Classification of Cancer Types Using Gene Expression Data

Zhu Wang
Connecticut Children's Medical Center
University of Connecticut School of Medicine
zwang@connecticutchildrens.org

This document presents data analysis similar to Wang (2012) using R package **bst**.

1 Classification of Small Round Blue Cell Tumors

Classifying the small round blue cell tumors (SRBCTs) of childhood into four categories is studied using gene expression profiles http://research.nhgri.nih.gov/microarray/Supplement/. With 2,308 gene profiles in 63 training samples and 20 test samples, perfect classification can be reached. We delete information not used in the analysis and set up the right data format. Take the logarithm base 10 of the gene levels, then standardize the results. We then select top 300 genes based on a marginal relevance measure.

```
R> library("bst")
R> datafile <- system.file("extdata", "supplemental_data", package="bst")
R> dat0 <- read.delim(datafile, header=TRUE, skip=1)[,-(1:2)]
R> genename <- read.delim(datafile, header=TRUE, skip=1)[,(1:2)]
R> dat0 <- t(dat0)
R> dat1 <- dat0[rownames(dat0) %in%
   c("TEST.9", "TEST.13","TEST.5", "TEST.3", "TEST.11"),]
R> dat2 <- dat0[!rownames(dat0) %in%
   c("TEST.9", "TEST.13","TEST.5", "TEST.3", "TEST.11"),]
R> dat2 <- rbind(dat2, dat1)
R> train <- dat2[1:63,] ### training samples
R> test <- dat2[64:83,] ### test samples
R> train.classes <- substr(rownames(train), 1,2)
R> test.classes <- c("NB","RM","NB","EW","RM","BL","EW","RM","EW","EW","EW",
 "RM","BL","RM","NB","NB","NB","NB","BL","EW")
R> train.classes <- as.numeric(factor(train.classes, levels=c("EW", "BL", "NB", "RM")))
R> test.classes <- as.numeric(factor(test.classes, levels=c("EW", "BL", "NB", "RM")))
R> ### pre-processing training data: standardize predictors after log-transformation
R> train <- log10(train)
R> x <- train
```

```
R> meanx <- colMeans(x)
R> one <- rep(1,nrow(x))
R> normx <- sqrt(drop(one %*% (x^2)))
R> train <- scale(train, meanx, normx)
R> ### compute a marginal relevance measure
R> tmp <- cbind(train, train.classes)
R> a0 <- b0 <- 0
R> for(k in 1:length(table(train.classes))){
   tmp1 <- subset(tmp, tmp[,2309]==k)
   xc.bar <- colMeans(tmp1[,-2309])   ###average of gene j across class k
   xa.bar <- colMeans(tmp[,-2309])    ###average of gene j across all samples
   a0 <- a0 + dim(tmp1)[1] * ((xc.bar - xa.bar)^2)
   b0 <- b0 + colSums((tmp[,-2309] - xc.bar)^2)
 }
R> bw <- a0/b0
R> ### select top 300 genes based on the ordered marginal relevance measure
R> npre <- 300
R> bw1 <- order(bw, decreasing=TRUE)[1:npre]
R> train <- train[,bw1]
R> ### pre-processing test data: standardize predictors after log-transformation
R> ### select the same 300 genes as in the training data
R> test <- log10(test)
R> test <- scale(test, meanx, normx)[, bw1]
R> test <- as.data.frame(test)
R> colnames(train) <- paste("x", 1:dim(train)[2], sep="")
R> colnames(test) <- paste("x", 1:dim(test)[2], sep="")
```

Multi-class HingeBoost with smoothing splines as base learner is applied to the data. A 5-fold cross-validation is used for tuning parameter selection.

```
R> m <- 30
R> set.seed(123)
R> dat.cvm <- cv.mhingebst(x=train, y=train.classes, balance=TRUE, K=5,
 ctrl = bst_control(mstop=m), family = "hinge", learner = "sm", type="error", n.cores=2)
```

Multi-class HingeBoost is applied with boosting iteration 20 based on the cross-validation results. Plot the evolution of the misclassification error on the test data versus the iteration counter, as the multi-class HingeBoost algorithm proceeds while working on the test set.

```
R> m1 <- 20
R> dat.m1 <- mhingebst(x=train, y=train.classes, ctrl = bst_control(mstop=m1),
 family = "hinge", learner = "sm")
R> risk.te1 <- predict(dat.m1, newdata=test, newy=test.classes, type="error")
R> plot(risk.te1, type="l", xlab="Iteration", ylab="Test Error")
```

Plot the evolution of the number of genes selected versus the iteration counter, as the multi-class HingeBoost algorithm proceeds while working on the training set.

```
R> plot(nsel(dat.m1, m1), ylab="No. Genes", xlab="Iteration", lty="solid", type="l")
```

Multi-class twin HingeBoost is applied. Plot the evolution of the misclassification error on the test data versus the iteration counter, as the multi-class twin HingeBoost algorithm proceeds while working on the test set.

```
R> m2 <- 20
R> xinit <- unlist(dat.m1$ensemble)
R> xinit <- subset(xinit, !is.na(xinit))
R> dat.m2 <- mhingebst(x=train, y=train.classes, family = "hinge", learner = "sm",
 ctrl = bst_control(mstop=m2, twinboost=TRUE, f.init=dat.m1$yhat, xselect.init=xinit))
R> risk.te2 <- predict(dat.m2,newdata=test,newy=test.classes,type="error")
R> plot(risk.te2, type="l", xlab="Iteration", ylab="Test Error")
```

Plot the evolution of the number of genes selected versus the iteration counter, as the multi-class twin HingeBoost algorithm proceeds while working on the training set.

```
R> plot(nsel(dat.m2, m2), ylab="No. Genes", xlab="Iteration", lty="solid", type="l")
```

2 Classification of 14 Cancer Types

We use **bst** to classify patients into 14 cancer types using gene expression levels obtained from http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi. The Global Cancer Map (GCM) dataset has 16,063 genes for 144 training and 46 test samples. To proceed the analysis, pre-processing steps are taken before standardization: (a) thresholding (truncated below 20 and above 16,000), (b) filtering (removal of genes with $\max / \min \leq 5$ and $\max - \min \leq 500$, (c) logarithmic base 10 transformation. With 10,884 genes remaining after filtering, the multi-class HingeBoost and twin HingeBoost with componentwise linear least squares are applied to the data.

```
R> ### training data
R> filename <- paste("http://www.broadinstitute.org/mpr/publications/projects/",
 "Global_Cancer_Map/GCM_Training.res", sep="")
R> dat0 <- read.delim(filename, sep="\t", header=FALSE, skip=3, quote="")
R> tmp <- dat0[,1:290]
R> tmp <- tmp[, -seq(4, 290, by=2)]
R> tmp <- tmp[, -(1:2)]
R> train <- t(tmp)
R> filename <- paste("http://www.broadinstitute.org/mpr/publications/projects/",
 "Global_Cancer_Map/GCM_Training.cls", sep="")
R> train.classes <- read.table(filename, skip=2)+1
R> train.classes <- unlist(train.classes)
R> ### test data
R> filename <- paste("http://www.broadinstitute.org/mpr/publications/projects/",
 "Global_Cancer_Map/GCM_Test.res", sep="")
R> dat0 <- read.delim(filename, sep="\t", header=FALSE, skip=3, quote="")
R> tmp <- dat0[,1:110]
R> tmp <- tmp[, -seq(4, 110, by=2)]
R> tmp <- tmp[, -(1:2)]
R> test <- t(tmp)[1:46,]
R> filename <- paste("http://www.broadinstitute.org/mpr/publications/projects/",
```

```
 "Global_Cancer_Map/GCM_Test.cls", sep="")
R> test.classes <- read.table(filename, skip=2)+1
R> test.classes <- test.classes[test.classes!=15]
R> test.classes <- unlist(test.classes)
R> ### pre-processing data
R> train[train < 20] <- 20
R> train[train > 16000] <- 16000
R> filter <- apply(train, 2,
 function(x) if(max(x)/min(x) > 5 && max(x)-min(x) > 500)
 return(1)
 else 0)
R> train <- train[, filter==1]
R> train <- log10(train)
R> x <- train
R> meanx <- colMeans(x)
R> one <- rep(1,nrow(x))
R> normx <- sqrt(drop(one %*% (x^2)))
R> train <- scale(train, meanx, normx)
R> tmp <- cbind(train, train.classes)
R> tmp <- cbind(train, train.classes)
R> nx <- dim(tmp)[2]
R> a0 <- b0 <- 0
R> for(k in 1:length(table(train.classes))){
   tmp1 <- subset(tmp, tmp[,nx]==k)
   xc.bar <- colMeans(tmp1[,-nx])   ###average of gene j across class k
   xa.bar <- colMeans(tmp[,-nx])    ### average of gene j across all samples
   a0 <- a0 + dim(tmp1)[1] * ((xc.bar - xa.bar)^2)
   b0 <- b0 + colSums((tmp[,-nx] - xc.bar)^2)
 }
R> bw <- a0/b0
R> npre <- nx - 1  ### use all genes and ignore bw values
R> bw1 <- order(bw, decreasing=TRUE)[1:npre]
R> train <- train[,bw1]
R> test[test < 20] <- 20
R> test[test > 16000] <- 16000
R> test <- test[, filter==1]
R> test <- log10(test)
R> test <- scale(test, meanx, normx)[, bw1]
R> test <- as.data.frame(test)
R> colnames(train) <- paste("x", 1:dim(train)[2], sep="")
R> colnames(test) <- paste("x", 1:dim(test)[2], sep="")
```

Multi-class HingeBoost is applied for 200 boosting iterations. Plot the evolution of the misclassification error on the test data versus the iteration counter, as the multi-class HingeBoost algorithm proceeds while working on the test set.

```
R> m1 <- m2 <- 200
R> dat.m1 <- mhingebst(x=train, y=train.classes, ctrl = bst_control(mstop=m1),
 family = "hinge", learner = "ls")
R> risk.te1 <- predict(dat.m1, newdata=test, newy=test.classes, mstop=m1, type="error")
R> plot(risk.te1, type="l", xlab="Iteration", ylab="Test Error", ylim=c(0.15, 0.4))
```

Plot the evolution of the number of genes selected versus the iteration counter, as the multi-class HingeBoost algorithm proceeds while working on the training set.

```
R> plot(nsel(dat.m1, m1), ylab="No. Genes", xlab="Iteration", lty="solid", type="l")
```

Multi-class twin HingeBoost is applied based on the results from the first round HingeBoost for 150 iterations. Plot the evolution of the misclassification error on the test data versus the iteration counter, as the multi-class twin HingeBoost algorithm proceeds while working on the test set.

```
R> fhat1 <- predict(dat.m1, mstop=150, type="response")
R> xinit <- unlist(dat.m1$ensemble[1:150])
R> xinit <- subset(xinit, !is.na(xinit))
R> ### How many genes selected with mstop=150
R> length(unique(xinit))
R> dat.m2 <- mhingebst(x=train, y=train.classes, ctrl = bst_control(mstop=m2,
 twinboost=TRUE, f.init=fhat1, xselect.init=xinit), family = "hinge", learner = "ls")
R> risk.te1 <- predict(dat.m2, newdata=test, newy=test.classes, mstop=m2, type="error")
R> plot(risk.te1, type="l", xlab="Iteration", ylab="Test Error", ylim=c(0.15, 0.4))
```

Plot the evolution of the number of genes selected versus the iteration counter, as the multi-class twin HingeBoost algorithm proceeds while working on the training set.

```
R> plot(nsel(dat.m2, m2), ylab="No. Genes", xlab="Iteration", lty="solid", type="l")
```

References

Zhu Wang. Multi-class HingeBoost: Method and application to the classification of cancer types using gene expression data. *Methods of Information in Medicine*, 51(2):162–167, 2012.