Report

Contents

1 Abstract

Correct indentification of subgroups of a heterogeneous population is an important step in developing precision medicine. In this report, we develop a two-stage data-driven efficient method to detect the subgroups existing in the lung cancer population under the $p >> n$ situation. We assume the heterogeneity mainly comes from the unobserved latent factors. After adjusting for the effects of a subset of the covariates $\mathbf{x}_i = (X_{i1}, \ldots, X_{ip})$ using the Lasso procudure, we apply the k-means algorithm to divide the heterogenous population into subgroups using dynamic programming based the elbow rule with the modified BIC criterion. Finally we verify the proposed method by applying our method to the publicly available lung cancer genomic data.

2 Introduction

Biological heterogeneity is common in many diseases; heterogeneity complicates clinical management, as it is often the main reason for prognostic and therapeutic failures. Thus there have been many attempts to identify subgroups from a heterogenous population. One of the most popular methods for finding subgroups from a heterogenous population is the mixture model. The mixture model-based approach often needs to specify the underlying distribution for data and the number of subgroups, which is impractical. Moreover, this method is not suitable for the high dimensional setting where the number of covariates p is much larger than the sample size n.

3 Model

In this project, we consider the following model:

$$Y_i = \mu_i + \mathbf{x}_i^T \boldsymbol{\beta} + \epsilon_i, i = 1, \cdots, n,$$

where Y_i is the response for the i-th subject, μ_i's are unknown subject-specific intercepts, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ is the coefficients vector for the covariates $\mathbf{x}_i$ and ϵ_i is the error term independent of $\mathbf{x}_i$ with $\mathbb{E}(\epsilon_i) = 0$ and $\mathrm{Var}(\epsilon_i) = \sigma^2$. In a biomedical study, e.g., the lung cancer genomic data, Y_i could be a certain phenotype associated or suvival time with some disease such as "Disease-Free Survival Time", $\mathbf{x}_i$ could be a set of clinical variables such as gender, age, race, and the expression levels of Agilent miRNA probes. When considering the expression levels of miRNA, the sample size n is usually smaller than the dimension of covariates p. Thus the sparsity principle applied naturally to address the high dimension issue $p > n$ where only a subset $\mathcal{S}$ of $\boldsymbol{\beta}$ is nonzero. We assume $|\mathcal{S}| = s$.

In the situation $p < n$, Ma and Huang (2015) considered optimizing the following objective fucntion

$$Q_n(\boldsymbol{\mu}, \boldsymbol{\beta}; \lambda) = 1/2 \sum_{i=1}^{n} (Y_i - \mu_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \sum_{1 \le i < j \le n} p(|\mu_i - \mu_j|, \lambda),$$

where $p(\cdot, \lambda)$ is a concave penalty function with a tuning parameter $\lambda > 0$, and applied the alternating direction method of multipliers (ADMM) algorithm to minimize $Q_n(\boldsymbol{\mu}, \boldsymbol{\beta}; \lambda)$. ADMM algorithm is essentially a first order algorithm, hence its convergence rate is very slow. And for n is large, this issue becomes more worse because we should optimize at lease $\binom{n}{2}$ parameters. A nature extension to the spirit of Shujie Ma, Huang Jian (2016) is to add $\boldsymbol{\beta}$ to the penalty term by optimizing the following objective function

$$\widetilde{Q}_n(\boldsymbol{\mu}, \boldsymbol{\beta}; \lambda) = 1/2 \sum_{i=1}^{n} (Y_i - \mu_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \alpha_1 \sum_{1 \le i < j \le n} p(|\mu_i - \mu_j|, \lambda) + \alpha_2 p(|\boldsymbol{\beta}|, \omega),$$

where α_1 and α_2 is the weight, and apply the ADMM algorithm, which makes the issue more severe. At this time, the ADMM algorithm has to iterate on at lease $\binom{n}{2} + p$ parameters sequentially where the slow convergence rate issue stands out.

4 Algorithm

Instead we first use Lasso procedure to select related covariates from $\mathbf{x}_i$, then apply k-means algorithm to the adjusted $\widetilde{Y}_i$. For the Lasso procedure, we minizize the following objective function

$$(\widehat{\mu}(\lambda), \widehat{\boldsymbol{\beta}}(\lambda)) = \arg \min_{(\mu, \boldsymbol{\beta}) \in \mathbb{R}^{p+1}} 1/2 \sum_{i=1}^{n} (Y_i - \mu - \mathbf{x}_i^T \boldsymbol{\beta})^2 + p(\boldsymbol{\beta}, \lambda). \tag{1}$$

Based on this, we can adjust the common effect existing in $\mathbf{x}_i$. Specifically, let $\widehat{\lambda}$ be the value of the tuning parameter selected based on a data-driven procedure such as Bayesian information criterion (BIC). For brevity, write $\widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}(\widehat{\lambda})$. Then let $\widetilde{Y}_i = Y_i - \mathbf{x}_i^T \widehat{\boldsymbol{\beta}}$. We apply the k-means algorithm to the adjusted $\widetilde{Y}_i$.

4.1 The choice of concave penalty function

For the choice of penalty function, we can simply use the ℓ_1-norm penalty (Tibshirani, 1996, LASSO), smoothly clipped absolute deviations penalty (Fan and Li, 2001, SCAD) or minimax concave penalty (Zhang, 2010, MCP). Interested readers may refer to Hastie et al. (2015) and references therein for comprehensive reviews on recent developments. In the present context we adopt ℓ_1-norm penalty $p(\boldsymbol{\beta}, \lambda) = \lambda |\boldsymbol{\beta}|_1$ for simplicity.

4.2 Optimization methods

The first order methods, such as Combettes and Pesquet (2011), Bach et al. (2012) and Tropp and Wright (2010), and Newton type algorithms, such as Fountoulakis et al. (2014) and Dassios et al. (2015), can be used to optimize (1). We suggest to use the primal dual active set method (Fan et al., 2014), which is in spirit a generalized version of Newton type method. It usually converges after one-step iteration if there is a very good initial value. We globalize it with continuation on regularization parameter and take a maximum vote

regularization parameter selection rule incorporated along with the continuation procedure without extra computation overhead (see Huang et al., 2018).

It remains to analysis the subgroups in $\widetilde{\mathbf{Y}} = (\widetilde{Y_1}, \ldots, \widetilde{Y_n})^T$. This is an unsupervised cluster problem. Many loss function can be adopted to find the cluster modes, e.g., k-means, k-median, k-modes, even the deep learing algorithm. The k-means problem is to partition data into k groups such that the sum of squared Euclidean distances to each group mean is minimized. However, the problem is NP-hard in a general Euclidean space, even when the number of clusters k is 2 (Aloise et al., 2009). The standard iterative k-means algorithm (Llyod, 1982) is a widely used heuristic solution. The algorithm iteratively calculates the within-cluster sum of squared distances, modifies group membership of each point to reduce the within-cluster sum of squared distances, and computes new cluster centers until local convergence is achieved. The time complexity of this standard k-means algorithm is $O(qknp)$, where q is the number of iterations, k is the number of clusters. In our context, we take k-means for its simplicity. There exists dynamic programming algorithm for optimal one-dimensional clustering (see Wang and Song, 2011). They implement this algorithm as an R package called Ckmeans.1d.dp. This exact dynamic programming solution with a runtime of $O(n^2k)$ to the 1-D k-means problem.

4.3 Selection of the tuning parameters k and λ

We select the tuning parameter λ by validation in terms of mse $1/n \sum_{i=1}(Y_i - \widehat{\mu} - \mathbf{x}_i^T \widehat{\boldsymbol{\beta}})^2$. And the the number of cluster k by decided by bayesian information criterion (BIC)

$$\mathrm{BIC}(k) = \log \left[\sum_{i=1}^{n} \{Y_i - \widehat{\mu_i}(k) - \mathbf{x}_i^T \widehat{\boldsymbol{\beta}}\}^2 / n \right] + C_n \log n / n(k+s),$$

where $C_n = c \log\{\log(n+s)\}$ with some positive constant c. We take $c = 10$ by defaults.

5 Emperical example

5.1 Chemores Data Example

Lung cancer is one of the most prevalent and deadiest cancers, which can be classified into two major subtypes, small cell lung cancer (SCLC) and non-small cell lung cancer (NSCLC). NSCLC accouting 80% of all primary lung cancers, is a known heterogeneous group and its prognosis is generally poor (Tsuboi et al., 2007). In the current clinical practice, it is difficult to perform histopathological classification with small biopsies (Orenstein, 2012). There is an urgent to analysis the subgroups existing in NSCLC population.

Publicly available lung cancer genomic data from the Chemores Cohort Study. This data is part of an integrated study of mRNA, miRNA and clinical variables to characterize the molecular distinctions between squamous cell carcinoma (SCC) and adenocarcinoma (AC) in Non Small Cell Lung Cancer (NSCLC) aside large cell lung carcinoma (LCC). Tissue samples were analysed from a cohort of 123 patients, who underwent complete surgical resection at the Institut Mutualiste Montsouris (Paris, France) between 30 January 2002 and 26 June 2006. All the patients belong to NSCLC. The studied outcome was the "Disease-Free Survival Time". Patients were followed until the first relapse occurred or administrative censoring. In this genomic dataset, the expression levels of Agilent miRNA probes (p=939) were included from the n=123 cohort samples. The miRNA data contains normalized expression levels. See below the paper by Lazar et al. (2013) and Array Express data repository for complete description of the samples, tissue preparation, Agilent array technology, and data normalization. In addition to the genomic data, five clinical variables, also evaluated on the cohort samples, are included as continuous variable ('Age') and nominal variables ('Type','KRAS.status','EGFR.status','P53.status'). Data is available here.

5.2 Data description

```r
knitr::opts_chunk$set(
  collapse = TRUE,
  comment = "#>"
)
options(warn = -1) # suppress warings globally

library(PRIMsrc)
```

```
## Loading required package: survival
```

```
## Loading required package: glmnet
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 3.0-1
```

```
## Loading required package: superpc
```

```
## Loading required package: Hmisc
```

```
## Loading required package: lattice
```

```
## Loading required package: Formula
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
## Loading required package: quantreg
```

```
## Loading required package: SparseM
```

```
##
## Attaching package: 'SparseM'
```

```
## The following object is masked from 'package:base':
##
##     backsolve
```

```
##
## Attaching package: 'quantreg'
```

```
## The following object is masked from 'package:Hmisc':
##
##     latex
```

```
## The following object is masked from 'package:survival':
##
##     untangle.specials
```

```
## PRIMsrc 0.8.2 and > 0.7.0 are major releases with significant user-visible changes.
```

```
## Type PRIMsrc.news() to see new features, changes, and bug fixes.
```

```
##
## Attaching package: 'PRIMsrc'
```

```
## The following objects are masked from 'package:Matrix':
##
##     print, summary
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --
```

```
## v tibble  2.1.3     v purrr   0.3.3
## v tidyr   1.0.2     v dplyr   0.8.3
## v readr   1.3.1     v stringr 1.4.0
## v tibble  2.1.3     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::expand()    masks Matrix::expand()
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()       masks stats::lag()
## x tidyr::pack()      masks Matrix::pack()
## x dplyr::src()       masks Hmisc::src()
## x dplyr::summarize() masks Hmisc::summarize()
## x tidyr::unpack()    masks Matrix::unpack()
```

```
library(glmnet)
library(Amelia) # for missmap
```

```
## Loading required package: Rcpp
```

```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.6, built: 2019-11-24)
## ## Copyright (C) 2005-2020 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```
library(Ckmeans.1d.dp)
```

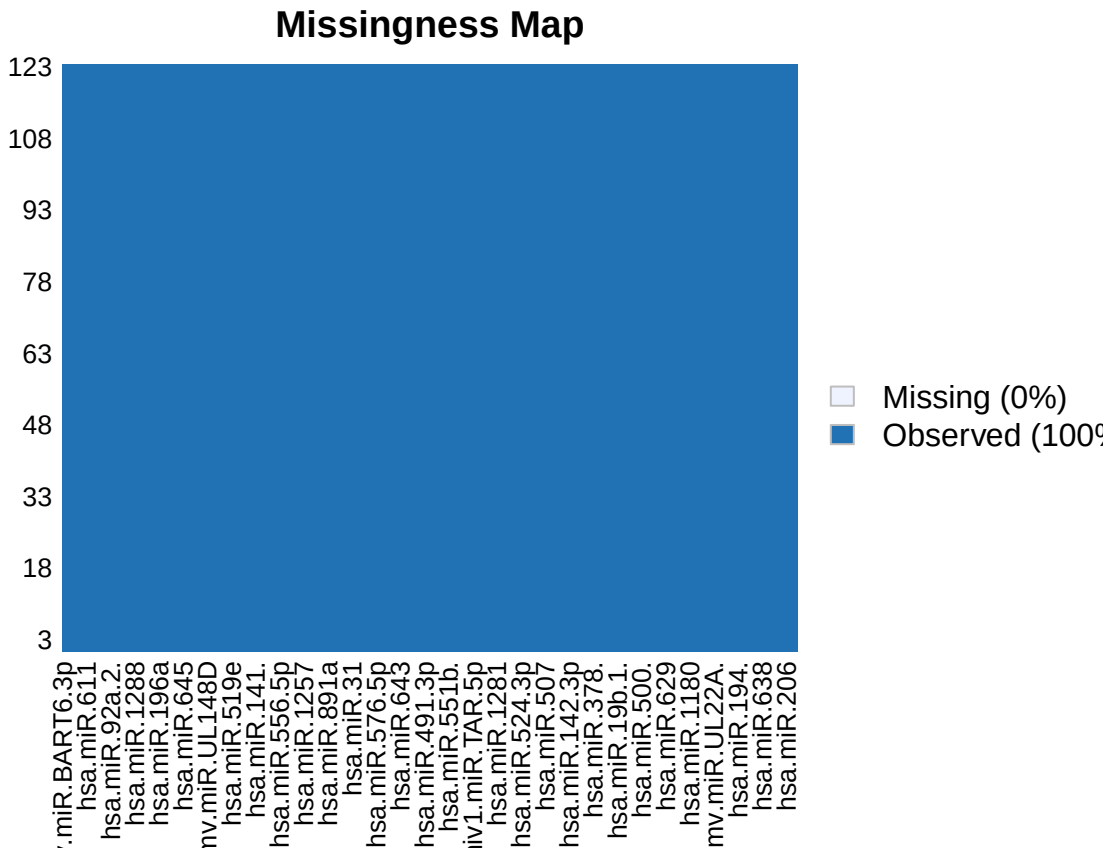
There are $n = 123$ individuals with covariates $p = 946$. This meets our expectation $p > n$.

```
head(Real.2[,1:10])
#>             y delta   Age Type KRAS.status EGFR.status P53.status hsa.miR.555
#> AGG600716 1.9     1 45.26    1           1           0          1   1.2584033
#> ANO420520 4.3     0 63.81    1           1           0          1   1.4273631
#> ARC270517 5.1     0 78.44    3           0           0          1   0.4226154
#> AVI260916 7.7     0 76.39    1           0           0          0   0.7036005
#> AZE450213 3.5     1 59.84    1           0           0          0   1.5351541
#> BAR331123 4.5     0 72.47    3           0           0          0   1.1982788
#>           hsa.miR.223. hsa.miR.346
#> AGG600716    -2.107231    2.201386
#> ANO420520    -2.540758    2.784025
#> ARC270517     2.444766    2.633147
#> AVI260916    -2.315074    2.158156
#> AZE450213    -2.888803    2.233118
#> BAR331123    -2.934864    3.079454
tail(Real.2[,1:10])
#>             y delta   Age Type KRAS.status EGFR.status P53.status hsa.miR.555
#> UST500306 6.3     0 53.57    3           0           0          0   1.0848988
#> VAL271009 7.0     0 75.30    3           0           0          0   0.3438152
#> VIL310309 5.9     0 72.66    3           0           0          1   0.8035270
```

```
#> WIS320823 6.0     0 70.75    3           0           0          0   0.4342302
#> YOT471216 0.2     1 56.96    1           0           0          0   1.9427728
#> ZIT420630 3.3     1 62.73    1           1           0          1   1.1894675
#>           hsa.miR.223. hsa.miR.346
#> UST500306   -0.9443907    2.938560
#> VAL271009   -1.1335018    2.507751
#> VIL310309   -3.5112075    2.891635
#> WIS320823   -3.0243737    3.046549
#> YOT471216   -1.8190561    2.638531
#> ZIT420630   -2.4424078    3.244648
dim(Real.2)
#> [1] 123 946
str(Real.2[,1:10])
#> 'data.frame':    123 obs. of  10 variables:
#>  $ y           : num  1.9 4.3 5.1 7.7 3.5 4.5 3.5 1 6.4 1 ...
#>  $ delta       : int  1 0 0 0 1 0 0 1 0 1 ...
#>  $ Age         : num  45.3 63.8 78.4 76.4 59.8 ...
#>  $ Type        : int  1 1 3 1 1 3 1 1 3 3 ...
#>  $ KRAS.status : int  1 1 0 0 0 0 0 1 0 0 ...
#>  $ EGFR.status : int  0 0 0 0 0 0 0 0 0 0 ...
#>  $ P53.status  : int  1 1 1 0 0 0 0 0 0 0 ...
#>  $ hsa.miR.555 : num  1.258 1.427 0.423 0.704 1.535 ...
#>  $ hsa.miR.223.: num  -2.11 -2.54 2.44 -2.32 -2.89 ...
#>  $ hsa.miR.346 : num  2.2 2.78 2.63 2.16 2.23 ...
summary(Real.2[,1:10])
#>        y             delta             Age             Type
#>  Min.   :0.000   Min.   :0.0000   Min.   :40.88   Min.   :1.000
#>  1st Qu.:1.150   1st Qu.:0.0000   1st Qu.:57.01   1st Qu.:1.000
#>  Median :3.300   Median :0.0000   Median :62.97   Median :2.000
#>  Mean   :3.246   Mean   :0.4797   Mean   :63.90   Mean   :1.992
#>  3rd Qu.:5.100   3rd Qu.:1.0000   3rd Qu.:70.94   3rd Qu.:3.000
#>  Max.   :7.700   Max.   :1.0000   Max.   :84.65   Max.   :4.000
#>   KRAS.status      EGFR.status       P53.status      hsa.miR.555
#>  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :-0.6131
#>  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 0.6958
#>  Median :0.0000   Median :0.0000   Median :0.0000   Median : 1.1983
#>  Mean   :0.1626   Mean   :0.1057   Mean   :0.2358   Mean   : 1.2192
#>  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.: 1.7325
#>  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   : 2.6162
#>   hsa.miR.223.      hsa.miR.346
#>  Min.   :-3.816   Min.   :1.172
#>  1st Qu.:-3.015   1st Qu.:2.288
#>  Median :-2.541   Median :2.651
#>  Mean   :-1.843   Mean   :2.634
#>  3rd Qu.:-1.555   3rd Qu.:3.008
#>  Max.   : 4.661   Max.   :3.691
(colSums(Real.2==0)/ncol(Real.2))[1:10] ## zero-inflate effect
#>           y       delta         Age        Type  KRAS.status  EGFR.status
#>  0.001057082  0.067653277  0.000000000  0.000000000  0.108879493  0.116279070
#>   P53.status  hsa.miR.555 hsa.miR.223.  hsa.miR.346
#>  0.099365751  0.000000000  0.000000000  0.000000000
```

Then we explore the missing patern of this dataset.

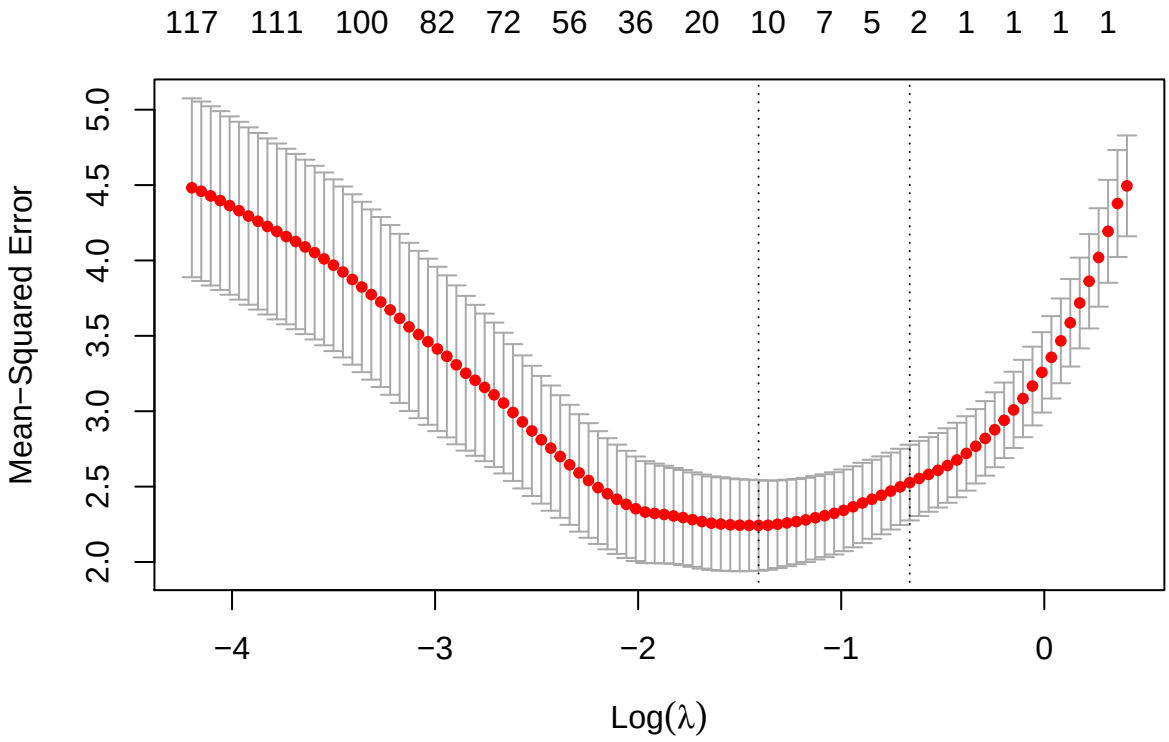
```
missmap(Real.2)
```



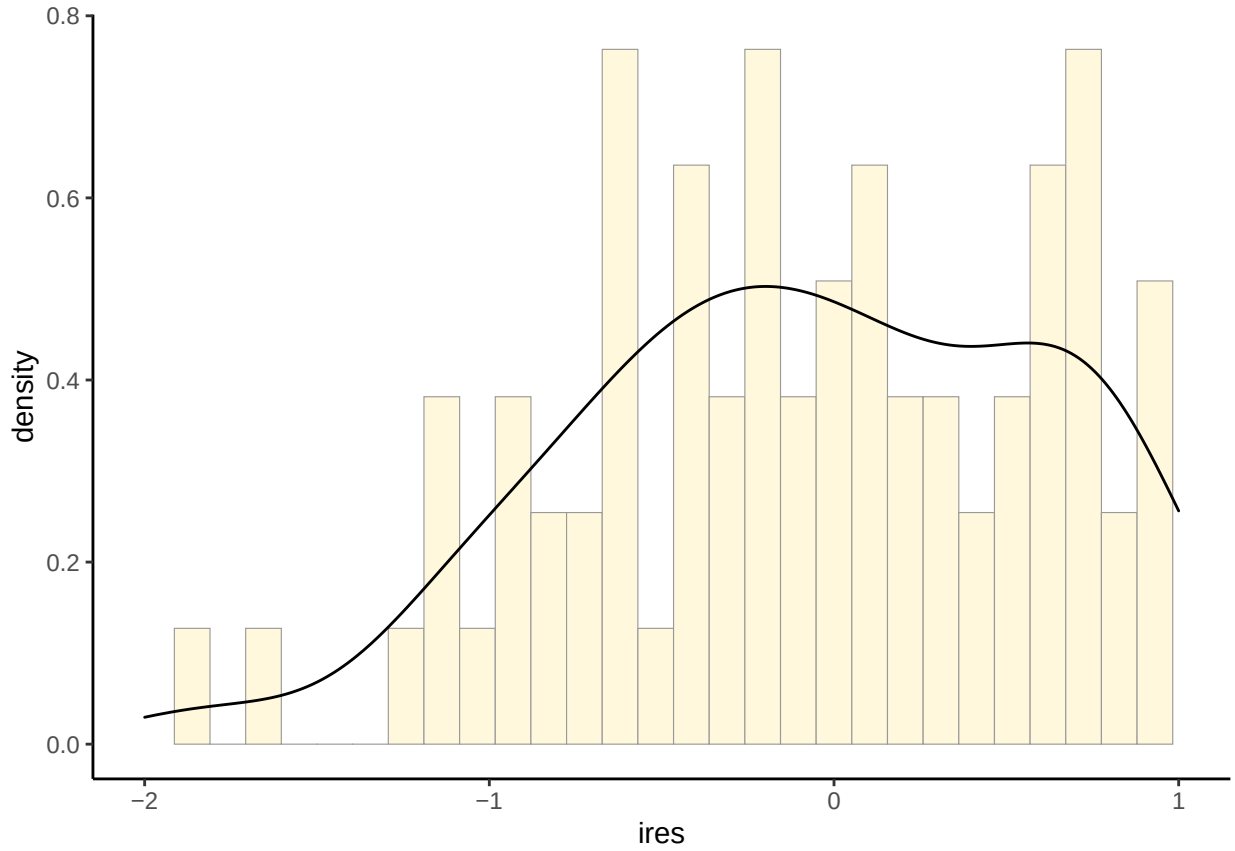
There are no missing data.

5.3 Results

```
x <- Real.2[,-c(1)] %>% apply(2, as.numeric)
y <- Real.2$y
cvfit <- cv.glmnet(x,y)
plot(cvfit)
```




```
beta <- coef(cvfit,s="lambda.min")
beta <- beta[2:length(beta)]
yhat <- x%*%beta
res = y - x %*% beta
res = res %>% as.data.frame
colnames(res) = "ires"
ggplot(res, aes(x = ires, y = ..density..)) +
  geom_histogram(fill = "cornsilk", colour = "grey60", size = 0.2)+ geom_density() + xlim(-2, 1) +
  theme_classic()
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

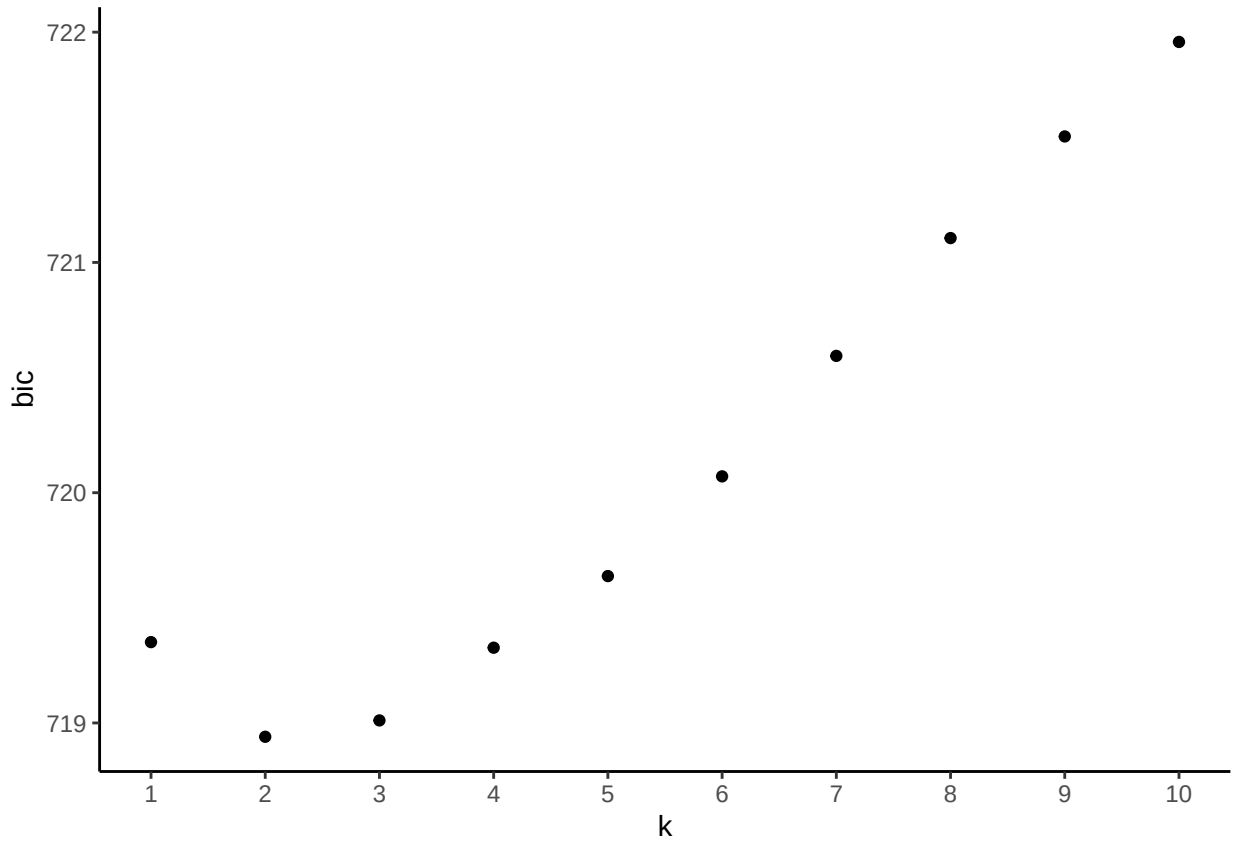


We can find two modes in the density plot. Thus, we apply kmeans to the residual.

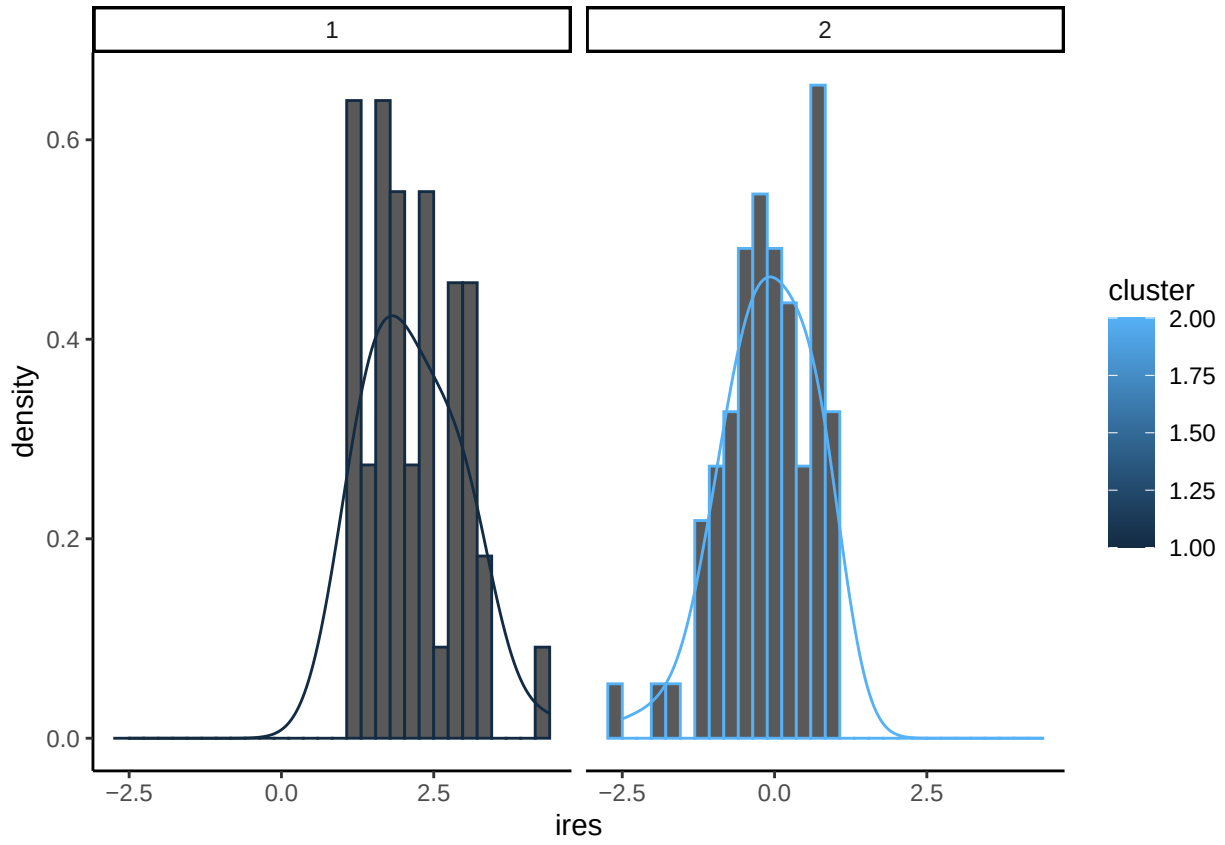
```
computeBIC <- function(X, y, muh, betah, k, c = 10) {

  s <- sum(abs(betah)>0)
  n <- dim(X)[1]
  p <- dim(X)[2]
  Qn <- sum((y-muh-X%*%betah)^2)/n
  df <- k+p;
  bic <- log(Qn) + c*log(log(n+p))*log(n)/n*df;
  return(bic)
}
n <- dim(x)[1]
p <- dim(x)[2]
kL <- 1
kU <- n-1
karr <- seq(kL,kU)
bicarr <- -1*rep(1,length(karr))
for ( i in 1:length(karr)) {
 k <- karr[i]
r <- Ckmeans.1d.dp(res[[1]],k)
muh <- r$centers[r$cluster]
bicarr[i] <- computeBIC(x,y,muh,beta,i)
}
df <- data.frame(k = karr[1:10],
                 bic = bicarr[1:10])
df %>% ggplot(aes(x = k, y = bic))+
```

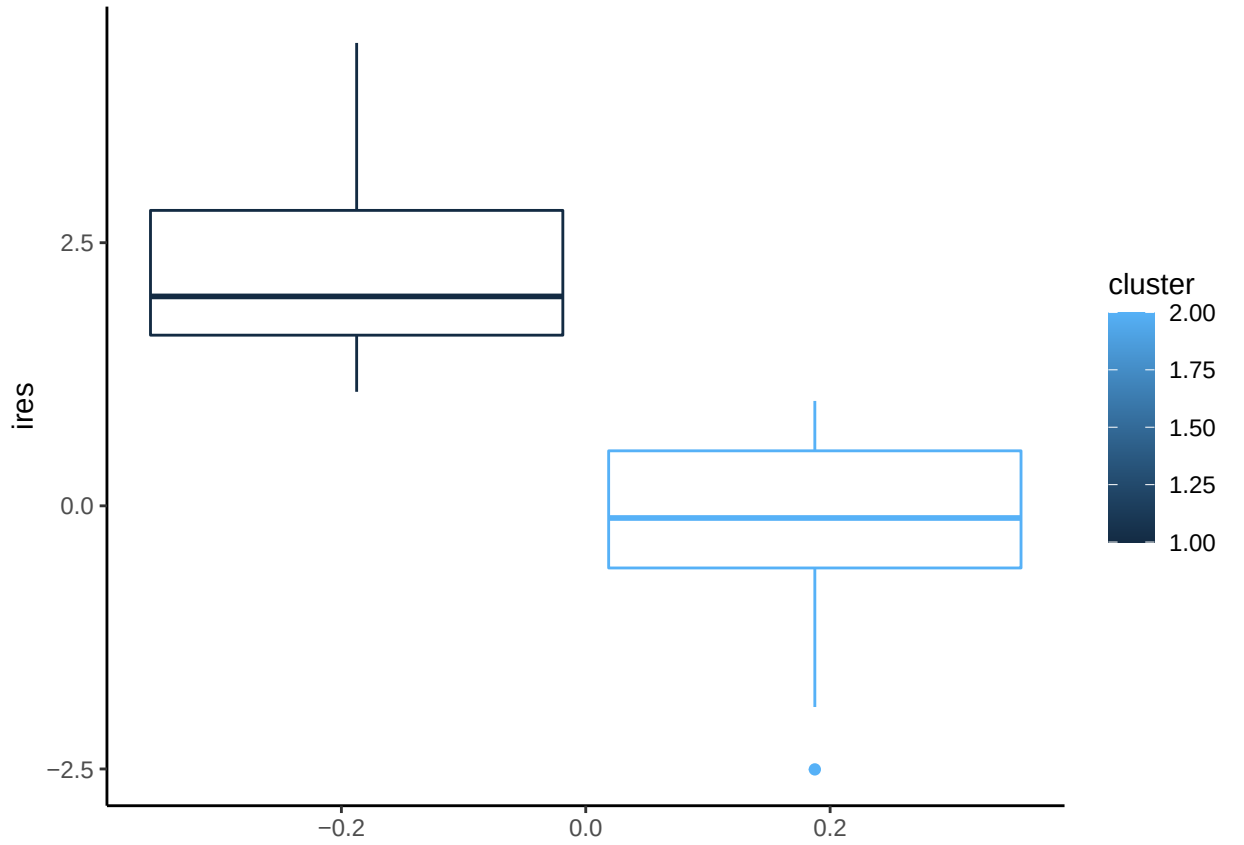
```
  geom_point() +
  scale_x_continuous(breaks= karr[1:10]) +
  theme_classic()
```




```
cl <- kmeans(res,2)
res$cluster <- cl$cluster
res1 <- res %>% as.data.frame() %>% filter(cl$cluster==1)
res2 <- res %>% as.data.frame() %>% filter(cl$cluster==2)
ggplot(res, aes(x = ires, y= ..density.., color = cluster, group = cluster))+
    geom_histogram()+
    geom_density(adjust=1.5) +
    facet_wrap(~cluster) +
    theme(
      legend.position="none",
      panel.spacing = unit(0.1, "lines"),
      axis.ticks.x=element_blank()
    )+
  theme_classic()
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(res, aes(x = ires, color = cluster, group = cluster))+
    geom_boxplot() +
  coord_flip()+
  theme_classic()
```



At this time, the distribution within each subgoup is more homogeneous.

```
t.test(res1$ires,res2$ires)
#>
#>  Welch Two Sample t-test
#>
#> data:  res1$ires and res2$ires
#> t = 16.184, df = 90.757, p-value < 2.2e-16
#> alternative hypothesis: true difference in means is not equal to 0
#> 95 percent confidence interval:
#>  1.988480 2.544924
#> sample estimates:
#>  mean of x  mean of y
#>  2.1612937 -0.1054083
```

T test and boxplot shows that the subgroup makes sense. External data also verifies our results, where 59 patients experienced a relapse (Lee et al., 2015).

5.4 R Shiny

We deploy our simulations with R Shiny in https://chency1997.shinyapps.io/shiny/.

References

Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. Np-hardness of euclidean sum-of-squares clustering. *Machine Learning*, 75:245–248, 2009.

Francis Bach, Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, et al. Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*, 4(1):1–106, 2012.

Patrick L. Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.

Ioannis Dassios, Kimon Fountoulakis, and Jacek Gondzio. A preconditioner for a primal-dual newton conjugate gradient method for compressed sensing problems. *SIAM Journal on Scientific Computing*, 37 (6):A2783–A2812, 2015.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001. doi: 10.1198/016214501753382273.

Qibin Fan, Yuling Jiao, and Xiliang Lu. A primal dual active set algorithm with continuation for compressed sensing. *IEEE Transactions on Signal Processing*, 62(23):6276–6285, 2014.

Kimon Fountoulakis, Jacek Gondzio, and Pavel Zhlobich. Matrix-free interior point method for compressed sensing problems. *Mathematical Programming Computation*, 6(1):1–31, 2014.

Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC monographs on statistics & applied probability. Chapman & Hall/CRC, Boca Raton, 1st edition, 2015. ISBN 9781498712163.

Jian Huang, Yuling Jiao, Xiliang Lu, and Liping Zhu. Robust decoding from 1-bit compressive sampling with ordinary and regularized least squares. *SIAM Journal on Scientific Computing*, 40(4):A2062–A2086, 2018.

Woojoo Lee, Andrey Alexeyenko, Maria Pernemalm, Justine Guégan, Philippe Dessen, Vladimir Lazar, Janne Lehtiö, and Yudi Pawitan. Identifying and assessing interesting subgroups in a heterogeneous population. *BioMed Research International*, 2015, 2015.

S. P. Llyod. Least squares quantization in pcm. 1982.

Shujie Ma and Jian Huang. A concave pairwise fusion approach to subgroup analysis. *Journal of the American Statistical Association*, 112:410–423, 2015.

Jan Orenstein. Revolution in lung cancer: new challenges for the surgical pathologist. *Archives of pathology & laboratory medicine*, 136 2:138, 2012.

Robert Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. ISSN 00359246.

Joel A. Tropp and Stephen J. Wright. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958, 2010.

Masahiro Tsuboi, Tatsuo Ohira, Hisashi Saji, Kuniharu Miyajima, Naohiro Kajiwara, Osamu Uchida, Jitsuo Usuda, and Harubumi Kato. The present status of postoperative adjuvant chemotherapy for completely resected non-small cell lung cancer. *Annals of thoracic and cardiovascular surgery : official journal of the Association of Thoracic and Cardiovascular Surgeons of Asia*, 13 2:73–7, 2007.

Haizhou Wang and Mingzhou Song. Ckmeans.1d.dp: Optimal k-means clustering in one dimension by dynamic programming. *The R journal*, 3 2:29–33, 2011.

Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942, 2010. ISSN 2168-8966. URL www.jstor.org/stable/25662264.