Robust Analysis of Cancer Heterogeneity for High-dimensional Data

Chao Cheng
Joint work with Xingdong Feng and Mengyun Wu

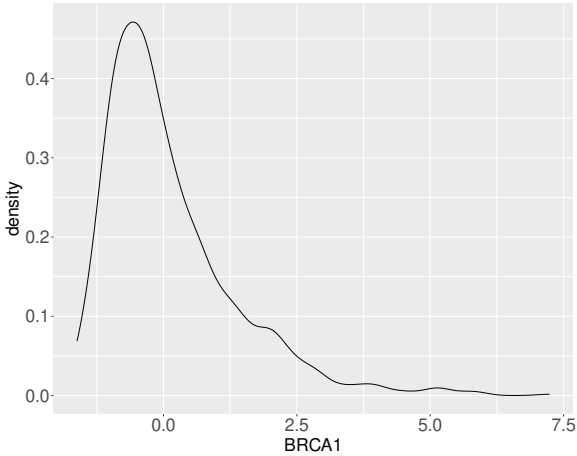
School of Statistics and Management
Shanghai University of Finance and Economics

April 11, 2022

Outline

Outline

Introduction



- Cancer heterogeneity may be attributed to many factors.
- Subgroup identification based on genomic pattern.

- High-dimensional data with variable selection need.
- Data is skewed and heavy-tailed .

Introduction

Subgroup analysis has been extensively conducted to accommodate heterogeneity.

- Pre-specified subgroups based on observed attributes.
- Explore the latent subgroups:
  - mixture model
  - penalty method

Related Work

Latent subgroup analysis

- Mixture model: Shen and He [2015], Ren et al. [2021]
- Pairwise penalty:
  - unsupervised: [Chi and Lange, 2015, Wu et al., 2016]
  - supervised: [Ma and Huang, 2017, Chen et al., 2020], for median regression[Zhang et al., 2019]

- A robust analysis tool for simultaneous subgroup identification and variable selection in high-dimensional data is still missing.

Outline

Model Setting

Consider the following model

$$y_i = \mu_i + \boldsymbol{x}_i^T \boldsymbol{\beta} + \varepsilon_i, \qquad i = 1, 2, \cdots, n, \tag{1}$$

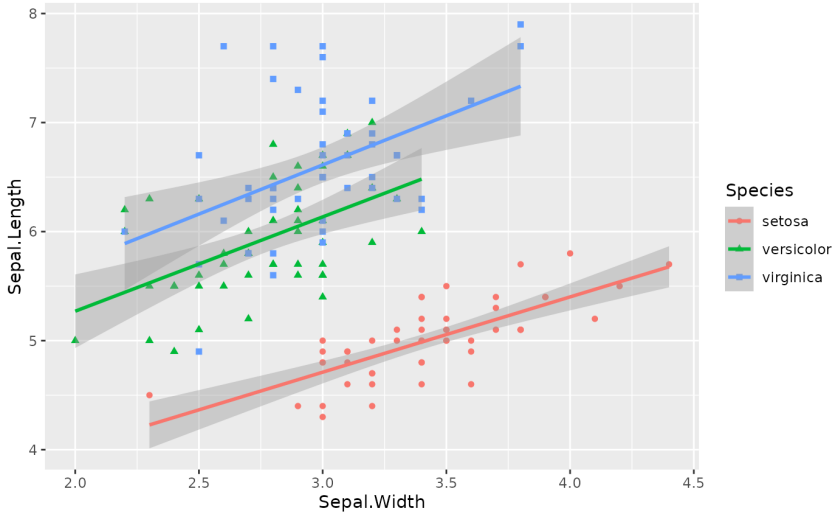
where $\boldsymbol{\beta}$ is a p-dimensional vector.

- Heterogeneous treatment effect:
  There is a partition $\boldsymbol{G}_0 = \{G_1, \cdots, G_{K_0}\}$ of $\{1, \cdots, n\}$, such that

  $$\mu_i = \alpha_k, \text{ for all } i \in G_k.$$

  Therefore α_k is the common value for subgroup G_k.
- Sparse covariate effect in high dimension:
  Only q entries of $\boldsymbol{\beta}$ are non-zero, where $q \ll n$.

The Iris Data



Proposed M-estimator

M-estimation with subgroup identification and variable selection

$$\left(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\beta}}\right) = \underset{\boldsymbol{\mu},\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \rho\left(y_i - \mu_i - \boldsymbol{x}_i^T \boldsymbol{\beta}\right) + \sum_{1 \le i < j \le n} P_{\lambda_1}\left(\mu_i - \mu_j\right) + \sum_{j=1}^{p} P_{\lambda_2}\left(\beta_j\right). \quad (2)$$

subgroup identification

robust M-estimation

variable selection

Proposed M-estimator: Loss Function

- $\mathrm{E}\psi(\varepsilon) = 0$ where ψ is the derivative of ρ.
- ρ is differentiable except at finite number of points.
- L_1: $\rho(x) = |x|$.
- L_2: $\rho(x) = x^2$.
- *Huber*:

$$\rho(x; c) = \begin{cases} \frac{1}{2}x^2 & |x| \le c \\ c|x| - \frac{1}{2}c^2 & |x| > c, \end{cases}$$

  where c is a positive constant.
- Quantile regression: $\rho(x; \tau) = x(\tau - 1_{x<0})$, where $\tau \in (0, 1)$.

Proposed M-estimator: Penalty Function

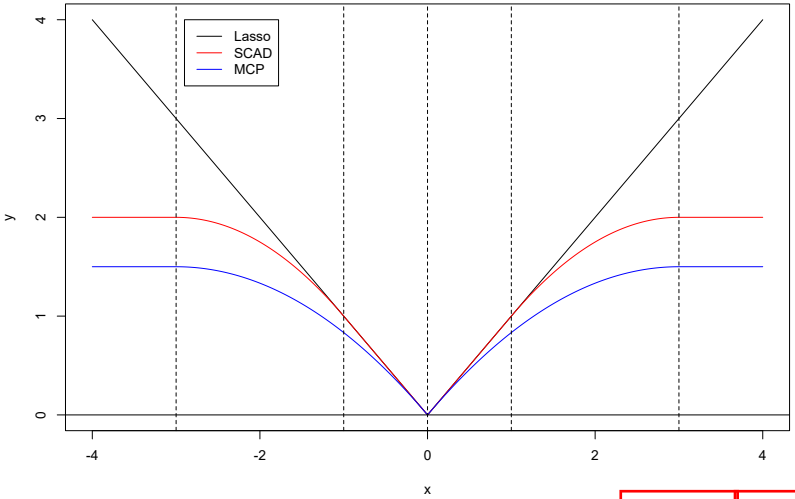
- LASSO: $P_\lambda(x) = \lambda |x|$.
- SCAD:

$$P'_{\lambda,\gamma}(x) = \lambda \left\{ I(x \le \lambda) + \frac{(\gamma\lambda - x)_+}{(\gamma - 1)\lambda} I(x > \lambda) \right\}, \quad x > 0, \quad \gamma > 2.$$

- MCP:

$$P'_{\lambda,\gamma}(x) = \lambda \left(1 - \frac{x}{\lambda\gamma} \right)_+, \quad x > 0, \quad \gamma > 1.$$

Proposed M-estimator: Penalty Function



Alternating Direction Method of Multipliers (ADMM)

We can rewrite the objective function in (2) to a optimization form:

$$\min \frac{1}{n}\sum_{i=1}^{n}\rho\left(z_i\right)+\sum_{1\le i<j\le n}P_{\lambda_1}\left(s_{ij}\right)+\sum_{j=1}^{p}P_{\lambda_2}\left(w_j\right)$$
$$\text{s.t.}\begin{cases}\boldsymbol{z}=\boldsymbol{y}-\boldsymbol{\mu}-\boldsymbol{X}\boldsymbol{\beta}\\ \boldsymbol{s}=\boldsymbol{D}\boldsymbol{\mu}\\ \boldsymbol{w}=\boldsymbol{\beta}\end{cases}, \tag{3}$$

where $\boldsymbol{D}$ is the $\frac{n(n-1)}{2}\times n$ pairwise difference matrix, hence $s_{ij}=\mu_i-\mu_j$.

Alternating Direction Method of Multipliers (ADMM)

The augmented lagrangian form of (3) is

$$
\begin{aligned}
& L(\beta, \mu, \boldsymbol{z}, \boldsymbol{s}, \boldsymbol{w}, \boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3) \\
=& \frac{1}{n} \sum_{i=1}^{n} \rho(z_i) + \sum_{1 \le i < j \le n} P_{\lambda_1}(s_{ij}) + \sum_{j=1}^{p} P_{\lambda_2}(w_j) \\
+& \frac{r_1}{2} \|\boldsymbol{y} - \mu - \boldsymbol{X}\beta - \boldsymbol{z}\|_2^2 + \frac{r_2}{2} \|\boldsymbol{D}\mu - \boldsymbol{s}\|_2^2 + \frac{r_3}{2} \|\beta - \boldsymbol{w}\|_2^2 \\
+& \langle \boldsymbol{y} - \mu - \boldsymbol{X}\beta - \boldsymbol{z}, \boldsymbol{q}_1 \rangle + \langle \boldsymbol{D}\mu - \boldsymbol{s}, \boldsymbol{q}_2 \rangle + \langle \beta - \boldsymbol{w}, \boldsymbol{q}_3 \rangle,
\end{aligned}
\tag{4}
$$

where r_1, r_2 and r_3 are positive constants, $\boldsymbol{q}_1$, $\boldsymbol{q}_2$ and $\boldsymbol{q}_3$ are multiplier vectors. $\|\boldsymbol{a}\|_2 = \sqrt{\boldsymbol{a}^T \boldsymbol{a}}$ and $\langle \boldsymbol{a}, \boldsymbol{b} \rangle = \boldsymbol{a}^T \boldsymbol{b}$.

Alternating Direction Method of Multipliers (ADMM)

The Augmented Lagrangian Multiplier(ALM) form

$$
\begin{aligned}
& L(\beta, \mu, \boldsymbol{z}, \boldsymbol{s}, \boldsymbol{w}, \boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3) \\
=& \frac{1}{n} \sum_{i=1}^{n} \rho(z_i) + \sum_{1 \le i < j \le n} P_{\lambda_1}(s_{ij}) + \sum_{j=1}^{p} P_{\lambda_2}(w_j) \\
+& \frac{r_1}{2} \|\boldsymbol{y} - \mu - \boldsymbol{X}\beta - \boldsymbol{z}\|_2^2 + \frac{r_2}{2} \|\boldsymbol{D}\mu - \boldsymbol{s}\|_2^2 + \frac{r_3}{2} \|\beta - \boldsymbol{w}\|_2^2 \\
+& \langle \boldsymbol{y} - \mu - \boldsymbol{X}\beta - \boldsymbol{z}, \boldsymbol{q}_1 \rangle + \langle \boldsymbol{D}\mu - \boldsymbol{s}, \boldsymbol{q}_2 \rangle + \langle \beta - \boldsymbol{w}, \boldsymbol{q}_3 \rangle .
\end{aligned}
$$

- L is quadratic in $\left(\mu^T, \beta^T\right)^T$.
- L is convex in $\boldsymbol{z}$, $\boldsymbol{s}$ and $\boldsymbol{w}$ when r_1, r_2, r_3 are properly chosen.
- L has the form of independent summation in $\boldsymbol{z}$, $\boldsymbol{s}$ and $\boldsymbol{w}$ when others are given.

Alternating Direction Method of Multipliers(ADMM)

We solve (4) in a coordinate descent fashion. For a given $\left(\boldsymbol{\beta}^{(k)}, \boldsymbol{\mu}^{(k)}, \boldsymbol{z}^{(k)}, \boldsymbol{s}^{(k)}, \boldsymbol{w}^{(k)}, \boldsymbol{q}_1^{(k)}, \boldsymbol{q}_2^{(k)}, \boldsymbol{q}_3^{(k)}\right)$ at step k. The update at step $k+1$ is given by:

- $\boldsymbol{\beta}^{(k+1)} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\, L\left(\boldsymbol{\beta}, \boldsymbol{\mu}^{(k)}, \boldsymbol{z}^{(k)}, \boldsymbol{s}^{(k)}, \boldsymbol{w}^{(k)}, \boldsymbol{q}_1^{(k)}, \boldsymbol{q}_2^{(k)}, \boldsymbol{q}_3^{(k)}\right)$:

  If $p \leq n$, then

$$\boldsymbol{\beta}^{(k+1)} = \left(r_1 \boldsymbol{X}^T \boldsymbol{X} + r_3 \boldsymbol{I}_p\right)^{-1} \boldsymbol{d}_1^{(k)}.$$

  If $p > n$, then

$$\boldsymbol{\beta}^{(k+1)} = \frac{1}{r_3}\left\{\boldsymbol{I}_p - r_1 \boldsymbol{X}^T \left(r_1 \boldsymbol{X}\boldsymbol{X}^T + r_3 \boldsymbol{I}_n\right)^{-1} \boldsymbol{X}\right\} \boldsymbol{d}_1^{(k)},$$

  where $\boldsymbol{d}_1^{(k)} = r_1 \boldsymbol{X}^T \left(\boldsymbol{y} - \boldsymbol{\mu}^{(k)} - \boldsymbol{z}^{(k)}\right) + r_3 \boldsymbol{w}^{(k)} + \boldsymbol{X}^T \boldsymbol{q}_1^{(k)} - \boldsymbol{q}_3^{(k)}$.

Alternating Direction Method of Multipliers(ADMM)

- $\boldsymbol{\mu}^{(k+1)} = \underset{\boldsymbol{\mu}}{\operatorname{argmin}}\, L\left(\beta^{(k+1)}, \boldsymbol{\mu}, \boldsymbol{z}^{(k)}, \boldsymbol{s}^{(k)}, \boldsymbol{w}^{(k)}, \boldsymbol{q}_1^{(k)}, \boldsymbol{q}_2^{(k)}, \boldsymbol{q}_3^{(k)}\right)$:

$$\boldsymbol{\mu} = \left(r_1 \boldsymbol{I}_n + r_2 \boldsymbol{D}^T \boldsymbol{D}\right)^{-1} \boldsymbol{d}_2^{(k)},$$

where $\boldsymbol{d}_2^{(k)} = r_1\left(\boldsymbol{y} - \boldsymbol{X}\beta^{(k+1)} - \boldsymbol{z}^{(k)}\right) + r_2 \boldsymbol{D}^T \boldsymbol{s}^{(k)} + \boldsymbol{q}_1^{(k)} - \boldsymbol{D}^T \boldsymbol{q}_2^{(k)}$.

Alternating Direction Method of Multipliers(ADMM)

- $\boldsymbol{z}^{(k+1)} = \underset{\boldsymbol{z}}{\operatorname{argmin}}\, L\left(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\mu}^{(k+1)}, \boldsymbol{z}, \boldsymbol{s}^{(k)}, \boldsymbol{w}^{(k)}, \boldsymbol{q}_1^{(k)}, \boldsymbol{q}_2^{(k)}, \boldsymbol{q}_3^{(k)}\right)$.

  The update of $\boldsymbol{z}$ depends on the choice of the loss ρ, and it can be computed elementwisely.

  - L_1:

$$z_i^{(k+1)} = S\left(y_i - \mu_i^{(k+1)} - \boldsymbol{x}_i^T\boldsymbol{\beta}^{(k+1)} + \frac{q_{1,i}^{(k)}}{r_1}, \frac{1}{nr_1}\right),$$

    where $S(x, \lambda)$ is the soft-thresholding function.

  - L_2:

$$z_i^{(k+1)} = \frac{y_i - \mu_i^{(k+1)} - \boldsymbol{x}_i^T\boldsymbol{\beta}^{(k+1)} + \frac{q_{1,i}^{(k)}}{r_1}}{1 + \frac{2}{nr_1}}.$$

    For L_2, $\boldsymbol{z}$ can be omitted in the algorithm.

Alternating Direction Method of Multipliers(ADMM)

- $\boldsymbol{z}^{(k+1)} = \underset{\boldsymbol{z}}{\operatorname{argmin}}\, L\left(\beta^{(k+1)}, \boldsymbol{\mu}^{(k+1)}, \boldsymbol{z}, \boldsymbol{s}^{(k)}, \boldsymbol{w}^{(k)}, \boldsymbol{q}_1^{(k)}, \boldsymbol{q}_2^{(k)}, \boldsymbol{q}_3^{(k)}\right)$.

  The update of $\boldsymbol{z}$ depends on the choice of the loss ρ, and it can be computed elementwisely. Let $d_{z,i}^{(k)} = y_i - \mu_i^{(k+1)} - \boldsymbol{x}_i^T \beta^{(k+1)} + \frac{q_{1,i}^{(k)}}{r_1}$.

  - *Huber*:

$$z_i^{(k+1)} = \begin{cases} S\left(d_{z,i}^{(k)}, \dfrac{c}{nr_1}\right) & \left(\dfrac{1}{nr_1} + 1\right) c < \left|d_{z,i}^{(k)}\right| \\ \dfrac{d_{z,i}^{(k)}}{1 + \frac{1}{nr_1}} & \left(\dfrac{1}{nr_1} + 1\right) c \geq \left|d_{z,i}^{(k)}\right| \end{cases}.$$

Alternating Direction Method of Multipliers(ADMM)

- $\boldsymbol{z}^{(k+1)} = \underset{\boldsymbol{z}}{\operatorname{argmin}}\, L\left(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\mu}^{(k+1)}, \boldsymbol{z}, \boldsymbol{s}^{(k)}, \boldsymbol{w}^{(k)}, \boldsymbol{q}_1^{(k)}, \boldsymbol{q}_2^{(k)}, \boldsymbol{q}_3^{(k)}\right)$.

  The update of $\boldsymbol{z}$ depends on the choice of the loss ρ, and it can be computed elementwisely. Let $d_{z,i}^{(k)} = y_i - \mu_i^{(k+1)} - \boldsymbol{x}_i^T \boldsymbol{\beta}^{(k+1)} + \frac{q_{1,i}^{(k)}}{r_1}$.

  - Quantile regression:

$$z_i^{(k+1)} = \begin{cases} d_{z,i}^{(k)} - \dfrac{\tau}{nr_1} & \dfrac{\tau}{nr_1} < d_{z,i}^{(k)} \\ 0 & \dfrac{\tau-1}{nr_1} \le d_{z,i}^{(k)} \le \dfrac{\tau}{nr_1} \\ d_{z,i}^{(k)} + \dfrac{1-\tau}{nr_1} & d_{z,i}^{(k)} \le \dfrac{\tau-1}{nr_1} \end{cases}.$$

Alternating Direction Method of Multipliers(ADMM)

- $\boldsymbol{s}^{(k+1)} = \underset{\boldsymbol{s}}{\operatorname{argmin}}\, L\left(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\mu}^{(k+1)}, \boldsymbol{z}^{(k+1)}, \boldsymbol{s}, \boldsymbol{w}^{(k)}, \boldsymbol{q}_1^{(k)}, \boldsymbol{q}_2^{(k)}, \boldsymbol{q}_3^{(k)}\right)$.

  The update of $\boldsymbol{s}$ depends on the choice of the loss P_{λ_1}, and it can be computed elementwisely. Let $d_{s,ij}^{(k)} = \mu_i^{(k+1)} - \mu_j^{(k+1)} + \frac{q_{2,ij}^{(k)}}{r_2}$.

  - LASSO:

$$s_{ij}^{(k+1)} = S\left(d_{s,ij}^{(k)}, \frac{\lambda_1}{r_2}\right).$$

  - MCP with $r_2\gamma_1 > 1$:

$$s_{ij}^{(k+1)} = \begin{cases} \dfrac{S\left(d_{s,ij}^{(k)}, \frac{\lambda_1}{r_2}\right)}{1 - \frac{1}{r_2\gamma_1}} & \left|d_{s,ij}^{(k)}\right| \le \gamma_1\lambda_1 \\ d_{s,ij}^{(k)} & \left|d_{s,ij}^{(k)}\right| > \gamma_1\lambda_1 \end{cases}.$$

Alternating Direction Method of Multipliers(ADMM)

- $\boldsymbol{s}^{(k+1)} = \underset{\boldsymbol{s}}{\operatorname{argmin}}\, L\left(\beta^{(k+1)}, \boldsymbol{\mu}^{(k+1)}, \boldsymbol{z}^{(k+1)}, \boldsymbol{s}, \boldsymbol{w}^{(k)}, \boldsymbol{q}_1^{(k)}, \boldsymbol{q}_2^{(k)}, \boldsymbol{q}_3^{(k)}\right)$.

  The update of $\boldsymbol{s}$ depends on the choice of the loss P_{λ_1}, and it can be computed elementwisely. Let $d_{s,ij}^{(k)} = \mu_i^{(k+1)} - \mu_j^{(k+1)} + \frac{q_{2,ij}^{(k)}}{r_2}$.

  - SCAD with $r_2\,(\gamma_1 - 1) > 1$:

$$s_{ij}^{(k+1)} = \begin{cases} S\left(d_{s,ij}^{(k)}, \frac{\lambda_1}{r_2}\right) & \left|d_{s,ij}^{(k)}\right| \le \left(1 + \frac{1}{r_2}\right)\lambda_1 \\ \dfrac{S\left(d_{s,ij}^{(k)}, \frac{\gamma_1\lambda_1}{r_2(\gamma_1-1)}\right)}{1 - \frac{1}{r_2(\gamma_1-1)}} & \left(1 + \frac{1}{r_2}\right)\lambda_1 < \left|d_{s,ij}^{(k)}\right| \le \gamma_1\lambda_1 \\ d_{s,ij}^{(k)} & \left|d_{s,ij}^{(k)}\right| > \gamma_1\lambda_1 \end{cases}.$$

Alternating Direction Method of Multipliers(ADMM)

- $\boldsymbol{w}^{(k+1)} = \underset{\boldsymbol{w}}{\operatorname{argmin}}\, L\left(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\mu}^{(k+1)}, \boldsymbol{z}^{(k+1)}, \boldsymbol{s}^{(k+1)}, \boldsymbol{w}, \boldsymbol{q}_1^{(k)}, \boldsymbol{q}_2^{(k)}, \boldsymbol{q}_3^{(k)}\right)$.

  The update of $\boldsymbol{w}$ depends on the choice of the loss P_{λ_2}, and it can be computed elementwisely. Let $d_{w,j}^{(k)} = \beta_j^{(k+1)} + \frac{q_{3,j}^{(k)}}{r_3}$.

  - LASSO:

$$w_j^{(k+1)} = S\left(d_{w,j}^{(k)}, \frac{\lambda_2}{r_3}\right).$$

  - MCP with $r_3\gamma_2 > 1$:

$$w_j^{(k+1)} = \begin{cases} \dfrac{S\left(d_{w,j}^{(k)}, \frac{\lambda_2}{r_3}\right)}{1 - \frac{1}{r_3\gamma_2}} & \left|d_{w,j}^{(k)}\right| \le \gamma_2\lambda_2 \\ d_{w,j}^{(k)} & \left|d_{w,j}^{(k)}\right| > \gamma_2\lambda_2 \end{cases}.$$

Alternating Direction Method of Multipliers(ADMM)

- $\boldsymbol{w}^{(k+1)} = \underset{\boldsymbol{w}}{\operatorname{argmin}}\, L\left(\beta^{(k+1)}, \boldsymbol{\mu}^{(k+1)}, \boldsymbol{z}^{(k+1)}, \boldsymbol{s}^{(k+1)}, \boldsymbol{w}, \boldsymbol{q}_1^{(k)}, \boldsymbol{q}_2^{(k)}, \boldsymbol{q}_3^{(k)}\right)$.

  The update of $\boldsymbol{w}$ depends on the choice of the loss P_{λ_2}, and it can be computed elementwisely. Let $d_{w,j}^{(k)} = \beta_j^{(k+1)} + \frac{q_{3,j}^{(k)}}{r_3}$.

  - SCAD with $r_3\,(\gamma_2 - 1) > 1$:

$$w_j^{(k+1)} = \begin{cases} S\left(d_{w,j}^{(k)}, \dfrac{\lambda_2}{r_3}\right) & \left|d_{w,j}^{(k)}\right| \le \left(1 + \dfrac{1}{r_3}\right)\lambda_2 \\ \dfrac{S\left(d_{w,j}^{(k)}, \frac{\gamma_2\lambda_2}{r_3(\gamma_2-1)}\right)}{1 - \frac{1}{r_3(\gamma_2-1)}} & \left(1 + \dfrac{1}{r_3}\right)\lambda_2 < \left|d_{w,j}^{(k)}\right| \le \gamma_2\lambda_2 \\ d_{w,j}^{(k)} & \left|d_{w,j}^{(k)}\right| > \gamma_2\lambda_2 \end{cases}.$$

Alternating Direction Method of Multipliers(ADMM)

- Update the multipliers by

$$\begin{cases} \boldsymbol{q}_1^{(k+1)} = \boldsymbol{q}_1^{(k)} + r_1 \left(\boldsymbol{y}^{(k+1)} - \boldsymbol{\mu}^{(k+1)} - \boldsymbol{X}\boldsymbol{\beta}^{(k+1)} - \boldsymbol{z}^{(k+1)} \right) \\ \boldsymbol{q}_2^{(k+1)} = \boldsymbol{q}_2^{(k)} + r_2 \left(\boldsymbol{D}\boldsymbol{\mu}^{(k+1)} - \boldsymbol{s}^{(k+1)} \right) \\ \boldsymbol{q}_3^{(k+1)} = \boldsymbol{q}_3^{(k)} + r_3 \left(\boldsymbol{\beta}^{(k+1)} - \boldsymbol{w}^{(k+1)} \right) \end{cases},$$

Searching Intervals for λ

We search the grid points of a rectange plane, from $(\lambda_{1,max}, \lambda_{2,max})$ to $(\lambda_{1,min}, \lambda_{2,min})$, for the solution pathes.
We choose $(\lambda_{1,max}, \lambda_{2,max})$ by

$$\lambda_{1,max} = \frac{1}{n} \left\| \boldsymbol{D} \left(\boldsymbol{D}^T \boldsymbol{D} \right)^{-1} \begin{pmatrix} \psi(y_1 - c) \\ \psi(y_2 - c) \\ \vdots \\ \psi(y_n - c) \end{pmatrix} \right\|_\infty \leq \frac{2}{n^2} \left\| \begin{pmatrix} \psi(y_1 - c) \\ \psi(y_2 - c) \\ \vdots \\ \psi(y_n - c) \end{pmatrix} \right\|_\infty ,$$

and

$$\lambda_{2,max} = \frac{1}{n} \left\| \sum_{i=1}^{n} (\psi(y_i - c)) \boldsymbol{x}_i \right\|_\infty ,$$

where $c = \underset{\mu}{\operatorname{argmin}} \sum_{i=1}^{n} \rho(y_i - \mu)$.

Modified BIC

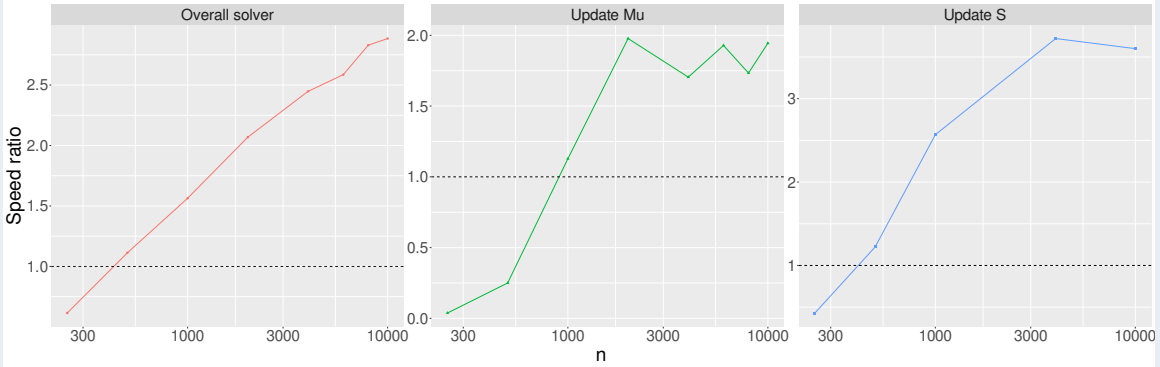
We use the modified BIC to select the tuning parameters.

$$\mathrm{BIC}\left(\hat{\boldsymbol{\mu}}\left(\boldsymbol{\lambda}\right),\hat{\boldsymbol{\beta}}\left(\boldsymbol{\lambda}\right)\right)=\log\left(\frac{1}{n}\sum_{i=1}^{n}\rho\left(y_i-\hat{\mu}_i\left(\boldsymbol{\lambda}\right)-\boldsymbol{x}_i^T\hat{\boldsymbol{\beta}}\left(\boldsymbol{\lambda}\right)\right)\right)+\mathrm{df}\left(\boldsymbol{\lambda}\right)\phi_n, \tag{5}$$

where $\mathrm{df}\left(\boldsymbol{\lambda}\right)=\hat{K}\left(\boldsymbol{\lambda}\right)+\left|\hat{\boldsymbol{\beta}}\left(\boldsymbol{\lambda}\right)\right|_0$ represents the degree of freedom of the model.
$\phi_n=c\frac{\log n}{n}\mathrm{loglog}\left(n+p\right)$ for some constant c.

Parallel Computation

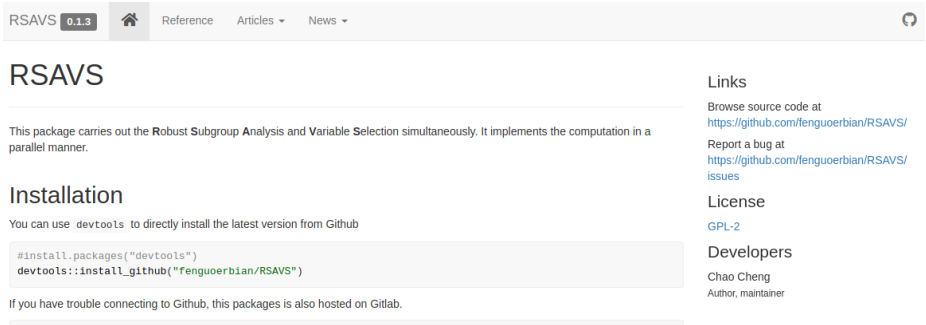
Parallel computation via OpenMP in C++



Practical Computing Techniques

- Warm start strategy.
- A clustering method as a post-processor.
  - Rounding or hard thresholding.
  - Find the grouping result from $\{\hat{s}_{ij}\}$.
  - A simple clustering method, like k-means.
- Search two pathes of λs respectively, not the whole plane.
- Parallel/Distributed computation. We use OpenMP with C++.

Available in an R package



RSAVS 0.1.3 Reference Articles News

RSAVS

This package carries out the **R**obust **S**ubgroup **A**nalysis and **V**ariable **S**election simultaneously. It implements the computation in a parallel manner.

Installation

You can use `devtools` to directly install the latest version from Github

```
#install.packages("devtools")
devtools::install_github("fenguoerbian/RSAVS")
```

If you have trouble connecting to Github, this packages is also hosted on Gitlab.

```
#install.packages("devtools")
devtools::install_gitlab("fenguoerbian/RSAVS")
```

Note: If you are interested in the original yet unoptimized version of this package used in the simulation of our paper, you can check the `simulation_archive` branch of this repo.

Note: It's recommended to read the vignettes on the package's website. But if you want to build the vignettes locally, you can run

```
devtools::install_github("fenguoerbian/RSAVS", force = TRUE, build_vignettes = TRUE)
```

Just keep in mind that this will take some really **long** time.

Links

Browse source code at https://github.com/fenguoerbian/RSAVS/

Report a bug at https://github.com/fenguoerbian/RSAVS/issues

License

GPL-2

Developers

Chao Cheng
Author, maintainer

Outline

Convergence of the Algorithm

- C1 ρ is continuous on $\mathcal{R}$, and differentiable except finite many points.
- C2 ρ is convex.
- C3 ρ has unique minimal point at 0 and $\rho(0) = 0$.

Convergence of the Algorithm

Denote the primal residual at step m by

$$\boldsymbol{r}^{(m)} = \begin{pmatrix} \boldsymbol{y} - \boldsymbol{u}^{(m)} - \boldsymbol{X}^T\beta^{(m)} - \boldsymbol{z}^{(m)} \\ \boldsymbol{D}\boldsymbol{\mu}^{(m)} - \boldsymbol{s}^{(m)} \\ \beta^{(m)} - \boldsymbol{w}^{(m)} \end{pmatrix}, \tag{6}$$

and the dual residual by

$$\boldsymbol{\eta}^{(m+1)} = \begin{pmatrix} r_1\left(\boldsymbol{z}^{(m+1)} - \boldsymbol{z}^{(m)}\right) - r_2\boldsymbol{D}^T\left(\boldsymbol{s}^{(m+1)} - \boldsymbol{s}^{(m)}\right) \\ r_1\boldsymbol{X}^T\left(\boldsymbol{z}^{(m+1)} - \boldsymbol{z}^{(m)}\right) - r_3\left(\boldsymbol{w}^{(m+1)} - \boldsymbol{w}^{(m)}\right) \end{pmatrix}. \tag{7}$$

Convergence of the Algorithm

Theorem

These residuals of the proposed algorithm satisfy that

$$\lim_{m\to\infty} \left\| \boldsymbol{r}^{(m)} \right\|_2^2 = 0, \qquad \lim_{m\to\infty} \left\| \boldsymbol{\eta}^{(m)} \right\|_2^2 = 0$$

for the Lasso, SCAD and MCP penalties if Conditions (C1)–(C3) hold.

Oracle Estimator

Given the real subgroup structure $\boldsymbol{G}_0 = \{G_1, \cdots, G_{K_0}\}$, the original model can be seen as

$$y_i = \boldsymbol{g}_i^T \boldsymbol{\alpha} + \boldsymbol{x}_i^T \boldsymbol{\beta} + \varepsilon_i,$$

where $\boldsymbol{g}_i$s are K_0-dimensional indicator vector, $g_{ij} = 1$ if $i \in G_j$. $\boldsymbol{\alpha}$ is the grouping effect, for all $i \in G_k$, $\mu_i = \alpha_k$.

Oracle Estimator

The oracle estimator is

$$\begin{aligned}\left(\tilde{\boldsymbol{\alpha}}, \tilde{\boldsymbol{\beta}}\right) &= \left(\tilde{\boldsymbol{\alpha}}, \left(\tilde{\boldsymbol{\beta}}_A^T, \mathbf{0}_{p-q}^T\right)^T\right) \\ &= \underset{\boldsymbol{\alpha}, \boldsymbol{\beta}_A}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \rho\left(y_i - \boldsymbol{g}_i^T \boldsymbol{\alpha} - \boldsymbol{x}_{A,i}^T \boldsymbol{\beta}_A\right),\end{aligned} \tag{8}$$

and $\tilde{\mu}_i = \boldsymbol{g}_i^T \tilde{\boldsymbol{\alpha}}$.

Oracle Property

Theorem

Suppose that Assumptions (A1)–(A5) hold. Also if $\max(\lambda_1, \lambda_2) = o\left(n^{-(1-c_2)/2}\right)$, $\sqrt{q(K_0+q)} = o\left(\sqrt{n}\lambda_2\right)$, $(K_0+q)\log n = o(n\lambda_2)$, $\log(p) = o\left(n\lambda_2^2\right)$, $n\lambda_2^2 \to \infty$ *and* $\sqrt{\log n} = o\left(n\lambda_1 |G_k|_{\min}\right)$. *Then there exists a local minimizer* $\left(\hat{\boldsymbol{\mu}}^T, \hat{\boldsymbol{\beta}}^T\right)^T$ *of the objective function (2) coupled with either SCAD or MCP penalty such that*

$$P\left(\left(\hat{\boldsymbol{\mu}}^T, \hat{\boldsymbol{\beta}}^T\right)^T = \left(\tilde{\boldsymbol{\mu}}^T, \tilde{\boldsymbol{\beta}}^T\right)^T\right) \to 1$$

as $n \to \infty$, *where* $\left(\tilde{\boldsymbol{\mu}}^T, \tilde{\boldsymbol{\beta}}^T\right)^T$ *is the oracle estimator.*

Oracle Property (Assumptions) I

(A1) Deisgn matrix is well behaved:
There exist constants M_1, C_1 and C_2 such that

$$C_1 \leq \lambda_{\min}\left(\frac{1}{n}\left(\boldsymbol{G} \quad \boldsymbol{X}_A\right)^T\left(\boldsymbol{G} \quad \boldsymbol{X}_A\right)\right)$$
$$\leq \lambda_{\max}\left(\frac{1}{n}\left(\boldsymbol{G} \quad \boldsymbol{X}_A\right)^T\left(\boldsymbol{G} \quad \boldsymbol{X}_A\right)\right) \leq C_2,$$

where $\boldsymbol{X}_A$ represents the submatrix of $\boldsymbol{X}$ formed by the columns indexed by A. Additionally we assume $\max_{1\leq i\leq n}\left\|\boldsymbol{x}_{i,A}\right\| = O_p(\sqrt{q})$ and $\max_{q+1\leq j\leq p} \mathrm{E}\left|x_j\right| \leq M_1$.

(A2) Model sparsity:
$\max\{K_0, q\} = O(n^{c_1})$ for some $0 \leq c_1 < \frac{1}{3}$.

Oracle Property (Assumptions) II

(A3) Minimal signal strength:
There exist constans c_2 and M_3 such that

$$2c_1 < c_2 \leq 1 \quad \text{and} \quad n^{(1-c_2)/2} b_n \geq M_3,$$

where

$$b_n = \min\left(\min_{i \neq j} \left| \alpha_{0,i} - \alpha_{0,j} \right|, \min_{1 \leq j \leq q} \left| \beta_{0,j} \right| \right).$$

(A4) Sub-gaussian:
There exists a positive constant c_3 such that for all constant $c \in [-c_3, c_3]$,

$$P\left(|\psi(\varepsilon_i + c)| > x \right) \leq 2\exp\left(-c_4 x^2 \right),$$

where c_4 is some positive constant.

Oracle Property (Assumptions) III

(A5) Uniformly continuous:
There exists a positive constant c_6 such that for any $\Delta \in [-c_6, c_6]$, $\mathrm{E}\psi(\varepsilon_i + \Delta)$ and $\mathrm{Var}\psi(\varepsilon_i + \Delta)$ is uniformly continuous with respect to Δ for all $1 \leq i \leq n$.

Selection Consistency

Theorem

Assume the Assumptions (A1)–(A8) hold. Then for any sequence $\phi_n \to 0$ satisfying $\log(n+p)/n = o(\phi_n)$, we have

$$P\left(\inf_{(\tilde{\alpha}(\boldsymbol{G})^T, \tilde{\beta}_{\mathrm{S}}^T) \neq (\tilde{\alpha}(\boldsymbol{G}^0)^T, \tilde{\beta}_{\mathrm{A}}^T)} \mathrm{BIC}\left(\tilde{\alpha}(\boldsymbol{G}), \tilde{\beta}_{\mathrm{S}}\right) > \mathrm{BIC}\left(\tilde{\alpha}(\boldsymbol{G}^0), \tilde{\beta}_{\mathrm{A}}\right)\right) \to 1.$$

Selction Consistency (Assumptions) I

(A6) The number of subgroups K and active covariates $|S|$ are bounded by K_U and q_U, respectively, where $K_U \in (K_0, \infty)$ and $q_U \in (q, \infty)$, and $\limsup_n (K_U + q_U) / n^{\kappa^\star} < 1$ for some constant $\kappa^\star < 1$. In addition, the matrix $(\boldsymbol{G}, \boldsymbol{X}_S)$ satisfies assumption (A1) for the identifiability.

(A7) The regular estimate $\left(\tilde{\alpha}(\boldsymbol{G})^T, \tilde{\beta}_S^T\right)^T$ satisfies

$$\left\| \left(\tilde{\alpha}(\boldsymbol{G})^T, \tilde{\beta}_S^T\right)^T - \left(\alpha(\boldsymbol{G})^T, \beta_S^T\right)^T \right\| = O_p\left(\frac{1}{\sqrt{n}}\right),$$

where $\left(\alpha(\boldsymbol{G})^T, \beta_S^T\right)^T = \arg\min_{\alpha,\beta} \mathrm{E}\rho(Y - \boldsymbol{G}\alpha - \boldsymbol{X}_S\beta_S)$.

Selction Consistency (Assumptions) II

(A8) The classic M-estimate satisfies

$$\sum_{i=1}^{n}\left\{\rho\left(y_i-\boldsymbol{g}_i^T\tilde{\boldsymbol{\alpha}}\left(\boldsymbol{G}\right)-\boldsymbol{x}_{i,S}^T\tilde{\boldsymbol{\beta}}_S\right)-\rho\left(y_i-\boldsymbol{g}_i^T\boldsymbol{\alpha}\left(\boldsymbol{G}\right)-\boldsymbol{x}_{i,S}^T\boldsymbol{\beta}_S\right)\right\}$$

$$=-\sum_{i=1}^{n}\psi\left(y_i-\boldsymbol{g}_i^T\boldsymbol{\alpha}\left(\boldsymbol{G}\right)-\boldsymbol{x}_{i,S}^T\boldsymbol{\beta}_S\right)\left\{\boldsymbol{g}_i^T\left(\tilde{\boldsymbol{\alpha}}\left(\boldsymbol{G}\right)-\boldsymbol{\alpha}\left(\boldsymbol{G}\right)\right)+\boldsymbol{x}_{i,S}^T\left(\tilde{\boldsymbol{\beta}}_S-\boldsymbol{\beta}_S\right)\right\}+O_p\left(1\right).$$

Outline

Simulation (Basic Settings) I

Consider $y_i = \mu_i + \boldsymbol{x}_i^T \beta + \varepsilon_i$, $i = 1, 2, \cdots, n$, under various scenarios.

- n ranges from 200 to 1000. p ranges from 5 to 500.
- $q = 5$, $\beta = \left(\mathbf{1}_5^T, \mathbf{0}_{p-5}^T\right)^T$.
- $K = 2$ with $\boldsymbol{\alpha} = (-1, 1)^T$ or $K = 3$ with $\boldsymbol{\alpha} = (-2, 0, 2)^T$, or $K = 5$ with $\boldsymbol{\alpha} = (-4, -2, 0, 2, 4)^T$.
- μ_is follow the independent multinormial distribution over $\boldsymbol{\alpha}$ with equal probability.
- $\boldsymbol{x}_i$ follows standard normal distribution.

Simulation (Basic Settings) II

Error term

$\varepsilon_i = 0.5\epsilon_i$ independently, where

1. $\epsilon_i \sim \mathrm{N}(0,1)$
2. $\epsilon_i \sim t(5)$.
3. $\epsilon_i \sim 0.95 \times \mathrm{N}(0,1) + 0.05 \times \mathrm{N}\left(0, 10^2\right)$.

Simulation (Basic Settings) III

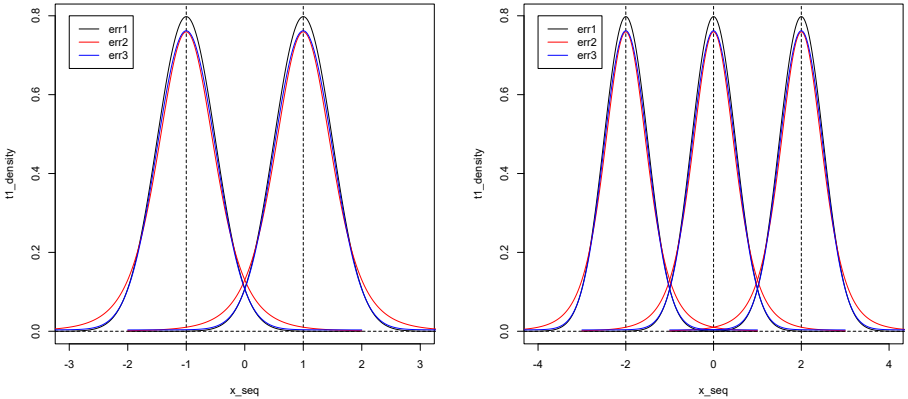


Figure 1: A demonstration about data points overlapping under our simulation settings.

Simulation (Basic Settings) IV

Candidate methods

- The proposed method, with L_1 and Huber loss.
- The proposed method, with L_2 loss[Ma and Huang, 2017].
- RSI(Robust Subgroup Identification, Zhang et al. [2019]).

Simulation (Basic Settings) V

Various metrics are considered:

- MAE_{μ} and MAE_{β}: the mean absolute error of μ and β.
- $\bar{K}$, $\tilde{K}$, $\bar{q}$ and $\tilde{q}$: the average and median value of the estimated number of subgroups and active covariates respectively.
- *RI* [Rand, 1971, Rand Index]: this index describes how close two grouping results are and is computed by

$$RI\left(\boldsymbol{G}_1, \boldsymbol{G}_2\right) = \frac{2}{n(n-1)} \times (TP + TN).$$

Simulation Results (Low-dimensional) I

Table 1: Error distribution is $0.5 \times t(5)$, $n = 200$, $p = q = 5$, $K \in \{2, 3\}$.

K	Method	MAE_{μ}	MAE_{β}	$\bar{K}$	$\bar{RI}$
2	L_1	0.172(0.044)	0.045(0.017)	2.000(0.000)	0.890(0.031)
	L_2	0.473(0.385)	0.056(0.026)	1.944(1.226)	0.748(0.186)
	Huber	0.169(0.068)	0.044(0.018)	1.996(0.063)	0.890(0.039)
	RSI	0.179(0.126)	0.069(0.059)	2.010(0.184)	0.891(0.040)
3	L_1	0.734(0.155)	0.105(0.042)	2.078(0.283)	0.747(0.049)
	L_2	0.738(0.501)	0.087(0.038)	3.504(2.326)	0.672(0.246)
	Huber	0.332(0.192)	0.075(0.038)	3.038(0.567)	0.868(0.057)
	RSI	0.400(0.274)	0.106(0.077)	2.746(0.508)	0.853(0.081)

Simulation Results (Low-dimensional) II

Table 2: Error distribution is $0.5 \times t(5)$, $n = 400$, $p = q = 5$, $K \in \{2, 3\}$.

K	Method	MAE_μ	MAE_β	$\bar{K}$	$\bar{RI}$
2	L_1	0.146(0.029)	0.031(0.011)	2.000(0.000)	0.899(0.019)
	L_2	0.365(0.180)	0.041(0.015)	4.916(1.969)	0.752(0.099)
	Huber	0.146(0.029)	0.030(0.010)	2.000(0.000)	0.898(0.019)
	RSI	0.179(0.189)	0.061(0.100)	2.122(0.851)	0.889(0.061)
3	L_1	0.323(0.236)	0.048(0.027)	2.788(0.409)	0.871(0.073)
	L_2	0.427(0.355)	0.054(0.024)	4.482(2.149)	0.806(0.171)
	Huber	0.219(0.106)	0.042(0.018)	3.002(0.233)	0.901(0.032)
	RSI	0.302(0.333)	0.103(0.171)	3.080(0.975)	0.885(0.069)

Simulation Results (High-dimensional) I

Table 3: Error distribution is $0.5 \times t(5)$, $n = 200$, $p = 100$, $q = 5$, $K \in \{2, 3\}$.

K	Method	MAE_μ	$\bar{K}$	$\tilde{K}$	$\bar{q}$	$\bar{q}_{TP}$	$\bar{RI}$
2	L_1	0.232(0.066)	2.000(0.000)	2	5.000(0.063)	4.998(0.045)	0.842(0.044)
	L_2	0.994(0.013)	1.000(0.000)	1	0.000(0.000)	0.000(0.000)	0.501(0.000)
	Huber	0.371(0.286)	1.922(0.461)	2	5.418(1.007)	4.798(0.240)	0.779(0.135)
	RSI	0.893(0.231)	6.398(2.666)	6	100.000(0.000)	5.000(0.000)	0.535(0.024)
3	L_1	0.817(0.084)	2.000(0.000)	2	4.780(0.670)	4.742(0.629)	0.720(0.026)
	L_2	1.425(0.036)	1.004(0.063)	1	0.000(0.000)	0.000(0.000)	0.333(0.005)
	Huber	0.720(0.261)	2.314(0.494)	2	4.844(1.011)	4.688(0.851)	0.742(0.087)
	RSI	1.325(0.302)	6.470(2.599)	7	100.000(0.000)	5.000(0.000)	0.610(0.043)

Simulation Results (High-dimensional) II

Table 4: Error distribution is $0.5 \times t(5)$, $n = 400$, $p = 100$, $q = 5$, $K \in \{2, 3\}$.

K	Method	MAE_μ	$\bar{K}$	$\breve{K}$	$\bar{q}$	$\bar{q}_{TP}$	$\bar{RI}$
2	L_1	0.161(0.035)	2.000(0.000)	2	5.000(0.000)	5.000(0.000)	0.883(0.024)
	L_2	1.000(0.001)	1.000(0.000)	1	4.790(1.004)	4.790(1.004)	0.499(0.000)
	Huber	0.179(0.076)	2.018(0.133)	2	5.004(0.063)	5.000(0.000)	0.873(0.037)
	RSI	0.853(0.349)	6.976(2.929)	8	100.000(0.000)	5.000(0.000)	0.578(0.079)
3	L_1	0.772(0.025)	2.000(0.000)	2	5.036(0.186)	5.000(0.000)	0.731(0.005)
	L_2	1.406(0.026)	1.006(0.100)	1	0.000(0.000)	0.000(0.000)	0.333(0.004)
	Huber	0.439(0.273)	2.672(0.470)	3	5.350(0.981)	5.000(0.000)	0.832(0.075)
	RSI	1.321(0.512)	7.500(2.651)	8.5	100.000(0.000)	5.000(0.000)	0.655(0.025)

Simulation Results (High-dimensional) III

Table 5: Error distribution is $0.5 \times t(5)$, $n = 200$, $p = 500$, $q = 5$, $K \in \{2, 3\}$.

K	Method	MAE_μ	$\bar{K}$	$\tilde{K}$	$\bar{q}$	$\tilde{q}_{TP}$	$\bar{RI}$
2	L_1	0.389(0.304)	2.000(0.000)	2	4.360(1.375)	4.360(1.375)	0.782(0.105)
	L_2	0.995(0.031)	1.010(0.100)	1	0.000(0.000)	0.000(0.000)	0.501(0.002)
	Huber	0.585(0.462)	2.140(0.450)	2	4.290(1.924)	4.050(1.666)	0.732(0.131)
3	L_1	0.938(0.140)	2.000(0.000)	2	3.610(1.456)	3.580(1.408)	0.677(0.049)
	L_2	1.425(0.036)	1.000(0.000)	1	0.000(0.000)	0.000(0.000)	0.333(0.004)
	Huber	0.925(0.236)	2.300(0.461)	2	3.410(1.718)	3.380(1.674)	0.687(0.065)

Simulation Results (High-dimensional) IV

Table 6: Error distribution is $0.5 \times t(5)$, $n = 400$, $p = 500$, $q = 5$, $K \in \{2, 3\}$.

K	Method	MAE_μ	$\bar{K}$	$\tilde{K}$	$\bar{q}$	$\tilde{q}_{TP}$	$\bar{RI}$
2	L_1	0.175(0.037)	2.000(0.000)	2	5.000(0.000)	5.000(0.000)	0.875(0.028)
	L_2	0.999(0.001)	1.000(0.000)	1	4.060(1.953)	4.060(1.953)	0.499(0.000)
	Huber	0.242(0.162)	2.120(0.409)	2	5.040(0.243)	5.000(0.000)	0.841(0.071)
3	L_1	0.776(0.038)	2.000(0.000)	2	4.960(0.315)	4.960(0.315)	0.729(0.010)
	L_2	1.404(0.021)	1.000(0.000)	1	0.000(0.000)	0.000(0.000)	0.333(0.000)
	Huber	0.587(0.274)	2.450(0.500)	2	5.350(0.968)	5.000(0.000)	0.788(0.075)

Simulation Results (Large-scale Data)

Table 7: Error distribution is $0.95 \times \mathrm{N}\left(0, 0.5^2\right) + 0.05 \times \mathrm{N}\left(0, 5^2\right)$, $n = 1000$, $p = 200$, $q = 5$, $K \in \{2, 3\}$.

K	Method	MAE_μ	$\bar{K}$	$\tilde{K}$	$\bar{q}$	$\bar{q}_{TP}$	$\bar{RI}$
2	L_1	0.339(0.374)	1.770(0.446)	2	5.330(0.865)	5.000(0.000)	0.808(0.175)
	L_2	1.022(0.029)	1.650(0.770)	1	5.000(0.000)	5.000(0.000)	0.500(0.000)
	Huber	0.887(0.292)	1.130(0.338)	1	5.140(0.725)	5.000(0.000)	0.552(0.136)
	RSI	1.069(0.034)	10.210(1.559)	10	200.000(0.000)	5.000(0.000)	0.501(0.001)
3	L_1	0.931(0.141)	1.940(0.239)	2	7.320(1.786)	5.000(0.000)	0.692(0.091)
	L_2	1.377(0.024)	1.140(0.450)	1	4.880(0.715)	4.800(0.715)	0.334(0.002)
	Huber	0.570(0.511)	2.380(0.874)	3	5.850(1.507)	5.000(0.000)	0.737(0.247)
	RSI	1.457(0.056)	11.980(1.484)	12	200.000(0.000)	5.000(0.000)	0.468(0.042)

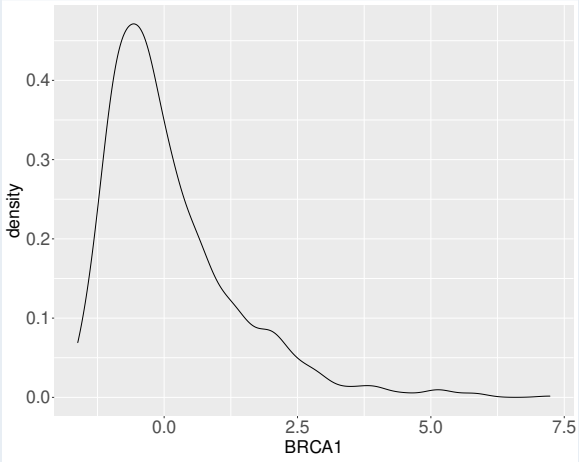
Outline

mRNA Expression of Breast Cancer Patients I

- The BReast CAncer(BRCA) data from The Cancer Genome Atlas(TCGA)
- 1074 patients and 8571 genes after pre-screening.
- The mRNA expression level of BRCA1 is predictive for patients' response/sensitivity to certain chemotherapy treatment.

mRNA Expression of Breast Cancer Patients II

Heavy-tail and skewed data



Subgroup Identification

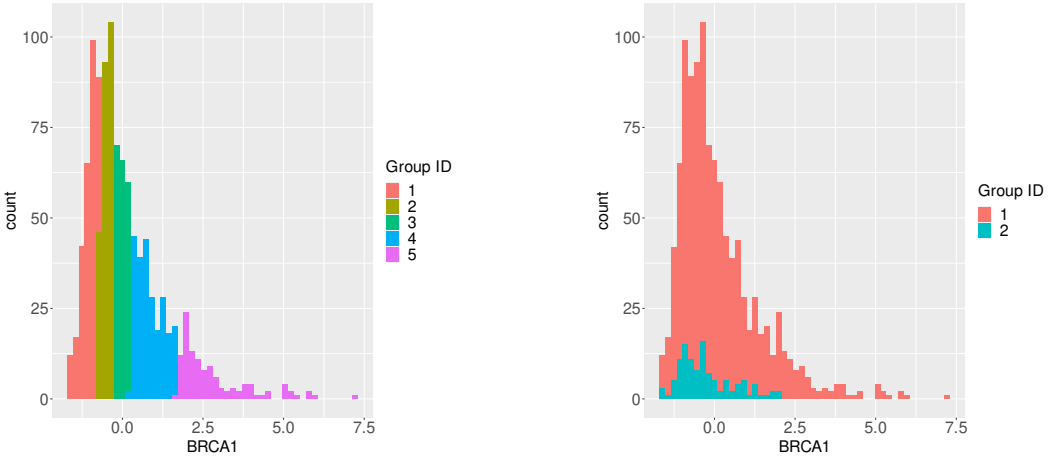
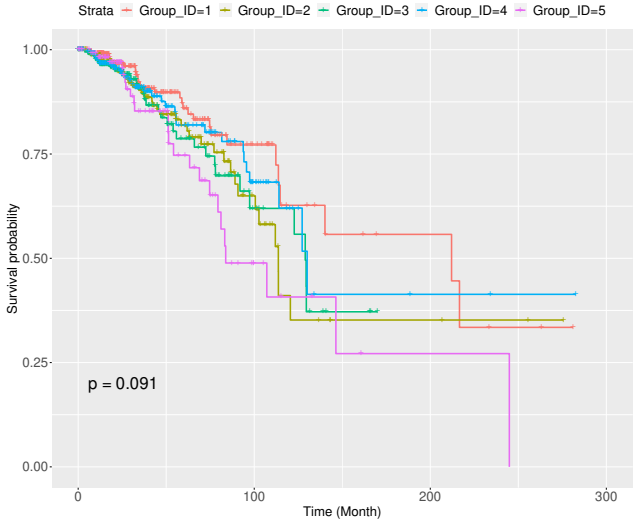


Figure 2: The estimated subgroups from different methods: Left panel is the proposed method. Right panel is RSI with divide-and-conquer strategy.

Characteristics of the Identified Subgroups



Characteristics of the Identified Subgroups

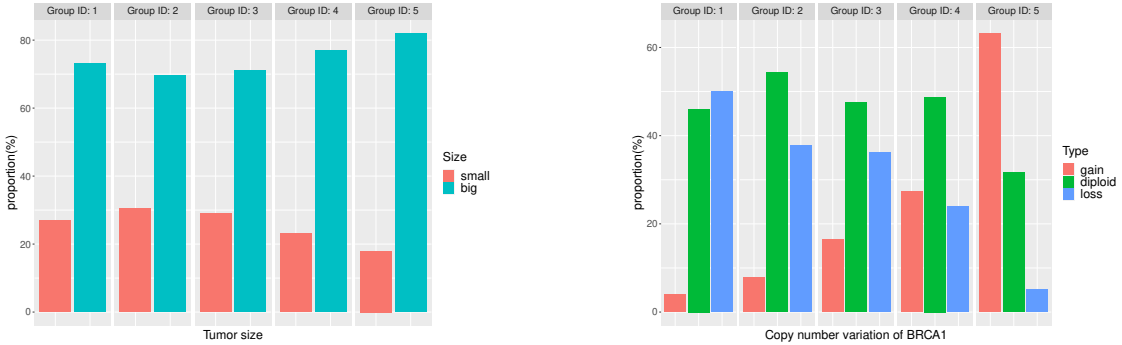
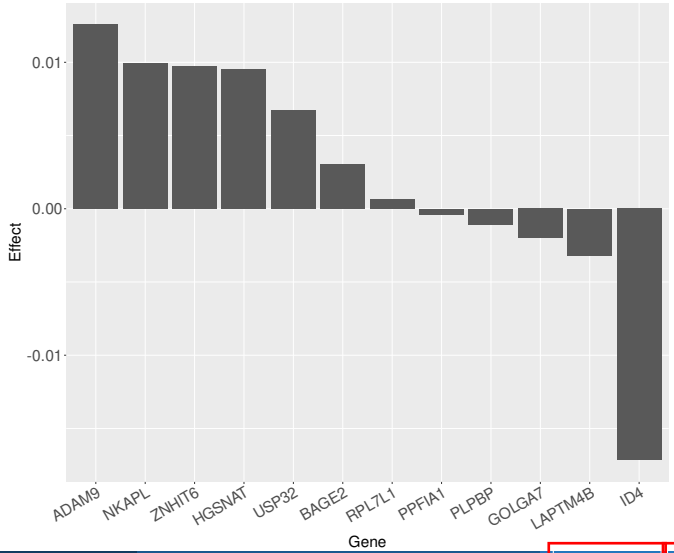


Figure 3: Left panel is tumor size. Right panel is copy number variation.

Selected Genes



Outline

Contributions

- A robust method for simultaneous subgroup identification and variable selection in high-dimensional data.
- Use parallel computation to speed up the computation.
- Theoretical results about oracle property and selection consistency is established.

Future Work

- It is conceptually straightforward to extend to other models, such as logistic regression.
- Speed up ADMM. It requires careful design and study of the algorithm for extra large datasets.
- The proposed method focuses on data exploration, how to extend it to prediction?
- Theoretical results for inference, especially posterior inference.

Reference I

Jingxiang Chen, Quoc Tran-Dinh, Michael R. Kosorok, and Yufeng Liu. Identifying heterogeneous effect using latent supervised clustering with adaptive fusion. *Journal of Computational and Graphical Statistics*, 30(1):43–54, jun 2020. doi: 10.1080/10618600.2020.1763808.

Eric C. Chi and Kenneth Lange. Splitting methods for convex clustering. *Journal of Computational and Graphical Statistics*, 24(4):994–1013, oct 2015. doi: 10.1080/10618600.2014.948181.

Shujie Ma and Jian Huang. A Concave Pairwise Fusion Approach to Subgroup Analysis. *Journal of the American Statistical Association*, 112(517):410–423, jan 2017. doi: 10.1080/01621459.2016.1148039.

William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, dec 1971. doi: 10.1080/01621459.1971.10482356.

Reference II

Mingyang Ren, Sanguo Zhang, Qingzhao Zhang, and Shuangge Ma. Gaussian graphical model-based heterogeneity analysis via penalized fusion. *Biometrics*, feb 2021. doi: 10.1111/biom.13426.

Juan Shen and Xuming He. Inference for subgroup analysis with a structured logistic-normal mixture model. *Journal of the American Statistical Association*, 110 (509):303–312, jan 2015. doi: 10.1080/01621459.2014.894763.

Chong Wu, Sunghoon Kwon, Xiaotong Shen, and Wei Pan. A new algorithm and theory for penalized regression-based clustering. *Journal of Machine Learning Research*, 17 (188):1–25, 2016. URL `http://jmlr.org/papers/v17/15-553.html`.

Yingying Zhang, Huixia Judy Wang, and Zhongyi Zhu. ROBUST SUBGROUP IDENTIFICATION. *Statistica Sinica*, 2019. doi: 10.5705/ss.202017.0179.

Acknowledgement

Thank you!