Regularized regression models
(Part 2)

1 LASSO (least absolute shrinkage and selection operator)[1]

- A penalty approach: the lasso estimate, denoted by $\hat{\beta}_{lasso}(\lambda)$, is the solution to minimization

$$\sum_{i=1}^{n}\{Y_i - \beta_0 - \sum_{j=1}^{p}\mathbf{x}_{ij}\beta_j\}^2 + \lambda\sum_{j=1}^{p}|\beta_j| \tag{1.1}$$

please compare it with ridge regression.

[1] A lasso (from Spanish la reata) is a loop of rope designed as a restraint to be thrown around a target and tightened when pulled. see Picture

We can write

$$||\beta||_1 = \sum_{j=1}^{p} |\beta_j|$$

Thus the penalty is called L_1 penalty

- A constraints approach: We can also write the lasso problem in an equivalent form

$$\sum_{i=1}^{n} \{Y_i - \beta_0 - \sum_{j=1}^{p} \mathbf{x}_{ij}\beta_j\}^2, \quad s.t. \quad \sum_{j=1}^{p} |\beta_j| \leq t \tag{1.2}$$

Denoted by $\hat{\beta}_{lasso}[t]$

- The two approaches are also equivalent:

$$\hat{\beta}_{lasso}[||\hat{\beta}_{lasso}(\lambda)||_1] = \hat{\beta}_{lasso}(\lambda)$$

Important facts:

- the L_2 ridge penalty $\sum_{j=1}^{p} \beta_j^2$ is replaced by the L_1 lasso penalty $\sum_{j=1}^{p} |\beta_j|$.
- Thus, it is sometimes called L_1-penalty method.
- The lasso is a shrinkage method like ridge, with subtle but important differences.
- Because of the nature of the constraint, letting t sufficiently small (or λ sufficiently large) will cause some of the coefficients to be exactly zero.
- Because there are zeros values in the estimated β, the actual assumption behind the method is the "sparsity". This will help us for variable (model) selection.

We can understand the sparsity by one example. Consider

$$\text{RIDGE}: \quad \sum_{i=1}^{n}\{Y_i - \beta_1 \mathbf{x}_{i1} - \beta_2 \mathbf{x}_{i1}\}^2, \quad s.t. \quad \sum_{j=1}^{2} \beta_j^2 \leq t$$

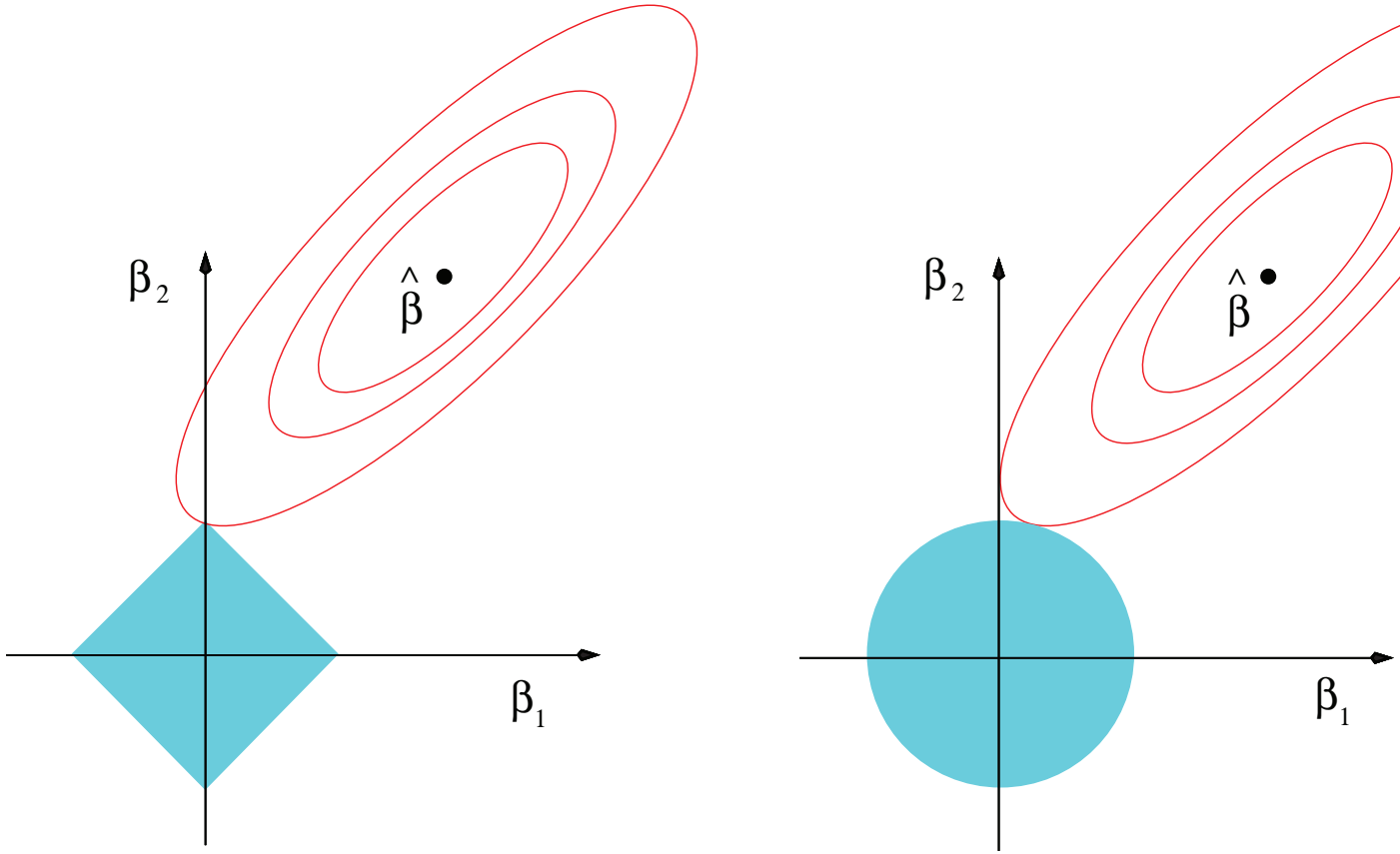
and

$$\text{LASSO}: \quad \sum_{i=1}^{n}\{Y_i - \beta_1 \mathbf{x}_{i1} - \beta_2 \mathbf{x}_{i1}\}^2, \quad s.t. \quad \sum_{j=1}^{2} |\beta_j| \leq t$$

The solution can be exact zero as shown in the figure below: the lasso (left) and ridge regression (right). The solid blue areas are the constraint regions $|\beta_1| + |\beta_2| \leq t$ and $\beta_1^2 + \beta_2^2 \leq t$, respectively, while the red ellipses are the contours of the least squares error function, i.e.

$$\{(\beta_1, \beta_2) : \sum_{i=1}^{n}\{Y_i - \beta_1 \mathbf{x}_{i1} - \beta_2 \mathbf{x}_{i1}\}^2 = SSE + \Delta\}$$

where $SSE = \min \sum_{i=1}^{n}\{Y_i - \beta_1 \mathbf{x}_{i1} - \beta_2 \mathbf{x}_{i1}\}^2$.


β2
β̂
β1
β2
β̂
β1

- Thus the lasso does a kind of continuous subset selection.
  - If t is chosen larger than $t_0 = \sum_{j=1}^{p} |\hat{\beta}_j|$ (where $\hat{\beta}_j$, the least squares estimates), then the lasso estimates are the $\hat{\beta}_j$.
  - if $t = 0$, then no variable is selected.
  - Let $s = t/t_0$, then when $s = 0$, no variable is selected, when $s = 1$ all are selected and the estimator is the LSE.
- on the other hand, it also inherits the properties of ridge regression and sometimes have more efficient estimator

Another understanding about the sparsity for the penalty approach

Consider a special case

$$Y_i = \beta x_i + \varepsilon_i, i = 1, ..., n$$

and for simplicity

$$\sum_{i=1}^{n} x_i = 0, \quad \sum_{i=1}^{n} x_i^2 = 1.$$

Then the LSE is

$$\hat{\beta}_{LSE} = \sum_{i=1}^{n} x_i y_i$$

Consider the Ridge regression, it minimizes

$$\sum_{i=1}^{n} \{Y_i - \beta x_i\}^2 + \lambda \beta^2$$

The solution, i.e. ridge regression estimator, is

$$\hat{\beta}_{ridge} = \hat{\beta}_{LSE}/(1+\lambda).$$

Now consider the LASSO estimator minimizes

$$\sum_{i=1}^{n}\{Y_i - \beta x_i\}^2 + \lambda|\beta|$$

or

$$\sum_{i=1}^{n}\{Y_i - \beta x_i\}^2 + 2\tilde{\lambda}|\beta|$$

where $\tilde{\lambda} = \lambda/2 \geq 0$. i.e.

$$\sum_{i=1}^{n} Y_i^2 + \beta^2 - 2\beta\sum_{i=1}^{n} x_i y_i + 2\tilde{\lambda}|\beta|$$

Because of $\hat{\beta}_{LSE} = \sum_{i=1}^{n} x_i y_i$, the above quantity becomes

$$\sum_{i=1}^{n} Y_i^2 + \beta^2 - 2\beta\hat{\beta}_{LSE} + 2\tilde{\lambda}|\beta|$$

Consider the function

$$g(\beta) = \beta^2 - 2\beta\hat{\beta}_{LSE} + 2\tilde{\lambda}|\beta|$$

It is easy to see that the lasso minimizes $g(\beta)$. we have

$$g'(\beta) = \begin{cases} 2\{\beta - (\hat{\beta}_{LSE} - \tilde{\lambda})\}, & \text{if } \beta > 0 \\ 0, & \text{if } \beta = 0, \\ 2\{\beta - (\hat{\beta}_{LSE} + \tilde{\lambda})\}, & \text{if } \beta < 0 \end{cases}$$

- If $\hat{\beta}_{LSE} - \tilde{\lambda} > 0$ i.e. $\hat{\beta}_{LSE} > \tilde{\lambda}$ (thus $\hat{\beta}_{LSE} + \tilde{\lambda} > 0$ too), then

$$\beta = \hat{\beta}_{LSE} - \tilde{\lambda}$$

is a solution to $g'(\beta) = 0$ for $\beta > 0$ region. Notice that $\beta = (\hat{\beta}_{LSE} + \tilde{\lambda})$, for

region $\beta < 0$, is not a solution. Thus

$$\hat{\beta}_{lasso} = \hat{\beta}_{LSE} - \tilde{\lambda}.$$

- If $\hat{\beta}_{LSE} + \tilde{\lambda} < 0$ i.e. $\hat{\beta}_{LSE} < -\tilde{\lambda}$, similarly,

$$\hat{\beta}_{lasso} = \hat{\beta}_{LSE} + \tilde{\lambda}.$$

- If $-\tilde{\lambda} \leq \hat{\beta}_{LSE} \leq \tilde{\lambda}$, then $(\hat{\beta}_{LSE} - \tilde{\lambda}) \leq 0$ and $(\hat{\beta}_{LSE} + \tilde{\lambda}) \geq 0$. Thus $g'(\beta) < 0$ for $\beta < 0$ and $g'(\beta) > 0$ for $\beta > 0$. It follows that $g(\beta)$ achieves its minimum at $\beta = 0$, i.e.

$$\hat{\beta}_{lasso} = 0.$$

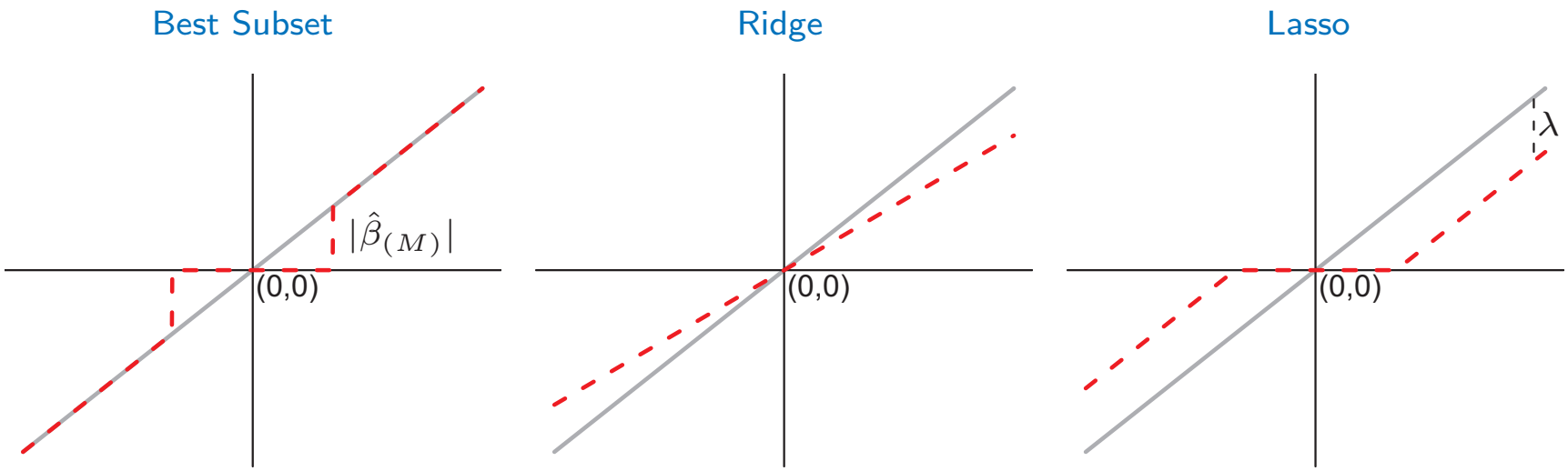
In conclusion, we can write the estimator as

$$\hat{\beta}_{lasso} = \begin{cases} \hat{\beta}_{LSE} - \tilde{\lambda}, & \text{if } \hat{\beta}_{LSE} \geq \tilde{\lambda} \\ 0, & \text{if } -\tilde{\lambda} \leq \hat{\beta}_{LSE} < \tilde{\lambda} \\ \hat{\beta}_{LSE} + \tilde{\lambda}, & \text{if } \hat{\beta}_{LSE} \leq -\tilde{\lambda} \end{cases}$$

Estimator	formula
variable selection (based on AIC, T-test, ...)	$\hat{\beta}_{LSE} I(\lvert\hat{\beta}_{LSE}\rvert > \beta_{(M)})$
ridge	$\hat{\beta}_{LSE}/(1+\lambda)$
Lasso	$\hat{\beta}_{LSE} - sign(\hat{\beta}_{LSE})\tilde{\lambda}$

Model (variable) selection and model estimation can be done simultaneously!

The estimators (Ridge, Lasso, and variable selection, in RED) and the LS estiamtor (in BLACK) are shown below



Calculation Issues[2] Assume X is standardized and y is centered

- quadratic approximation. Suppose we have old values of β, denote by β^{old}.

  The new value is β^{New} solution to

$$\sum_{i=1}^{n}\{Y_i - \sum_{j=1}^{p}\mathbf{x}_{ij}\beta_j\}^2 + \lambda\sum_{j=1}^{p}\beta_j^2/\beta_j^{old} \tag{1.3}$$

- LARS (least angle regression). Choose a small ϵ.

  1. Start with initial residual $r = y$, and $\beta_1 = \beta_2 = ... = \beta_p = 0$.

  2. Find the predictor $\mathbf{x}_j (j = 1, ..., p)$ most correlated with r

  3. Update $\beta_j \leftarrow \beta_j + \delta_j$, where $\delta_j = \epsilon \times sign(\mathbf{x}_j^\top r)$.

  4. Set $r \leftarrow r - \delta_j \mathbf{x}_j$, and repeat Steps 2 and 3 many times.

[2]Please skip this section

2 Selection of tuning parameter λ

Usually, 5-fold or 10-fold CV is used.

Question: Please state the procedure of selecting the tuning parameter based on 5-fold CV.

R package: cv.glmnet(...), glmnet(....), plot(...), predict(...)

mycv[3] = cv.glmnet(x, y, lambda, type.measure, nfolds)

- INPUT
  - x: matrix of design of n×p
  - y: response y as in glmnet.
  - lambda=: Optional user-supplied lambda sequence; default is NULL, and glmnet chooses its own sequence
  - nfolds=: number of folds - default is 10, nfolds=n is the delete-one-observation CV.

[3] any name

- OUTPUT
  - mycv$cvm: the mean cross-validated error - a vector of length(lambda).
  - mycv$cvsd: estimate of standard error of cvm.
  - mycv$cvup: upper curve = cvm+cvsd.
  - mycv$cvlo: lower curve = cvm-cvsd.
  - mycv$nzero: number of non-zero coefficients at each lambda.
  - mycv$lambda.min: value of lambda that gives minimum cvm.
  - mycv$lambda.1se: vlargest value of lambda such that error is within 1 standard error of the minimum.
- plot(mycv): to plot the CV values against lambda

```
mylasso = glmnet(x, y, family=c("gaussian","binomial","poisson",
                                "multinomial","cox","mgaussian"),
        alpha = c(0, ...., 1), nlambda = 100, lambda=NULL,
        standardize = TRUE, intercept=TRUE,
        standardize.response=FALSE)
```

- INPUT
  - x: input matrix, of dimension nobs x nvars; each row is an observation vector.
  - y: response variable.
  - family: Response type. Different choice corresponds to different general-

ized linear regression model. Default will be family="gaussian", i.e. linear regression model

- alpha:The elasticnet mixing parameter, with $0 \leq \alpha \leq 1$. The penalty is defined as

$$(1-\alpha)/2||\beta||_2^2 + \alpha||\beta||_1.$$

  alpha=1 is the lasso penalty, and alpha=0 the ridge penalty.

- nlambda: the number of lambda values - default is 100.

- lambda: a user supplied lambda sequence.

- standardize=TRUE/FALSE Logical flag for X variable standardization, prior to fitting the model sequence. The coefficients are always returned on the original scale. Default is standardize=TRUE.

- intercept: Should intercept(s) be fitted (default=TRUE) or set to zero (FALSE)
- dfmax: Limit the maximum number of variables in the model. Useful for very large p, if a partial path is desired.

- OUTPUT
  - mylasso$a0: Intercept sequence of length length(lambda)
  - mylasso$beta: all the estimated β with different λs.
  - mylasso$lambda: The actual sequence of lambda values used
  - mylasso$df: The number of nonzero coefficients for each value of lambda.

- plot(mylasso): to plot the path of the estimated coefficients against t

mypred = predict(mylasso, newx, type=c("link","response",

"coefficients","nonzero","class"))

OUTPUT

- mylasso: or any other name fitted "glmnet" model object.
- newx: Matrix of new values for x at which predictions are to be made.
- type: Type "link" gives the linear predictors for "binomial", "multinomial" models; for "gaussian" models it gives the fitted values. Type "response" gives the fitted probabilities for "binomial" or "multinomial"; for "binomial" models, results are returned only for the class corresponding to the second level. Type "class" applies only to "binomial" or "multinomial" models.

Example 2.1 For the diabetes data with 10 variables $\mathbf{x}_1, ..., \mathbf{x}_{10}$ (see **(diabetes.csv)**).

$$y_i = \beta_1 \mathbf{x}_{i,1} + ... + \beta_{10} \mathbf{x}_{i,10} + \varepsilon_i, \quad i = 1, ..., 442$$

After standardizing the predictors,

$$\mathbf{x}_{i,1} = (\mathbf{x}_{i,1} - 48.518100)/13.1090278; \quad ... \quad \mathbf{x}_{i,10} = (\mathbf{x}_{i,10} - 91.260181)/11.4963347$$

we use the lasso

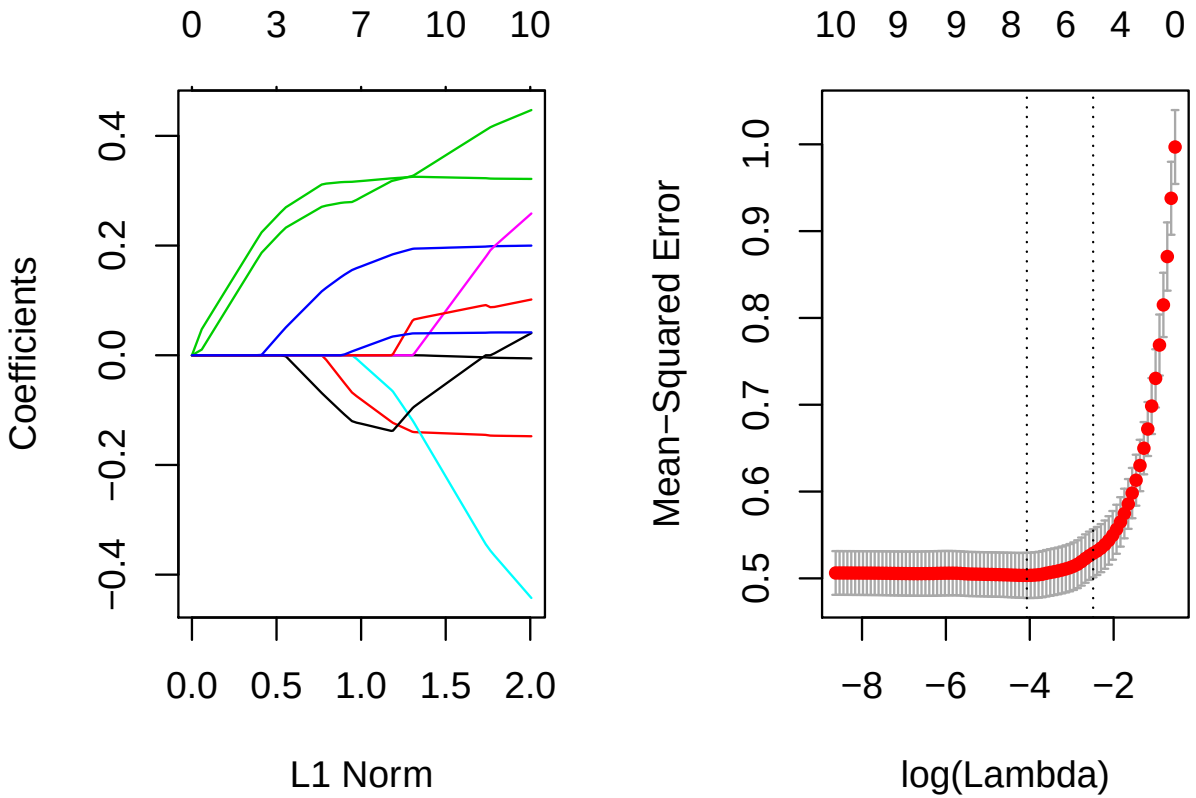
$$\sum_{i=1}^{n} [y_i - (\beta_1 \mathbf{x}_{i,1} + ... + \beta_{10} \mathbf{x}_{i,10})]^2 + \lambda \sum_{j=1}^{10} |\beta_j|.$$

By 10-fold CV, we select $\lambda = 0.01417678$. The estimated model is

$$\hat{y} = 0\mathbf{x}_1 - 0.1190\mathbf{x}_2 + 0.3221\mathbf{x}_3 + 0.1819\mathbf{x}_4 - 0.0602\mathbf{x}_5$$

$$+0\mathbf{x}_6 - 0.1373\mathbf{x}_7 + 0\mathbf{x}_8 + 0.3155\mathbf{x}_9 + 0.0322\mathbf{x}_{10}$$

We have the following coefficients paths ((code))



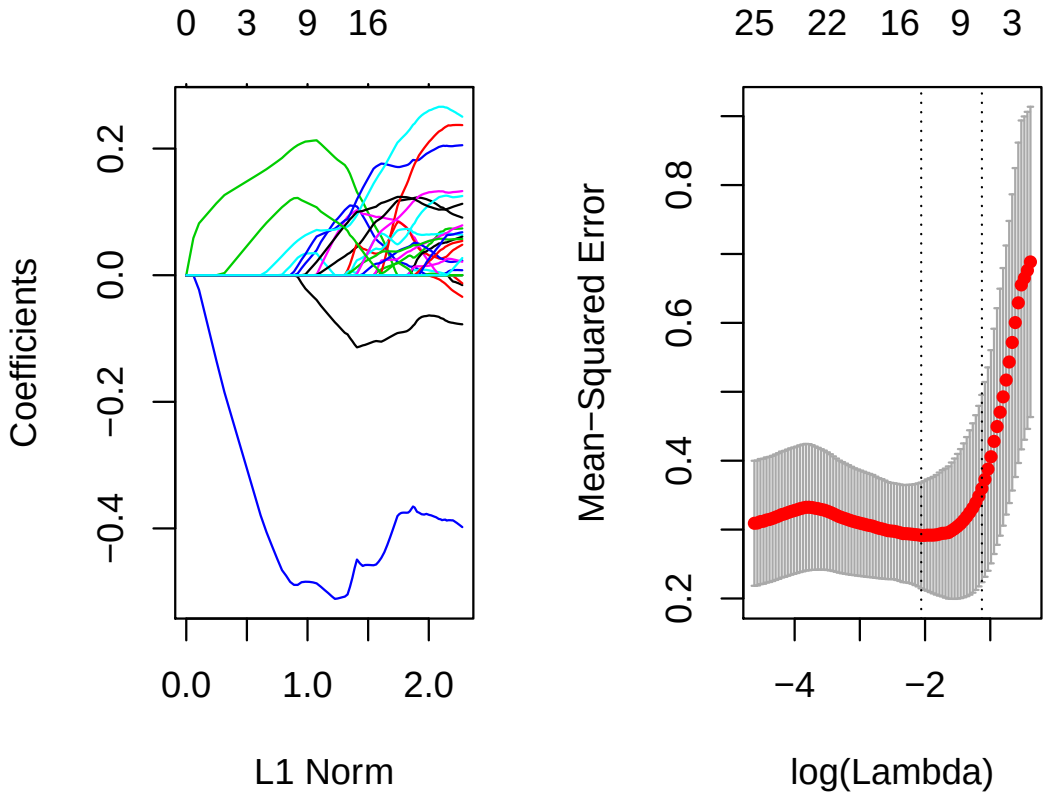
Example 2.2 For the leukemia gene expression data ((**(training data)**. There are n=25 cells with p=250 genes. they are from two types of cells.

To check the models, a new experiment was done and the data were collected ((**(testing set)**.

The prediction errors for the new data set based on LASSO regression: 0.1392248. (with tuning parameter $\lambda = 0.1276111$ by 5-fold cross-validation method).

From the last figure below, we can see that we can have a very accurate classification for the new data.

The path of selection is as follows.

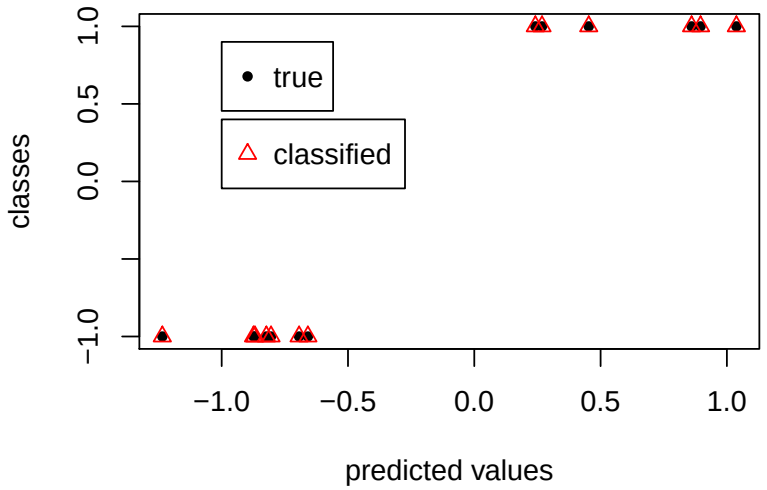


R code for the calculation **(code)**

The estimated model is

$$
\begin{aligned}
Y \;=\; & 0.1491\mathbf{x}_{31} + 0.1126\mathbf{x}_{48} + 0.0976\mathbf{x}_{66} + 0.03606\mathbf{x}_{69} + 0.03305\mathbf{x}_{73} \\
& +0.1099\mathbf{x}_{86} + 0.0091\mathbf{x}_{96} + 0.0020\mathbf{x}_{157} - 0.1100\mathbf{x}_{161} + 0.0110\mathbf{x}_{174} \\
& -0.4642\mathbf{x}_{176} + 0.0858\mathbf{x}_{183} + 0.0940\mathbf{x}_{208} + 0.05354\mathbf{x}_{217}
\end{aligned}
$$

The classification for the 13 new cells are shown below



3 A combination of LASSO and Ridge regression

The ridge regression estimator β_{ridge} is the solution to

$$\min_{\beta} \left\{ \sum_{i=1}^{n} (Y_i - \beta_0 - \beta_1 \mathbf{x}_{i1} - ... - \beta_p \mathbf{x}_{ip})^2 + \lambda ||\beta||_2^2 \right.$$

and β_{lasso} to

$$\min_{\beta} \left\{ \sum_{i=1}^{n} (Y_i - \beta_0 - \beta_1 \mathbf{x}_{i1} - ... - \beta_p \mathbf{x}_{ip})^2 + \lambda ||\beta||_1 \right.$$

A combination of them is the elastic net estimator, denoted by β_{net}, to

$$\min_{\beta} \left\{ \sum_{i=1}^{n} (Y_i - \beta_0 - \beta_1 \mathbf{x}_{i1} - ... - \beta_p \mathbf{x}_{ip})^2 + \lambda \{ 0.5(1-\alpha) ||\beta||_2^2 + \alpha ||\beta||_1 \} \right.$$

where $0 \leq \alpha \leq 1$; if $\alpha = 1$ it is lasso, and $\alpha = 0$ ridge.

R: glmnet(x, y, ..., α = ??, ...)