Cancer Classification Using Mass Spectrometry-based Proteomics Data

Zhu Wang
UT Health San Antonio
wangz1@uthscsa.edu

This document presents data analysis similar to Wang (2011) using R package **bst**. Serum samples were collected from 77 benign prostate hyperplasia (BPH), 84 early stage prostate cancer, 83 late stage prostate cancer and 82 age-matched healthy men (HM). Peak detection and alignment were based on surface-enhanced laser desorption/ionization (SELDI) mass spectrometry protein profiles. The data have 779 peaks (predictors) for each sample.

```
R> library("bst")
R> library("pROC")
R> ### benign prostate hyperplasia
R> bph <- read.table(file=system.file("extdata", "BPH.txt", package="bst"))
R> ### early stage prostate cancer,
R> ccd <- read.table(file=system.file("extdata", "CCD.txt", package="bst"))
R> ### late stage prostate cancer
R> cab <- read.table(file=system.file("extdata", "CAB.txt", package="bst"))
R> ### age-matched healthy men
R> hm <- read.table(file=system.file("extdata", "control.txt", package="bst"))
R> ### mass spectrometry protein
R> mz <- read.table(file=system.file("extdata", "prostate_mz.txt", package="bst"))
```

1 Cancer vs Non-cancer

Cancer includes early and late stage cancer while non-cancer includes healthy men and benign cancer.

```
R> dat <- t(cbind(hm, bph, cab, ccd))
R> y <- c(rep(-1, dim(hm)[2] + dim(bph)[2]), rep(1,  dim(cab)[2] + dim(ccd)[2]))
```

Peaks with variations only in a very small number of samples are less likely to be linked with cancer stages. Thus, we filter out the peaks whose total number of fixed value exceed 95% across samples.

```
R> myuniq <- function(x){
   if(sum(x == 0) <= length(x) * 0.95)
   return(TRUE)
   else return(FALSE)
 }
```

```
R> res <- apply(dat, 2, myuniq)
R> dat <- dat[,res]
R> rownames(dat) <- NULL
R> colnames(dat) <- mz[res,1]
```

we randomly select 75% of the samples as training data and the remaining samples as the test data.

```
R> ntrain <- floor(length(y)*0.75)
R> set.seed(13)
R> q <- sample(length(y))
R> y.tr <- y[q][1:ntrain]; y.te <- y[q][-(1:ntrain)]
R> X <- dat[q,]
R> x.tr <- X[1:ntrain,];  x.te <- X[-(1:ntrain),]
```

Apply HingeBoost to classify cancer vs non-cancer.

```
R> dat.m1 <- bst(x=x.tr, y=y.tr, ctrl = bst_control(mstop=400), family = "hinge2")
R> pred <- predict(dat.m1, x.te)
R> ### misclassification error
R> mean(abs(y.te-sign(pred))/2)

[1] 0.03659

R> ### area under the curve (AUC) of receiver operating characteristic (ROC)
R> auc(y.te, pred)

Area under the curve: 0.99

R> ### number of variables selected
R> length(dat.m1$xselect)

[1] 41
```

Apply twin HingeBoost to classify cancer vs non-cancer.

```
R> dat.m2 <- bst(x=x.tr, y=y.tr, family="hinge2", ctrl = bst_control(mstop=500,
 twinboost=TRUE, twintype=2, coefir=coef(dat.m1), f.init=predict(dat.m1),
 xselect.init = dat.m1$xselect))
R> pred <- predict(dat.m2, x.te)
R> ### misclassification error
R> mean(abs(y.te-sign(pred))/2)

[1] 0.04878

R> ### AUC with twin boosting
R> auc(y.te, pred)

Area under the curve: 0.989

R> ### number of variables selected
R> length(dat.m2$xselect)

[1] 18
```

2 Cancer vs Healthy Men

```
R> dat <- t(cbind(cab, ccd, hm))
R> y <- c(rep(1, dim(cab)[2] + dim(ccd)[2]), rep(-1, dim(hm)[2]))
R> res <- apply(dat, 2, myuniq)
R> dat <- dat[,res]
R> rownames(dat) <- NULL
R> colnames(dat) <- mz[res,1]
R> ntrain <- floor(length(y)*0.75)
R> set.seed(13)
R> q <- sample(length(y))
R> y.tr <- y[q][1:ntrain]; y.te <- y[q][-(1:ntrain)]
R> X <- dat[q,]
R> x.tr <- X[1:ntrain,];  x.te <- X[-(1:ntrain),]
```

Apply HingeBoost to classify cancer vs healthy men.

```
R> dat.m1 <- bst(x=x.tr, y=y.tr, ctrl = bst_control(mstop=400), family = "hinge2")
R> pred <- predict(dat.m1, x.te)
R> ### misclassification error
R> mean(abs(y.te-sign(pred))/2)

[1] 0.01587

R> ### AUC
R> auc(y.te, pred)

Area under the curve: 0.985

R> ### number of variables selected
R> length(dat.m1$xselect)

[1] 44
```

Apply twin HingeBoost to classify cancer vs healthy men.

```
R> dat.m2 <- bst(x=x.tr, y=y.tr, family="hinge2", ctrl = bst_control(mstop=200,
 twinboost=TRUE, twintype=2, coefir=coef(dat.m1), f.init=predict(dat.m1),
 xselect.init = dat.m1$xselect))
R> pred <- predict(dat.m2, x.te)
R> ### misclassification error
R> mean(abs(y.te-sign(pred))/2)

[1] 0.03175

R> ### AUC with twin boosting
R> auc(y.te, pred)

Area under the curve: 0.996

R> ### number of variables selected
R> length(dat.m2$xselect)

[1] 12
```

3 Cancer vs Benign Cancer

```
R> dat <- t(cbind(bph, cab, ccd))
R> y <- c(rep(-1, dim(bph)[2]), rep(1,  dim(cab)[2] + dim(ccd)[2]))
R> res <- apply(dat, 2, myuniq)
R> dat <- dat[,res]
R> rownames(dat) <- NULL
R> colnames(dat) <- mz[res,1]
R> ntrain <- floor(length(y)*0.75)
R> set.seed(13)
R> q <- sample(length(y))
R> y.tr <- y[q][1:ntrain]; y.te <- y[q][-(1:ntrain)]
R> X <- dat[q,]
R> x.tr <- X[1:ntrain,];  x.te <- X[-(1:ntrain),]
```

Apply HingeBoost to classify cancer vs benign cancer.

```
R> dat.m1 <- bst(x=x.tr, y=y.tr, ctrl = bst_control(mstop=400), family = "hinge2")
R> pred <- predict(dat.m1, x.te)
R> ### misclassification error
R> mean(abs(y.te-sign(pred))/2)

[1] 0

R> ### AUC
R> auc(y.te, pred)

Area under the curve: 1

R> ### number of variables selected
R> length(dat.m1$xselect)

[1] 52
```

Apply twin HingeBoost to classify cancer vs benign cancer.

```
R> dat.m2 <- bst(x=x.tr, y=y.tr, family="hinge2", ctrl = bst_control(mstop=500,
 twinboost=TRUE, twintype=2, coefir=coef(dat.m1), f.init=predict(dat.m1),
 xselect.init = dat.m1$xselect))
R> pred <- predict(dat.m2, x.te)
R> ### misclassification error
R> mean(abs(y.te-sign(pred))/2)

[1] 0.04918

R> ### AUC with twin boosting
R> auc(y.te, pred)

Area under the curve: 1

R> ### number of variables selected
R> length(dat.m2$xselect)

[1] 16
```

4 Benign Cancer vs Healthy Men

```
R> dat <- t(cbind(bph, hm))
R> y <- c(rep(1, dim(bph)[2]), rep(-1, dim(hm)[2]))
R> res <- apply(dat, 2, myuniq)
R> dat <- dat[,res]
R> rownames(dat) <- NULL
R> colnames(dat) <- mz[res,1]
R> ntrain <- floor(length(y)*0.75)
R> set.seed(13)
R> q <- sample(length(y))
R> y.tr <- y[q][1:ntrain]; y.te <- y[q][-(1:ntrain)]
R> X <- dat[q,]
R> x.tr <- X[1:ntrain,];  x.te <- X[-(1:ntrain),]
```

Apply HingeBoost to classify benign cancer vs healthy men.

```
R> dat.m1 <- bst(x=x.tr, y=y.tr, ctrl = bst_control(mstop=400), family = "hinge2")
R> pred <- predict(dat.m1, x.te)
R> ### misclassification error
R> mean(abs(y.te-sign(pred))/2)

[1] 0.025

R> ### AUC
R> auc(y.te, pred)

Area under the curve: 0.952

R> ### number of variables selected
R> length(dat.m1$xselect)

[1] 56
```

Apply twin HingeBoost to classify benign cancer vs healthy men.

```
R> dat.m2 <- bst(x=x.tr, y=y.tr, family="hinge2", ctrl = bst_control(mstop=500,
 twinboost=TRUE, twintype=2, coefir=coef(dat.m1), f.init=predict(dat.m1),
 xselect.init = dat.m1$xselect))
R> pred <- predict(dat.m2, x.te)
R> ### misclassification error
R> mean(abs(y.te-sign(pred))/2)

[1] 0.025

R> ### AUC with twin boosting
R> auc(y.te, pred)

Area under the curve: 0.988

R> ### number of variables selected
R> length(dat.m2$xselect)

[1] 19
```

References

Zhu Wang. HingeBoost: ROC-based boost for classification and variable selection. *The International Journal of Biostatistics*, 7(1):1–30, 2011.