form.

(Last Modified: September 30, 2019)

Key words: Clustering, Coordinate descent, Fused Lasso, Generalized Lasso, Spatial effect, Spatial data.

*Corresponding author
E-mail address: mineaki-ohishi@hiroshima-u.ac.jp (Mineaki Ohishi)

1. Introduction

In this paper, we deal with spatial data that the sample depends on space. That is, we consider pairs of a response variable $y_{j,i}$ and a vector $\boldsymbol{x}_{j,i}$ of explanatory variables for the ith sample ($i \in \{1, \dots, n_j\}$) in the space j ($\in \{1, \dots, m\}$). When we use such data, it is important to unravel spatial effects as described by Anselin (1990) and Anselin & Getis (1992). For example, it is widely known that the rent or price of apartments is influenced by their address and that ecological surveys of flora and fauna are influenced by the observation points.

Spatial effects can be estimated by a geographically weighted regression (GWR) proposed by Brunsdon *et al.* (1996). The GWR is a weighted estimation method that estimates at each sample point and the estimator can be obtained in closed form. In real data analysis, the GWR has been widely applied (e.g., Löchl & Axhausen, 2010; Nagamura & Kaneda, 2015). Although the GWR has advantages, there are also important shortcomings, not least. The first disadvantage of the GWR is that it may be intractable in large sample data. To calculate weights used for local estimation, a distance matrix needs to be calculated that has distances between each sample point as the elements. Since the size of the distance matrix depends on the sample size, the calculation is computationally onerous in large sample data. The second drawback is that placement and shape of kernel both need to be optimized. The weights are calculated by the kernel with distance as the argument. Hence, results of estimation and contour line depend on the placement and the shape. The third problem is that the GWR is not appropriate where data are markedly sparse or unbalanced. Since the estimation results depend on the sample points, the GWR cannot estimate well for such data. Fourth and finally, it is hard to use the GWR for prediction problem. Since the estimation results are obtained at each sample point, to obtain predictive values for new observed points, we must locally estimate again at the points. A predictive model that must estimate parameters every time new observed points are acquired is cumbersome to use and inefficient in practical aspect.

To overcome the above shortcomings of the GWR, we propose an estimation method that discretely evaluates spatial effects by splitting the space subjected to analysis. Specifically, since the split small spaces have adjacent relationships, we estimate spatial effects by using the idea of the fused Lasso proposed by Tibshirani *et al.* (2005). A key merit of discrete evaluation is that calculation of the distance matrix and the kernel are unnecessary. Therefore, estimation in large sample data becomes straightforward. Since adjacent relationships of small spaces are used rather than the sample points, this method is not vulnerable to markedly sparse or unbalanced issues. Moreover, by using the fused Lasso, we can obtain clustering of spatial effects by joining the adjacent small spaces. The clustering results have the potential to be used in business practice, e.g., area marketing. Furthermore, the proposed method is amenable to prediction problems. The reason is when we get a new observed point, the predictive value can be derived by using the estimate for the space included the point.

Specifically, we propose the spatial-fused Lasso, which is an extension of the fused Lasso, for adjacent relationships of spaces. The optimization problem of the spatial-fused Lasso can come down to the optimization problem of the generalized Lasso (Tibshirani & Taylor, 2011) and the optimal solution can be obtained by using the algorithm proposed by Tibshirani & Taylor (2011) that obtains the solution path by solving the dual problem. For modeling purposes with the spatial-fused Lasso, the algorithm implemented as the package genlasso (e.g., Arnold & Tibshirani, 2019) in R (e.g., R Core Team, 2019) is available to calculate the optimal solution. However, estimation with the genlasso package has a high calculation cost and cannot be practically executed in large sample data. Moreover, there are issues in terms of numerical error and estimates cannot be exactly joined. Accordingly, we focus on the coordinate descent algorithm. To accurately calculate the optimal solution of the spatial-fused Lasso, even in large sample data, we give an update equation of the coordinate descent algorithm in closed form.

This paper is organized as follows: In section 2, we describe the spatial-fused Lasso and its optimization. In section 3, we give closed form update equations of the coordinate descent algorithm for optimizing the spatial-fused Lasso. Numerical examples are discussed in section 4. Technical details are relegated to the Appendix.

2. Preliminaries

2.1. Spatial-Fused Lasso

We split the space subjected to analysis into m small spaces and let μ_j ($j \in \{1, \ldots, m\}$) be the spatial effect for space j. Then, we consider the following model for an n_j-dimensional vector $\boldsymbol{y}_j = (y_{j,1}, \ldots, y_{j,n_j})'$ of response variables for space j:

$$\boldsymbol{y}_j = \boldsymbol{X}_j\boldsymbol{\beta} + \mu_j \boldsymbol{1}_{n_j} + \boldsymbol{\varepsilon}_j \quad (j \in \{1, \ldots, m\}),$$

where $\boldsymbol{X}_j = (\boldsymbol{x}_{j,1}, \ldots, \boldsymbol{x}_{j,n_j})'$ is an $n_j \times p$ matrix of nonstochastic explanatory variables, $\boldsymbol{\beta}$ is a p-dimensional vector of regression coefficients that does not depend on space, $\boldsymbol{1}_n$ is an n-dimensional vector of ones, and $\boldsymbol{\varepsilon}_j$ is an n_j-dimensional vector of independent error variables from a distribution with mean 0 and variance σ^2. In addition, $\boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_m$ are independent vectors. Then, an n-dimensional vector $\boldsymbol{y} = (\boldsymbol{y}_1', \ldots, \boldsymbol{y}_m')'$ of response variables for all spaces is expressed as

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{R}\boldsymbol{\mu} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{X} = (\boldsymbol{X}_1', \ldots, \boldsymbol{X}_m')'$ is an $n \times p$ matrix of nonstochastic explanatory variables, $\boldsymbol{R} = \mathrm{diag}(\boldsymbol{1}_{n_1}, \ldots, \boldsymbol{1}_{n_m})$ is an $n \times m$ block diagonal matrix, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_m)'$ is an m-dimensional vector of spatial effects, $\boldsymbol{\varepsilon} = (\boldsymbol{\varepsilon}_1', \ldots, \boldsymbol{\varepsilon}_m')'$ is an n-dimensional vector of independent error variables, and $n = \sum_{j=1}^{m} n_j$. Without loss of generality, we assume that a norm of a column vector of $\boldsymbol{X}$ is 1. Moreover, for the purposes of applications with a dummy variable with 3 or more categories as one explanatory variable, let the number of explanatory variables be k ($\leq p$) and we express $\boldsymbol{X}$ and $\boldsymbol{\beta}$ as

$$\boldsymbol{X} = (\boldsymbol{A}_1, \dots, \boldsymbol{A}_k), \quad \boldsymbol{\beta} = (\boldsymbol{\beta}_1', \dots, \boldsymbol{\beta}_k')',$$

where $\boldsymbol{A}_\ell$ is an $n \times p_\ell$ matrix that expresses the ℓth explanatory variable, $\boldsymbol{\beta}_\ell$ is a p_ℓ-dimensional vector of regression coefficients for the ℓth explanatory variable, and p_ℓ satisfies $p_\ell \geq 1$ and $p = \sum_{\ell=1}^{k} p_\ell$. In particular, when $p_\ell = 1$, we denote $\boldsymbol{A}_\ell = \boldsymbol{a}_\ell$ and $\boldsymbol{\beta}_\ell = \beta_\ell$. Notice that $\boldsymbol{X}$ is scaled and $\boldsymbol{A}_\ell$ (ℓ *s.t.* $p_\ell \geq 2$) is a matrix of a dummy variable, i.e., the number of non-zero elements of a row vector is one at most. Hence, the following equations hold:

$$||\boldsymbol{a}_\ell||_2 = 1 \ (\ell \ s.t. \ p_\ell = 1), \quad \boldsymbol{A}_\ell' \boldsymbol{A}_\ell = \boldsymbol{I}_{p_\ell} \ (\ell \ s.t. \ p_\ell \geq 2).$$

We estimate unknown parameters $\boldsymbol{\beta}$ and $\boldsymbol{\mu}$ by minimizing the following penalized residual sum of squares:

$$||\boldsymbol{y} - \boldsymbol{X\beta} - \boldsymbol{R\mu}||_2^2 + \lambda_1 \sum_{j=1}^{k} w_{1,j} ||\boldsymbol{\beta}_j||_2 + \lambda_2 \sum_{j=1}^{m} \sum_{\ell \in D_j} w_{2,j\ell} |\mu_j - \mu_\ell|, \tag{2.1}$$

where λ_1 and λ_2 are tuning parameters, $w_{1,j}$ and $w_{2,j\ell}$ are adaptive Lasso weights proposed by Zou (2006), and D_j is an index set of adjacent spaces for space j that satisfies $D_j \subseteq \{1, \dots, m\} \backslash \{j\}$. For example, if space 1 adjoins space 2 and 3, $D_1 = \{2, 3\}$. The second term in (2.1) is the group Lasso-type penalty proposed by Yuan & Lin (2006). This is an extension of Lasso (Tibshirani, 1996) to variable selection in terms of whether several variables are simultaneously zero. In this paper, since we consider variable selection in terms of whether the elements of $\boldsymbol{\beta}_\ell$ are simultaneously zero, the penalty is invoked. The third term in (2.1) is an extended penalty of the fused Lasso (Tibshirani *et al.*, 2005) for considering spatial effects. The ordinary fused Lasso is an extension of Lasso to analyze variables that have an order relationship and can equally join anteroposterior estimates. Since spatial effects have an adjacent relationship that is more complex than an anteroposterior relationship, we use the extended penalty of the fused Lasso. To distinguish it from the ordinary fused Lasso, we call it the spatial-fused Lasso-type penalty. By using the spatial-fused Lasso-type penalty, the estimation of spatial effects with the join of adjacent spaces becomes possible and we can equally estimate spatial effects of adjacent spaces. Moreover, these are adaptive penalties weighted by using an idea of the adaptive Lasso proposed by Zou (2006). By using weights based on the least-squares estimator, the adaptive Lasso estimator has the oracle property (Fan & Li, 2001). We estimate spatial effects by optimizing $\boldsymbol{\beta}$ and $\boldsymbol{\mu}$ via objective function (2.1). Furthermore, since the method is based on the fused Lasso, a clustering of spatial effects can be performed by joining adjacent spaces. Thus, the results obtained using this method have a secondary use in terms of local modeling in a clustered space, area marketing, and so on.

2.2. Optimization Procedure

We describe the procedure to optimize $\boldsymbol{\beta}$ and $\boldsymbol{\mu}$ via the objective function (2.1). Since the function includes two unknown parameters $\boldsymbol{\beta}$ and $\boldsymbol{\mu}$ and two tuning parameters λ_1 and λ_2, we must consider not only optimization with respect to the unknown parameters but also the optimal selection of the tuning parameters. Although the function (2.1) is convex with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\mu}$ with fixed λ_1 and λ_2, it is not convex with respect to λ_1 and λ_2. Hence, we must select the optimal pair (λ_1, λ_2) by optimizing $\boldsymbol{\beta}$ and $\boldsymbol{\mu}$ for any pairs (λ_1, λ_2). However, although each maximum in search points of λ_1 and λ_2 can be calculated from the data, these maxima cannot be simultaneously obtained, because the maxima of λ_1 and λ_2 depend on $\boldsymbol{\mu}$ and $\boldsymbol{\beta}$, respectively. Then, we obtain the optimal solutions of $\boldsymbol{\beta}$ and $\boldsymbol{\mu}$ by using the following algorithm whereby optimizations of λ_1 and $\boldsymbol{\beta}$ and the optimizations of λ_2 and $\boldsymbol{\mu}$ are alternately repeated.

- **Alternate Optimization Algorithm**

Input: *Initial vectors of $\boldsymbol{\beta}$ and $\boldsymbol{\mu}$*
Output: *Optimal solutions of $\boldsymbol{\beta}$ and $\boldsymbol{\mu}$*

Step 1: Optimize λ_1 and $\boldsymbol{\beta}$ by using fixed $\boldsymbol{\mu}$.

Step 2: Optimize λ_2 and $\boldsymbol{\mu}$ by using fixed $\boldsymbol{\beta}$.

Step 3: Repeat Steps 1 and 2 until $\boldsymbol{\beta}$ and $\boldsymbol{\mu}$ converge.

The following describes optimizations of $\boldsymbol{\beta}$ and $\boldsymbol{\mu}$, respectively.

Optimization of $\boldsymbol{\beta}$

Let $\boldsymbol{\mu}$ be fixed (let $\boldsymbol{\mu} = \hat{\boldsymbol{\mu}}$). Then, the optimal solution of $\boldsymbol{\beta}$ under fixed λ_1 can be obtained by minimizing the following function:

$$||\tilde{\boldsymbol{y}}_1 - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + \lambda_1 \sum_{j=1}^{k} w_{1,j} ||\boldsymbol{\beta}_j||_2, \tag{2.2}$$

where $\tilde{\boldsymbol{y}}_1 = \boldsymbol{y} - \boldsymbol{R}\hat{\boldsymbol{\mu}}$. As above, the optimization problem of $\boldsymbol{\beta}$ comes down to the optimization problem of the ordinary group Lasso as (2.2). To optimize the group Lasso, Yuan & Lin (2006) proposed an update equation of the coordinate descent algorithm, an extension of LARS (Efron *et al.*, 2004), and an extension of non-negative garrote (Breiman, 1995). Herein, we optimize $\boldsymbol{\beta}$ by using the update equation of the coordinate descent algorithm because it can be obtained in closed form and the algorithm is tractable. Although, the update equation proposed by Yuan & Lin (2006) requires a orthogonality of explanatory variables, the condition is satisfied, i.e., $\boldsymbol{A}_\ell'\boldsymbol{A}_\ell = \boldsymbol{I}_{p_\ell}$. Hence, we obtain the following update equation of the coordinate descent algorithm for (2.2):

$$\hat{\boldsymbol{\beta}}_{\lambda_1,i} = \left(1 - \frac{\lambda_1 w_{1,i}}{2\|\boldsymbol{c}_i\|_2}\right)_+ \boldsymbol{c}_i \quad (i = 1, \ldots, k), \tag{2.3}$$

where $\boldsymbol{c}_i = \boldsymbol{A}_i'(\tilde{\boldsymbol{y}}_1 - \sum_{j\neq i}^k \boldsymbol{A}_j\hat{\boldsymbol{\beta}}_j)$ and $(x)_+ = \max\{0, x\}$. In particular, when $p_i = 1, \hat{\beta}_{\lambda_1,i}$ is given as

$$\hat{\beta}_{\lambda_1,i} = S\left(c_i, \lambda_1 w_{1,i}/2\right), \tag{2.4}$$

where $S(x, a)$ is a soft-thresholding operator (e.g., Donoho & Johnstone, 1994), i.e., $S(x, a) = \mathrm{sign}(x)(|x| - a)_+$. The equation (2.4) is equal to an update equation given by Friedman *et al.* (2007). We decide a search point set Λ_1 of λ_1 and by using the following algorithm (which repeats updating of $\boldsymbol{\beta}$ with (2.3) or (2.4) for any $\lambda_1 \in \Lambda_1$), λ_1 and $\boldsymbol{\beta}$ can be optimized.

- **Coordinate Descent Algorithm for $\boldsymbol{\beta}$ (CDA$_{\boldsymbol{\beta}}$)**

Input: *Initial vector of $\boldsymbol{\beta}$ and search point set* Λ_1

Output: *Optimal solutions of $\boldsymbol{\beta}$ and λ_1*

```
Step 1:  Fix $\lambda_1$ and update $\hat{\boldsymbol{\beta}}_{\lambda_1,i}$ by using (2.3) or (2.4) for $i \in \{1, \ldots, k\}$.
Step 2:  Repeat Step 1 for fixed $\lambda_1$ until $\hat{\boldsymbol{\beta}}_{\lambda_1}$ converges.
Step 3:  Repeat Steps 1 and 2 for all $\lambda_1 \in \Lambda_1$.
Step 4:  Select the optimal $\lambda_1$.
```

When we use the CDA$_{\boldsymbol{\beta}}$, Λ_1 can be decided by defining $\lambda_{1,\max}$ and splitting the range $[0, \lambda_{1,\max}]$. For an example of $\lambda_{1,\max}$, we use the λ_1 that satisfies $\hat{\boldsymbol{\beta}}_{\lambda_1} = \boldsymbol{0}_p$, where $\boldsymbol{0}_p$ is a p-dimensional vector of zeros. From (2.3) and (2.4), $\lambda_{1,\max}$ satisfies

$$\forall \ell \in \{1, \ldots, k\},\ 1 - \frac{\lambda_1 w_{1,\ell}}{2\|\boldsymbol{c}_\ell\|_2} \leq 0.$$

Thus, we have

$$\lambda_{1,\max} = \max_{\ell\in\{1,\ldots,k\}} \frac{2\|\boldsymbol{c}_\ell\|_2}{w_{1,\ell}}. \tag{2.5}$$

Optimization of $\boldsymbol{\mu}$

Let $\boldsymbol{\beta}$ be fixed (let $\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}$). Then, the optimal solution of $\boldsymbol{\mu}$ under fixed λ_2 can be obtained by minimizing the following function:

$$\|\tilde{\boldsymbol{y}}_2 - \boldsymbol{R}\boldsymbol{\mu}\|_2^2 + \lambda_2 \sum_{j=1}^{m} \sum_{\ell \in D_j} w_{2,j\ell}|\mu_j - \mu_\ell|, \tag{2.6}$$

where $\tilde{\boldsymbol{y}}_2 = \boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}$. Since the spatial-fused Lasso can be reinterpreted as the generalized Lasso proposed by Tibshirani & Taylor (2011), the former can be optimized by using an optimization method for the latter. The penalty of the generalized Lasso is expressed as $\|\boldsymbol{D}\boldsymbol{\mu}\|_1$ by using known penalty matrix $\boldsymbol{D}$. For example, if $m = 3$, $D_1 = \{2\}$, $D_2 = \{1, 3\}$ and $D_3 = \{2\}$, then we use the following $\boldsymbol{D}$:

$$\boldsymbol{D} = \begin{pmatrix} w_{2,12} & -w_{2,12} & 0 \\ -w_{2,21} & w_{2,21} & 0 \\ 0 & w_{2,23} & -w_{2,23} \\ 0 & -w_{2,32} & w_{2,32} \end{pmatrix}.$$

Regarding optimization of the generalized Lasso, an algorithm exists for obtaining the solution path by solving a dual problem (Tibshirani & Taylor, 2011). The algorithm is implemented via the genlasso package in R; thus $\boldsymbol{\mu}$ can be optimized therein. However, the genlasso package has the following problems:

(P1) The numerical error is large.

(P2) The calculation cost is high.

Although we can obtain a sparse solution that includes an exact zero by using the ordinary Lasso, the zero may not be an exact zero when Lasso is optimized using the genlasso package. Moreover, although we can obtain a solution which joins elements by using the ordinary fused Lasso, optimization by the genlasso package may not exactly join. In terms of calculation cost, running the algorithm takes a long time and it cannot be executed large sample data. Clearly, this raises non-trivial concerns. Consequently, in this paper, we propose an efficient algorithm to minimize (2.6) for the purpose of estimating spatial effects.

3. Coordinate Descent Algorithm

In this section, we describe the optimization method of the spatial-fused Lasso by using the coordinate descent algorithm. For the ordinary fused Lasso, Friedman *et al.* (2007) proposed the coordinate descent algorithm. The algorithm consists of descent cycle and fusion cycle. The descent cycle successively minimizes along coordinate directions. The ordinary coordinate descent algorithm only consists of the descent cycle. However, when the ordinary coordinate descent algorithm is applied for fused Lasso, some estimates are joined and then the solution gets stuck, failing to reach the minimum. To avoid this problem, Friedman *et al.* (2007) invoked the fusion cycle. Since this problem can also occur when using the spatial-fused Lasso, we give update equations of $\boldsymbol{\mu}$ for the descent cycle and the fusion cycle.

descent cycle

The descent cycle minimizes along coordinate directions. That is to say, let μ_j ($j \in \{1, \ldots, m\}\backslash\{i\}$, $i \in \{1, \ldots, m\}$) be fixed (let $\mu_j = \hat{\mu}_j$) and we minimize (2.6) with respect to μ_i. The update equation for μ_i is given as follows: We denote the elements of D_i by

$$D_i = \{d_{i,1}, \ldots, d_{i,r_i}\} \ (\subseteq \{1, \ldots, m\}\backslash\{i\}),$$

where r_i is the number of elements of D_i, i.e., $r_i = \#(D_i) \leq m-1$. Moreover, let $t_{i,0} = -\infty$ and let $t_{i,1}, \ldots, t_{i,r_i}$ be the order statistics of $\hat{\mu}_j$ ($j \in D_i$), i.e.,

$$t_{i,j} = \begin{cases} \min\left\{\hat{\mu}_{d_{i,1}}, \ldots, \hat{\mu}_{d_{i,r_i}}\right\} & (j = 1) \\ \min\left\{\left\{\hat{\mu}_{d_{i,1}}, \ldots, \hat{\mu}_{d_{i,r_i}}\right\} \backslash \left\{t_{i,1}, \ldots, t_{i,j-1}\right\}\right\} & (j = 2, \ldots, r_i) \end{cases}, \tag{3.1}$$

and $J^+_{i,a}$ and $J^-_{i,a}$ be index sets for $a \in \{0, \ldots, r_i\}$ defined by

$$J^+_{i,a} = \left\{j \in D_i \mid \hat{\mu}_j \leq t_{i,a}\right\}, \quad J^-_{i,a} = \left\{j \in D_i \mid t_{i,a} < \hat{\mu}_j\right\}. \tag{3.2}$$

By using these equations, we define $\tilde{w}_{i,a}$ and $v_{i,a}$ as

$$\tilde{w}_{i,a} = \sum_{j \in J^+_{i,a}} w_{2,ij} - \sum_{j \in J^-_{i,a}} w_{2,ij}, \quad v_{i,a} = \frac{\tilde{\boldsymbol{y}}'_{2,i}\mathbf{1}_{n_i} - \lambda_2 \tilde{w}_{i,a}}{n_i}, \tag{3.3}$$

where $\tilde{\boldsymbol{y}}_{2,i}$ is the ith block of $\tilde{\boldsymbol{y}}_2$, i.e., $\tilde{\boldsymbol{y}}_{2,i} = \boldsymbol{y}_i - \boldsymbol{X}_i\hat{\boldsymbol{\beta}}$. Then, the update equation for μ_i is given as

$$\hat{\mu}_i = \begin{cases} v_{i,a^*_i} & (a^*_i \text{ exists}) \\ t_{i,a^\star_i} & (a^\star_i \text{ exists}) \end{cases}, \tag{3.4}$$

where a^*_i and $a^\star_i$ are nonnegative values defined by

$$a^*_i \in \{0, \ldots, r_i\} \ s.t. \ v_{i,a^*_i} \in R_{i,a^*_i}, \quad a^\star_i \in \{1, \ldots, r_i\} \ s.t. \ t_{i,a^\star_i} \in [v_{i,a^\star_i}, v_{i,a^\star_i - 1}),$$

and $R_{i,a}$ is the range defined by

$$R_{i,a} = \begin{cases} (t_{i,a}, t_{i,a+1}] & (a = 0, \ldots, r_i - 1) \\ (t_{i,r_i}, \infty) & (a = r_i) \end{cases}. \tag{3.5}$$

By updating μ_i ($i \in \{1, \ldots, m\}$) in order with (3.4), we can obtain the solution of $\boldsymbol{\mu}$ for the descent cycle. If $\hat{\mu}_i = t_{i,a^\star_i}$, spaces i and j ($j \in D_i \ s.t. \ t_{i,a^\star_i} = \hat{\mu}_j$) are joined. The uniqueness of (3.4) holds by the following theorem (the proof is given in Appendix A.1):

Theorem 1. *Let $\phi(x)$ be the continuous piecewise function defined by*

$$\phi(x) = \phi_a(x) = c_2 x^2 + c_{1,a} x + c_0 \ (c_2 > 0, \ x \in R_a),$$

where R_a $(a \in \{0, \dots, q\})$ is the range defined with (3.5) *by using monotonically increasing sequence $t_0 = -\infty, t_1, \dots, t_q$. Suppose that v_a, the x-coordinate of the vertex of $\phi_a(x)$, is monotonically decreasing with respect to a. Then, $\hat{x} = \arg\min_{x \in \mathbb{R}} \phi(x)$ is given by*

$$\hat{x} = \begin{cases} v_{a^*} & (a^* \text{ exists}) \\ t_{a^\star} & (a^\star \text{ exists}) \end{cases},$$

where a^* and $a^\star$ are nonnegative values defined by

$$a^* \in \{0, \dots, q\} \text{ s.t. } v_{a^*} \in R_{a^*}, \quad a^\star \in \{1, \dots, q\} \text{ s.t. } t_{a^\star} \in [v_{a^\star}, v_{a^\star - 1}),$$

and satisfy the following statements:

(i) *$a^\star$ does not exist $\Leftrightarrow$ a^* exists.*

(ii) *If a^* or $a^\star$ exists, it is unique.*

The equation (2.6) can be expressed as a piecewise function with respect to μ_i that satisfies the condition in Theorem 1 by using Lemmas A.1 and A.2 (details are given in Appendix A.2.1). Consequently, the update equation of μ_i is given in closed form as (3.4) by using Theorem 1.

fusion cycle

The fusion cycle avoids a solution getting stuck when some estimates are joined at the descent cycle. Suppose that we obtain $\hat{\mu}_j = \hat{\mu}_\ell$ as estimates of μ_j and μ_ℓ $(j \neq \ell)$ at the descent cycle. Then, to avoid $\hat{\mu}_j$ and $\hat{\mu}_\ell$ getting stuck, let $\eta_i = \mu_j = \mu_\ell$ and minimize toward the η_i-axis direction.

After the descent cycle, suppose that we obtain $\hat{\mu}_1, \dots, \hat{\mu}_m$ as estimates of $\mu_1, \dots, \mu_m$. Let $\hat{\eta}_1, \dots, \hat{\eta}_b$ $(b \leq m)$ be distinct values of $\hat{\mu}_1, \dots, \hat{\mu}_m$ and we define index sets $E_1, \dots, E_b$ as

$$E_j = \left\{\ell \in \{1, \dots, m\} \mid \hat{\mu}_\ell = \hat{\eta}_j\right\} \ (\subseteq \{1, \dots, m\}).$$

These E_j satisfy $E_j \neq \emptyset$ and $E_j \cap E_\ell = \emptyset$ $(j \neq \ell)$. If $b < m$, we execute the fusion cycle. In the fusion cycle, let η_j $(j \in \{1, \dots, b\} \backslash \{i\}, \ i \in \{1, \dots, b\})$ be fixed (let $\eta_j = \hat{\eta}_j$) and we minimize (2.6) with respect to η_i as in the descent cycle. We define a positive constant q_i as

$$q_i = \sum_{j \in E_i} q_{ij}, \quad q_{ij} = \#\left(D_j \backslash E_i\right).$$

Let $t_{i,0} = -\infty$ and let $t_{i,1}, \ldots, t_{i,q_i}$ be the order statistics of $\hat{\mu}_\ell$ $(\ell \in D_j \backslash E_i,\ j \in E_i)$ and $J^+_{ij,a}$ and $J^-_{ij,a}$ $(j \in E_i)$ be index sets for $a \in \{0, \ldots, q_i\}$ defined by

$$J^+_{ij,a} = \left\{\ell \in D_j \backslash E_i \mid \hat{\mu}_\ell \leq t_{i,a}\right\}, \quad J^-_{ij,a} = \left\{\ell \in D_j \backslash E_i \mid t_{i,a} < \hat{\mu}_\ell\right\}.$$

By using these equations, we define $\tilde{w}_{i,a}$ and $v_{i,a}$ as

$$\tilde{w}_{i,a} = \sum_{j \in E_i} \sum_{\ell \in J^+_{ij,a}} w_{2,ij} - \sum_{j \in E_i} \sum_{\ell \in J^-_{ij,a}} w_{2,ij}, \quad v_{i,a} = \frac{c_{1,i} - \lambda_2 \tilde{w}_{i,a}}{c_{2,i}}, \tag{3.6}$$

where $c_{1,i}$ and $c_{2,i}$ are constants defined by

$$c_{1,i} = \sum_{j \in E_i} \tilde{\boldsymbol{y}}_j' \mathbf{1}_{n_j}, \quad c_{2,i} = \sum_{j \in E_i} n_j. \tag{3.7}$$

Then, the update equation of η_i is given as

$$\hat{\eta}_i = \begin{cases} v_{i,a_i^*} & (a_i^* \text{ exists}) \\ t_{i,a_i^\star} & (a_i^\star \text{ exists}) \end{cases}, \tag{3.8}$$

where a_i^* and $a_i^\star$ are nonnegative values defined by

$$a_i^* \in \{0, \ldots, q_i\}\ s.t.\ v_{i,a_i^*} \in R_{i,a_i^*}, \quad a_i^\star \in \{1, \ldots, q_i\}\ s.t.\ t_{i,a_i^\star} \in [v_{i,a_i^\star}, v_{i,a_i^\star - 1}),$$

and $R_{i,a}$ $(a \in \{0, \ldots, q_i\})$ is the range defined as with (3.5) by using $t_{i,0}, \ldots, t_{i,q_i}$. By updating η_i $(i \in \{1, \ldots, b\})$ in order with (3.8), we can obtain the solution of $\boldsymbol{\mu}$ in the fusion cycle. If $\hat{\eta}_i = t_{i,a_i^\star}$, E_i and corresponding E_j are joined, and the fusion cycle is repeated until a join of spaces does not occur. The uniqueness of (3.8) holds by Theorem 1 as with (3.4). The equation (2.6) can be expressed as a piecewise function with respect to η_i that satisfies the condition in Theorem 1 by Lemmas A.3 and A.4 (details are given in Appendix A.2.2). Consequently, the update equation of η_i is given in closed form as (3.8) by using Theorem 1.

As above, we obtain update equations of the descent cycle and the fusion cycle. We decide a search point set Λ_2 of λ_2 and by using the following algorithm (which repeats updating of $\boldsymbol{\mu}$ in the descent cycle and the fusion cycle by using (3.4) and (3.8) for any $\lambda_2 \in \Lambda_2$), λ_2 and $\boldsymbol{\mu}$ can be optimized.

- **Coordinate Descent Algorithm for $\boldsymbol{\mu}$ (CDA$_{\boldsymbol{\mu}}$)**

  Input: *Initial vector of $\boldsymbol{\mu}$ and search points set Λ_2*

  Output: *Optimal solutions of $\boldsymbol{\mu}$ and λ_2*

  Step 1: (descent cycle) Fix λ_2, update $\hat{\mu}_{\lambda_2,i}$ by using (3.4) *for $i \in \{1, \ldots, m\}$, and define b.*

Step 2: *(fusion cycle) If $b < m$ in Step 1, update $\hat{\eta}_{\lambda_2,i}$ by using* (3.8) *for $i \in \{1, \ldots, b\}$ and repeat until the following statement is not satisfied.*

$$\exists(i_1, i_2) \ s.t. \ \hat{\eta}_{\lambda_2,i_1} = \hat{\eta}_{\lambda_2,i_2} \ (i_1 \neq i_2).$$

Step 3: *Repeat Steps 1 and 2 for fixed λ_2 until $\hat{\boldsymbol{\mu}}_{\lambda_2}$ converges.*

Step 4: *Repeat Steps 1, 2, and 3 for all $\lambda_2 \in \Lambda_2$.*

Step 5: *Select the optimal λ_2.*

Λ_2 can be decided by defining $\lambda_{2,\max}$ and splitting the range $[0, \lambda_{2,\max}]$ as with Λ_1. As an example of $\lambda_{2,\max}$, we use the λ_2 that satisfies $\hat{\boldsymbol{\mu}}_{\lambda_2} = \hat{\mu}_\infty \mathbf{1}_m$, where $\hat{\mu}_\infty = \mathbf{1}_n' \tilde{\boldsymbol{y}}_2/n$. From the descent cycle, $\lambda_{2,\max}$ satisfies

$$\forall j \in \{1, \ldots, m\}, \ \hat{\mu}_\infty \in [v_{j,r_j}, v_{j,0}].$$

Thus, we have

$$\lambda_{2,\max} = \max\left\{ \max_{j\in\{1,\ldots,m\}} \frac{\hat{\mu}_\infty n_j - \tilde{\boldsymbol{y}}_{2,j}' \mathbf{1}_{n_j}}{\sum_{\ell\in D_j} w_{2,j\ell}}, \ \max_{j\in\{1,\ldots,m\}} \frac{\tilde{\boldsymbol{y}}_{2,j}' \mathbf{1}_{n_j} - \hat{\mu}_\infty n_j}{\sum_{\ell\in D_j} w_{2,j\ell}} \right\}. \tag{3.9}$$

The optimal solutions of $\boldsymbol{\beta}$ and $\boldsymbol{\mu}$ can be obtained by using the Alternate Optimization Algorithm with CDA$_{\boldsymbol{\beta}}$ and CDA$_{\boldsymbol{\mu}}$ for the objective function (2.1).

4. Numerical Studies

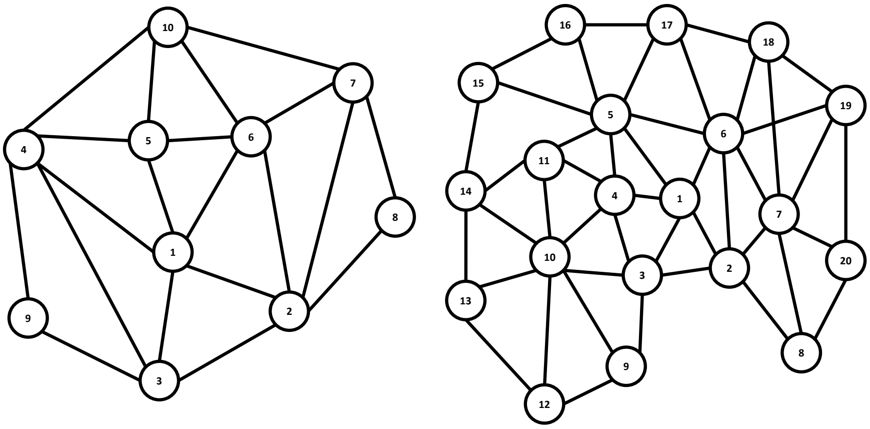
In this section, we present numerical simulations, discuss estimation accuracy, and consider an illustrative application to an actual data set. We use a computer with a Windows 10 Pro operating system, an Intel (R) Core (TM) i7-7700 processor, and 16 GB of RAM and R (ver. 3.6.0).

4.1. Simulation

In this subsection, we compare the estimation accuracies of the following Method 1 and Method 2 by simulation.

Method 1: The Alternate Optimization Algorithm using CDA$_{\boldsymbol{\mu}}$ to optimize $\boldsymbol{\mu}$.

Method 2: The Alternate Optimization Algorithm using the genlasso package (ver. 1.4) to optimize $\boldsymbol{\mu}$.



(a) $m = 10$ (b) $m = 20$

Figure 1. Simulation spaces and adjacent relationship

In applying both methods, we use the same CDA$_{\boldsymbol{\beta}}$ to optimize $\boldsymbol{\beta}$. The number of split spaces is $m = 10, 20$, the correlation between explanatory variables is $\rho = 0.5, 0.8$, and the sample sizes of split small spaces are $n_1 = \cdots = n_m = n_0$. Then, total sample size is $n = mn_0$ and we use $n_0 = 100, 200, 500, 1{,}000$. Figure 1 shows simulation spaces when $m = 10, 20$ with adjacent relationships indicated by lines. We generated data from the simulation model $N_n(\boldsymbol{X\beta} + \boldsymbol{R\mu}, \boldsymbol{I}_n)$ with the following $\boldsymbol{X}$:

$$\boldsymbol{X} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_8, \boldsymbol{A}_9, \ldots, \boldsymbol{A}_{13}),$$

where column vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_8$ and block matrices $\boldsymbol{A}_9, \ldots, \boldsymbol{A}_{13}$ are calculated as using the following procedure. Let $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{14}$ be independent n-dimensional vectors that the elements are identically and independently distributed according to $U(0, 1)$ and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{13}$ be n-dimensional vectors defined by

$$\boldsymbol{v}_j = \omega \boldsymbol{u}_{14} + (1 - \omega)\boldsymbol{u}_j,$$

where ω is the parameter determining the correlation of $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ $(i \neq j)$ as ρ, defined by

$$\omega = \begin{cases} \dfrac{\rho \pm \sqrt{\rho^2 - \rho(2\rho - 1)}}{2\rho - 1} & (\rho \neq 1/2) \\ \dfrac{1}{2} & (\rho = 1/2) \end{cases}.$$

By using these vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{13}$, we define the blocks in $\boldsymbol{X}$ as follows: Let $\boldsymbol{a}_j = \boldsymbol{v}_j$ for $j = 1, \ldots, 5$; let $\boldsymbol{a}_j$ $(j = 6, 7, 8)$ be dummy variables that take the value 1 or 0 defined by

$$a_{j,i} = \begin{cases} 1 & (v_{j,i} > 0.6) \\ 0 & (v_{j,i} \leq 0.6) \end{cases} \quad (i \in \{1, \dots, n\})\,;$$

and let $\boldsymbol{A}_j$ ($j = 9, \dots, 13$) be $(j-7)$-dimensional dummy variables that are categorized, defined by

$$(\text{The } i\text{th row vector of } \boldsymbol{A}_j) = \begin{cases} \boldsymbol{e}_{j-7,\ell} & (v_{j,i} \in Q_{j-6,\ell},\ \ell \neq j-6) \\ \boldsymbol{0}_{j-7} & (v_{j,i} \in Q_{j-6,j-6}) \end{cases} \quad (i = 1, \dots, n),$$

where $\boldsymbol{e}_{j,\ell}$ is a j-dimensional vector in which the ℓth element is 1 and the others are 0 and $Q_{j,\ell}$ is the ℓth range when $[0, 1]$ is split into j ranges. The following 2 cases are used as $\boldsymbol{\beta}$ and $\boldsymbol{\mu}$.

Case 1: Let the number of true explanatory variables be $k_* = 9$ and the number of true joins of spaces be

$$m_* = \begin{cases} 3 & (m = 10) \\ 6 & (m = 20) \end{cases},$$

and we use the following $\boldsymbol{\beta}$ and $\boldsymbol{\mu}$:

$$\boldsymbol{\beta} = \left(1, 2, 3, 0, 0, 1, 1, 2, \boldsymbol{1}_2', \boldsymbol{0}_3', 2 \times \boldsymbol{1}_4', \boldsymbol{0}_5', 3 \times \boldsymbol{1}_6'\right)',$$
$$\forall j \in E_\ell,\ \mu_j = \ell \quad (\ell = 1, \dots, m_*).$$

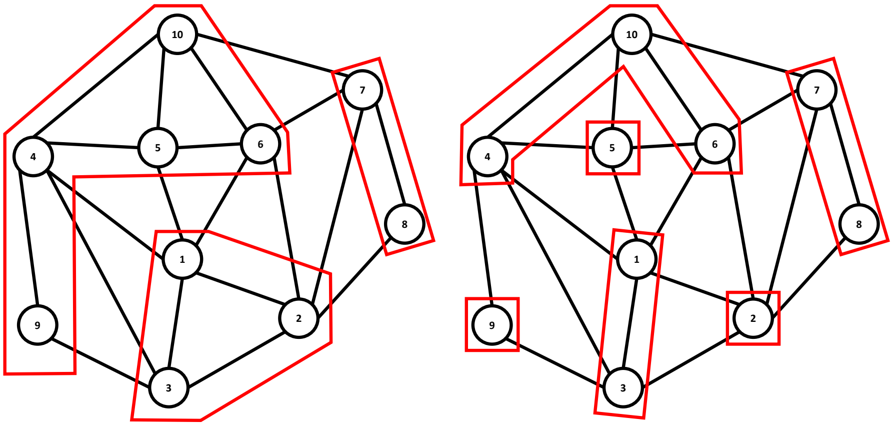
Case 2: Let the number of true explanatory variables be $k_* = 3$ and the number of true joins of spaces be

$$m_* = \begin{cases} 6 & (m = 10) \\ 12 & (m = 20) \end{cases},$$

and we use the following $\boldsymbol{\beta}$ and $\boldsymbol{\mu}$:

$$\boldsymbol{\beta} = \left(1, 0, 0, 0, 0, 1, 0, 0, \boldsymbol{0}_2', \boldsymbol{0}_3', 2 \times \boldsymbol{1}_4', \boldsymbol{0}_5', \boldsymbol{0}_6'\right)',$$
$$\forall j \in E_\ell,\ \mu_j = \ell \quad (\ell = 1, \dots, m_*).$$

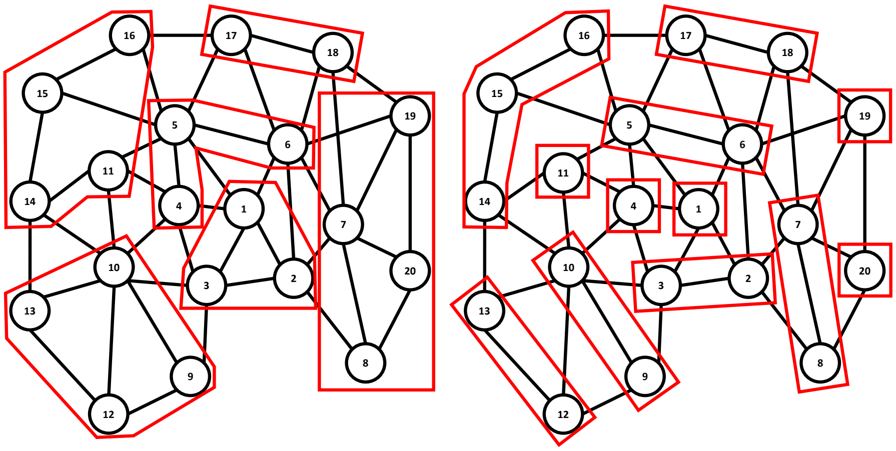
Figures 2 and 3 show true joins of spaces when $m = 10, 20$, respectively. Estimation accuracy is evaluated by the selection probabilities of true variables and true joins by Monte Carlo simulation with 1,000 iterations. One hundred search points of tuning parameters λ_1 and λ_2 are split by $\lambda_{\max}(3/4)^{j-1}$ ($j = 1, \dots, 100$) using $\lambda_{1,\max}$ and $\lambda_{2,\max}$ in (2.5) and (3.9), respectively. In terms of selecting of the optimal tuning parameters, we use the following Extended GCV



(a) Case 1

(b) Case 2

Figure 2. True joins when $m = 10$



(a) Case 1

(b) Case 2

Figure 3. True joins when $m = 20$

(EGCV) criterion (Ohishi *et al.*, 2020) minimization method:

$$\mathrm{EGCV} = \frac{(\text{residual sum of squares})/n}{\{1 - (\text{degrees of freedom})/n\}^{\alpha}},$$

where α is some positive value expressing the strength of the model complexity penalty. The EGCV criterion coincides with the GCV criterion (Craven & Wahba, 1979) when $\alpha = 2$. Moreover, we use the following general weights for penalty terms:

$$w_{1,j} = \frac{1}{\|\hat{\boldsymbol{\beta}}_j\|_2} \ (j \in \{1, \dots, k\}), \quad w_{2,j\ell} = \frac{1}{|\hat{\mu}_j - \hat{\mu}_\ell|} \ (j \in \{1, \dots, m\}, \ \ell \in D_j),$$

where $\hat{\boldsymbol{\beta}}_j$ and $\hat{\mu}_j$ are the least-squares estimators of $\boldsymbol{\beta}_j$ and μ_j, respectively. Tables 1 and 2 show the selection probabilities (SP) of true variables and true joins and running times (RT) of programs in Cases 1 and 2, respectively. The SP is displayed as the combined probability and separate probability about variables and joins. From the tables, since the SP of Method 1 approaches 100% as sample size increases, we found that Method 1 has high estimation accuracy. On the other hand, Method 2 struggles to select the true variables and true joins and its SP is 7.4% (when $m = 10$, $\rho = 0.8$, and $n = 5{,}000$ in Case 1) at most. In particular, it struggles to select the true joins and its SP is only 7.8% (when $m = 10$, $\rho = 0.8$, and $n = 5{,}000$ in Case 1) at most. Moreover, in terms of running time, Method 1 is about 134 times faster than Method 2 (when $m = 20$, $\rho = 0.5$, and $n = 20{,}000$ in Case 1) at most.

Table 1. Selection probabilities and running times in Case 1

			Method 1				Method 2			
m	ρ	n	SP (combined)	SP ($\boldsymbol{\beta}$)	SP ($\boldsymbol{\mu}$)	RT (s)	SP (combined)	SP ($\boldsymbol{\beta}$)	SP ($\boldsymbol{\mu}$)	RT (s)
10	0.5	1,000	74.20	92.40	79.90	0.65	4.30	92.10	5.00	0.39
		2,000	88.60	97.60	91.00	0.60	4.10	97.60	4.10	0.68
		5,000	93.80	98.50	95.20	0.58	6.00	98.50	6.10	4.14
		10,000	95.00	97.30	97.60	0.64	3.90	97.30	4.00	14.27
	0.8	1,000	59.60	73.70	79.90	1.01	3.20	73.60	5.10	0.53
		2,000	82.60	92.40	89.90	0.86	4.80	92.20	5.20	0.91
		5,000	91.40	95.40	95.80	0.82	7.40	95.40	7.80	4.68
		10,000	93.20	95.30	97.90	0.67	3.70	95.30	3.80	14.70
20	0.5	2,000	72.70	98.60	73.80	1.21	0.00	98.60	0.00	1.48
		4,000	85.20	98.80	86.30	1.23	0.00	98.70	0.00	6.14
		10,000	93.80	99.10	94.70	1.22	0.00	99.00	0.00	27.01
		20,000	97.60	98.90	98.70	1.10	0.10	98.90	0.10	148.04
	0.8	2,000	69.20	92.90	74.40	1.60	0.30	92.90	0.30	1.77
		4,000	83.30	95.90	86.80	1.47	0.00	96.00	0.00	6.63
		10,000	90.00	95.30	94.30	1.23	0.10	95.40	0.10	29.64
		20,000	94.10	95.30	98.70	1.18	0.00	95.40	0.00	146.24

4.2. A Real Data Example

In this subsection, we present an illustrative application of the proposed method (Method 1 in subsection 4.1) to an actual data set. Search points of tuning parameters and the model selection criterion are as per subsection 4.1. Since the applied data have a large sample and

Table 2. Selection probabilities and running times in Case 2

			Method 1				Method 2			
m	ρ	n	SP (combined)	SP ($\boldsymbol{\beta}$)	SP ($\boldsymbol{\mu}$)	RT (s)	SP (combined)	SP ($\boldsymbol{\beta}$)	SP ($\boldsymbol{\mu}$)	RT (s)
10	0.5	1,000	74.00	91.60	81.20	1.39	2.90	91.20	3.20	0.41
		2,000	87.30	97.20	89.90	0.84	0.90	97.10	0.90	0.52
		5,000	94.80	98.80	95.90	0.76	4.50	98.80	4.50	3.15
		10,000	95.90	98.50	97.30	0.71	1.40	98.40	1.40	11.41
	0.8	1,000	55.10	69.10	80.30	2.35	1.80	69.20	2.80	0.66
		2,000	81.20	91.30	89.40	1.05	1.80	91.50	2.00	0.61
		5,000	92.40	96.00	96.10	0.81	2.80	96.00	2.80	3.24
		10,000	94.10	96.60	97.40	0.87	2.20	96.60	2.30	12.62
20	0.5	2,000	63.40	97.90	65.00	1.53	0.10	97.90	0.10	1.06
		4,000	80.90	98.50	81.90	1.37	0.00	98.60	0.00	3.97
		10,000	92.70	99.20	93.40	1.32	0.40	99.20	0.40	20.63
		20,000	96.40	99.30	97.10	1.32	0.10	99.40	0.10	144.11
	0.8	2,000	59.60	91.20	66.20	1.77	0.30	91.20	0.30	1.20
		4,000	80.10	96.70	82.80	1.43	0.00	96.60	0.00	4.38
		10,000	90.70	97.00	93.50	1.47	0.70	97.00	0.70	27.11
		20,000	94.80	97.50	97.10	1.41	0.00	97.60	0.00	148.19

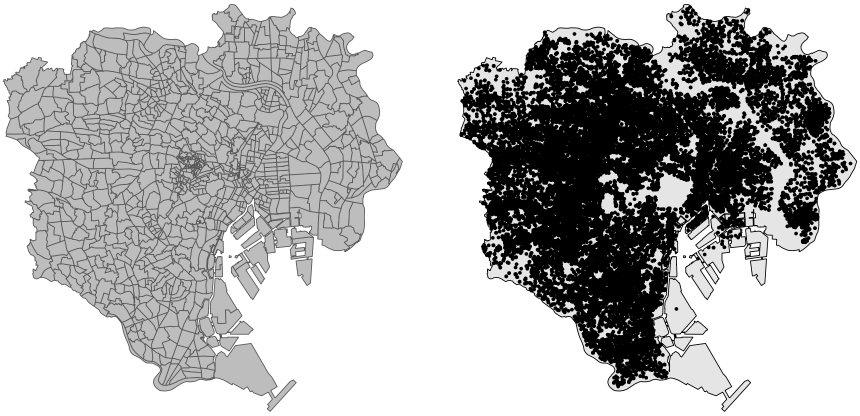
because the genlasso package causes memory shortage, Method 2 in subsection 4.1 cannot run the program. We compare the proposed method, which discretely evaluates spatial effects, with the GWR, which continuously evaluates spatial effects. Details of the estimation method by the GWR are described in Appendix A.4. The data pertain to studio apartment rents and environmental conditions in Tokyo's 23 wards collected by Tokyo Kantei Co., Ltd. Here, n = 61,999 and all data were collected between April 2014 and April 2015 (Table 3). In this application, let the response variable be monthly rent with the remainder set as explanatory variables. We estimate regional effects at 852 areas split Tokyo's 23 wards using the proposed method and at all sample points using the GWR. Figure 4 (a) and (b) show respectively the split of Tokyo's 23 wards into 852 areas and all sample points. Figure 5 shows that estimation results in the form of choropleth maps are similar using the proposed method and the GWR. Moreover, as Figure 6 shows, the proposed method can perform clustering of regional effects and the GWR can draw the contours. In terms of the former, as with Figure 6, the 852 areas in Tokyo's 23 wards are clustered to form 190 areas. Table 4 summarizes estimates of regression coefficients. As a result of variable selection, the proposed method did not select B5 and C1 and the GWR did not select C1. Figures 7 and 8 are residual plots for quantitative variables according to the proposed method and the GWR, respectively. Table 5 provides information concerning coefficients of determination (R^2), median error rate (MER), and running time. From the results, R^2 is more than 0.8, MER is less than 10%, and the residual plots are unproblematic. Thus,

Table 3. Data items

Y	Monthly rent of an apartment (yen)		
A	Land area of an apartment (m^2)		
B1	Whether an apartment has a parking lot		
B2	Whether an apartment is a condominium		
B3	Whether an apartment is a corner apartment		
B4	Whether an apartment is a fixed-term tenancy agreement		
B5	Whether an apartment is on the top floor		
C1	Facing direction		
	base: South	C1a: North	C1b: Northeast
	C1c: East	C1d: Southeast	C1e: Southwest
	C1f: West	C1g: Northwest	
C2	Building structure		
	base: Reinforced concrete	C2a: Wooden	C2b: Light steel frame
	C2c: Steel frame	C2d: Steel framed rein-forced concrete	C2e: ALC
	C2f: Steel framed precast concrete	C2g: Precast concrete	C2h: Reinforced block
	C2i: Other		
C3	Building age (years)		
	base: 0 (new-build)	C3a: 1 – 5	C3b: 6 – 10
	C3c: 11 – 15	C3d: 16 – 20	C3e: 21 – 25
	C3f: 26 – 30	C3g: 31 – 35	C3h: 36 – 40
	C3i: 41 – 45	C3j: 46 – 50	
C4	Iteration of logarithmic transformations of the top floor and a room floor		
	base: 0	C4a: 0 – 1	C4b: 1 – 2
	C4c: 2 – 3	C4d: 3 – 4	C4e: 4 – 5
	C4f: 5 <		
C5	Walking time (min) to the nearest station		
	base: 1 – 5	C5a: 6 – 10	C5b: 11 – 15
	C5c: 16 – 20	C5d: 21 ≤	

Y and A are continuous variables. B1 to B5 are dummy variables that take the value of 1 or 0. C1 to C5 are multidimensional dummy variables. C3 to C5 were transformed from continuous variables to categorical variables.

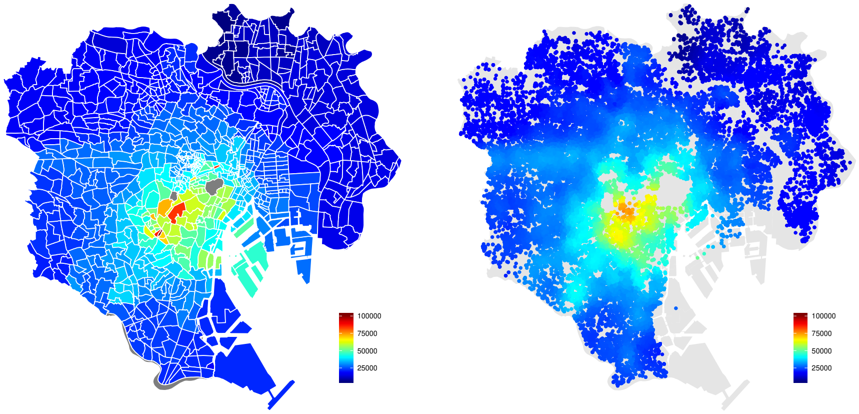
the proposed method and the GWR both perform well. Although we have applied large sample data, because the data exhibit low sparseness and contour line spread concentrically, we were able to obtain a good result using the GWR. However, there was a high calculation cost involved. Specifically, the GWR took about 126 times longer to run than the proposed method (Table 5). Since the proposed method discretely evaluates spatial effects and strongly depends on the number of split spaces rather than the sample size, the proposed method is a viable and practical option in large sample data.



(a) 852 areas

(b) All sample points

Figure 4. Tokyo's 23 wards



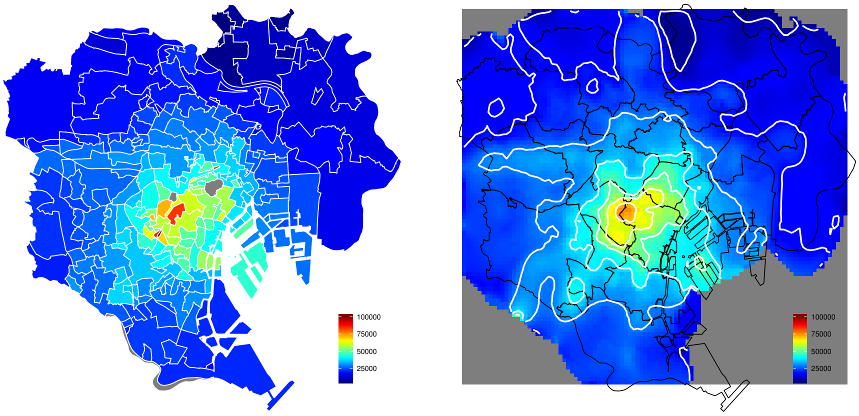
(a) Proposed method

(b) GWR

Figure 5. Regional effects estimation results I

5. Conclusion

In this paper, we proposed an algorithm for solving the optimization problem of the spatial-fused Lasso. This algorithm discretely evaluates and estimates spatial effects. Although the optimization problem can be solved using the genlasso package in R, since there are problems in terms of calculation cost and accuracy, we provided an update equation of the coordinate descent algorithm for the spatial-fused Lasso in closed form. From numerical studies, we found



(a) Clustering (proposed method)

(b) Contours (GWR)

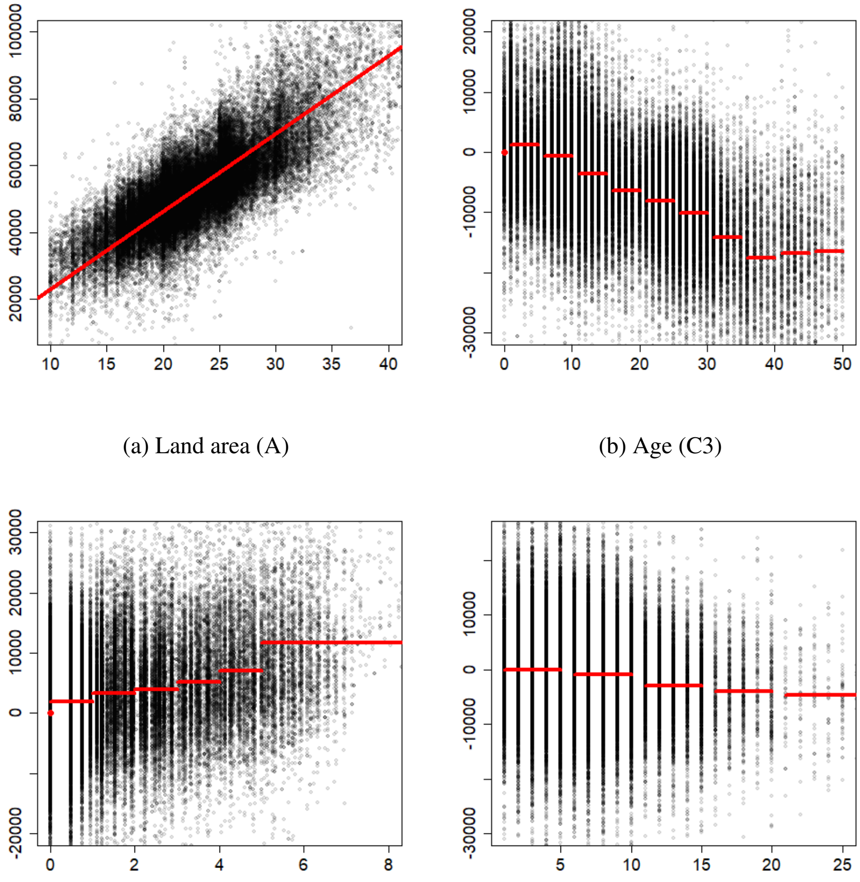
Figure 6. Regional effects estimation results II

Table 4. Regression coefficient estimates

	estimate			estimate			estimate	
	Proposed method	GWR		Proposed method	GWR		Proposed method	GWR
A	0.67942	0.67774	C2b	-0.01053	-0.01144	C3g	-0.04101	-0.04716
B1	0.00666	0.00757	C2c	-0.00844	-0.00938	C3h	-0.03809	-0.04243
B2	-0.00925	-0.01021	C2d	-0.00675	-0.00686	C3i	-0.03330	-0.03732
B3	-0.00954	-0.01118	C2e	-0.00689	-0.00748	C3j	-0.02307	-0.02579
B4	0.00468	0.00632	C2f	-0.00016	-0.00021	C4a	0.01151	0.01192
B5	0.00000	-0.00336	C2g	-0.00211	-0.00195	C4b	0.02015	0.02026
C1a	0.00000	0.00000	C2h	-0.00012	-0.00011	C4c	0.01583	0.01531
C1b	0.00000	0.00000	C2i	-0.00297	-0.00312	C4d	0.01803	0.01752
C1c	0.00000	0.00000	C3a	0.00596	-0.00342	C4e	0.02212	0.02161
C1d	0.00000	0.00000	C3b	-0.00286	-0.01307	C4f	0.03702	0.03700
C1e	0.00000	0.00000	C3c	-0.01729	-0.02610	C5a	-0.00774	-0.00805
C1f	0.00000	0.00000	C3d	-0.02185	-0.02910	C5b	-0.01239	-0.01262
C1g	0.00000	0.00000	C3e	-0.03560	-0.04395	C5c	-0.00708	-0.00709
C2a	-0.01291	-0.01413	C3f	-0.04550	-0.05479	C5d	-0.00422	-0.00427

that our proposed method exhibits higher calculation accuracy than the genlasso package and it is also much faster. Importantly, our proposed method is viable and practical in large sample data, unlike the genlasso package. Although the determination of adjacent relationships may be a non-trivial endeavor, we were able to obtain valuable results.

Moreover, although the spatial-fused Lasso was used herein for overcoming disadvantages associated with continuous evaluation of spatial effects, since it can be easily applied even in



(a) Land area (A)

(b) Age (C3)

(c) Iteration (C4)

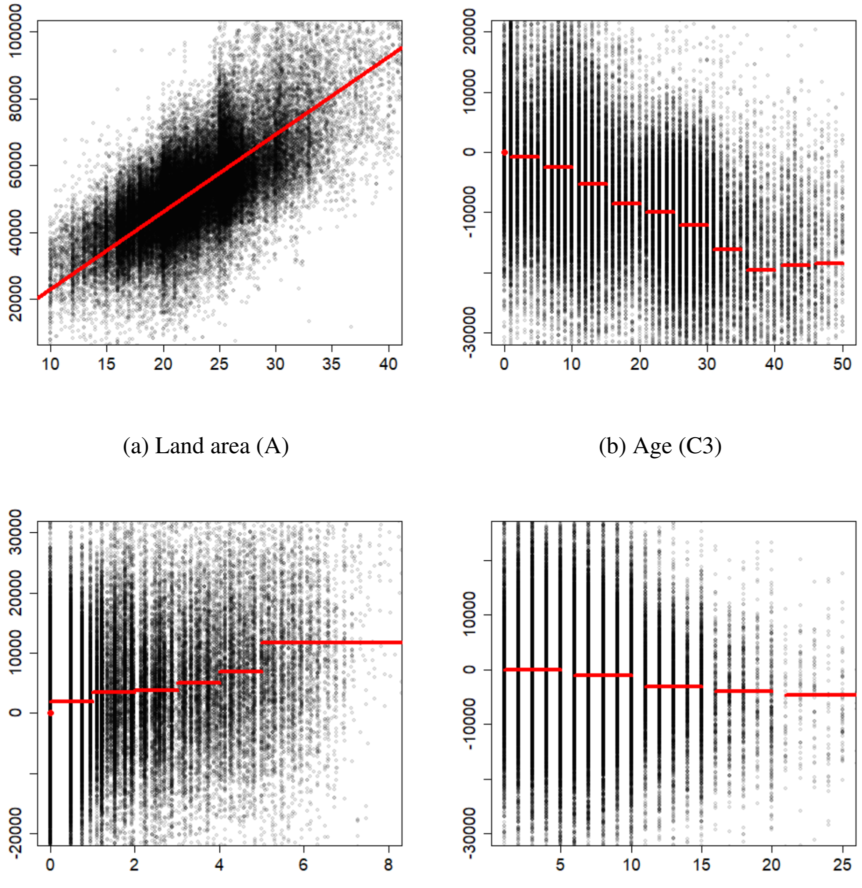
(d) Walking time (C5)

Figure 7. Residual plots (proposed method)

Table 5. Model fit and run time

	R^2	MER (%)	run time (min)
Proposed method	0.835	6.720	2.406
GWR	0.832	6.717	303.350

a large sample data and offers high accuracy, we can expect it to extend as spatial statistics method. In addition, since estimates are obtained at each joined space, our method also has advantages in prediction problem. Furthermore, since we can obtain clustering of spatial effects



(a) Land area (A)

(b) Age (C3)

(c) Iteration (C4)

(d) Walking time (C5)

Figure 8. Residual plots (GWR)

by joining small spaces, our method could have secondary uses in business practice.

Acknowledgments

The authors thank Mr. Ryoya Oda and Mr. Yuya Suzuki of Hiroshima University, Mr. Sato Yuhei and Mr. Yu Matsumura of Tokyo Kantei Co., Ltd., and Koki Kirishima of Computer Management Co., Ltd., for helpful comments.

References

Anselin, L. (1990). What is special about spatial data? Alternative perspectives on spatial data analysis. *Spatial Statistics: Past, Present, and Future*, Institute of Mathematical Geography, Ann Arbor, Michigan, 63-77.

Anselin, L. & Getis, A. (1992). Spatial statistical analysis and geographic information systems. *Ann. Reg. Sci.*, **26**, 19–33.

Arnold, T. & Tibshirani, R. (2019). genlasso: path algorithm for generalized lasso problems. R package version 1.4. https://CRAN.R-project.org/package=genlasso.

Breiman, L. (1995). Better subset regression using the nonnegative garrote. *Technometrics*, **37**, 373–384.

Brunsdon, C., Fotheringham, S. & Charlton, M. (1996). Geographically weighted regression: a method for exploring spatial nonstationarity. *Geogr. Anal.*, **28**, 281–298.

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.

Donoho, D. L. & Johnstone, I. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, **81**, 425–455.

Efron, B., Johnstone, I., Hastie, T. & Tibshirani, R. (2004). Least angle regression. *Ann. Statist.*, **32**, 407–499.

Fan, J. & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Statist. Assoc.*, **96**, 1348–1360.

Friedman, J., Hastie, T., Höfling, H. & Tibshirani, R. (2007). Pathwise coordinate optimization. *Ann. Appl. Stat.*, **1**, 302–332.

Löchl, M. & Axhausen, K. W. (2010). Modeling hedonic residential rents for land use and transport simulation while considering spatial effects. *J. Transp. Land Use*, **3**, 39–63.

Nakamura, T. & Kaneda, T. (2015). Analyses of factors of land price by applying mixed geographically weighted regression models in residential area in Nagoya in 2002 and 2012. *AIJ J. Technol. Des.*, **21**, 307–310 (in Japanese).

Ohishi, M., Yanagihara, H. & Fujikoshi, Y. (2020). A fast algorithm for optimizing ridge parameters in a generalized ridge regression by minimizing a model selection criterion. *J. Statist. Plann. Inference*, **204**, 187–205.

R Core Team. (2019). R: A language and environment for statistical computing. R foundation for statistical computing, Vienna, Austria. https://www.R-project.org/.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B*, **58**, 267–288.

Tibshirani, R., Saunders, M. & Rosset, S. (2005). Sparsity and smoothness via the fused Lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **67**, 91–108.

Tibshirani, R. & Taylor, J. (2011). The solution path of the generalized Lasso. *Ann. Statist.*, **39**, 1335–1371.

Tokyo Kantei Co., Ltd. https://www.kantei.ne.jp.

Yuan, M. & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **68**, 49–67.

Zou, H. (2006). The adaptive Lasso and its oracle properties. *J. Amer. Stat. Assoc.*, **101**, 1418–1429.

Appendix

A.1. Proof of Theorem 1

First, we present (i) and (ii) concerning a^* and $a^\star$. The (i) is proved as follows:

$$a^\star \text{ does not exist} \Leftrightarrow \forall a \in \{1, \ldots, q\},\ t_a \notin [v_a, v_{a-1})$$

$$\Leftrightarrow \begin{cases} \forall a \in \{1,\dots,q\},\ v_{a-1} \le t_a \\ \forall a \in \{1,\dots,q\},\ t_a < v_a \\ \exists ! a_0 \in \{1,\dots,q-1\}\ s.t. \begin{cases} t_a < v_{a_0} & (a \le a_0 - 1) \\ v_{a_0} \le t_a & (a_0 \le a) \end{cases} \end{cases}$$

$$\Leftrightarrow \begin{cases} v_0 \in R_0 & (a^* = 0) \\ v_k \in R_q & (a^* = q) \\ v_{a_0} \in R_{a_0} & (a^* = a_0 \in \{1,\dots,q-1\}) \end{cases}$$

$$\Leftrightarrow \exists a^* \in \{0,\dots,q\}\ s.t.\ v_{a^*} \in R_{a^*}$$

$$\Leftrightarrow a^* \text{ exists.}$$

Regarding the uniqueness of a^* in (ii), assume that a ($\in \{0,\dots,q\}$) and b ($\in \{0,\dots,q\}$) exist such that $v_a \in R_a$ and $v_b \in R_b$, and they satisfy $a + 1 \le b$ without loss of generality. Then, although $t_a < v_a \le t_{a+1} \le t_b < v_b \le t_{b+1}$ holds, this is in conflict with $v_b < v_a$. Thus, we have $a = b$.

Regarding the uniqueness of $a^\star$ in (ii), assume that a ($\in \{1,\dots,q\}$) and b ($\in \{1,\dots,q\}$) exist such that $t_a \in [v_a, v_{a-1})$ and $t_b \in [v_b, v_{b-1})$ and they satisfy $a + 1 \le b$ without loss of generality. Then, although $v_a \le t_a \le t_b < v_{b-1}$ holds, this is in conflict with $v_{b-1} \le v_a$. Thus, we have $a = b$.

The (i) and (ii) mean that either a^* or $a^\star$ uniquely exists and v_{a^*} or $t_{a^\star}$ is a local minimizer. In addition, from $\phi(x)$ is a continuous function, the local minimizer is the minimizer of $\phi(x)$. Consequently, Theorem 1 is proved.

A.2. Transformations to Piecewise Function

A.2.1. Descent Cycle

To minimize (2.6) with respect to μ_i ($i \in \{1,\dots,m\}$), we rewrite (2.6) as a function of μ_i. This function is given by the following lemma (the proof is given in Appendix A.3.1):

Lemma A.1. *The equation* (2.6) *can be expressed as the following function of* μ_i ($i \in \{1,\dots,m\}$)*:*

$$\phi_1(\mu_i \mid \lambda_2) = n_i\mu_i^2 - 2\tilde{\boldsymbol{y}}_{2,i}'\mathbf{1}_{n_i}\mu_i + 2\lambda_2 \sum_{j\in D_i} w_{2,ij}|\mu_i - \hat{\mu}_j| + u_i, \tag{A.1}$$

where u_i *is the term that does not depend on* μ_i*.*

Moreover, we rewrite (A.1) in non-absolute form. By using the order statistics $t_{i,1},\dots,t_{i,r_i}$ and the range $R_{i,a}$ ($a \in \{0,\dots,r_i\}, t_{i,0} = -\infty$) defined by (3.1) and (3.5), respectively, the piecewise function of μ_i for (2.6) is given by the following lemma (the proof is given in Appendix A.3.2):

Lemma A.2. *The equation* (A.1) *can be expressed as the following piecewise function:*

$$\begin{aligned}\phi_1(\mu_i \mid \lambda_2) &= \phi_{1,a}(\mu_i \mid \lambda_2)\\ &= n_i\mu_i^2 - 2(\tilde{\boldsymbol{y}}_{2,i}'\mathbf{1}_{n_i} - \lambda_2\tilde{w}_{i,a})\mu_i + u_{i,a} \quad (\mu_i \in R_{i,a},\ a \in \{0,\ldots,r_i\}),\end{aligned} \tag{A.2}$$

where $\tilde{w}_{i,a}$ is defined by (3.3) *and $u_{i,a}$ is the term that does not depend on μ_i. Moreover, $\phi_{1,a}(\mu_i \mid \lambda_2)$ satisfies the following properties:*

- *The $\phi_1(\mu_i \mid \lambda_2)$ is continuous in $\mu_i \in \mathbb{R}$, i.e., $\phi_{1,a}(t_{a+1} \mid \lambda_2) = \phi_{1,a+1}(t_{a+1} \mid \lambda_2)$ $(a = 0,\ldots,r_i - 1)$.*
- *The $v_{i,a}$, the μ_i-coordinate of the vertex of $\phi_{1,a}(\mu_i \mid \lambda_2)$, is a monotonically decreasing sequence with respect to a.*

A.2.2. Fusion Cycle

To minimize (2.6) with respect to η_i $(i \in \{1,\ldots,b\})$, we rewrite (2.6) as a function of η_i. This function is given by the following lemma (the proof is given in Appendix ap lem3):

Lemma A.3. *The equation* (2.6) *can be expressed as the following function of η_i $(i \in \{1,\ldots,b\})$:*

$$\phi_2(\eta_i \mid \lambda_2) = c_{2,i}\eta_i^2 - 2c_{1,i}\eta_i + 2\lambda_2 \sum_{j\in E_i}\sum_{\ell\in D_j\backslash E_i} w_{2,j\ell}|\eta_i - \hat{\mu}_\ell| + u_i, \tag{A.3}$$

where $c_{1,i}$ and $c_{2,i}$ are constants defined by (3.7) *and u_i is the term that does not depend on η_i.*

Moreover, by using the order statistics $t_{i,0},\ldots,t_{i,q_i}$ and the range $R_{i,a}$ $(a \in \{0,\ldots,q_i\})$ defined by using the order statistics, a piecewise function of η_i for (2.6) is given by the following lemma (the proof is given in Appendix A.3.4):

Lemma A.4. *The* (A.3) *can be expressed as the following piecewise function:*

$$\begin{aligned}\phi_2(\eta_i \mid \lambda_2) &= \phi_{2,a}(\eta_i \mid \lambda_2)\\ &= c_{2,i}\eta_i^2 - 2(c_{1,i} - \lambda\tilde{w}_{i,a})\eta_i + u_{i,a} \quad (\eta_i \in R_{i,a},\ a \in \{0,\ldots,q_i\}),\end{aligned} \tag{A.4}$$

where $\tilde{w}_{i,a}$ is defined by (3.6) *and $u_{i,a}$ is the term that does not depend on η_i. Moreover, $\phi_{2,a}(\eta_i \mid \lambda_2)$ satisfies the following properties:*

- *The $\phi_2(\eta_i \mid \lambda_2)$ is continuous in $\eta_i \in \mathbb{R}$, i.e., $\phi_{2,a}(t_{a+1} \mid \lambda_2) = \phi_{2,a+1}(t_{a+1} \mid \lambda_2)$ $(a = 0,\ldots,q_i - 1)$.*
- *The $v_{i,a}$, the η_i-coordinate of the vertex of $\phi_{2,a}(\eta_i \mid \lambda_2)$, is a monotonically decreasing sequence with respect to a.*

A.3. Proofs of Lemmas

A.3.1. Proof of Lemma A.1

We partition (2.6) into terms that do and do not depend on μ_i. The first term in (2.6) can be partitioned as follows:

$$\begin{aligned}
\|\tilde{\boldsymbol{y}}_2 - \boldsymbol{R}\boldsymbol{\mu}\|_2^2 &= \tilde{\boldsymbol{y}}_2'\tilde{\boldsymbol{y}}_2 - 2\tilde{\boldsymbol{y}}_2'\boldsymbol{R}\boldsymbol{\mu} + \boldsymbol{\mu}'\boldsymbol{R}'\boldsymbol{R}\boldsymbol{\mu} \\
&= \tilde{\boldsymbol{y}}_2'\tilde{\boldsymbol{y}}_2 - 2\left(\sum_{j\neq i}^{m} \hat{\mu}_j \tilde{\boldsymbol{y}}_{2,j}'\boldsymbol{1}_{n_j} + \mu_i \tilde{\boldsymbol{y}}_{2,i}'\boldsymbol{1}_{n_i}\right) + \sum_{j\neq i}^{m} n_j\hat{\mu}_j^2 + n_i\mu_i^2 \\
&= n_i\mu_i^2 - 2\tilde{\boldsymbol{y}}_{2,i}'\boldsymbol{1}_{n_i}\mu_i + \sum_{j\neq i}^{m}(n_j\hat{\mu}_j^2 - 2\tilde{\boldsymbol{y}}_{2,j}'\boldsymbol{1}_{n_j}\hat{\mu}_j) + \tilde{\boldsymbol{y}}_2'\tilde{\boldsymbol{y}}_2.
\end{aligned}$$

Moreover, since $w_{2,j\ell} = w_{2,\ell j}$ and $|\mu_j - \mu_\ell| = |\mu_\ell - \mu_j|$, the second term in (2.6) can be partitioned as follows:

$$\begin{aligned}
\sum_{j=1}^{m}\sum_{\ell\in D_j} w_{2,j\ell}|\mu_j - \mu_\ell| &= \sum_{j\neq i}^{m}\sum_{\ell\in D_j} w_{2,j\ell}|\mu_j - \mu_\ell| + \sum_{\ell\in D_i} w_{2,i\ell}|\mu_i - \mu_\ell| \\
&= \sum_{j\neq i}^{m}\sum_{\ell\in D_j\backslash\{i\}} w_{2,j\ell}|\hat{\mu}_j - \hat{\mu}_\ell| + 2\sum_{\ell\in D_i} w_{2,i\ell}|\mu_i - \hat{\mu}_\ell|.
\end{aligned}$$

Consequently, since $D_i = \{d_{i,1}, \ldots, d_{i,r_i}\}$, Lemma A.1 is proved and u_i is given by

$$u_i = \sum_{j\neq i}^{m}(n_j\hat{\mu}_j^2 - 2\tilde{\boldsymbol{y}}_{2,j}'\boldsymbol{1}_{n_j}\hat{\mu}_j) + \tilde{\boldsymbol{y}}_2'\tilde{\boldsymbol{y}}_2 + \lambda_2\sum_{j\neq i}^{m}\sum_{\ell\in D_j\backslash\{i\}} w_{2,j\ell}|\hat{\mu}_j - \hat{\mu}_\ell|.$$

A.3.2. Proof of Lemma A.2

First, we prove (A.2). Since $t_1, \ldots, t_{r_i}$ are the order statistics of $\hat{\mu}_{d_{i,1}}, \ldots, \hat{\mu}_{d_{i,r_i}}$, the following equation holds when $\mu_i \in R_{i,a}$:

$$\begin{aligned}
\sum_{j\in D_i} w_{2,ij}|\mu_i - \hat{\mu}_j| &= \sum_{j\in J_{i,a}^{+}} w_{2,ij}(\mu_i - \hat{\mu}_j) + \sum_{j\in J_{i,a}^{-}} w_{2,ij}(\hat{\mu}_j - \mu_i) \\
&= \tilde{w}_{i,a}\mu_i - \left(\sum_{j\in J_{i,a}^{+}} w_{2,ij}\hat{\mu}_j - \sum_{j\in J_{i,a}^{-}} w_{2,ij}\hat{\mu}_j\right),
\end{aligned}$$

where $J_{i,a}^{+}$ and $J_{i,a}^{-}$ are index sets defined by (3.2) and $\tilde{w}_{i,a}$ is defined by (3.3). Thus, we have (A.2) and $u_{i,a}$ is given by

$$u_{i,a} = u_i - 2\lambda_2\left(\sum_{j\in J_{i,a}^{+}} w_{2,ij}\hat{\mu}_j - \sum_{j\in J_{i,a}^{-}} w_{2,ij}\hat{\mu}_j\right).$$

Next, we prove that $v_{i,a}$ is a monotonically decreasing sequence. The following equation with respect to $\tilde{w}_{i,a}$ holds:

$$\begin{aligned}\tilde{w}_{i,a} &= \sum_{j \in J^+_{i,a}} w_{2,ij} - \sum_{j \in J^-_{i,a}} w_{2,ij}\\ &= \left(\sum_{j \in J^+_{i,a+1}} w_{2,ij} - w_{2,ij_*} \right) - \left(\sum_{j \in J^-_{i,a+1}} w_{2,ij} + w_{2,ij_*} \right)\\ &= \tilde{w}_{i,a+1} - 2w_{2,ij_*},\end{aligned}$$

where $j_* = \arg\min_{j \in J^-_{i,a}} \hat{\mu}_j = \arg\max_{j \in J^+_{i,a+1}} \hat{\mu}_j$. Since $w_{2,ij} > 0$, $\tilde{w}_{i,a}$ is a monotonically increasing sequence with respect to a. Thus, $v_{i,a}$ is a monotonically decreasing sequence with respect to a.

Finally, we prove that $\phi_1(\mu_i \mid \lambda_2)$ is a continuous function. The following equation holds about the term of $\phi_{1,a}(t_{i,a+1} \mid \lambda_2)$ that depends on a:

$$\begin{aligned}&2\lambda_2 \tilde{w}_{i,a} t_{i,a+1} - 2\lambda_2 \left(\sum_{j \in J^+_{i,a}} w_{2,ij}\hat{\mu}_j - \sum_{j \in J^-_{i,a}} w_{2,ij}\hat{\mu}_j \right)\\ &= 2\lambda_2 \tilde{w}_{i,a+1} t_{i,a+1} - 4\lambda_2 w_{2,ij_*} t_{i,a+1} - 2\lambda_2 \left(\sum_{j \in J^+_{i,a+1}} w_{2,ij}\hat{\mu}_j - \sum_{j \in J^-_{i,a+1}} w_{2,ij}\hat{\mu}_j - 2w_{2,ij_*} t_{i,a+1} \right)\\ &= 2\lambda_2 \tilde{w}_{i,a+1} t_{i,a+1} - 2\lambda_2 \left(\sum_{j \in J^+_{i,a+1}} w_{2,ij}\hat{\mu}_j - \sum_{j \in J^-_{i,a+1}} w_{2,ij}\hat{\mu}_j \right).\end{aligned}$$

Thus, we have $\phi_{1,a}(t_{a+1} \mid \lambda_2) = \phi_{1,a+1}(t_{a+1} \mid \lambda_2)$. Consequently, Lemma A.3.2 is proved.

A.3.3. Proof of Lemma A.3

We partition (2.6) into terms that do and do not depend on η_i. Then, the first term in (2.6) can be partitioned as follows:

$$\begin{aligned}\|\tilde{\boldsymbol{y}}_2 - \boldsymbol{R}\boldsymbol{\mu}\|_2^2 &= \tilde{\boldsymbol{y}}_2'\tilde{\boldsymbol{y}}_2 - 2\tilde{\boldsymbol{y}}_2'\boldsymbol{R}\boldsymbol{\mu} + \boldsymbol{\mu}'\boldsymbol{R}'\boldsymbol{R}\boldsymbol{\mu}\\ &= \tilde{\boldsymbol{y}}_2'\tilde{\boldsymbol{y}}_2 - 2\left(\sum_{j \notin E_i} \hat{\mu}_j \tilde{\boldsymbol{y}}_{2,j}' \boldsymbol{1}_{n_j} + \sum_{j \in E_i} \mu_j \tilde{\boldsymbol{y}}_{2,j}' \boldsymbol{1}_{n_j} \right) + \sum_{j \notin E_i} n_j \hat{\mu}_j^2 + \sum_{j \in E_i} n_j \mu_j^2\\ &= c_{2,i}\eta_i^2 - 2c_{1,i}\eta_i + \sum_{j \notin E_i} (n_j \hat{\mu}_j^2 - 2\tilde{\boldsymbol{y}}_{2,j}' \boldsymbol{1}_{n_j} \hat{\mu}_j) + \tilde{\boldsymbol{y}}_2'\tilde{\boldsymbol{y}}_2.\end{aligned}$$

Moreover, the second term in (2.6) can be partitioned as follows:

$$\sum_{j=1}^{m} \sum_{\ell \in D_j} w_{2,j\ell}|\mu_j - \mu_\ell| = \sum_{j \notin E_i} \sum_{\ell \in D_j} w_{2,j\ell}|\mu_j - \mu_\ell| + \sum_{j \in E_i} \sum_{\ell \in D_j} w_{2,j\ell}|\mu_j - \mu_\ell|$$

$$
\begin{aligned}
&= \sum_{j \notin E_i} \sum_{\ell \in D_j \backslash E_i} w_{2,j\ell} |\hat{\mu}_j - \hat{\mu}_\ell| + \sum_{j \notin E_i} \sum_{\ell \in D_j \cap E_i} w_{2,j\ell} |\hat{\mu}_j - \mu_\ell| \\
&\quad + \sum_{j \in E_i} \sum_{\ell \in D_j \backslash E_i} w_{2,j\ell} |\mu_j - \hat{\mu}_\ell| + \sum_{j \in E_i} \sum_{\ell \in D_j \cap E_i} w_{2,j\ell} |\mu_j - \mu_\ell| \\
&= \sum_{j \notin E_i} \sum_{\ell \in D_j \backslash E_i} w_{2,j\ell} |\hat{\mu}_j - \hat{\mu}_\ell| + \sum_{j \notin E_i} \sum_{\ell \in D_j \cap E_i} w_{2,j\ell} |\hat{\mu}_j - \eta_i| \\
&\quad + \sum_{j \in E_i} \sum_{\ell \in D_j \backslash E_i} w_{2,j\ell} |\eta_i - \hat{\mu}_\ell| + \sum_{j \in E_i} \sum_{\ell \in D_j \cap E_i} w_{2,j\ell} |\eta_i - \eta_i| \\
&= \sum_{j \notin E_i} \sum_{\ell \in D_j \backslash E_i} w_{2,j\ell} |\hat{\mu}_j - \hat{\mu}_\ell| + 2 \sum_{j \in E_i} \sum_{\ell \in D_j \backslash E_i} w_{2,j\ell} |\eta_i - \hat{\mu}_\ell|.
\end{aligned}
$$

Consequently, Lemma A.3 is proved and u_i is given by

$$
u_i = \sum_{j \notin E_i} (n_j \hat{\mu}_j^2 - 2\tilde{\boldsymbol{y}}_{2,j}' \mathbf{1}_{n_j} \hat{\mu}_j) + \tilde{\boldsymbol{y}}_2' \tilde{\boldsymbol{y}}_2 + \lambda_2 \sum_{j \notin E_i} \sum_{\ell \in D_j \backslash E_i} w_{2,j\ell} |\hat{\mu}_j - \hat{\mu}_\ell|.
$$

A.3.4. Proof of Lemma A.4

We omit details of the proof because Lemma A.4 can be proved as was Lemma A.2.

A.4. Estimation method using the GWR

With the GWR, spatial effects are estimated for each sample point. Let ξ_i $(i = 1, \ldots, n)$ be the spatial effect for the ith sample. Then, we consider the following model for y_i:

$$
y_i = \boldsymbol{x}_i' \boldsymbol{\beta} + \xi_i + \varepsilon_i,
$$

where $\boldsymbol{x}_i$ is the ith row vector of $\boldsymbol{X}$ and ε_i is the ith element of ε. The estimators of $\boldsymbol{\beta}$ and ξ_i are given by

$$
\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left\| \boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} - \hat{\boldsymbol{\xi}} \right\|^2, \quad \hat{\xi}_i = \arg\min_{\xi} \left\| \boldsymbol{W}_i^{1/2} (\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}} - \xi \mathbf{1}_n) \right\|^2, \tag{A.5}
$$

where $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_n)'$, $\boldsymbol{W}_i = \mathrm{diag}(w_{i,1}, \ldots, w_{i,n})$ is a diagonal matrix of order n, and $w_{i,j}$ is the weight of the jth sample for the ith sample. By solving (A.5), the estimators of $\boldsymbol{\beta}$ and $\boldsymbol{\xi}$ are expressed as

$$
\begin{aligned}
\hat{\boldsymbol{\beta}} &= (\boldsymbol{X}'\boldsymbol{X} - \boldsymbol{X}'\boldsymbol{W}\boldsymbol{X})^{-1} \boldsymbol{X}'(\boldsymbol{I}_n - \boldsymbol{W})\boldsymbol{y}, \\
\hat{\boldsymbol{\xi}} &= \boldsymbol{W} \left\{ \boldsymbol{I}_n - \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X} - \boldsymbol{X}'\boldsymbol{W}\boldsymbol{X})^{-1} \boldsymbol{X}'(\boldsymbol{I}_n - \boldsymbol{W}) \right\} \boldsymbol{y},
\end{aligned}
$$

where $\boldsymbol{W} = (\{\mathrm{tr}(\boldsymbol{W}_1)\}^{-1} \boldsymbol{w}_1, \ldots, \{\mathrm{tr}(\boldsymbol{W}_n)\}^{-1} \boldsymbol{w}_n)'$ and $\boldsymbol{w}_i = (w_{i,1}, \ldots, w_{i,n})'$. In this simulation, we use following weight:

$$w_{i,j} = \begin{cases} \dfrac{\cos(\theta_{i,j}) + 1}{2} & (0 \leq \theta_{i,j} < \pi) \\ 0 & (\pi \leq \theta_{i,j}) \end{cases}, \quad \theta_{i,j} = \frac{\pi d_{i,j}}{d_{\max}},$$

where $d_{i,j}$ is the distance between the ith sample point and the jth sample point and we define $d_{\max}$ by the following procedure. Let $d_{i,(j)}$ be the jth-smallest of $d_{i,1}, \ldots, d_{i,n}$ and we decide $r_{\min}$ and $r_{\max}$. Then, we calculate an $n \times r_{\max}$ distance matrix where the ith row vector is $(d_{i,(1)}, \ldots, d_{i,(r_{\max})})$. By using the distance matrix, we define the distance $d_{\max}$ so that the number of sample points used as weights is at least $r_{\min}$. In this simulation, let $r_{\max} = 50, 100, 300, 500, 1{,}000, 1{,}500, 2{,}000, 2{,}500$ and let $r_{\min}$ increase in steps of 50 up to 1,000 and then steps of 100 from 1,000. For example, $r_{\min} = 50, 100, \ldots, 250, 300$ when $r_{\max} = 300$ and $r_{\min} = 50, 100, \ldots, 950, 1{,}000, 1{,}100, 1{,}200, 1{,}300, 1{,}400, 1{,}500$ when $r_{\max} = 1{,}500$. The result from subsection 4.2 is that we estimate $\boldsymbol{\beta}$ and $\boldsymbol{\xi}$ for all pairs $(r_{\max}, r_{\min})$ and optimize the pairs $(r_{\max}, r_{\min})$ based on the EGCV criterion minimization method. The reason for calculating the $n \times r_{\max}$ distance matrix is that since the sample size is very large ($n = 61{,}999$), the full-size ($n \times n$) distance matrix cannot be calculated. Moreover, because of calculation cost, we set $r_{\max}$ up to 2,500.