Solving Fused Penalty Estimation Problems via Block Splitting Algorithms

Tso-Jung Yen

Reporter: Ren Xiaonan

2021 年 11 月 6 日

Contents

Concerned problem

Concerned with the following penalized estimation problem:

$$\hat{\beta} = \arg\min_{\beta} \left\{ \sum_{i=1}^{m} l_i\left(\beta_i\right) + \sum_{(i,j)\in\mathcal{H}} \lambda_{ij} g_{ij}\left(\beta_i - \beta_j\right) \right\}$$

where $\hat{\beta} = \left(\hat{\beta}_1, \hat{\beta}_2, \cdots, \hat{\beta}_m\right)$.

- $l_i\left(\beta_i\right)$: loss function associated with data from data center i.
- β_i: p-dimensional vector.
- $\mathcal{H}$: undirected graph.
- $\lambda_{ij} \geq 0$: tuning parameter.

Concerned problem

Concerned with the following penalized estimation problem:

$$\hat{\beta} = \arg\min_{\beta} \left\{ \sum_{i=1}^{m} l_i\left(\beta_i\right) + \sum_{(i,j)\in\mathcal{H}} \lambda_{ij} g_{ij}\left(\beta_i - \beta_j\right) \right\}$$

where $\hat{\beta} = \left(\hat{\beta}_1, \hat{\beta}_2, \cdots, \hat{\beta}_m\right)$.

- $g_{ij}\left(\beta_i - \beta_j\right)$: a function of the difference between β_i and β_j.

  A commonly-seen example is the l_q-norm distance:
  $g_{ij}\left(\beta_i - \beta_j\right) = \left\|\beta_i - \beta_j\right\|_q$ for $q \in [1, \infty]$.

  $q = 1$ is called the fused lasso estimator.

  $q = 2$ is called the fused group lasso estimator.

Problem setting

When $g_{ij}(\beta_i - \beta_j)$ is not separable in terms of β_i and β_j:

Reformulate the optimization problem and then solve the reformulated optimization problem via the following **iterative scheme**.

The iterative scheme is an example of the Alternating Direction Method of Multipliers (ADMM) (Boyd et al., 2011).

Problem setting

$$\beta^{r+1} = \arg\min_{\beta} \left(\sum_{i=1}^{m} l_i\left(\beta_i\right) + \frac{\rho}{2} \left\| G\beta - \gamma^r + \eta^r \right\|_2^2 \right)$$

$$\gamma^{r+1} = \arg\min_{\gamma} \left(\sum_{(i,j)\in\mathcal{H}} g_{ij}\left(\gamma_{ij}\right) + \frac{\rho}{2\lambda_{ij}} \left\| \gamma - G\beta^{r+1} - \eta^r \right\|_2^2 \right)$$

$$\eta^{r+1} = \eta^r + G\beta^{r+1} - \gamma^{r+1}$$

where $\gamma = \{\gamma_{ij}\}_{(i,j)\in\mathcal{H}}$, $\eta = \{\eta_{ij}\}_{(i,j)\in\mathcal{H}}$, G is a matrix such that $\left\| G\beta - \gamma^r + \eta^r \right\|_2^2 = \sum_{(i,j)\in\mathcal{H}} \left\| \beta_i - \beta_j - \gamma_{ij}^r + \eta_{ij}^r \right\|_2^2$, and r is the iteration number. Here G is a $q_{\mathcal{H}}p \times mp$ matrix, where $q_{\mathcal{H}} = |\mathcal{H}|$ is the number of edges of $\mathcal{H}$.

The coupling quadratic term is not separable.

Bottleneck

Also consider: the loss function $l_i(\beta_i) = 2^{-1}\|y_i - X_i\beta_i\|_2^2$, where y_i is an n_r dimensional vector and X_i is an $n_i \times p$ matrix. The first line has a closed-form representation

$$\beta^{r+1} = \left(X^T X + \rho G^T G\right)^{-1}\left[X^T y + \rho G^T\left(\gamma^r - \eta^r\right)\right]$$

where $X = \mathrm{diag}(X_1, X_2, \cdots, X_m)$ and $y = (y_1, y_2, \cdots, y_m)$, $X^T X + \rho G^T G$ is an $mp \times mp$ matrix.

$X^T X + \rho G^T G$ costs $O\left(np^2\right) + O(\mathrm{nnz}(G) \cdot mp) + O\left(mp^2\right)$ flops, where $n = \sum_{i=1}^m n_i, n_i$ is the number of rows of X_i, and $\mathrm{nnz}(G)$ is the number of non-zero valued elements in G.

Bottleneck

1. Direct method by performing inverse of $X^T X + \rho G^T G$ via the GaussJordan elimination. Cost $O\left(m^3 p^3\right)$.
2. Use the Cholesky-forward-backward-substitution.

$$V z^{r+1} = X^T y + \rho G^T \left(\gamma^r - \eta^r\right)$$
$$V^T \beta^{r+1} = z^{r+1}$$

where V is the lower triangular matrix associated with the Cholesky decomposition of the form $VV^T = X^T X + \rho G^T G$.

Run fast due to the triangular structure of V. Computing the Cholesky decomposition costs $O\left(m^3 p^3\right)$ flops.

Bottleneck

Recently ADMM-based algorithms for solving the similar optimization problem.

1. Hallac (2015) proposed an ADMM algorithm by introducing a set of auxiliary variables to decouple the linear constraints and derived a closed form representation for the proximal operator of the l_2-norm distance function.
2. Ramdas and Tibshirani (2016) solved the l_1 fused lasso problem by transforming it to its dual problem and used a clever linear algebra technique to decouple the Gram matrix $G^T G$ for fast computation.
3. Zhu (2017) reformulated the problem by introducing a diagonal matrix $D \succ G^T G$ using the pre-conditioned technique and adopting an iterative scheme based on the idea of primal-dual algorithms. This iterative scheme can run fast since D is a diagonal matrix

This paper

1. Without carrying out numerical computation involving the linear operator G.
2. Computational cost.
3. Parallel computing.
4. Convergence properties.

The primal problem

Redefine the optimization problem:

$$\min_{p,a,\theta} \sum_{i=1}^{m} l_i(\beta_i) + \sum_{(i,j)\in\mathcal{H}} \lambda_{ij} g_{ij}(\alpha_{ij})$$

subject to $\alpha_{ij} = \theta_{ij} - \theta_{ji}$ for $(i,j) \in \mathcal{H}$

$\theta_{ij} = \beta_i$ for $j \in \mathcal{N}(i)$ and $i = 1, 2, \cdots, m$

where $\mathcal{N}(i) = \{j : (i,j) \in \mathcal{H} \text{ or } (j,i) \in \mathcal{H}\}$.

Call the constrained optimization problem the primal problem: $\{\beta_i\}_{i=1}^m$ and $\{\alpha_{ij}\}_{(i,j)\in H}$ the primal variables, and $\{\theta_{ij}, \theta_{ji}\}_{(i,j)\in\mathcal{H}}$ the auxiliary variables.

The primal problem

Derive the Lagrangian of the primal problem:

$$
\begin{aligned}
& L(\beta, \alpha, \theta, \tau, \xi) \\
& = \sum_{i=1}^{m} l_i\left(\beta_i\right) + \sum_{(i,j)\in\mathcal{H}} \lambda_{ij} g_{ij}\left(\alpha_{ij}\right) \\
& + \sum_{(i,j)\in\mathcal{H}} \rho\left\langle \tau_{ij}, \alpha_{ij} - \left(\theta_{ij} - \theta_{ji}\right)\right\rangle + \sum_{i=1}^{m} \sum_{j\in\mathcal{N}(i)} \rho\left\langle \xi_{ij}, \beta_i - \theta_{ij}\right\rangle
\end{aligned}
$$

where $\{\tau_{ij}\}_{(i,j)\in\mathcal{H}}$ and $\{\xi_{ij}, \xi_{ji}\}_{(i,j)\in\mathcal{H}}$ are the dual variables, $\rho \geq 0$ is a scale parameter.

The primal problem

Derive an augmented Lagrangian by introducing two sets of quadratic coupling terms to the constraints $\alpha_{ij} = \theta_{ij} - \theta_{ji}$ for $(i,j) \in \mathcal{H}$, and $\theta_{ij} = \beta_i$ for $j \in \mathcal{N}(i)$:

$$
\begin{aligned}
L_{\text{aug}}\left(\beta, \alpha, \theta, \tau, \xi\right) = & \sum_{i=1}^{m} l_i\left(\beta_i\right) + \sum_{(i,j)\in\mathcal{H}} \lambda_{ij} g_{ij}\left(\alpha_{ij}\right) \\
& + \frac{\rho}{2} \sum_{(i,j)\in\mathcal{H}} \left\|\alpha_{ij} - \left(\theta_{ij} - \theta_{ji}\right) + \tau_{ij}\right\|_2^2 - \frac{\rho}{2} \sum_{(i,j)\in\mathcal{H}} \left\|\tau_{ij}\right\|_2^2 \\
& + \frac{\rho}{2} \sum_{i=1}^{m} \sum_{j\in\mathcal{N}(i)} \left\|\beta_i - \theta_{ij} + \xi_{ij}\right\|_2^2 - \frac{\rho}{2} \sum_{i=1}^{m} \sum_{j\in\mathcal{N}(i)} \left\|\xi_{ij}\right\|_2^2
\end{aligned}
$$

Iterative scheme

Propose the following iterative scheme by incorporating mixing strategy:

1. $\beta_i^{r+1} = \arg\min_{\beta_i} \left(l_i(\beta_i) + \frac{\rho|\mathcal{N}(i)|}{2} \left\| \beta_i - \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \left(\theta_{ij}^r - \xi_{ij}^r \right) \right\|_2^2 \right)$ for $i = 1, 2, \cdots, m$
2. $\alpha_{ij}^{r+1} = \arg\min_{\alpha_{ij}} \left(g_{ij}(\alpha_{ij}) + \frac{\rho}{2\lambda_{ij}} \left\| \alpha_{ij} - \left(\theta_{ij}^r - \theta_{ji}^r \right) + \tau_{ij}^r \right\|_2^2 \right)$ for $(i,j) \in \mathcal{H}$
3. $\theta_{ij}^{r+1} = \frac{1}{3} \left(\theta_{ji}^r - \alpha_{ij}^{r+1} - \tau_{ij}^r + \beta_i^{r+1} + \xi_{ij}^r + \theta_{ij}^r \right)$ for $(i,j) \in \mathcal{H}$
4. $\theta_{ji}^{r+1} = \frac{1}{3} \left(\theta_{ij}^r - \alpha_{ij}^{r+1} - \tau_{ij}^r + \beta_j^{r+1} + \xi_{ji}^r + \theta_{ji}^r \right)$ for $(i,j) \in \mathcal{H}$
5. $\tau_{ij}^{r+1} = \tau_{ij}^r - \left(\theta_{ij}^{r+1} - \theta_{ji}^{r+1} \right) + \alpha_{ij}^{r+1}$ for $(i,j) \in \mathcal{H}$
6. $\xi_{ij}^{r+1} = \xi_{ij}^r - \theta_{ij}^{r+1} + \beta_i^{r+1}$ for $(i,j) \in \mathcal{H}$
7. $\xi_{ji}^{r+1} = \xi_{ji}^r - \theta_{ji}^{r+1} + \beta_j^{r+1}$ for $(i,j) \in \mathcal{H}$

Stopping criterion

Given that $3-7$ hold, have

$$\xi_{ij}^{r+1}+\tau_{ij}^{r+1}=\xi_{ji}^{r+1}-\tau_{ij}^{r+1}=\theta_{ij}^{r+1}-\theta_{ij}^{r}+\theta_{ji}^{r+1}-\theta_{ji}^{r}$$

In practice, use

$$\Delta_{\text{primal}}\left(\beta^{r+1},\alpha^{r+1},\theta^{r+1}\right)=\frac{1}{5q_{\mathcal{H}}\sqrt{p}}\sum_{(i,j)\in\mathcal{H}}\Big(\left\|\theta_{ij}^{r+1}-\theta_{ij}^{r}\right\|_2+\left\|\theta_{ji}^{r+1}-\theta_{ji}^{r}\right\|_2$$

$$+\left\|\beta_{i}^{r+1}-\theta_{ij}^{r+1}\right\|_2+\left\|\beta_{j}^{r+1}-\theta_{ji}^{r+1}\right\|_2+\left\|\alpha_{ij}^{r+1}-\left(\theta_{ij}^{r+1}-\theta_{ji}^{r+1}\right)\right\|_2\Big)$$

$\leq \epsilon_{\text{primal}}$ use $5q_{\mathcal{H}}\sqrt{p}$

and

$$\Delta_{\text{dual}}\left(\xi^{r+1},\tau^{r+1}\right)=\frac{1}{2q_{\mathcal{H}}\sqrt{p}}\sum_{(i,j)\in\mathcal{H}}\Big(\left\|\xi_{ij}^{r+1}+\tau_{ij}^{r+1}\right\|_2+\left\|\xi_{ji}^{r+1}-\tau_{ij}^{r+1}\right\|_2\Big)$$

$\leq \epsilon_{\text{dual}}$ use $2q_{\mathcal{H}}\sqrt{p}$

Computational Complexity at Each Iteration

The computational cost at each iteration is proportional to the number of the data centers m or the number of pairwise comparisons $q_{\mathcal{H}}$. If $q_{\mathcal{H}}$ is proportional to m, the computation at each iteration will increase linearly in terms of the number of data centers m.

The computational cost for one iteration is

$$O\left(m \max_i R_i\right) + O\left(q_{\mathcal{H}} \max_{(i,j)\in\mathcal{H}} R_{ij}^{\text{prox}}\right) + O\left(q_{\mathcal{H}} p\right)$$

flops, where R_i is the computation cost for obtaining $step-1$ for i, R_{ij}^{prox} is the computational cost for computing $step-2$ for pairwise comparison (i,j), and p is the dimension of the parameter vector.

Data generation–linear regression model

The response is $y_{ik} \sim \text{Normal}\left(x_{ik}^T \beta_i, 0.25\right)$, $i = 1, 2, \ldots, m$ and $k = 1, 2, \ldots, n_i$. x_{ik} is a p-dimensional vector of covariates corresponding to data point k from data center i, and n_i is the size of the data from data center i.

Assume $\beta_i = \sum_{g=1}^{5} \omega_g \mathbb{I}\{a_i = g\}$, where $a_i \sim$ Uniform $(\{1, 2, \ldots, 5\})$.

In practice, let $p = 10$, and the covariate vector $\left(\omega_1^T, \omega_2^T, \ldots, \omega_5^T\right)$ as

$$\omega = \begin{bmatrix} -2 & 2 & -2 & 2 & -2 & 2 & -2 & 2 & -2 & 2 \\ 2 & -2 & 2 & -2 & 2 & -2 & 2 & -2 & 2 & -2 \\ 2 & 2 & 2 & 2 & 2 & -2 & -2 & -2 & -2 & -2 \\ -2 & -2 & -2 & -2 & -2 & 2 & 2 & 2 & 2 & 2 \\ -1 & 3 & -1 & 3 & -1 & 3 & -1 & 3 & -1 & 3 \end{bmatrix}$$

Data generation–logistic regression model

The response is

$$y_{ik} \sim \text{Bernoulli}\left(\exp\left(x_{ik}^T\beta_i\right) / \left(1+\exp\left(x_{ik}^T\beta_i\right)\right)\right)$$

for $i = 1, 2, \ldots, m$, and $k = 1, 2, \ldots, n_i$.
Set $\omega = \left(\omega_1^T, \omega_2^T, \ldots, \omega_5^T\right)$

$$\omega = \begin{bmatrix} 1 & 1 & -0.1 & 0.1 & -0.1 & 0.1 & -0.1 & 0.1 & -0.1 & 0.1 \\ -0.1 & 0.1 & 1 & 1 & 0.2 & -0.2 & 0.2 & -0.2 & 0.2 & -0.2 \\ -0.1 & 0.1 & -0.1 & 0.1 & 1 & 1 & -0.1 & 0.1 & -0.1 & 0.1 \\ -0.1 & 0.1 & -0.1 & 0.1 & -0.1 & 0.1 & 1 & 1 & -0.1 & 0.1 \\ -0.1 & 0.1 & -0.1 & 0.1 & -0.1 & 0.1 & -0.1 & 0.1 & 1 & 1 \end{bmatrix}$$

Convergence of the Algorithm

Pay attention on the primal error $\Delta_{\text{primal}}\ (\beta^r, \alpha^r, \theta^r)$ and the dual error $\Delta_{\text{dual}}\ (\xi^r, \tau^r)$.

There exist relationships:

$$\Delta_1^r + \Delta_2^r \leq 5q_{\mathcal{H}}\sqrt{p} \cdot \Delta_{\text{primal}}\ (\beta^r, \alpha^r, \theta^r)$$

$$\Delta_3^r + \Delta_4^r \leq 2q_{\mathcal{H}}\sqrt{p} \cdot \Delta_{\text{dual}}\ (\xi^r, \tau^r)$$

If we can show that both $\Delta_{\text{primal}}\left(\beta^r, \alpha^r, \hat{\theta}^r\right)$ and $\Delta_{\text{dual}}\ (\xi^r, \tau^r)$ decrease at a rate of $r^{-1/2}$, then we can confirm that $\Delta_1^r + \Delta_2^r = O\left(r^{-1/2}\right)$ and $\Delta_3^r + \Delta_4^r = O\left(r^{-1/2}\right)$, which further verify the theoretical results in Theorem B.2 ($\Delta_i^r \leq C\sqrt{r^{-1/2}}$) up to constants.

Convergence of the Algorithm

Data generation: linear regression model to generate the data for $m = 500$ data centers. For data center i, assume the number of data points $n_i \sim Poisson(100)$. For each data point, the corresponding covariate vector $x_{ik} \sim \text{Normal}\,(0, I_{10\times 10})$.

Estimation: loss function $l_i\left(\beta_i\right) = \left(2n_i\right)^{-1} \left\|y_i - X_i\beta_i\right\|_2^2$ and penalty function $\lambda \sum_{(i,j)\in\mathcal{H}} \left\|\beta_i - \beta_j\right\|_2$. Define $\mathcal{H}$ as

$$\mathcal{N}(i) = \begin{cases} \{i+1, i+2, \cdots, i+\mathrm{d}\} \text{ for } i = 1, 2, \cdots, m-d \\ \{i+1, i+2, \cdots, m\} \text{ for } i = m-d+1, \cdots, m-1 \end{cases}$$

The number of edges of $\mathcal{H}$ is equal to $2^{-1}\left(2md - d^2 - d\right)$. Let $d = 20$, and the number of edges in $\mathcal{H}$ is $q_{\mathcal{H}} = 9790$.

Convergence of the Algorithm

Performance measures: Evaluate the primal error, the dual error, and the relative optimal error, which is defined as

$$\Delta_{\text{opt}}\left(\beta^{\text{erg},r}, \alpha^{\text{erg},r}\right) = \frac{\left|\sum_{i=1}^{m} l_i\left(\beta_i^{\text{erg},r}\right) + \sum_{(i,j)\in\mathcal{H}} \lambda \left\|\alpha_{ij}^{\text{erg},r}\right\|_2 - \Psi_{\text{primal}}^{\text{ADMM}}\right|}{\left|\Psi_{\text{primal}}^{\text{ADMM}}\right|}$$

where $\beta_i^{\text{erg},r}$ and $\alpha_{ij}^{\text{erg},r}$ are the ergodic average of sequences $\{\beta_i^s\}_{s=1}^r$ and $\left\{\alpha_{ij}^s\right\}_{s=1}^r$, respectively, and $\Psi_{\text{primal}}^{\text{ADMM}} = \sum_{i=1}^{m} l_i\left(\beta_i^{\text{ADMM}}\right) + \sum_{(i,j)\in\mathcal{H}} \lambda \left\|\gamma_{ij}^{\text{ADMM}}\right\|_2$.

Stop at $r = 2000$.

Convergence of the Algorithm

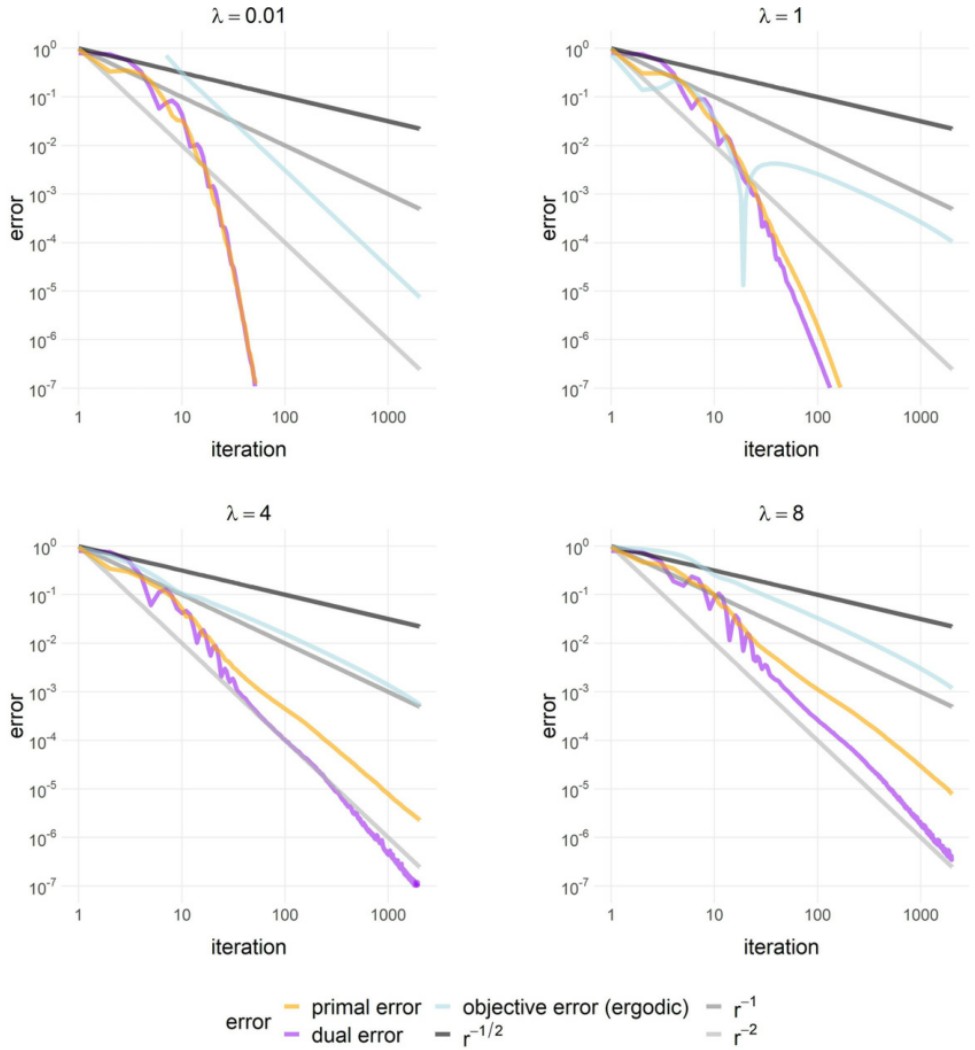

Figure 1. Plots for the optimal error, primal error, and the dual error against the iteration number r. The three reference lines $r^{-1/2}$, r^{-1}, and r^{-2} are shown in black, dark gray, and light gray colors, respectively. Top left: $\lambda = 0.01$; top right: $\lambda = 1$; bottom left: $\lambda = 4$; bottom right: $\lambda = 8$. The plots show that the objective error (plotted in light blue color) has an order magnitude $O(r^{-1})$ while the primal error (plotted in orange color) and dual error (plotted in purple color) are well below the line $r^{-1/2}$.

Runtime of the algorithm

Whether has an advantage in computational time over the ADMM-based iterative scheme when the number of data centers m and the number of pairwise comparisons q increase.

Data generation: linear regression model to generate the data. Vary the number of data centers from $m = 100$ to $m = 1000$ to generate the data. For data center i, the number of data points $n_i \sim Poisson(100)$. For each data point, $x_{ik} \sim \text{Normal}\,(0, I_{10\times 10})$.

Parameter estimation: The edges of the comparison graph $\mathcal{H}$ vary from 1790 for $m = 100$ to 19790 for $m = 1000$.

Set the tolerance errors $\epsilon_{\text{primal}} = \epsilon_{\text{dual}} = \epsilon_{\text{ADMM}} = 5 \times 10^{-3}$.

Runtime of the algorithm

Performance measures:

- (a) Runtime of computing initial values: It includes the runtime of coding the linear operator G, computing the Cholesky decomposition of $X^T X + \rho G^T G$ for primal model and the runtime of computing $\left(X_i^T X_i\right)^{-1}$ for $i = 1, 2, \ldots, m$ for our method.
- (b) Aggregated runtime of iteration: It is the sum of the runtime of carrying out estimation under the five different tuning parameter values.
- (c) Total runtime: The sum of (a) and (b).

Runtime of the algorithm

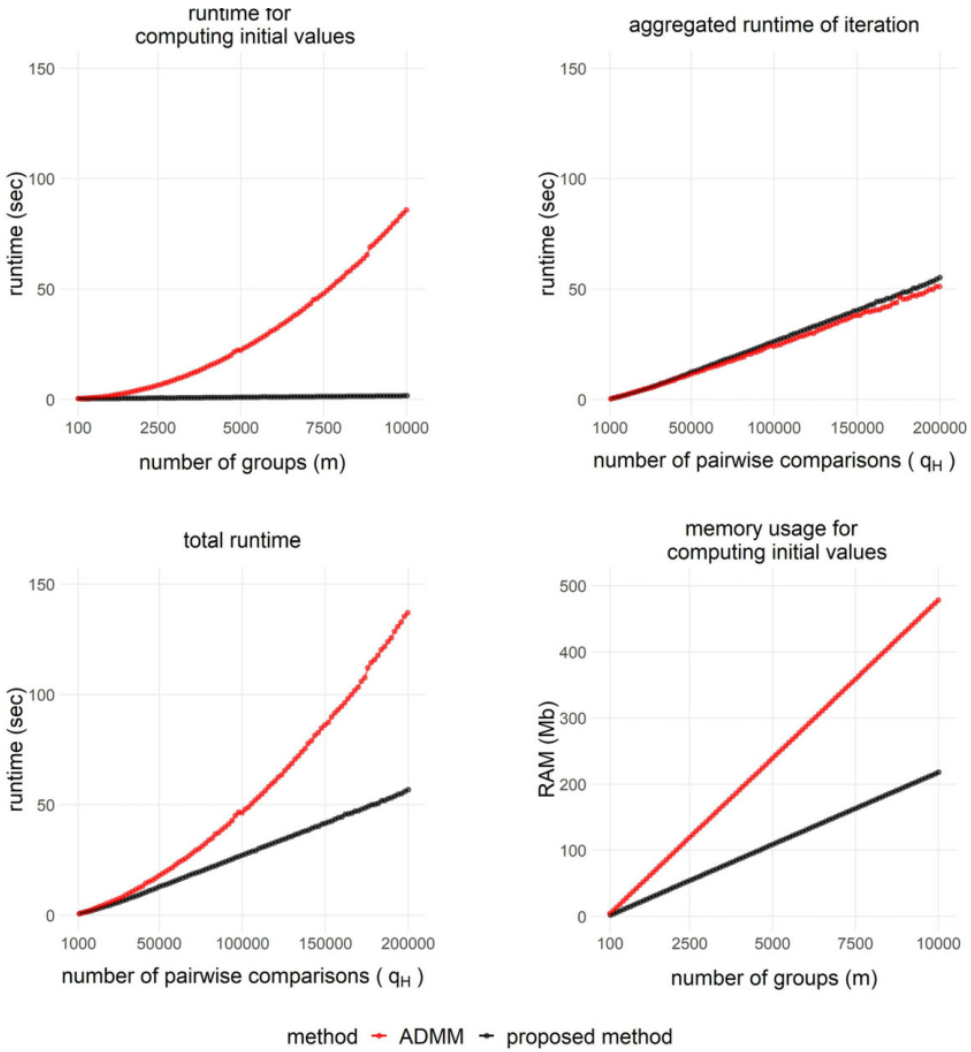

Figure 2. Plots for runtime and memory usage. Top left: Runtime of computing initial values against the number of data center m; top right: aggregated runtime of carrying out iteration against the number of pairwise comparisons $q_{\mathcal{H}}$; bottom left: total runtime against the number of pairwise comparisons $q_{\mathcal{H}}$. Bottom right: Memory usage for storing necessary quantities against the number of groups m. The plots suggest that the proposed method (plotted in black color) has advantages in runtime and memory usage when comparing with the algorithm based on the alternating direction method of multipliers (plotted in red color).

Runtime of the Algorithm: Parallel Implementation

Data generation: use logistic regression model and vary the number of data centers from $m = 1000$ to $m = 1000000$. For each data center, the number of data points $n_i \sim Poisson(100)$. For each data point, fix the first covariate equal to 1 and generated the rest of 9 covariates from $Uniform\{0, 1\}$.

Estimation: the fused group penalty function. Set comparison graph with $d = 1$.

Computational environment: Carry out $step - 1$ in parallel under multiple cores and do computation of the rest under a single core.

Algorithm settings: two different approaches to obtain $step - 1$.

Runtime of the Algorithm: Parallel Implementation

$$\beta_i^{s+1,r} = \arg\min_{\beta_i} \left\{ \frac{1}{2} \left(\phi_i^{s,r} - \Sigma_i^{s,T} X_i \beta_i \right)^T \left(\Sigma_i^{-1} \right)^{s,r} \left(\phi_i^{s,r} - \Sigma_i^{s,r} X_i \beta_i \right) \right.$$
$$\left. + \frac{\rho |\mathcal{N}(i)|}{2} \left\| \beta_i - \mid \mathcal{N}(i) \mid^{-1} b_i \right\|^2 \right\}$$
$$= \left(X_i^T \Sigma_i^{s,r} X_i + \rho |\mathcal{N}(i)| I \right)^{-1} \left(X_i^T \phi_i^{s,r} + \rho b_i \right)$$

where $\Sigma^{s,r}$ is an $n_i \times n_i$ diagonal matrix with the (k,k) th entry $\left(\Sigma_i^{s,r}\right)_{kk} = \mu_{ik}^{s,r}\left(1-\mu_{ik}^{s,r}\right), \mu_{ik}^{s,r} = \exp\left(x_{ik}^T \beta_i^{s,r}\right) / \left[1+\exp\left(x_{ik}^T \beta_i^{s,r}\right)\right]$, $\phi_i^{s,r}$ is an n_i dimensional vector with the kth entry $\left(\phi_i^{s,r}\right)_k = y_{ik} - \mu_{ik}^{s,r} + \left(\sum_i^{s,r}\right)_{kk} x_{ik}^T \beta_i^{s,r}$, and $b_i = \sum_{j \in \mathcal{N}(i)} \left(\theta_{ij}^r - \xi_{ij}^r\right)$ is a p-dimensional vector that is fixed throughout the iteration.

Stopping criterion: $\left\| \beta_i^{s+1,r} - \beta_i^{s,r} \right\|_2 / \sqrt{p} \leq \epsilon$.

Runtime of the Algorithm: Parallel Implementation

$$\beta_i^{s+1,r} = \arg\min_{\beta_i} \left\{ \left\langle \nabla l_i\left(\beta_i^{s,r}\right), \beta_i - \beta_i^{s,r} \right\rangle \right.$$
$$\left. + \frac{A_i}{2} \left\| \beta_i - \beta_i^{s,r} \right\|^2 + \frac{\rho |\mathcal{N}(i)|}{2} \left\| \beta_i - |\mathcal{N}(i)|^{-1} b_i \right\|^2 \right\}$$

where A_i is the gradient Lipschitz constant associated with the loss function $l_i(\beta_i)$.

From the KKT conditions we can obtain a closed form representation for $\beta_i^{s+1,r}$ in terms of $\beta_i^{s,r}, y_i, A_j$ and ρ in a way such that

$$0 \in \nabla l_i\left(\beta_i^{s,r}\right) + A_i \left(\beta_i^{s+1,r} - \beta_i^{s,r}\right) + \rho \mid \mathcal{N}(i) \mid \beta^{s+1,r} - \rho b_i$$
$$\Leftrightarrow \left(A_i + \rho |\mathcal{N}(i)|\right) \beta_i^{s+1,r} \in A_i \beta_i^{s,r} + \rho b_i - \nabla l_i\left(\beta_i^{s,r}\right)$$
$$\Leftrightarrow \beta_i^{s+1,r} \in \frac{A_i \beta_i^{s,r} + \rho b_i}{A_i + \rho |\mathcal{N}(i)|} - \frac{1}{A_i + \rho |\mathcal{N}(i)|} \nabla l_i\left(\beta_i^{s,r}\right)$$

Runtime of the Algorithm: Parallel Implementation

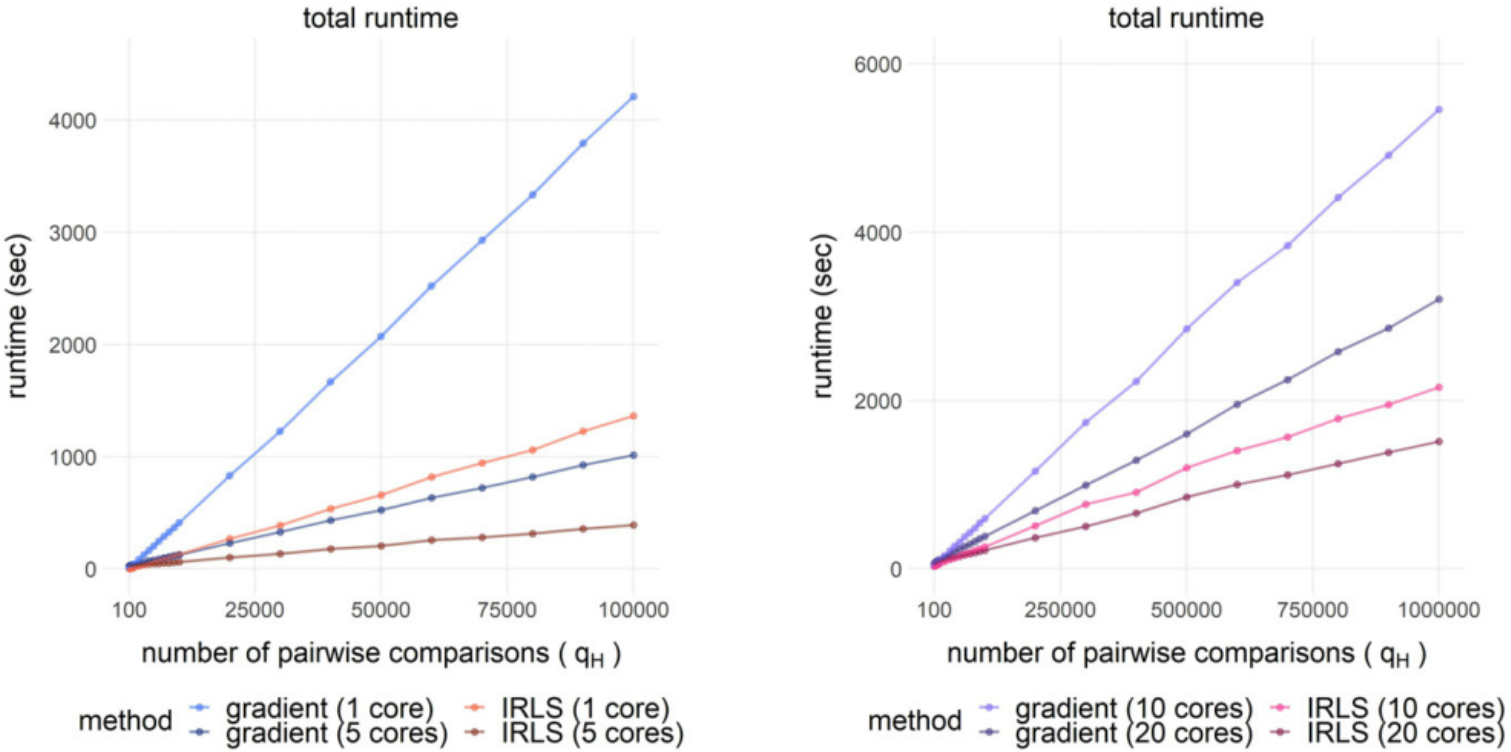

Figure 3. Plots of total runtime against the number of pairwise comparisons $q_{\mathcal{H}}$. Left: Results under computational environments with 1 core and 5 cores; right: results under computational environments with 10 cores and 20 cores. The plots show that the proposed method has an advantage in runtime under a parallel computing framework.

Accuracy of the algorithm

Study whether sequences generated by the algorithm can provide good performance in estimation and prediction. Investigate how performance of the estimation varies as the number of data points in each data center varies.

Data generation: use logistic regression model to generate the data for $m = 100$ data centers. For each data center, the number of data points $n_i \sim Poisson(N)$, where N is the mean number of data points collected at each data center. Vary the mean number of data points from $N = 100$ to $N = 2000$.

Accuracy of the algorithm

Estimation: Carry out the estimation under 20 tuning parameter values and then used a model selection criterion to select the best one. First carry out maximum likelihood estimation for each β_i and denote $\hat{\beta}_i^{\text{SepMLE}}$. Then define the 20 tuning parameter values as the 20 different quantiles of the sequence $\left\{\left\|\hat{\beta}_i^{\text{SepMLE}} - \hat{\beta}_j^{\text{SepMLE}}\right\|_2\right\}_{(i,j)\in\mathcal{H}}$. The maximum value of λ was $\max_{(i,j)\in\mathcal{H}}\left\|\hat{\beta}_i^{\text{SepMLE}} - \hat{\beta}_j^{\text{SepMLE}}\right\|_2$, the minimum value of λ as the 0.1 percent quantile of $\left\{\left\|\hat{\beta}_i^{\text{SepMLE}} - \hat{\beta}_j^{\text{SepMLE}}\right\|_2\right\}_{(i,j)\in\mathcal{H}}$.

Given that λ was fixed, use a two-stage procedure to obtain an estimate of β_i for each data center.

Accuracy of the algorithm

First stage: estimate $\alpha_{ij} = \beta_i - \beta_j$. With estimate $\hat{\alpha}_{ij}^{\lambda}$, start a post-processing process by constructing the graph

$$\mathcal{U}^{\lambda} = \left\{(i,j) : \left\|\hat{\alpha}_{ij}^{\lambda}\right\|_2 = 0\right\}$$

Then label the m data centers by applying community detection techniques ('cluster fast greedy' function in R package 'igraph') to partition nodes of $\mathcal{U}^{\lambda}$. Let $\widehat{K}^{\mathrm{BS},\lambda}$ denote the number of clusters in the partition.

Second stage: group data points from data centers in the same cluster and carry out maximum likelihood estimation separately with the $\widehat{K}^{\mathrm{BS},\lambda}$ clustered datasets. Let $\widehat{\beta}_i^{\mathrm{BS},\lambda}$ denote the regression estimate corresponding to data center i.

Accuracy of the algorithm

Estimation: Model selection criteria:

$$\mathrm{AIC}(\lambda) = 2\sum_{i=1}^{m} l_i\left(\widehat{\beta}_i^{\mathrm{BS},\lambda}\right) + 2p\widehat{K}^{\mathrm{BS},\lambda}$$

$$\mathrm{BIC}(\lambda) = 2\sum_{i=1}^{m} l_i\left(\widehat{\beta}_i^{\mathrm{BS},\lambda}\right) + p\widehat{K}^{\mathrm{BS},\lambda}\log\left(\sum_{i=1}^{m} n_i\right)$$

Algorithm settings: IRLS scheme to obtain $step - 1$.

Accuracy of the algorithm

Performance measures: Let $Q^{\text{true}} = \{Q_u^{\text{tue}}\}_u$ be the true partition of nodes and $\hat{Q}^{\text{BS}} = \left\{\hat{Q}_v^{\text{BS}}\right\}_v$ be the partition of nodes generated by community detection techniques.

(a) Estimated number of clusters $\left(\hat{K}^{\text{BS}}\right)$.

(b) Information quality ratio (IQR):

$$\text{IQR}\left(Q^{\text{true}}, \widehat{Q}^{BS}\right) = \frac{\sum_{u,v} \mathbb{P}\left(Q_u^{\text{true}} \cap \widehat{Q}_v^{\text{BS}}\right) \log \mathbb{P}\left(Q_u^{\text{true}}\right) \mathbb{P}\left(\widehat{Q}_v^{\text{BS}}\right)}{\sum_{u,v} \mathbb{P}\left(Q_u^{\text{true}} \cap \widehat{Q}_v^{\text{BS}}\right) \log \mathbb{P}\left(Q_u^{\text{true}} \cap \widehat{Q}_v^{\text{BS}}\right)} - 1$$

(c) Mean squared error (MSE):

$$\text{MSE}\left(\hat{\beta}^{\text{B}}\right) = \frac{1}{mp} \sum_{i=1}^{m} \left\| \hat{\beta}_i^{\text{BS}} - \beta_i^{\text{true}} \right\|_2^2$$

Accuracy of the algorithm

Performance measures:

(d) Relative mean squared prediction error (relative MSPE):

$$\text{relative MSPE}\left(\hat{y}^{BS}\right) = \frac{\sum_{i=1}^{m} \left\| \hat{y}_i^{\text{new,BS}} - y_i^{\text{new}} \right\|_2^2}{\sum_{i=1}^{m} \left\| \hat{y}_i^{\text{new,null}} - y_i^{\text{new}} \right\|_2^2}$$

where y_i^{new} are newly generated data points not being used in the estimation, and $\hat{y}_i^{\text{new,null}}$ is a vector of the predicted values of y_i^{new} using a logistic regression model only with the intercept term. Let the number of newly generated data points $n_i^{\text{new}} = n_i$ to generate y_i^{new}.

Accuracy of the algorithm

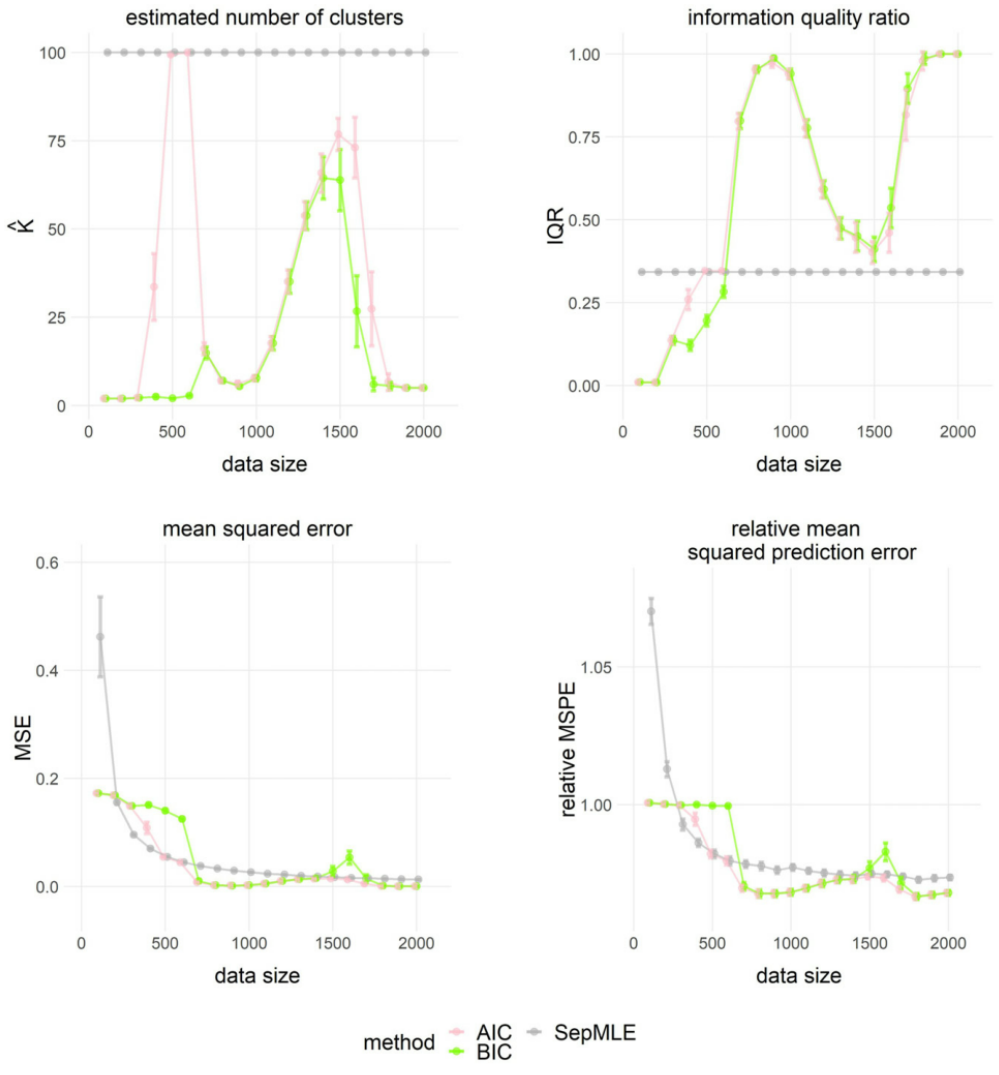

Figure 4. Plots of performance measures against the mean data size N at each data center. In each plot, the x-axis is the data size while the y-axis is the corresponding performance measure. Top left: Estimated number of clusters; top right: information quality ratio (IQR); bottom left: mean squared error of the estimated parameter (MSE); bottom right: relative mean squared prediction error (relative MSPE). The plots show that performance of the estimation method does not monotonically increase with the data size. There may exist some interesting phase transition pattern in the estimation procedure.

Advantages

1. Without carrying out numerical computation involving the linear operator G.
2. Computational cost: scalable in terms of the number of data centers m or the number of pairwise comparisons q at each iteration.
3. If the loss function is separable in terms of the data centers, parallel computing.
4. The sequence generated by the iterative scheme can make the objective function approach to its optimal value with a rate similar to the one obtained for the ADMM-based iterative schemes.

Discussion

1. Did not consider the problem in which the loss function is not separable in terms of the parameter vectors.
2. Did not deal with the problem in which the loss function is strongly convex.
3. Did not pay attention on the cases in which the dimension of parameter vector p is large.
4. (Reviewers) How the convergence of the iterative scheme depends on the topology of comparison graph $\mathcal{H}$.
5. Although it can run the iterative scheme under a parallel computing framework, it can only manage to run it in a synchronous fashion.

Causes of death in 367 towns in Taiwan

The data contain yearly numbers of dead people in terms of their death causes in 367 towns in Taiwan. They are collected over a 5-year period between 2008 and 2012. There are 29 death causes in the data. The 29th cause is a combination of the main death causes for about 80 to 95% of dead people in each town. The rest of 28 causes are the 'rare' death causes. By rare we mean those occurring with small probabilities.

The dataset used in this section can be found in Open Government Data (2012) and downloaded from http://data.gov.tw/node/5965.

Aim is to cluster towns that have a similar pattern in the rare death causes.

Model estimation

Assume there are $p+1$ death causes. Let π_{ja} denote the probability of a dead individual in town j who dies in cause a. Model the logarithm of the odds for the death cause a against the death cause $p+1$ in town j by

$$\log\left(\frac{\pi_{ja}}{\pi_{j(p+1)}}\right) = \beta_{ja}$$

We have

$$\pi_{ja} = \frac{\exp\left(\beta_{ja}\right)}{1+\sum_{a'=1}^{p}\exp\left(\beta_{ja'}\right)} \text{ for } a = 1, 2, \cdots, p$$

and $\pi_{ja} = \left[1+\sum_{a'=1}^{p}\exp\left(\beta_{ja'}\right)\right]^{-1}$ for $a = p+1$. The parameter vector $\beta_j = (\beta_{j1}, \beta_{j2}, \cdots, \beta_{jp})$ characterizes the distribution of death causes in town j.

Model estimation

There are m towns, and each town has c observations. The minus logarithm of the likelihood function is

$$
\begin{aligned}
l(\beta) &= \sum_{j=1}^{m} l\left(\beta_j\right) \\
&= -\sum_{j=1}^{m}\left(\sum_{t=1}^{c}\left\{\sum_{a'=1}^{p} y_{jta'}\beta_{ja'} - n_{jt}\log\left[1+\sum_{a'=1}^{p}\exp\left(\beta_{ja'}\right)\right]\right\}\right)
\end{aligned}
$$

where $n_{jt} = \sum_{a'=1}^{p+1} y_{jta'}$ is the number of deaths in observation t in town j. The fused group lasso estimate of $\beta = (\beta_1, \beta_2, \cdots, \beta_m)$ is defined by

$$
\widehat{\beta} = \arg\min_{\beta_1,\beta_2,\cdots,\beta_m}\left\{\sum_{j=1}^{m} l\left(\beta_j\right) + \lambda\sum_{(j,k)\in\mathcal{H}}\left\|\beta_j - \beta_k\right\|_2\right\}
$$

Model estimation

The comparison graph $\mathcal{H}$ is defined by

$$\mathcal{H} = \left\{ (j,k) : d\left(\widehat{\phi}_j, \widehat{\phi}_k\right) \leq b_{\text{thr}} \right\}$$

$d\left(\widehat{\phi}_j, \widehat{\phi}_k\right)$ is a distance function, $b_{\text{thr}} \geq 0$ is a threshold value, and $\widehat{\phi}_j = \left(\widehat{\phi}_{j1}, \widehat{\phi}_{j2}, \cdots, \widehat{\phi}_{jp}\right)$ is a p-dimensional vector in which each $\widehat{\phi}_{ja}$ is defined by

$$\widehat{\phi}_{ja} = \frac{\sum_{t=1}^{s} y_{jta}}{\sum_{t=1}^{s} y_{jt,p+1}}$$

Define $d\left(\widehat{\phi}_j, \widehat{\phi}_k\right) = \left\| \widehat{\phi}_j - \widehat{\phi}_k \right\|_{\infty}$.

Model estimation

Use the Silhouette coefficient to evaluate quality of a partition:

$$\text{Silhouette coefficient} = \left(\frac{1}{|\{j : |\mathcal{C}_j| \geq 2\}|} \sum_{j:|\mathcal{C}_j|\geq 2} \frac{v_{2j} - v_{1j}}{\max\{v_{1j}, v_{2j}\}} \right)$$

where $v_{1j} = (|\mathcal{C}_j| - 1)^{-1} \sum_{k \in \mathcal{C}_j} \left(\widehat{\phi}_{ja} - \widehat{\phi}_{ka} \right)^2$, $v_{2j} = (m - |\mathcal{C}_j|)^{-1} \sum_{k \notin \mathcal{C}_j} \left(\widehat{\phi}_{ja} - \widehat{\phi}_{ka} \right)^2$, $\widehat{\phi}_{ja}$ is the empirical odds of death cause a for town j, $\mathcal{C}_j$ is the cluster that town j belongs to, and $m = 367$ is the number of towns.

Result

Method	BIC	# of clusters	Silhouette coefficient
Maximum likelihood estimation (MLE)	165,264.5	367	0
Fused group lasso (FGL)	110,496.9	33	0.746
MLE with k-means clustering (kMLE)	108,079.2	12	0.295

Table 1: Performance results for partitions based on the three estimations.