R Package *fairselect* for Features Annealed Independence Rules

Ying Lu

School of Statistics, University of Minnesota

Work by Jianqing Fan and Yingying Fan

Department of Operational Research and Financial Engineering

Princeton University

July 15, 2011

1 Introduction

This R package is about implementing the Features Annealed Independence Rules proposed by Fan, J. and Fan, Y. (2008) on high dimensional classification. The problem is as follows:

For the p-dimensional classification problem between two classes C_1 and C_2, suppose we have n_k observations $Y_{k1}...Y_{kn_k}$ in R^p in the kth class. The jth feature of the ith sample from class C_k satisfies the model

$$Y_{kij} = \mu_{kij} + \epsilon_{kij}, k = 1, 2, i = 1, 2, ..n_k, j = 1, 2...p$$

In matrix form, it is

$$Y_{ki} = \mu_k + \epsilon_{ki}, k = 1, 2, i = 1, 2, ..n_k, j = 1, 2...p$$

where $\mu_k = (\mu_{k1}...\mu_{kp})'$ is the mean vector from class C_k, and $\epsilon_{ki} = (\epsilon_{ki1}, ..\epsilon_{kip})'$ has distribution $N(0, \Sigma_k)$.

We further denote that $\hat{\mu}_k = \sum_{i=1}^{n_k} Y_{ki}/n_k$, $\hat{\mu} = (\mu_1 + \mu_2)/2$, $\hat{D} = diag\{(S_{1j}^2 + S_{2j}^2)/2\}$. where S_{kj}^2 is the sample variance of the jth feature in class k. so the classification rule is

$$\hat{\delta}(x) = (x - \hat{\mu})'\hat{D}^{-1}(\hat{\mu}_1 - \hat{\mu}_2)$$

If $\hat{\delta}(x) > 0$, we classify the observation in class 1, otherwise we put it in class 2.

When the number of covariates p is way larger than the sample size n, classification is difficult since it is not easy to know which variables are important and which are unimportant. Because of the high dimension issue, the Fisher discriminant analysis gives poor performance because the p*p variance covariance matrix is huge and estimation of this matrix is nearly impossible. Bickel and Levina (2004) propose the independence rule by using only the diagonal elements of the variance covariance matrix to construct the Fisher's rule. Fan, J. and Fan, Y. (2008) show that this is still needs to be improved because the classification effect is not satisfactory. To enhance the classification power, Fan, J. and Fan, Y (2008) propose the Features Annealed Independence Rules. In their opinion, the dissatisfactory performance of the independence rule is due to the inclusion of unimportant predictors(features). As a matter of fact, these unimportant features accumulate the errors and add to much noise. To extract important features in the first step is the key to mitigate this problem. They proposed the Features Annealed Independence Rules. Two methods are used and named as "t test" and "oracle".

$$T_j = \frac{\hat{Y}_{1j} - \hat{Y}_{2j}}{\sqrt{S_{1j}^2/n_1 + S_{2j}^2/n_2}}$$

$$O_j = \hat{Y}_{1j} - \hat{Y}_{2j}$$

where

$$\hat{Y}_{kj} = \sum_{i=1}^{n_k} y_{kij}/n_k$$

The first method is about ranking features according to the absolute value of the sample t statistic for each predictor in the training sample. The second method is about ranking features according to the absolute value of the sample mean difference for each predictors in the training sample. These two methods to some extent reflect whether the predictors are important or not in the classification process. Fan and Fan propose that firstly, we need to select important features, and secondly, we only need to construct the independence rules based on the selected features.

2 Theorems

Key Assumptions for the model are: All observations are independent across samples and within each class C_k; Observations $Y_{k1}...Y_{kn_k}$ are identically distributed; $c_1 \leq \frac{n_1}{n_2} \leq c_2$, where c_1 and c_2 are positive constants and lastly, $\Sigma_1 = \Sigma_2 = \Sigma$.

Theorem 2.1. *Suppose a is a p-dimensional uniformly distributed random vector on a $(p-1)$ dimensional sphere. Let $\lambda_1..\lambda_p$ be the eigenvalue of the covariance matrix Σ. Suppose $\lim_p \frac{1}{p^2} \sum_{j=1}^{p} \lambda_j^2 < \infty$, and $\lim_p \frac{1}{p} \sum_{j=1}^{p} \lambda_j = \tau$ with τ a constant, moreover assume that $p^{-1}\alpha'\alpha \to 0$, then if we define*

$$\hat{\delta}_a(x) = (a'x - a'\hat{\mu})(a'\hat{\mu}_1 - a'\hat{\mu}_2) \tag{1}$$

The misclassification error $P(\hat{\delta}_a(X) \leq 0 | Y_{ki}, i = 1, ..n_k, k = 1, 2) \to \dfrac{1}{2}$ in probability.

Theorem 2.2. *Let s be a sequence such that $\log(p - s) = o(n^\gamma)$, and $\log(s) = o(n^{1/2-\gamma}\beta_n)$ for some $\beta_n \to 0$ and $0 < \gamma < \frac{1}{3}$. Suppose that $\min_{1 \leq j \leq s} \frac{|a_j|}{\sqrt{\sigma_{1j}^2 + \sigma_{2j}^2}} = n^{-\gamma}\beta_n$. Then under some conditions, for $x \sim cn^{\gamma/2}$ with c some positive constant, we have*

$$P(\min_{j \leq s} |T_j| \geq x \quad and \quad \max_{j > s} |T_j| < x) \to 1$$

3 R Functions

```
>fairselect(training, testing, method)
```

This function does feature selection in the binary classification on the high dimensional data using FAIR. See reference for more details. Arguments are shown below:

training: training dataset should be in the form of matrix. It is used to build models.

testing: testing dataset should be in the form of matrix. It is used to check model accuracy.

method: method has two options: ”ttest” or ”oracle”. They are different criteria for feature selection on high dimensional data.

The function will return several values:

value: the minimum misclassification error by employing the model onto the testing data.

feature: the optimal number of important features to select.

m_1: the sample mean of the first class in the training data.

m_2: the sample mean of the second class in the training data.

cova: variance covariance matrix of the training sample data.

```
>classify(newdata,fairobject)
```

This function does binary classification on the new dataset based on the model.

newdata: newdata set should be in the form of numeric.

fairobject: The *fair* object with which we use to construct the classifier.

4 Examples

Here is an artificial example to illustrate how the functions work.

```
>x=matrix(rnorm(30*100),nrow=30)

>x[,1]=rbinom(30,1,prob=0.5)

>y=matrix(rnorm(30*100,0,1),nrow=30)

>y[,1]=rbinom(30,1,prob=0.5)

>training=x

>testing=y

>newdata=rnorm(99)

>a=fairselect(x,y,"ttest")  # returns the method we use, features selected,
sample mean in the training samples for two classes, and sample
variance covariance of the features.
>b=fairselect(x,y,"oracle")
>classify(newdata,a) #returns the value of 0 or 1
```

References

[1] Fan, J. and Fan, Y. (2008). High-Dimensional Classification Using Features Annealed Independence Rules. *The Annals of Statistics*, **36(6)**, 2008.

[2] Bickel,P.J. and Levina,E.(2004). Some theory for Fisher's linear discriminant function, "naive Bayes," and some alternatives when here are many more variables than observations. *Bernolli*, **10**, 989-1010, 2004.