TR-No. 21-06, Hiroshima Statistical Research Group, 1–17

Coordinate Descent Algorithm of Generalized Fused Lasso Logistic Regression for Multivariate Trend Filtering

Mineaki Ohishi[1]*, **Mariko Yamamura**[2] **and Hirokazu Yanagihara**[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]Department of Statistics, Radiation Effects Research Foundation
5-2 Hijiyama Park, Minami-ku, Hiroshima 732-0815, Japan
[3]Graduate School of Advanced Science and Engineering, Hiroshima University
1-3-1 Kagamiyama, Higashi-Hiroshima 739-8526, Japan

Abstract

Generalized fused Lasso (GFL) is an extension of fused Lasso and performs multivariate trend filtering based on adjacent information among parameters. This paper deals with an optimization problem for GFL logistic regression. Model parameters for the generalized linear model including the logistic regression model are usually optimized by minimizing a linear approximation of an objective function because the minimizer of the objective function cannot be obtained in closed form. In this paper, we propose an algorithm for solving the optimization problem for GFL logistic regression without approximating the objective function, for the purpose of optimizing fast and accurately. Specifically, we derive update equations of a coordinate descent algorithm for solving the optimization problem in closed form. Moreover, we show an example for spatio-temporal data analysis.

(Last Modified: October 6, 2021)

Key words: Coordinate descent algorithm, Generalized fused Lasso, Logistic regression, Spatio-temporal analysis, Trend filtering.

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

1. Introduction

We consider the following logistic regression model:

$$y_i \sim B(m_i, \pi_i), \quad \pi_i = \frac{\exp(\mu_i)}{1 + \exp(\mu_i)} \quad (i = 1, \ldots, n), \tag{1}$$

where μ_i is an unknown parameter. The maximum likelihood estimation method is one of the most basic methods for estimating μ_i, and the estimator is calculated by minimizing a negative log-likelihood function (NLF):

$$\sum_{i=1}^{n} \left[m_i \log\{1 + \exp(\mu_i)\} - y_i \mu_i \right]. \tag{2}$$

Hence, the maximum likelihood estimator (MLE) of μ_i is given by $\hat{\mu}_i = \log\{y_i/(m_i - y_i)\}$. Although an MLE has several good properties, e.g., efficiency, asymptotic normality, and consistency, the MLE for model (1) may induce overfitting because the sample size and the number of parameters are equal. We can expect to avoid such overfitting by estimating under some constraint. One of the typical methods is a penalized estimation method that is based on minimizing the function obtained by adding a penalty term to an NLF. For example, we can use ridge regression (Hoerl & Kennard, 1970) or Lasso (Tibshirani, 1996) (e.g., see Cessie & van Houwelingen, 1992; Shevade & Keerthi, 2003; Pereira *et al.*, 2016, for ridge regression and Lasso for logistic regression). In this paper, we consider multivariate trend filtering (e.g., Tibshirani, 2014) by generalized fused Lasso (GFL; e.g., Tibshirani, 2014; Xin *et al.*, 2014, 2016; Wang *et al.*, 2016; Ohishi *et al.*, 2021), where multivariate trend filtering is an extension of trend filtering (e.g., Leser, 1961; Osborne, 1995; Kim *et al.*, 2009) for multiple factors and GFL is an extension of fused Lasso (Tibshirani *et al.*, 2005) for general adjacent information. Lee *et al.* (2014), Xin *et al.* (2014), Yu *et al.* (2015), and Yamamura *et al.* (2021) studied GFL logistic regression (including the ordinary fused Lasso logistic regression). To estimate the parameters μ_i $(i = 1, \ldots, n)$ in the model (1), we consider minimizing the GFL-penalized NLF defined by adding the GFL penalty to NLF (2):

$$\ell(\boldsymbol{\mu}) = \sum_{i=1}^{n} \left[m_i \log\{1 + \exp(\mu_i)\} - y_i \mu_i \right] + \lambda \sum_{i=1}^{n} \sum_{j \in D_i} w_{ij} |\mu_i - \mu_j|, \tag{3}$$

where $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)'$, $\lambda \geq 0$ is a tuning parameter, $D_i \subseteq \{1, \ldots, n\} \backslash \{i\}$ is an index set expressing adjacent information among individuals and satisfying $j \in D_i \Leftrightarrow i \in D_j$, and $w_{ij} > 0$ is a penalty weight based on Zou (2006) and satisfies $w_{ij} = w_{ji}$. Since the purpose of this paper is to discuss trend filtering, we do not adopt an ℓ_1 penalty to shrink μ_i towards 0, unlike in Lee *et al.* (2014), Xin *et al.* (2014), and Yu *et al.* (2015).

The ordinary fused Lasso, which performs trend filtering by order relation as adjacent information, deals with the following limited adjacent information:

$$D_i = \begin{cases} \{2\} & (i = 1) \\ \{i - 1, i + 1\} & (i = 2, \ldots, n - 1) \\ \{n - 1\} & (i = n) \end{cases}.$$

Such relations can be seen in variables with respect to time, genomic sequence, and so on. On the other hand, GFL can deal with general adjacent information. Figure 1 is an example of

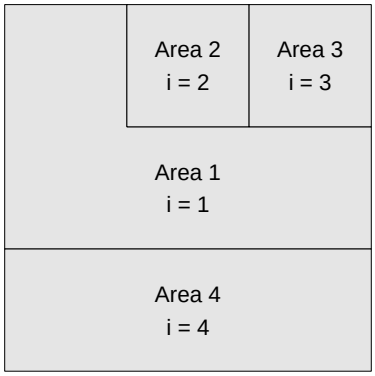

Figure 1. Example for spatial adjacent relationships when $n = 4$

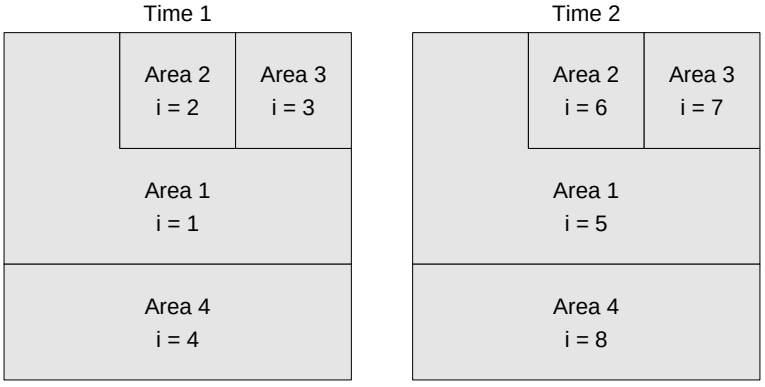

Figure 2. Example for spatio-temporal adjacent relationships when $n = 8$

spatial adjacency when $n = 4$, and the adjacent information is expressed as

$$D_1 = \{2, 3, 4\},\ D_2 = \{1, 3\},\ D_3 = \{1, 2\},\ D_4 = \{1\}.$$

Furthermore, Figure 2 is an example of spatio-temporal adjacency when $n = 8$, which is by adding a time factor to the spatial adjacency in Figure 1. The adjacent information in Figure 2 is expressed as

$$D_1 = \{2, 3, 4, 5\},\ D_2 = \{1, 3, 6\},\ D_3 = \{1, 2, 7\},\ D_4 = \{1, 8\},$$
$$D_5 = \{1, 6, 7, 8\},\ D_6 = \{2, 5, 7\},\ D_7 = \{3, 5, 6\},\ D_8 = \{4, 5\}.$$

Although Figure 1 expresses an example for only one factor and Figure 2 expresses an example for two factors, GFL can deal with the general case of adjacent information based on p factors. For example, it is possible to consider adjacent information based on the three factors space, time, and building age when modeling real estate price. GFL can be applied to various data analyses. For instance, Lee *et al.* (2014) and Xin *et al.* (2014) applied it diagnosing Alzheimer's disease, Yu *et al.* (2015) applied it classifying spectral data, Ohishi *et al.* (2021) applied it estimating regional effects on apartment rents, and Yamamura *et al.* (2021) applied it estimating spatio-temporal trends in crime rate.

Multivariate trend filtering based on GFL has various advantages, such as that optimizations of location, number, and bandwidth of basis functions are not required, as opposed to smoothing by splines. On the other hand, in parameter estimation for a generalized linear model including a logistic regression model, since the estimator cannot usually be obtained in closed form, the parameters are often estimated by minimizing a simple function transformed from an objective function, e.g., a linear approximation. Actually, by minimizing a linear approximation of an objective function, Lee *et al.* (2014) and Yu *et al.* (2015) estimated parameters for fused Lasso logistic regression and Xin *et al.* (2014) and Yamamura *et al.* (2021) estimated parameters for GFL logistic regression. However, in such a case, there is the concern that a gap between the minimizer of the approximation and the true minimizer occurs and that the minimization is slower. Hence, it is mathematically and practically better to minimize an objective function without any approximation. Therefore, in this paper, we propose an algorithm to minimize the objective function (3) without any approximation. Specifically, we focus on a coordinate descent algorithm and derive the update equations in closed form. Moreover, we show an example for spatio-temporal data analysis.

The remainder of the paper is organized as follows. In section 2, we describe a coordinate descent algorithm and derive the closed-form update equations of the coordinate descent algorithm for GFL logistic regression. In section 3, we demonstrate the performance of our algorithm and apply GFL logistic regression to an actual dataset. Section 4 presents a summary of the paper.

2. Main Result

2.1. What is a Coordinate Descent Algorithm?

In this section, we describe the optimization problem for GFL logistic regression, specifi-

cally a coordinate descent algorithm for minimizing the objective function (3). In general, a coordinate descent algorithm finds the minimizer of an objective function by repeating minimization along each coordinate direction. In our problem, we minimize the objective function (3) along each μ_i ($i = 1, \ldots, n$)-direction and repeat this until the solution converges. Friedman *et al.* (2007) and Ohishi *et al.* (2021) proposed coordinate descent algorithms for fused Lasso and GFL, respectively, in the case of a linear regression model. However, as Friedman *et al.* (2007) described, minimizing only along each simple coordinate direction fails in the case of minimizing the objective function in the GFL optimization problem. GFL shrinks a difference $\mu_i - \mu_j$ and the two parameters are often equal. When the current solutions for μ_i and μ_j are equal, the two solutions get stuck and cannot reach the minimizers in a coordinate descent algorithm. To avoid such a problem, the coordinate descent algorithms proposed by Friedman *et al.* (2007) and Ohishi *et al.* (2021) consist of two cycles called a descent cycle and a fusion cycle. The descent cycle performs minimization along each simple coordinate direction as above. Then, the fusion cycle is executed when current solutions for several parameters are equal in the descent cycle. In the fusion cycle, the descent cycle is executed by regarding parameters with equal current solutions as a single parameter. By executing the fusion cycle, we can avoid solutions getting stuck and the objective function can be minimized. Hence, in this paper, the descent cycle and the fusion cycle are adopted as in a similar way to the two previous studies and we derive each update equation in closed form.

First, we consider the descent cycle. In the descent cycle, the objective function (3) is minimized along μ_i-direction. That is, we partially minimize the objective function with respect to μ_i. To do this, we extract terms which depend on μ_i from the objective function. Regarding the penalty term of the objective function, Ohishi *et al.* (2021) derived the following equation:

$$\sum_{i=1}^{n} \sum_{j \in D_i} w_{ij} |\mu_i - \mu_j| = 2 \sum_{j \in D_i} w_{ij} |\mu_i - \mu_j| + \sum_{\ell \neq i}^{n} \sum_{j \in D_\ell \backslash \{i\}} w_{\ell j} |\mu_\ell - \mu_j|.$$

From this equation, the following function is the objective function along the μ_i-direction in the descent cycle:

$$\ell_i(\mu) = m_i \log\{1 + \exp(\mu)\} - y_i \mu + 2\lambda \sum_{j \in D_i} w_{ij} |\mu - \hat{\mu}_j|, \tag{4}$$

where the notation $\hat{\mu}_j$ ($j \in \{1, \ldots, n\} \backslash \{i\}$) means μ_j is fixed.

Next, we consider the fusion cycle. In the fusion cycle, the objective function (3) is minimized by regarding parameters with equal current solutions as a single parameter. Now suppose that we have $\hat{\mu}_1, \ldots, \hat{\mu}_n$ as current solutions of $\mu_1, \ldots, \mu_n$ and that there exist combinations of parameters with equal solutions. Then, let $\hat{\xi}_1, \ldots, \hat{\xi}_b$ ($b < n$) be distinct values of $\hat{\mu}_1, \ldots, \hat{\mu}_n$ and define the following index sets:

$$E_l = \left\{ i \in \{1, \ldots, n\} \mid \hat{\mu}_i = \hat{\xi}_l \right\} \quad (l = 1, \ldots, b).$$

These index sets express the combinations of parameters with equal solutions, and it is clear that $E_l \neq \emptyset$, $E_l \cap E_j = \emptyset$ $(l \neq j)$, and $\cup_{l=1}^{b} E_l = \{1, \ldots, n\}$. For example, $E_1 = \{1, 2, 3\}$ means $\hat{\mu}_1 = \hat{\mu}_2 = \hat{\mu}_3 = \hat{\xi}_1$. Then, parameters for individuals in E_l, that is μ_i $(i \in E_l)$, are regarded as a single parameter ξ_l and the objective function (3) is minimized along the ξ_l-direction. That is, we partially minimize the objective function with respect to ξ_l. To do this, we extract terms which depend on ξ_l from the objective function. The first term of the objective function can be decomposed as

$$\begin{aligned}
&\sum_{i=1}^{n} \left[m_i \log\{1 + \exp(\mu_i)\} - y_i \mu_i \right] \\
&\qquad = \sum_{i \in E_l} m_i \log\{1 + \exp(\xi_l)\} - \sum_{i \in E_l} y_i \xi_l + \sum_{i \notin E_l} \left[m_i \log\{1 + \exp(\mu_i)\} - y_i \mu_i \right],
\end{aligned}$$

where $i \notin E_l$ in the last term of the above means $i \in \{1, \ldots, n\} \backslash E_l$. Regarding the penalty term of the objective function, at the beginning, we define index sets expressing adjacent information for E_l. Let D_l^* be the index set defined by

$$D_l^* = \{ j \in \{1, \ldots, b\} \backslash \{l\} \mid E_j \cap F_l \neq \emptyset \}, \quad F_l = \bigcup_{i \in E_l} D_i \backslash E_l,$$

where D_l^* satisfies $D_l^* \neq \emptyset$ and $j \in D_l^* \Leftrightarrow l \in D_j^*$, and F_l is an index set expressing individuals which are adjacent to $i \in E_l$. The definition of D_l^* means E_l is adjacent to E_j $(j \in D_l^*)$. After that, we transform the weights w_{ij} for $|\mu_i - \mu_j|$ $(j \in D_i)$ to weights for $|\xi_l - \xi_j|$ $(j \in D_l^*)$. For $j \in D_l^*$, we define this as

$$w_{lj}^* = \sum_{(i,s) \in \mathcal{J}_{lj}} w_{is}, \quad \mathcal{J}_{lj} = \bigcup_{i \in E_l} \{i\} \times (E_j \cap D_i).$$

Then, from Ohishi *et al.* (2021), the following equation holds:

$$\sum_{i=1}^{n} \sum_{j \in D_i} w_{ij} |\mu_i - \mu_j| = 2 \sum_{j \in D_l^*} w_{lj}^* |\xi_l - \xi_j| + \sum_{i \notin E_l} \sum_{j \in D_i \backslash E_l} w_{ij} |\mu_i - \mu_j|.$$

From the above, the following function is the objective function along the ξ_l-direction in the fusion cycle:

$$\ell_l^*(\xi) = \sum_{i \in E_l} m_i \log\{1 + \exp(\xi)\} - \sum_{i \in E_l} y_i \xi + 2\lambda \sum_{j \in D_l^*} w_{lj}^* |\xi - \hat{\xi}_j|, \tag{5}$$

where the notation $\hat{\xi}_j$ means ξ_j is fixed.

Thus, we have to minimize the two functions (4) and (5) to obtain the update equations for the descent cycle and the fusion cycle, respectively. Fortunately, the two functions are essentially equal. Hence, it is sufficient to minimize the following function, which is a generalization of (4) and (5):

$$f(x) = m\log\{1+\exp(x)\} - yx + 2\lambda\sum_{j=1}^{r} w_j|x - z_j| \quad (x \in \mathbb{R}), \tag{6}$$

where m, y, λ, w_j are positive constants, z_j is a constant, and $m \geq y$.

2.2. Update Equations

In this section, we derive the update equations of the coordinate descent algorithm for GFL logistic regression by minimizing (6). Since (6) has multiple non-differentiable points, first, we judge it is minimized at each non-differentiable point or not by considering a subdifferential of f at each non-differentiable point. A subdifferential of f at $\tilde{x} \in \mathbb{R}$ is given by

$$\partial f(\tilde{x}) = \{u \in \mathbb{R} \mid f(x) \geq f(\tilde{x}) + u(x - \tilde{x})\ (\forall x \in \mathbb{R})\} = \left[g_-(\tilde{x}), g_+(\tilde{x})\right],$$

where $g_-(x)$ and $g_+(x)$ are left and right derivatives defined by

$$g_-(x) = \lim_{\delta \to -0} g(x,\delta), \quad g_+(x) = \lim_{\delta \to +0} g(x,\delta), \quad g(x,\delta) = \frac{f(x+\delta) - f(x)}{\delta}.$$

The left and right derivatives at a non-differentiable point z_j ($j \in \{1,\ldots,r\}$) are given by

$$g_-(z_j) = \frac{m\exp(z_j)}{1+\exp(z_j)} - y - 2\lambda w_j + 2\lambda\sum_{l\neq j}^{r} w_l \operatorname{sign}(z_j - z_l),$$
$$g_+(z_j) = \frac{m\exp(z_j)}{1+\exp(z_j)} - y + 2\lambda w_j + 2\lambda\sum_{l\neq j}^{r} w_l \operatorname{sign}(z_j - z_l).$$

Then, (6) is minimized at $x = z_{j_\star}$ if there exists $j_\star \in \{1,\ldots,r\}$ such that $0 \in \partial f(z_{j_\star})$. Notice that (6) is convex. Hence, $j_\star$ is unique if it exists. Next, we consider when $j_\star$ does not exist. Let t_j ($j = 1,\ldots,r$) be the jth order statistic of $z_1,\ldots,z_r$ and let $s(x) = (\operatorname{sign}(g_-(x)), \operatorname{sign}(g_+(x)))$. Then, only one of the following statements is true.

(S1) $\forall j \in \{1,\ldots,r\},\ s(t_j) = (1,1)$.

(S2) $\forall j \in \{1,\ldots,r\},\ s(t_j) = (-1,-1)$.

(S3) $\exists! j_* \in \{1,\ldots,r-1\}\ s.t.\ \forall j \in \{1,\ldots,r\},\ s(t_j) = \begin{cases}(-1,-1) & (j \leq j_*)\\ (1,1) & (j > j_*)\end{cases}$.

The above statements tell us which interval includes the minimum. The interval including the minimum is given by

$$\mathcal{I} = (\mathcal{I}_L, \mathcal{I}_R) = \begin{cases} (-\infty, t_1) & \text{(S1)} \\ (t_r, \infty) & \text{(S2)} \\ (t_{j_*}, t_{j_*+1}) & \text{(S3)} \end{cases}.$$

Hence, it is sufficient to search the minimizer in the interval $\mathcal{I}$.

When $x \in \mathcal{I}$, we can easily rewrite (6) in non-absolute form. That is, $x - z_j$ is positive for $j \in J_+$ and negative for $j \in J_-$, where J_+ and J_- are index sets given by

$$J_+ = \{j \in \{1, \dots, r\} \mid z_j \le \mathcal{I}_L\}, \quad J_- = \{j \in \{1, \dots, r\} \mid \mathcal{I}_R \le z_j\}.$$

Then, we have

$$\sum_{j=1}^{r} w_j |x - z_j| = \sum_{j \in J_+} w_j (x - z_j) + \sum_{j \in J_-} w_j (z_j - x) = \tilde{w} x - u,$$
$$\tilde{w} = \sum_{j \in J_+} w_j - \sum_{j \in J_-} w_j, \quad u = \sum_{j \in J_+} w_j z_j - \sum_{j \in J_-} w_j z_j.$$

Hence, when $x \in \mathcal{I}$, (6) is expressed as

$$f(x) = m \log\{1 + \exp(x)\} + cx - 2\lambda u, \quad c = 2\lambda \tilde{w} - y,$$

and we have

$$\frac{d}{dx} f(x) = \frac{m \exp(x)}{1 + \exp(x)} + c \quad (x \in \mathcal{I}).$$

From the above, a stationary point which is a point satisfying $df(x)/dx = 0$ uniquely exists and this is the minimizer of (6). That is, (6) is minimized at $x = \log\{-c/(m + c)\}$.

Consequently, the minimizer of $f(x)$ is given by the following theorem.

Theorem 1. *Let $\hat{x}$ be the minimizer of $f(x)$. Then, $\hat{x}$ is given by*

$$\hat{x} = \begin{cases} z_{j_\star} & (j_\star \textit{ exists}) \\ \log \dfrac{-c}{m + c} & (j_\star \textit{ does not exist}) \end{cases}.$$

Thus, the theorem gives us the minimizer of $f(x)$ in closed form. By applying Theorem 1 to equations (4) and (5), which are the objective functions for the descent cycle and the fusion cycle, respectively, we can obtain the update equations of the coordinate descent algorithm for minimizing (3), both in closed form.

Here, we give the update equation for the descent cycle; i.e., we apply Theorem 1 to equation (4). Let $j_{i\star}$ be an index defined by

$$j_{i\star} \in D_i \ s.t.\ 0 \in \partial \ell_i(\hat{\mu}_{j_{i\star}}),$$

where $\partial \ell_i(\cdot)$ is a subdifferential of ℓ_i. If $j_{i\star}$ exists, it is unique. If not, the interval $\mathcal{I}_i = (\mathcal{I}_{iL}, \mathcal{I}_{iR})$ including the minimum is defined by checking three statements (S1), (S2), and (S3). Then, the update equation for the descent cycle is given in closed form as follows:

$$\hat{\mu}_i = \begin{cases} \hat{\mu}_{j_{i\star}} & (j_{i\star} \text{ exists}) \\ \log \dfrac{-c_i}{m_i + c_i} & (j_{i\star} \text{ does not exist}) \end{cases},$$

where $c_i = 2\lambda \tilde{w}_i - y_i$ and $\tilde{w}_i$ is given by

$$\tilde{w}_i = \sum_{j \in J_{i+}} w_{ij} - \sum_{j \in J_{i-}} w_{ij}, \quad J_{i+} = \{j \in D_i \mid \hat{\mu}_j \leq \mathcal{I}_{iL}\}, \quad J_{i-} = \{j \in D_i \mid \mathcal{I}_{iR} \leq \hat{\mu}_j\}.$$

In the process updating μ_i, if μ_i is updated by $\hat{\mu}_{j_0}$ ($j_0 \in D_i$), the current solutions for the individuals i and j_0 are equal and the two individuals are joined together. The update equation for the fusion cycle is obtained in closed form in a similar way.

2.3. Optimization Algorithm

In the previous section, we derived the closed-form update equations of the coordinate descent algorithm for GFL logistic regression. Using these equations, the algorithm for minimizing the objective function (3) is summarized as

Algorithm 1.

Step 1. *(Initialization) Set λ and an initial vector of $\boldsymbol{\mu}$.*

Step 2. *(Descent cycle) For $i = 1, \ldots, n$, update μ_i by applying Theorem 1 to* (4)*, and define b. If $b < n$, go to Step 3. If not, go to Step 4.*

Step 3. *(Fusion cycle) For $l = 1, \ldots, b$, update ξ_l ($= \mu_i, i \in E_l$) by applying Theorem 1 to* (5)*.*

Step 4. *(Convergence judgment) If the solution converges, the algorithm terminates. If not, return to Step 2.*

When executing Algorithm 1, we have to decide a value for λ. The λ is a tuning parameter to adjust the strength of the penalty term, i.e., the degree of smoothing. For example, if λ is too small, most parameters will not be joined together and overfitting will not be improved. In

opposition, if λ is too large, most parameters will be joined together and model fitting will deteriorate. Hence, the optimization of λ is an important problem for obtaining better estimates. One simple strategy is executing Algorithm 1 for each of some candidate set of λ values and selecting the optimal value by, e.g., minimizing a model selection criterion.

Our idea is to define $\lambda_{\max}$ and to divide an interval $(0, \lambda_{\max}]$. Here, we give how to define $\lambda_{\max}$ based on the update equation for the descent cycle. More parameters are joined together as λ increases and finally, all parameters are equal. It is best that $\lambda_{\max}$ be the value such that all parameters are equal. Let $\hat{\mu}_{\max}$ be the estimator when all parameters are equal, which is given by $\hat{\mu}_{\max} = \log\{\sum_{i=1}^{n} y_i / \sum_{i=1}^{n}(m_i - y_i)\}$. Then, we define $\lambda_{\max}$ such that all parameters are updated by $\hat{\mu}_{\max}$ when all initial values are $\hat{\mu}_{\max}$. Such a condition is given by

$$\frac{m_i \exp(\hat{\mu}_{\max})}{1 + \exp(\hat{\mu}_{\max})} - y_i - 2\lambda w_i \leq 0 \leq \frac{m_i \exp(\hat{\mu}_{\max})}{1 + \exp(\hat{\mu}_{\max})} - y_i + 2\lambda w_i \quad (\forall i \in \{1, \ldots, n\}),$$

where $w_i = \sum_{j \in D_i} w_{ij}$. Therefore, a sufficient condition that μ_i be updated to $\hat{\mu}_{\max}$ is given by

$$\lambda \geq \lambda_{i,\max} = \frac{|(m_i - y_i)\exp(\hat{\mu}_{\max}) - y_i|}{2w_i\{1 + \exp(\hat{\mu}_{\max})\}}.$$

Consequently, we define $\lambda_{\max}$ as

$$\lambda_{\max} = \max_{i \in \{1,\ldots,n\}} \lambda_{i,\max}. \tag{7}$$

3. Numerical Studies

3.1. Simulations

First, we compare our algorithm to an existing algorithm (used in Yamamura *et al.*, 2021; it minimizes a linear approximation of the objective function) with respect to runtime and minimum. We randomly define adjacent information D_i ($i = 1, \ldots, n$) among n individuals and true joins E_j^* ($j = 1, \ldots, b^* = n/10$) of the individuals, where $\max_{i \in \{1,\ldots,n\}} r_i \leq 10$ and $r_i = \#(D_i)$. Then, we generate simulation data by

$$y_i \sim B(m_i, \pi_j), \quad \pi_j = \frac{\exp(\xi_j^*)}{1 + \exp(\xi_j^*)}, \quad \xi_j^* = (-1)^j / j \quad (i \in E_j^*,\ j = 1, \ldots, b^*).$$

In this simulation, the minimum of the objective function (3) and runtime of an algorithm are separately compared via Monte Carlo simulation with 1,000 iterations.

Table 1 shows results for $n \in \{300, 1{,}000\}$, under a fixed λ, where the numbers of trials m_i ($i = 1, \ldots, n$) were defined by a common setting or a random setting—the common setting is $m_1 = \cdots = m_n = m_0 \in \{1{,}000, 10{,}000\}$; and the random setting (denoted by $m_0 =$ "random")

Table 1. Minimization and runtime

				runtime (s)		degrees of freedom	
n	m_0	δ	min (%)	proposed	existing	proposed	existing
300	1,000	100	100	0.99	0.83	128.45	1.65
		1,000	100	0.47	1.21	231.99	8.85
		10,000	100	0.14	2.95	277.69	73.68
	10,000	100	100	1.02	1.13	82.74	2.20
		1,000	100	0.59	1.45	206.40	7.45
		10,000	100	0.14	2.33	268.26	41.32
	random	100	100	1.10	1.34	98.86	1.91
		1,000	100	0.63	2.06	213.79	8.33
		10,000	100	0.28	7.94	269.59	56.18
1,000	1,000	100	100	6.90	2.31	310.06	1.17
		1,000	100	3.16	2.75	686.90	3.78
		10,000	100	0.78	7.44	911.87	44.30
	10,000	100	100	6.60	4.40	192.95	1.31
		1,000	100	3.81	4.71	603.69	4.54
		10,000	100	0.95	6.80	860.66	21.20
	random	100	100	7.04	6.54	125.65	1.12
		1,000	100	5.73	7.91	530.91	2.97
		10,000	100	2.74	16.67	821.38	18.33

is defining m_i ($i = 1, \ldots, n$) by sampling without replacement from $\{100, 101, \ldots, 10{,}000\}$. Regarding the value of λ, we set $\lambda = \lambda_{\max}/\delta$ for $\delta \in \{100, 1{,}000, 10{,}000\}$, where $\lambda_{\max}$ is given by (7). The table displays three indexes: *min*, *runtime*, and *degrees of freedom*, where *min* expresses the ratio (%) that the minimum obtained by our algorithm is smaller than that obtained by the existing algorithm, *runtime* expresses the mean value of runtime, and *degrees of freedom* expresses the mean value of the degrees of freedom of the estimates. Since the values of *min* are all 100%, we found that our algorithm can better minimize the objective function and provide better estimates than the existing algorithm. The values of *degrees of freedom* tell us that the existing algorithm joins parameters together at smaller λ. In particular, we can guess that most parameters were equal at $\delta = 100, 1{,}000$ in the existing algorithm. Moreover, in the case that *degrees of freedom* is extremely small, the existing algorithm was fast and sometimes faster than our algorithm. We can guess the reasons to be as follows: in the existing algorithm, parameters are easily joined together, all parameters are often equal, and joined parameters are hardly separated in the descent cycle. Although it can be seen that there are many gaps between our algorithm and the existing algorithm, we consider that the results by our algorithm are more trustworthy than those of the existing algorithm because our algorithm better minimized the objective function.

Next, we evaluate the selection probability of the true joins under the optimal tuning parameter selected by minimizing BIC (Schwarz, 1978), where the candidates for the optimal tuning parameter are the 100 points defined by $\lambda_{\max}(3/4)^{(j-1)}$ ($j = 1, \ldots, 100$). We consider the same situation as above except for the true parameters: here, the true parameters are defined by

$$\xi_j^* = -2 + \frac{4(j-1)}{b^* - 1} \quad (j = 1, \ldots, b^*).$$

Moreover, the numbers of trials are defined by $m_1 = \cdots = m_n = n\zeta$ and $\zeta \in \{1{,}000, 10{,}000, 100{,}000\}$. Table 2 shows the results for selection probability (%) via 1,000 iterations. From the table, we found that the selection probability increases as the numbers of trials increase.

Table 2. Selection probability of true join

		Selection probability (%)	
b^*	ζ	$n = 300$	$n = 1{,}000$
$n/10$	1,000	45.8	18.0
	10,000	88.1	84.2
	100,000	92.1	95.7
$\lfloor n/3 \rfloor$	1,000	12.5	2.4
	10,000	76.5	64.0
	100,000	89.0	91.3

$\lfloor \cdot \rfloor$ is the floor function.

3.2. Real Data Analysis

In this section, we apply GFL logistic regression to spatio-temporal analysis. We use a dataset about the crime rate in the Kinki region of Japan. The dataset is same as that used in Yamamura *et al.* (2021) and comprises the municipality data K4201 and A1101 of the System of Social and Demographic Statistics downloaded from e-Stat of the Statistics Bureau of the Ministry of Internal Affairs and Communications (see `https://www.e-stat.go.jp/`). The data items are the number of crimes (K4201), the total population (A1101), city, and year, where city indicates one of the 227 cities of the Kinki region (see Figure 3) and year indicates one of the 14 years between 1995 and 2008. Note that the total population was reported for 1995, 2000, and 2005. Therefore, for the years in which there were no observed values for the total population, the values for the most recent past year are used. Hence, the sample size is $n = 227 \times 14 = 3{,}178$. In this analysis, we focus on a spatio-temporal factor, the city and year pair, and estimate the spatio-temporal trend for the crime rate. That is,

Figure 3. The 227 cities in the Kinki region

we consider a logistic regression model in which m_i is the population and y_i is the number of crimes for spatio-temporal i, and we estimate μ_i expressing spatio-temporal trend by GFL. The adjacent information in the dataset is obtained as combined adjacent information of that for city and year, similar to as in Figure 2. Regarding city, for example, Osaka is adjacent to 11 cities, which are Sakai, Toyonaka, Suita, Moriguchi, Yao, Matsubara, Daito, Kadoma, Settsu, Higashi-Osaka, and Amagasaki. Regarding year, for example, 2000 is adjacent to 1999 and 2001. Therefore, regarding the spatio-temporal relation, for example, Osaka in 2000 is adjacent to Osaka in 1999 and 2001 and 11 cities in 2000 which are adjacent to Osaka. Moreover, the optimal tuning parameter is selected from 100 points, which are the same as those in section 3.1, by minimizing BIC.

Figures 4 and 5 are choropleth maps showing the estimates of the crime rate for each year. From the figures, the crime rates of the central part tend to be high for any year and particularly, the crime rates are high as a whole in the early 2000s. Figure 6 shows crime rate estimates for 8 cities, which are the prefectural capitals of the prefectures of the Kinki region, as representative cities. Since years are not joined together very much in the figure, we can consider that the trend of time changes is large changes each year.

4. Conclusion

In this paper, we proposed a coordinate descent algorithm for GFL logistic regression and derived the update equations in closed form. Although parameters are often estimated by minimizing an approximation of an objective function, our algorithm can minimize the objective function without any approximation. Since our algorithm does not use approximation

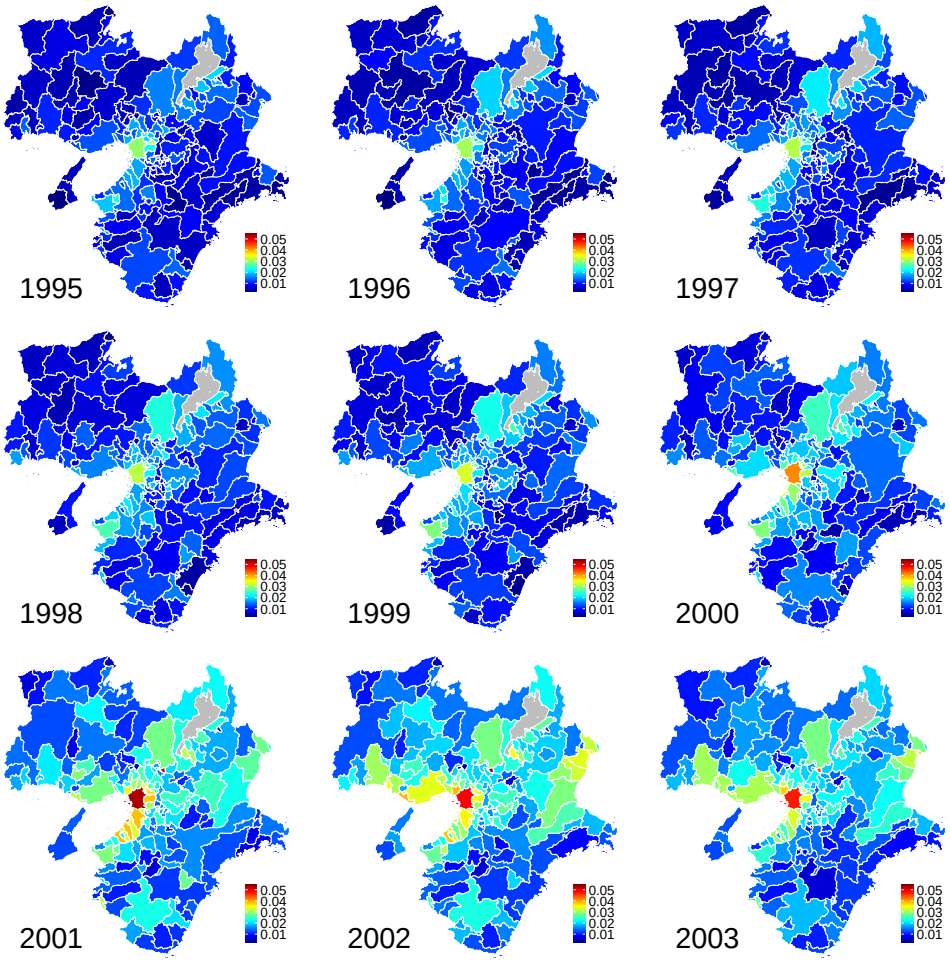

Figure 4. Crime rate estimates for each year (1/2)

and the minimizer along a coordinate direction is obtained exactly, our algorithm was able to minimize the objective function better than the existing algorithm (which minimizes a linear approximation of the objective function) in simulations. Moreover, although we applied GFL to multivariate trend filtering based on two factors—time and space—in a real data analysis, GFL can deal with the general case of multivariate trend filtering based on p factors.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP18K03415, JP20H04151, JP21K13834. The Radiation Effects Research Foundation (RERF), Hiroshima and Nagasaki,

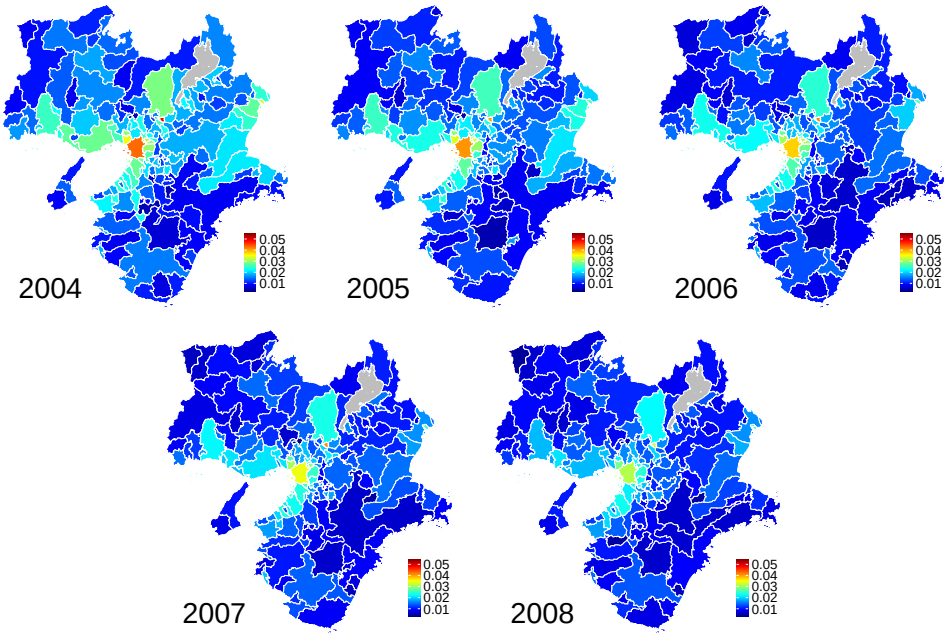

Figure 5. Crime rate estimates for each year (2/2)

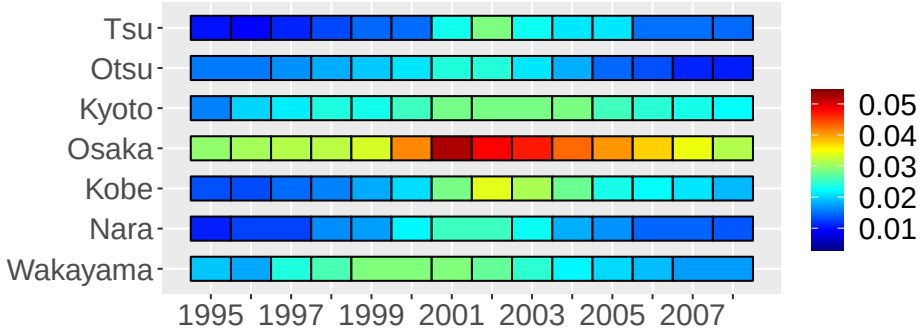

Figure 6. Crime rate estimates for each city

Japan is a public interest foundation funded by the Japanese Ministry of Health, Labor and Welfare (MHLW) and the US Department of Energy (DOE). This publication was supported by RERF. The views of the authors do not necessarily reflect those of the two governments.

References

Cessie, S. L. & van Houwelingen, J. C. (1992). Ridge estimators in logistic regression. *J. R. Stat. Soc. Ser.* **C**. *Appl. Stat.*, **41**, 191–201.

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

Hoerl, A. E. & Kennard, R. W. (1970). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, **12**, 55–67.

Kim, S.-J., Koh, K., Boyd, S. & Gorinevsky, D. (2009). ℓ_1 trend filtering. *SIAM Rev.*, **51**, 339–360.

Lee, S. H., Yu, D., Bachman, A. H., Lim, J. & Ardekani, B. A. (2014). Application of fused Lasso logistic regression to the study of corpus callosum thickness in early alzheimer's disease. *J. Neurosci. Methods*, **221**, 78–84.

Leser, C. (1961). A simple method of trend construction. *J. R. Stat. Soc. Ser.* **B**. *Stat. Methodol.*, **23**, 91–107.

Ohishi, M., Fukui, K., Okamura, K., Itoh, Y. & Yanagihara, H. (2021). Coordinate optimization for generalized fused Lasso. *Comm. Statist. Theory Methods*, in press.

Osborne, D. (1995). Moving average detrending and the analysis of business cycles. *Oxford Bull. Econom. Statist.*, **57**, 547–558.

Pereira, J. M., Basto, M. & da Silva, A. F. (2016). The logistic Lasso and ridge regression in predicting corporate failure. *Procedia Econ. Financ*, **39**, 634–641.

Schwarz, G. (1978). Estimating the dimension of a model. *Ann. Statist.*, **6**, 461–464.

Shevade, S. K. & Keerthi, S. S. (2003). A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics*, **19**, 2246–2253.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *J. R. Stat. Soc. Ser.* **B**. *Stat. Methodol.*, **58**, 267–288.

Tibshirani, R. J. (2014). Adaptive piecewise polynomial estimation via trend filtering. *Ann. Statist.*, **42**, 285–323.

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

Wang, Y. X., Sharpnack, J., Smola, A. J. & Tibshirani, R. J. (2016). Trend filtering on graphs. *J. Mach. Learn. Res*, **17**, 3651–3691.

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. AAAI press. California. 2163–2169.

Xin, B., Kawahara, Y., Wang, Y., Hu, L. & Gao, W. (2016). Efficient generalized fused Lasso and its applications. *ACM T. Intel. Syst. Tec.*, **7**, 1–22.

Yamamura, M., Ohishi, M. & Yanagihara, H. (2021). Spatio-temporal adaptive fused Lasso for proportion data. I. Czarnowski, R. J. Howlett & L. C. Jain, eds, Intelligent Decision Technologies. Springer Singapore. Singapore. 479–489.

Yu, D., Lee, S. J., Lee, W. J., Kim, S. C., Lim, J. & Kwon, S. W. (2015). Classification of spectral data using fused lasso logistic regression. *Chemometr. Intell. Lab.*, **142**, 70–77.

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