TR-No. 21-02, Hiroshima Statistical Research Group, 1–16

Coordinate Descent Algorithm for Generalized Group Fused Lasso

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

[1]Education and Research Center for Artificial Intelligence and Data Innovation, Hiroshima University
1-1-89 Higashi-Senda-machi, Naka-ku, Hiroshima 730-0053, Japan
[2]Tokyo Kantei Co., Ltd.
Meguro Nishiguchi Bldg., 8F, 2-24-15 Kami-Osaki, Shinagawa 141-0021, Japan
[3]Graduate School of Advanced Science and Engineering, Hiroshima University
1-3-1 Kagamiyama, Higashi-Hiroshima 739-8526, Japan

Abstract

Group fused Lasso is an extension of the fused Lasso to problems involving grouped variables; it has the advantage of allowing consideration of mergers for grouped variables with adjacency relations. To date, however, studies of group fused Lasso have been restricted to handling only a limited adjacency relations. In this paper, we discuss generalized group fused Lasso (GGFL), an extension of group fused Lasso designed to accommodate more general adjacency relations. For example, in cases where models are defined separately for individual groups, GGFL allows groups with similar characteristics to be joined together, thus facilitating the classification of groups. This makes GGFL a powerful technique, but to date there has been no effective algorithm for obtaining the solutions. Here we propose an algorithm for obtaining GGFL solutions based on the method of coordinate descent algorithm.

(Last Modified: January 19, 2021)

Key words: Coordinate descent algorithm, Linear model, Fused Lasso.

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

1. Introduction

We consider a problem involving m groups and assume that, for the jth group ($j \in \{1, \ldots, m\}$), we are given a dataset $\{\boldsymbol{y}_j, \boldsymbol{X}_j\}$, where $\boldsymbol{y}_j$ is an n_j-dimensional vector of a response variable and $\boldsymbol{X}_j$ is an $n_j \times k$ matrix of explanatory variables. Then a linear regression

model for group j may be expressed in the form

$$\boldsymbol{y}_j = \boldsymbol{X}_j\boldsymbol{\beta}_j + \boldsymbol{\varepsilon}_j, \tag{1}$$

where $\boldsymbol{\beta}_j$ is a k-dimensional vector of regression coefficients and $\boldsymbol{\varepsilon}_j$ is an n_j-dimensional vector of error variable. We assume that $n_j > k$ and that the first column of $\boldsymbol{X}_j$ is an intercept—that is, the first column of $\boldsymbol{X}_j$ is $\mathbf{1}_{n_j}$, where $\mathbf{1}_n$ denotes the n-dimensional vector with all elements equal to 1. The simplest strategy for estimating the m unknown parameter vectors $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_m$ is the least squares method; applying this method individually to each group j, the least-squares estimator (LSE) for $\boldsymbol{\beta}_j$ takes the form

$$\hat{\boldsymbol{\beta}}_j = \boldsymbol{M}_j^{-1}\boldsymbol{c}_j; \quad \boldsymbol{M}_j = \boldsymbol{X}_j'\boldsymbol{X}_j, \quad \boldsymbol{c}_j = \boldsymbol{X}_j'\boldsymbol{y}_j. \tag{2}$$

When the modeling procedure is carried out separately for each group, it is of interest to determine which groups have (or do not have) similar characteristics. For example, although equation (1) describes a group-by-group modeling approach involving m distinct models, groups 1 and 2 may have similar characteristics, in which case we would like the models for these two groups to agree—that is, we would like the estimates obtained for $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ to be equal. The fused Lasso proposed in Tibshirani *et al.* (2005) is a technique for establishing this sort of relationship between pairs of unknown parameters with adjacent subscripts. For equation (1), we may apply the group fused Lasso, an extension of fused Lasso in which the solution is obtained as the minimizer of a penalized residual sum of squares (PRSS) of the form

$$\sum_{j=1}^{m} \|\boldsymbol{y}_j - \boldsymbol{X}_j\boldsymbol{\beta}_j\|^2 + \lambda \sum_{j=1}^{m-1} \|\boldsymbol{\beta}_j - \boldsymbol{\beta}_{j+1}\|,$$

where λ is a non-negative tuning parameter. The optimization problem for group fused Lasso may, by a suitable transformation of penalty terms, be reduced to the optimization problem for the ordinary group Lasso (Yuan & Lin, 2006), and can then be solved (Bleakley & Vert, 2011). However, although group fused Lasso is capable of addressing *one-to-one* relationships between groups—such as relationships between groups 1 and 2, or between groups 2 and 3—it has no provision for taking into account *one-to-many* relationships, such as a relationship existing between group 1 and groups 2, 3, and 4. Here we fill this gap by generalized group fused Lasso (GGFL), an extension of group fused Lasso that we use to estimate $\boldsymbol{\beta}_j$ ($j = 1, \ldots, m$). Specifically, the GGFL estimators of $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_m$ are obtained as the minimizer of the following PRSS:

$$\sum_{j=1}^{m} \|\boldsymbol{y}_j - \boldsymbol{X}_j\boldsymbol{\beta}_j\|^2 + \lambda \sum_{j=1}^{m} \sum_{\ell \in D_j} w_{j\ell} \|\boldsymbol{\beta}_j - \boldsymbol{\beta}_\ell\|, \tag{3}$$

where $D_j \subset \{1,\ldots,m\}\backslash\{j\}$ is the index set and $w_{j\ell}$ is a penalty weight based on adaptive Lasso (Zou, 2006). When $D_j = \{j+1\}$ $(j = 1,\ldots,m-1)$, $D_m = \emptyset$, and $w_{j\ell} = 1$, GGFL coincides with group fused Lasso. GGFL may also be viewed as an extension of generalized fused Lasso (e.g., Xin *et al.*, 2014; Ohishi *et al.*, 2019) for use with grouped variables. The GGFL optimization problem, like that of group fused Lasso, may be reduced to the group Lasso optimization problem by a suitable transformation of penalty terms; however, the *inverse* of this transformation is not uniquely defined in GGFL, and for that reason, it is not possible to obtain a unique solution. Thus, in this work, we propose a GGFL optimization strategy based on the coordinate descent algorithm—that is, we give an objective function for $\boldsymbol{\beta}_j$ $(j \in \{1,\ldots,m\})$ and derive the minimizer of the objective function. Because GGFL involves multiple non-differentiable points, we first derive a condition for the objective function to attain minimum at a non-differentiable point. When there are no non-differentiable points at which the objective function attains a minimum, we may proceed to search for the minimizer to this condition.

The remainder of this paper is organized as follows. In Section 2, we first derive condition for the objective function to exhibit minimum at a non-differentiable point, and then discuss a method for finding solutions in the absence of such pathologies and propose a coordinate descent algorithm for GGFL. In Section 3, we conduct simulations to characterize the performance of GGFL, and then present an application case study involving actual data. Technical detail is provided in the Appendix.

2. Main Result

In this section, we discuss a coordinate descent algorithm for minimizing the PRSS (3). The coordinate descent algorithm computes the minimizer by repeatedly minimizing the following objective function for $\boldsymbol{\beta}_j$ $(j \in \{1,\ldots,m\})$, in which all terms that do not depend on $\boldsymbol{\beta}_j$ are neglected:

$$f_j(\boldsymbol{\beta}_j) = \boldsymbol{\beta}_j' \boldsymbol{M}_j \boldsymbol{\beta}_j - 2\boldsymbol{c}_j' \boldsymbol{\beta}_j + \sum_{\ell \in D_j} \lambda_{j\ell} \|\boldsymbol{\beta}_j - \boldsymbol{\beta}_\ell\|, \tag{4}$$

where $\lambda_{j\ell} = 2\lambda w_{j\ell}$. We note that $f_j(\boldsymbol{\beta}_j)$ fails to be differentiable at $\boldsymbol{\beta}_j = \boldsymbol{\beta}_\ell$ $(\ell \in D_j)$. In this work, we use the following procedure to achieve minimization of the objective function (4). First, we determine whether the objective function attains a minimum at a non-differentiable point. If not, we proceed to seek the minimizer to this condition.

2.1. Conditions indicating the presence of groups to be joined

Let $f(\boldsymbol{z})$ $(\boldsymbol{z} \in \mathbb{R}^q)$ be a convex function that is differentiable for $\boldsymbol{z} \neq \boldsymbol{z}_0$ and consider the set $\mathcal{A}_q$ defined by $\mathcal{A}_q = \{\boldsymbol{\alpha} \in \mathbb{R}^q \mid \|\boldsymbol{\alpha}\| = 1\}$. Letting δ denote a non-negative real number, a

necessary and sufficient condition for $f(\boldsymbol{z})$ to attain a minimum at the non-differentiable point $\boldsymbol{z} = \boldsymbol{z}_0$ is

$$\forall \boldsymbol{\alpha} \in \mathcal{A}_q, \lim_{\delta \to +0} \left\{ \frac{\partial}{\partial \delta} f(\boldsymbol{z}_0 + \delta \boldsymbol{\alpha}) \right\} \geq 0. \tag{5}$$

We now use this condition to derive a condition for the objective function (4) to achieve a minimum at a non-differentiable point.

For an arbitrary vector $\boldsymbol{\alpha} \in \mathcal{A}_k$ and $s \in D_j$, we consider the function defined by

$$\begin{aligned} g_{js}(\delta) = f_j(\boldsymbol{\beta}_s + \delta \boldsymbol{\alpha}) = \boldsymbol{\alpha}' \boldsymbol{M}_j \boldsymbol{\alpha} \delta^2 + 2(\boldsymbol{M}_j \boldsymbol{\beta}_s - \boldsymbol{c}_j)' \boldsymbol{\alpha} \delta + \boldsymbol{\beta}_s' \boldsymbol{M}_j \boldsymbol{\beta}_s - 2\boldsymbol{c}_j' \boldsymbol{\beta}_s \\ + \lambda_{js} |\delta| + \sum_{\ell \in D_j \setminus \{s\}} \lambda_{j\ell} \|\delta \boldsymbol{\alpha} + \boldsymbol{\beta}_s - \boldsymbol{\beta}_\ell\|. \end{aligned}$$

The $g_{js}(\delta)$ is not differentiable at $\delta = 0$. For $\delta \neq 0$, the first derivative of $g_{js}(\delta)$ may be expressed as

$$\begin{aligned} \dot{g}_{js}(\delta) = \frac{d}{d\delta} g_{js}(\delta) = 2\boldsymbol{\alpha}' \boldsymbol{M}_j \boldsymbol{\alpha} \delta + 2(\boldsymbol{M}_j \boldsymbol{\beta}_s - \boldsymbol{c}_j)' \boldsymbol{\alpha} \\ + \lambda_{js} \operatorname{sign}(\delta) + \sum_{\ell \in D_j \setminus \{s\}} \lambda_{j\ell} \frac{\delta + (\boldsymbol{\beta}_s - \boldsymbol{\beta}_\ell)' \boldsymbol{\alpha}}{\|\delta \boldsymbol{\alpha} + \boldsymbol{\beta}_s - \boldsymbol{\beta}_\ell\|}. \end{aligned}$$

Thus, we have

$$\lim_{\delta \to +0} \dot{g}_{js}(\delta) = 2(\boldsymbol{M}_j \boldsymbol{\beta}_s - \boldsymbol{c}_j)' \boldsymbol{\alpha} + \lambda_{js} + \sum_{\ell \in D_j \setminus \{s\}} \lambda_{j\ell} \frac{(\boldsymbol{\beta}_s - \boldsymbol{\beta}_\ell)' \boldsymbol{\alpha}}{\|\boldsymbol{\beta}_s - \boldsymbol{\beta}_\ell\|},$$

and hence, from (5),

$$\begin{aligned} & f_j(\boldsymbol{\beta}_j) \text{ attains minimum for } \boldsymbol{\beta}_j = \boldsymbol{\beta}_s \ (s \in D_j) \\ \Leftrightarrow\ & \forall \boldsymbol{\alpha} \in \mathbb{R}^k, \lambda_{js} \geq -\boldsymbol{v}_j(\boldsymbol{\beta}_s)' \boldsymbol{\alpha}, \end{aligned}$$

where

$$\boldsymbol{v}_j(\boldsymbol{\beta}_s) = 2(\boldsymbol{M}_j \boldsymbol{\beta}_s - \boldsymbol{c}_j) + \sum_{\ell \in D_j \setminus \{s\}} \lambda_{j\ell} \frac{\boldsymbol{\beta}_s - \boldsymbol{\beta}_\ell}{\|\boldsymbol{\beta}_s - \boldsymbol{\beta}_\ell\|}.$$

Also, noting that $\|\boldsymbol{\alpha}\| = 1$, from the Cauchy–Schwarz inequality, we find

$$-\boldsymbol{v}_j(\boldsymbol{\beta}_s)' \boldsymbol{\alpha} \leq \left\| \boldsymbol{v}_j(\boldsymbol{\beta}_s) \right\|,$$

with equality holding only for $\boldsymbol{\alpha} = \boldsymbol{v}_j(\boldsymbol{\beta}_s) / \|\boldsymbol{v}_j(\boldsymbol{\beta}_s)\|$. From these results, we obtain the following theorem.

Theorem 1. The function $f_j(\boldsymbol{\beta}_j)$ defined by (4) attains a minimum at the non-differentiable point $\boldsymbol{\beta}_j = \boldsymbol{\beta}_s$ if there exists $s \in D_j$ for which the following inequality holds:

$$\lambda_{js} \geq \left\| \boldsymbol{v}_j(\boldsymbol{\beta}_s) \right\|.$$

Repeatedly applying Theorem 1 for $j \in \{1, \ldots, m\}$ allows us to determine whether the objective function $f_j(\boldsymbol{\beta}_j)$ attains a minimum at each non-differentiable point. If there exists $s \in D_j$ for which the inequality of Theorem 1 holds, then the estimates for $\boldsymbol{\beta}_j$ and $\boldsymbol{\beta}_s$ will be exactly equal, indicating that groups j and s are joined.

2.2. Searching minimizer in the absence of groups to be joined

We now assume that the objective function $f_j(\boldsymbol{\beta}_j)$ of equation (4) does not attain a minimum at any non-differentiable point $\boldsymbol{\beta}_j = \boldsymbol{\beta}_s$ ($s \in D_j$)—that is, that the inequality in Theorem 1 does not hold for any $s \in D_j$—and consider the task of seeking solution under this condition. Then $f_j(\boldsymbol{\beta}_j)$ is a differentiable convex function and hence, we may search for solution by computing its gradient that takes the form

$$\frac{\partial}{\partial \boldsymbol{\beta}_j} f_j(\boldsymbol{\beta}_j) = 2\boldsymbol{M}_j\boldsymbol{\beta}_j - 2\boldsymbol{c}_j + \sum_{\ell \in D_j} \lambda_{j\ell} \frac{\boldsymbol{\beta}_j - \boldsymbol{\beta}_\ell}{\|\boldsymbol{\beta}_j - \boldsymbol{\beta}_\ell\|}.$$

From this expression, it follows that

$$\frac{\partial}{\partial \boldsymbol{\beta}_j} f_j(\boldsymbol{\beta}_j) = \boldsymbol{0}_k \Longleftrightarrow \left(2\boldsymbol{M}_j + \sum_{\ell \in D_j} \frac{\lambda_{j\ell}}{\|\boldsymbol{\beta}_j - \boldsymbol{\beta}_\ell\|} \boldsymbol{I}_k \right) \boldsymbol{\beta}_j = 2\boldsymbol{c}_j + \sum_{\ell \in D_j} \frac{\lambda_{j\ell}\boldsymbol{\beta}_\ell}{\|\boldsymbol{\beta}_j - \boldsymbol{\beta}_\ell\|},$$

and we derive the following update equation:

$$\boldsymbol{\beta}_j^{\text{new}} = Q(\boldsymbol{\beta}_j^{\text{old}})^{-1} h(\boldsymbol{\beta}_j^{\text{old}}), \tag{6}$$

where

$$Q(\boldsymbol{\beta}_j) = 2\boldsymbol{M}_j + \sum_{\ell \in D_j} \frac{\lambda_{j\ell}}{\|\boldsymbol{\beta}_j - \boldsymbol{\beta}_\ell\|} \boldsymbol{I}_k, \quad h(\boldsymbol{\beta}_j) = 2\boldsymbol{c}_j + \sum_{\ell \in D_j} \frac{\lambda_{j\ell}\boldsymbol{\beta}_\ell}{\|\boldsymbol{\beta}_j - \boldsymbol{\beta}_\ell\|}.$$

2.3. Coordinate descent algorithm for GGFL

Friedman *et al.* (2007) and Ohishi *et al.* (2019) respectively proposed coordinate descent algorithms for ordinary fused Lasso and generalized fused Lasso. These papers note the following phenomenon. When several groups are joined together at intermediate stages of the algorithm, the optimization process gets stuck: the corresponding objective-function values stagnate—ceasing to improve on subsequent iterations—and the algorithm fails to achieve minimization. To avoid this difficulty, as similar to the two papers, our coordinate descent

algorithm incorporates two cycles: a *descent cycle* and a *fusion cycle*.

The descent cycle repeatedly minimizes $f_j(\boldsymbol{\beta}_j)$ in (4) for $j \in \{1, \ldots, m\}$. More specifically, for GGFL, the descent cycle proceeds as follows:

Algorithm 1.（**Descent cycle for GGFL**）

Step 1. Apply Theorem 1 to $f_j(\boldsymbol{\beta}_j)$. If the condition holds, proceed to Step 2; otherwise proceed to Step 3.

Step 2. For s satisfying the condition of Theorem 1, update the solution as $\boldsymbol{\beta}_j = \boldsymbol{\beta}_s$.

Step 3. Use equation (6) to seek a solution.

Step 4. Repeat steps 1–3 for all $j \in \{1, \ldots, m\}$.

Step 5. Repeat step 4 until the solution converges.

The fusion cycle is designed to avoid the phenomenon in which the joining of groups during the descent cycle causes the optimization process to founder, obstructing the progress of the algorithm. Suppose that, following a descent cycle, we have solutions $\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_m$ and that some groups have been joined. Let $\hat{\boldsymbol{\xi}}_1, \ldots, \hat{\boldsymbol{\xi}}_t$ $(t < m)$ be the distinct vectors of $\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_m$ and define an index set E_ℓ $(\ell = 1, \ldots, t)$ according to

$$E_\ell = \left\{ j \in \{1, \ldots, m\} \mid \hat{\boldsymbol{\xi}}_\ell = \hat{\boldsymbol{\beta}}_j \right\},$$

where $E_\ell \neq \emptyset$ and $E_\ell \cap E_j = \emptyset$ $(\ell \neq j)$. The two terms in the PRSS (3) may respectively be rewritten as follows (see Ohishi *et al.*, 2019):

$$\sum_{j=1}^{m} \|\boldsymbol{y}_j - \boldsymbol{X}_j\boldsymbol{\beta}_j\|^2 = \sum_{\ell=1}^{t} \sum_{j \in E_\ell} \|\boldsymbol{y}_j - \boldsymbol{X}_j\boldsymbol{\xi}_\ell\|^2,$$

$$\sum_{j=1}^{m} \sum_{\ell \in D_j} w_{j\ell} \|\boldsymbol{\beta}_j - \boldsymbol{\beta}_\ell\| = 2 \sum_{j \in E_\ell} \sum_{i \in D_j \backslash E_\ell} w_{ji} \|\boldsymbol{\beta}_j - \boldsymbol{\beta}_i\| + \sum_{j \notin E_\ell} \sum_{i \in D_j \backslash E_\ell} w_{ji} \|\boldsymbol{\beta}_j - \boldsymbol{\beta}_i\|. \tag{7}$$

Note that the first term in (7) is the sum with respect to the elements in the set

$$\bigcup_{j \in E_\ell} \{j\} \times (D_j \backslash E_\ell).$$

Next define $D_\ell^* \subseteq \{1, \ldots, t\} \backslash \{\ell\}$ $(\ell \in \{1, \ldots, t\})$ and $w_{\ell i}^*$ $(\ell \in \{1, \ldots, t\};\ i \in D_\ell^*)$ as follows:

$$D_\ell^* = \{ s \in \{1, \ldots, t\} \backslash \{\ell\} \mid E_s \cap F_\ell \neq \emptyset \}, \quad F_\ell = \bigcup_{j \in E_\ell} D_j \backslash E_\ell,$$

$$w_{\ell i}^* = \sum_{(j,s) \in \mathcal{J}_{\ell i}} w_{js}, \quad \mathcal{J}_{\ell i} = \bigcup_{j \in E_\ell} \{j\} \times (E_i \cap D_j),$$

where D_ℓ^* satisfies $\ell \notin D_\ell^*$, $D_\ell^* \neq \emptyset$, and $s \in D_\ell^* \Leftrightarrow \ell \in D_s^*$. In words, D_ℓ^* is the set of fused-group indexes that have an adjacency relationship with fused group ℓ. Then we have the following lemma (the proof is given in Appendix A.1).

Lemma 1. An equality of sets exists:

$$\bigcup_{j \in E_\ell} \{j\} \times (D_j \backslash E_\ell) = \bigcup_{i \in D_\ell^*} \mathcal{J}_{\ell i}.$$

As a consequence, we find

$$\sum_{j \in E_\ell} \sum_{i \in D_j \backslash E_\ell} w_{ji} \|\boldsymbol{\beta}_j - \boldsymbol{\beta}_i\| = \sum_{i \in D_\ell^*} \sum_{(j,s) \in \mathcal{J}_{\ell i}} w_{js} \|\boldsymbol{\beta}_j - \boldsymbol{\beta}_s\| = \sum_{i \in D_\ell^*} w_{\ell i}^* \|\boldsymbol{\xi}_\ell - \boldsymbol{\xi}_i\|.$$

Thus, excluding from the PRSS (3) all terms that do not depend on $\boldsymbol{\xi}_\ell$, we may express the objective function for the fusion cycle as

$$f_\ell^*(\boldsymbol{\xi}_\ell) = \sum_{j \in E_\ell} (\boldsymbol{\xi}_\ell' \boldsymbol{M}_j \boldsymbol{\xi}_\ell - 2\boldsymbol{c}_j' \boldsymbol{\xi}_\ell) + \sum_{i \in D_\ell^*} \lambda_{\ell i}^* \|\boldsymbol{\xi}_\ell - \boldsymbol{\xi}_i\|,$$

where $\lambda_{\ell i}^* = 2\lambda w_{\ell i}^*$. In the fusion cycle, we execute the descent cycle for $f_\ell^*(\boldsymbol{\xi}_\ell)$. In analogy to what we found above for the descent cycle, we may formulate a theorem expressing the condition for $f_\ell^*(\boldsymbol{\xi}_\ell)$ to be minimized at $\boldsymbol{\xi}_\ell = \boldsymbol{\xi}_s$ ($s \in D_\ell^*$) as follows.

Theorem 2. If there exists $s \in D_\ell^*$ such that the following inequality holds, then $f_\ell^*(\boldsymbol{\xi}_\ell)$ attains a minimum at the non-differentiable point $\boldsymbol{\xi}_\ell = \boldsymbol{\xi}_s$:

$$\lambda_{\ell s}^* \geq \|v_\ell^*(\boldsymbol{\xi}_s)\|, \quad v_\ell^*(\boldsymbol{\xi}_s) = 2 \sum_{j \in E_\ell} (\boldsymbol{M}_j \boldsymbol{\xi}_s - \boldsymbol{c}_j) + \sum_{i \in D_\ell^* \backslash \{s\}} \lambda_{\ell i}^* \frac{\boldsymbol{\xi}_s - \boldsymbol{\xi}_i}{\|\boldsymbol{\xi}_s - \boldsymbol{\xi}_i\|}.$$

Assembling the steps discussed above yields the following coordinate descent algorithm for GGFL.

Algorithm 2.（Coordinate descent algorithm for GGFL）

Step 1. Choose initial vectors for $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_m$.

Step 2. Execute the descent cycle.

Step 3. If any groups were joined, execute the fusion cycle.

Step 4. Repeat steps 2 and 3 until the solution converges.

When executing Algorithm 1, the condition of Theorem 1 cannot be evaluated if there are any fused groups within D_j. To avoid this situation, we revise the objective function (4) before

executing the algorithm. For $\ell_1, \ell_2 \in D_j$, we set $\boldsymbol{\beta}_{\ell_1} = \boldsymbol{\beta}_{\ell_2}$, and then make the transformations

$$\lambda_{j\ell_1} \leftarrow \lambda_{j\ell_1} + \lambda_{j\ell_2}, \quad D_j \leftarrow D_j \backslash \{\ell_2\}.$$

Because GGFL depends on the tuning parameter λ, for practical applications, it is important to optimize the value of λ. Let $\hat{\boldsymbol{\beta}}_{\max}$ denote the solution with all groups joined; that is, $\hat{\boldsymbol{\beta}}_{\max}$ is the LSE for $\boldsymbol{\beta}_j$ obtained by setting $\boldsymbol{\beta}_1 = \cdots = \boldsymbol{\beta}_m$. Then we define $\lambda_{\max}$ as follows:

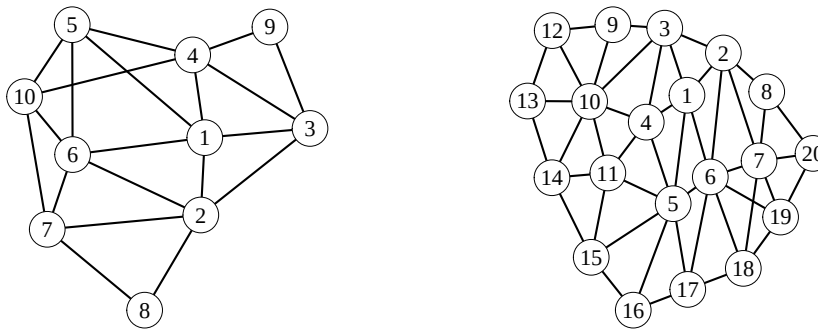
$$\lambda_{\max} = \max_{j=1,\ldots,m} \frac{\|\boldsymbol{M}_j \hat{\boldsymbol{\beta}}_{\max} - \boldsymbol{c}_j\|}{\sum_{\ell \in D_j} w_{j\ell}}.$$

The significance of $\lambda_{\max}$ may be seen by noting that the solution obtained for $\lambda = \lambda_{\max}$ satisfies $\hat{\boldsymbol{\beta}}_1 = \cdots = \hat{\boldsymbol{\beta}}_m = \hat{\boldsymbol{\beta}}_{\max}$. Thus, for each candidate λ value in the range $(0, \lambda_{\max}]$, we execute Algorithm 2 and select the optimal λ value.

3. Numerical Studies

3.1. Simulation

For this simulation, we construct simulation spaces similar to those used in Ohishi *et al.* (2019). We consider two problem sizes, with group counts $m = 10$ and $m = 20$ and the adjacency relationships depicted in Figure 1. Thus, for the $m = 10$ problem, group 1 is adjacent to



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

Figure 1. Adjacency relationships among groups.

groups 2, 3, 4, 5, and 6, i.e., $D_1 = \{2, 3, 4, 5, 6\}$. For each m value, we consider two possible cases of group-join configurations; each configuration is specified by enumerating the true sets

E_ℓ ($\ell = 1, \ldots, m^*$) of joined groups (here m^* is the true number of sets of joined groups), as follows:

- $m = 10$, case 1:

$$E_1 = \{1, 2, 3\}, \quad E_2 = \{4, 5, 6, 9, 10\}, \quad E_3 = \{7, 8\}.$$

- $m = 10$, case 2:

$$E_1 = \{1, 3\}, \quad E_2 = \{2\}, \quad E_3 = \{4, 6, 10\}, \quad E_4 = \{5\}, \quad E_5 = \{7, 8\}, \quad E_6 = \{9\}.$$

- $m = 20$, case 1:

$$E_1 = \{1, 2, 3\}, \quad E_2 = \{4, 5, 6\}, \quad E_3 = \{7, 8, 19, 20\},$$
$$E_4 = \{9, 10, 12, 13\}, \quad E_5 = \{11, 14, 15, 16\}, \quad E_6 = \{17, 18\}.$$

- $m = 20$, case 2:

$$E_1 = \{1\}, \quad E_2 = \{2, 3\}, \quad E_3 = \{4\}, \quad E_4 = \{5, 6\},$$
$$E_5 = \{7, 8\}, \quad E_6 = \{9, 10\}, \quad E_7 = \{11\}, \quad E_8 = \{12, 13\},$$
$$E_9 = \{14, 15, 16\}, \quad E_{10} = \{17, 18\}, \quad E_{11} = \{19\}, \quad E_{12} = \{20\}.$$

These group-join configurations are illustrated in Figures 2 and 3. In this section, we use simulation data to assess whether our GGFL technique successfully determines the true join configuration in each case.

The model from which our simulation data are generated takes the following form:

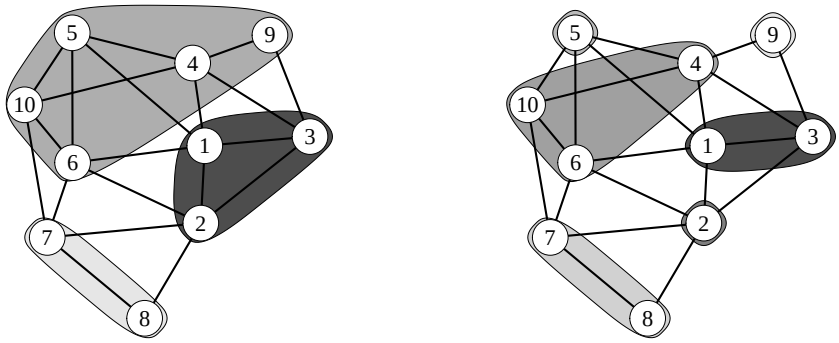
$$\boldsymbol{y}_j \sim N_{n_j}(\boldsymbol{X}_j \boldsymbol{\beta}_j, \boldsymbol{I}_{n_j}) \quad (j = 1, \ldots, m),$$

where $\boldsymbol{X}_j$ is an $n_j \times k$ matrix whose the first column is $\boldsymbol{1}_{n_j}$ and whose all remaining elements are identically and independently distributed according to $U(-1, 1)$, and $\boldsymbol{\beta}_j$ is a k-dimensional vector defined by

$$\forall j \in E_\ell, \ \boldsymbol{\beta}_j = \ell \boldsymbol{1}_k \quad (\ell = 1, \ldots, m^*).$$

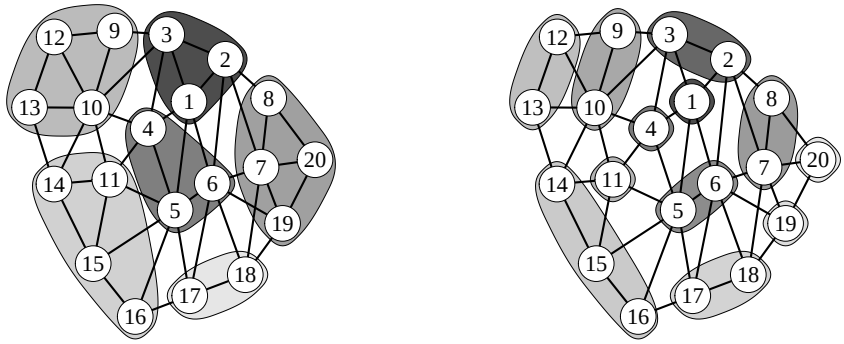
In our simulation, we characterize not only the selection probability (SP) of the true join configuration but also the mean square error (MSE). For $\hat{\boldsymbol{\beta}} = (\hat{\boldsymbol{\beta}}_1', \ldots, \hat{\boldsymbol{\beta}}_m')'$ and $\hat{\boldsymbol{y}} = (\hat{\boldsymbol{y}}_1', \ldots, \hat{\boldsymbol{y}}_m')'$, we compute the following two MSEs:

$$\mathrm{MSE}_\beta[\hat{\boldsymbol{\beta}}] = \mathrm{E}\left[\sum_{j=1}^{m} \|\boldsymbol{\beta}_j - \hat{\boldsymbol{\beta}}_j\|^2\right] / (km), \quad \mathrm{MSE}_y[\hat{\boldsymbol{y}}] = \mathrm{E}\left[\sum_{j=1}^{m} \|\boldsymbol{y}_j - \hat{\boldsymbol{y}}_j\|^2\right] / n,$$



(a) Case 1 (b) Case 2

Figure 2. True join configurations for $m = 10$.



(a) Case 1 (b) Case 2

Figure 3. True join configurations for $m = 20$.

where $\hat{\boldsymbol{\beta}}_j$ is an estimator of $\boldsymbol{\beta}_j$ and $\hat{\boldsymbol{y}}_j$ is given by $\hat{\boldsymbol{y}}_j = \boldsymbol{X}_j\hat{\boldsymbol{\beta}}_j$. The expected value of the MSE is characterized by 1,000 repetitions of a Monte Carlo simulation. We consider three estimators for $\boldsymbol{\beta}_j$:

- GGFL: The estimator produced by the GGFL method proposed in this paper.
- LSE 1: The LSEs defined by (2), i.e., $\hat{\boldsymbol{\beta}}_j = \boldsymbol{M}_j^{-1}\boldsymbol{c}_j$.
- LSE 2: The LSE for a common set $\boldsymbol{\beta}_1 = \cdots = \boldsymbol{\beta}_m$, i.e., $\hat{\boldsymbol{\beta}}_j = \hat{\boldsymbol{\beta}}_{\max}$.

Actually, instead of MSE, we use the following relative MSE (RMSE).

$$\text{RMSE} = 100 \times \begin{cases} (\text{MSE for GGFL})/(\text{MSE for LSE 2}) & (\text{for GGFL}) \\ (\text{MSE for LSE 1})/(\text{MSE for LSE 2}) & (\text{for LSE 1}) \end{cases}.$$

Denoting estimators for LSE 1 by $\tilde{\boldsymbol{\beta}}_1, \ldots, \tilde{\boldsymbol{\beta}}_m$, we use the following weights for GGFL penalties:

$$w_{j\ell} = \frac{1}{||\tilde{\boldsymbol{\beta}}_j - \tilde{\boldsymbol{\beta}}_\ell||} \quad (j = 1, \ldots, m,\ \ell \in D_j).$$

We use the EGCV criterion (Ohishi *et al.*, 2020) to determine the optimal tuning parameter and set the penalty strength to $\log n$. The RMSE and SP values for cases 1 and 2 are tabulated

Table 1. Simulation results

			case 1					case2				
			RMSE_y		RMSE_β			RMSE_y		RMSE_β		
m	k	n	GGFL	LSE 1	GGFL	LSE 1	SP	GGFL	LSE 1	GGFL	LSE 1	SP
10	20	500	**2.67**	4.44	**5.07**	26.54	95.52	**0.69**	0.85	**1.86**	5.00	97.09
		1,000	**0.97**	2.36	**2.45**	11.07	99.85	**0.29**	0.44	**0.92**	2.10	99.91
		2,000	**0.39**	1.15	**1.30**	5.26	100.00	**0.13**	0.22	**0.48**	0.99	100.00
	40	500	6.02	**4.28**	**9.53**	67.93	53.51	9.80	**0.82**	**12.60**	12.61	56.31
		1,000	**0.88**	2.11	**2.40**	14.18	99.99	**0.25**	0.40	**0.99**	2.71	99.99
		2,000	**0.37**	1.09	**1.30**	5.86	100.00	**0.12**	0.20	**0.52**	1.14	100.00
20	20	1,000	**0.44**	0.93	**1.03**	5.83	62.12	**0.14**	0.23	**0.44**	1.46	73.81
		2,000	**0.16**	0.45	**0.49**	2.35	98.95	**0.06**	0.11	**0.22**	0.58	99.10
		4,000	**0.07**	0.22	**0.26**	1.05	99.99	**0.03**	0.05	**0.12**	0.26	99.99
	40	1,000	**0.67**	0.84	**1.58**	19.30	30.65	4.11	**0.20**	4.65	**4.56**	12.12
		2,000	**0.15**	0.42	**0.52**	3.17	99.72	**0.06**	0.11	**0.24**	0.77	99.79
		4,000	**0.07**	0.22	**0.26**	1.19	100.00	**0.03**	0.05	**0.13**	0.29	100.00

in Table 1. As this table demonstrates, in most case, our proposed method achieves the smallest RMSE values of all the estimation methods considered for both predicted and estimated values. The table also indicate a possibility that our GGFL method have a consistency for the selection of true join configurations.

3.2. A Real Data Example

In this section, we present an application of the method proposed in this paper to actual data. The dataset we use is similar to that used in Ohishi *et al.* (2019); it consists of rental prices—and additional data describing environmental conditions—for studio apartments in Tokyo's 23 wards as observed between April 2014 and April 2015. The dataset, compiled by Tokyo Kantei

Co., Ltd., has a sample size of $n = 61{,}999$ and contains $m = 852$ groups; the specific data items it covers are listed in Table 2. For this dataset, the territory covered by Tokyo's 23 wards was

Table 2. Data items

Y	Monthly rent of an apartment (yen)
A1	Floor area of an apartment (m^2)
A2	Building age (years)
A3	Interaction of logarithmic transformations of the top floor and a room floor
A4	Walking time (min) to the nearest station
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 a facing direction is south
B6	Whether a building structure is a reinforced concrete

Y and A1 through A4 are continuous variables. B1 through B6 are dummy variables that take the value of 1 or 0.

Figure 4. The 852 subregions in Tokyo's 23 wards.

divided into 852 geographical subregions, corresponding to the 852 groups in the dataset (Figure 4). In our analysis, we take the monthly apartment rent to be the response variable, using all other data items as explanatory variables; however, problems arise when dummy variables are modeled on a group-by-group basis. For this reason, our dummy variables are common to all groups. More specifically, our modeling proceeds as follows. Let $\boldsymbol{y}_j$ be an n_j-dimensional

vector of a response variable for group j. Let $\boldsymbol{X}_j$ be an $n_j \times 5$ matrix of explanatory variables for group j, whose first column is $\mathbf{1}_{n_j}$ and whose remaining 4 columns correspond to items A1 through A4. Let $\boldsymbol{Z}_j$ be an additional $n_j \times 6$ matrix of explanatory variables for group j, whose columns correspond to items B1 through B6. Then we consider the following model:

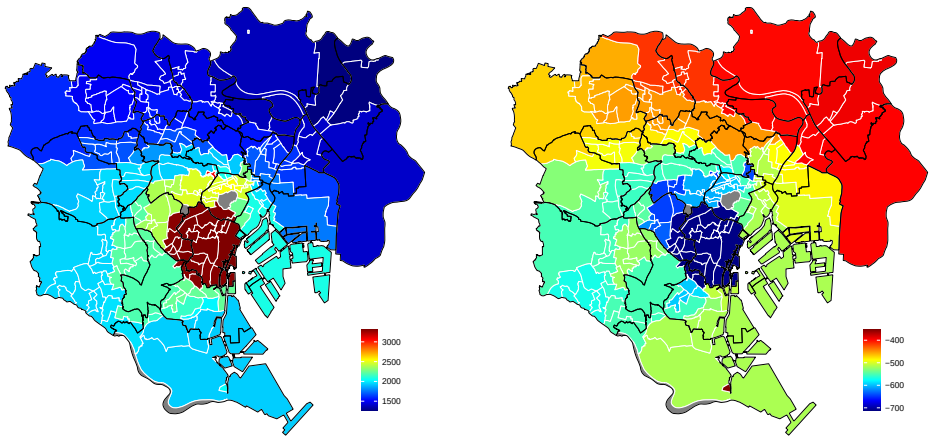
$$\boldsymbol{y}_j = \boldsymbol{X}_j\boldsymbol{\beta}_j + \boldsymbol{Z}_j\boldsymbol{\gamma} + \boldsymbol{\varepsilon}_j \quad (j = 1, \ldots, m),$$

where $\boldsymbol{\beta}_j$ and $\boldsymbol{\gamma}$ are five- and six-dimensional vectors of regression coefficients. This means that Tokyo's 23 wards are expressed by m (= 852) submodels. We estimate regression coefficients as follows:

$$\hat{\boldsymbol{\beta}}_\lambda = (\hat{\boldsymbol{\beta}}'_{\lambda,1}, \ldots, \hat{\boldsymbol{\beta}}'_{\lambda,m})' = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{j=1}^{m} \|\boldsymbol{y}_j - \boldsymbol{X}_j\boldsymbol{\beta}_j - \boldsymbol{Z}_j\hat{\boldsymbol{\gamma}}_\lambda\|^2 + \lambda \sum_{j=1}^{m} \sum_{\ell \in D_j} w_{j\ell} \|\boldsymbol{\beta}_j - \boldsymbol{\beta}_\ell\| \right\},$$

$$\hat{\boldsymbol{\gamma}}_\lambda = \arg\min_{\boldsymbol{\gamma}} \sum_{j=1}^{m} \|\boldsymbol{y}_j - \boldsymbol{X}_j\hat{\boldsymbol{\beta}}_{\lambda,j} - \boldsymbol{Z}_j\boldsymbol{\gamma}\|^2.$$

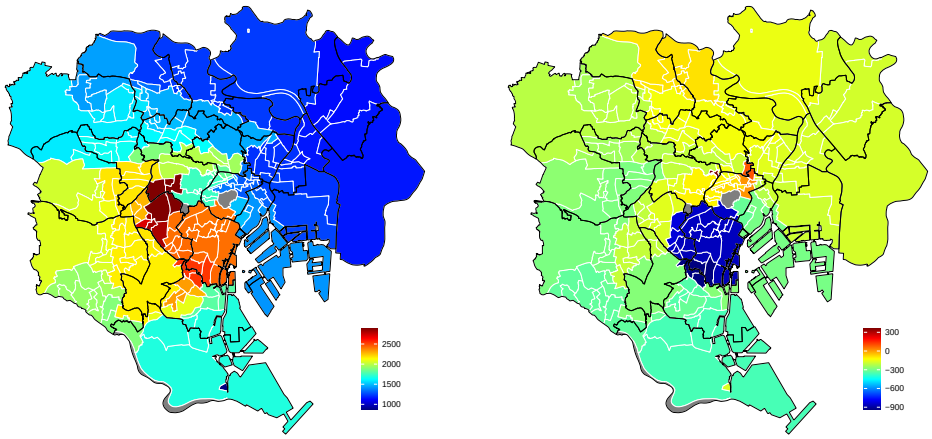
We estimate $\boldsymbol{\beta}$ (= $(\boldsymbol{\beta}'_1, \ldots, \boldsymbol{\beta}'_m)'$) and $\boldsymbol{\gamma}$ for each λ value, and then select the optimal λ value by minimizing the EGCV criterion, where candidate λ values are given by $\lambda_{\max}(3/4)^{(j-1)}$ (j = $1, \ldots, 100$). Figures 5 and 6 are choropleth maps illustrating the estimates of the regres-



(a) Floor area (A1) (b) Building age (A2)

Figure 5. Estimation results 1

sion coefficients for the continuous variables (A1–A4). In these maps, the 852 subregions are covered by 189 subregions. In other words, Tokyo's 23 wards can be expressed by just the 189-submodels. Moreover, for this model, the coefficient of determination is 0.86 and the MER is



(a) Interaction (A3) (b) Walking time (A4)

Figure 6. Estimation results 2

0.065, confirming that our modeling approach successfully captures key features of the data.

Acknowledgment The first author's research was partially supported by JSPS KAKENHI Grant Number JP20H04151. The last author's research was partially supported by JSPS KAKENHI Grant Numbers JP16H03606, JP18K03415, and JP20H04151.

References

Bleakley, K. & Vert, J.-P. (2011). The group fused Lasso for multiple change-point detection. arXiv 1106.4199v1.

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

Ohishi, M., Fukui, K., Okamura, K., Itoh, Y. & Yanagihara, H. (2019). Estimation for spatial effects by using the fused Lasso. Technical Report TR-No. 19-07. Hiroshima Statistical Research Group. Hiroshima.

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.

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.

Xin, B., Kawahara, Y., Wang, Y. & Gao, W. (2014). Efficient generalized fused Lasso and its application to the diagnosis of Alzheimer's disease. Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence. Association for the Advancement of Artificial Intelligence. California. 2163–2169.

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. Statist. Assoc.*, **101**, 1418–1429.

Appendix

A.1. The proof of Lemma 1

First, we show that

$$\bigcup_{j\in E_\ell}\{j\}\times(D_j\backslash E_\ell)\subset\bigcup_{i\in D_\ell^*}\mathcal{J}_{\ell i}.$$

Let (i, s) be an element of the above LHS. Then, the following statement is true:

$$\begin{aligned}(i,s)\in\bigcup_{j\in E_\ell}\{j\}\times(D_j\backslash E_\ell)&\Leftrightarrow\exists j_0\in E_\ell\ s.t.\ (i,s)\in\{j_0\}\times(D_{j_0}\backslash E_\ell)\\&\Leftrightarrow\exists j_0\in E_\ell\ s.t.\ i=j_0\wedge s\in D_{j_0}\backslash E_\ell.\end{aligned}$$

The $s\in D_{j_0}\backslash E_\ell$ leads $s\in F_\ell$ and

$$s\in D_{j_0}\wedge s\notin E_\ell\Leftrightarrow s\in D_{j_0}\wedge\exists! i_0\in\{1,\ldots,b\}\backslash\{\ell\}\ s.t.\ s\in E_{i_0}.$$

These results say $s\in E_{i_0}\cap F_\ell$ and hence $i_0\in D_\ell^*$. Notice that $(i,s)\in\{j_0\}\times E_{i_0}\cap D_{j_0}$. Hence, we have

$$(i,s)\in\bigcup_{i\in D_\ell^*}\mathcal{J}_{\ell i}.$$

Next, we show that

$$\bigcup_{j\in E_\ell}\{j\}\times(D_j\backslash E_\ell)\supset\bigcup_{i\in D_\ell^*}\mathcal{J}_{\ell i}.$$

Let (i, s) be an element of the above RHS. Then, the following statement is true:

$$(i,s)\in\bigcup_{i\in D_\ell^*}\mathcal{J}_{\ell i}\Leftrightarrow\exists i_0\in D_\ell^*\ s.t.\ \left(\exists j_0\in E_\ell\ s.t.\ (i,s)\in\{j_0\}\times E_{i_0}\cap D_{j_0}\right)$$

$$\Rightarrow i = j_0 \wedge s \in E_{i_0} \cap D_{j_0}.$$

Moreover, we found that $s \notin E_\ell$ because $i_0 \in D_\ell^*$. Hence, we have

$$(i, s) \in \bigcup_{j \in E_\ell} \{j\} \times (D_j \backslash E_\ell).$$

Consequently, Lemma 1 is proved.