Math7340 Final

Chengbo Gu

Problem 1. (10 points) MLE and boostrap for Poisson data

A random sample from the Poisson distribution $poisson(\lambda = e^{\theta})$ is provieded in the file "DataPois.txt"

(a) What if the sample size n? What is the sample mean $\overline{Y}$?

```
y<-as.numeric(t(read.table(file="DataPois.txt", header=TRUE)))
length(y)

## [1] 120

mean(y)

## [1] 1.816667
```

Sample size is 120, sample mean is 1.816667.

(b) Find the value of MLE $\hat{\theta}$ on this data set using numerical method.

```
nloglik <- function(theta) -sum(log(dpois(y, lambda=exp(theta))))
optim(par=1, nloglik)

## Warning in optim(par = 1, nloglik): one-dimensional optimization by Nelder-Mead is unr
eliable:
## use "Brent" or optimize() directly

## $par
## [1] 0.5970703
##
## $value
## [1] 201.1172
##
## $counts
## function gradient
##       26       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

The value of MLE $\hat{\theta}$ on this data set is 0.5970703.

(c) Test the null hypothesis that $\theta = 1$ at level 0.05, using a bootstrap confidence interval. What is your conclusion? Can you find the p-value?

```
n <- length(y)
nboot <- 1000
boot.theta <- rep(NA, nboot)
for (i in 1:nboot) {
  y.star <- y[sample(1:n, replace=TRUE)]
  nloglik.boot <- function(theta) -sum(log(dpois(y.star, lambda=exp(theta))))
  boot.theta[i] <- optim(par=1, nloglik.boot)$par
}

quantile(boot.theta, c(0.025, 0.975))

##      2.5%     97.5%
## 0.4542969 0.7259766

t.test(y, mu=exp(1))

##
##  One Sample t-test
##
## data:  y
## t = -7.1048, df = 119, p-value = 9.57e-11
## alternative hypothesis: true mean is not equal to 2.718282
## 95 percent confidence interval:
##  1.565388 2.067945
## sample estimates:
## mean of x
##  1.816667

tstat <- (mean(y) - exp(1))/(sd(y)/sqrt(n))
2*pt(-abs(tstat), df=n-1)

## [1] 9.569858e-11
```

The 95% boostrap confidence interval is (0.4542969, 0.7259766) which doesn't include 1.

Thus, we reject the null hypothesis and conclude that $\theta \neq 1$.

To say $\theta = 1$ is equal to say the $\overline{y} = e$, so we could apply a t-test to get the precise p-value. Of course we could do it manually.

The p-value here is 9.57e-11 which is extremely small. So we could reject the null hypothesis using p-value too.

Problem 2. (20 points) ANOVA

We analyze the data set NCI60 data from the ISLR library.

(a) Delete the cancer types with only one or two cases ("K562A-repro", etc.). Keep only the cancer types with more than 3 cases.

```
library(ISLR)
library(lmtest)

ncidata <- NCI60$data
ncilabs <- as.factor(NCI60$labs)

filter <- as.vector(sapply(ncilabs, function(x) table(ncilabs)[x] > 2))
mydata <- ncidata[filter,]
mylabs <- ncilabs[filter]
```

(b) Analyze the expression values of the first gene in the data (first column). Does the first gene express differently in different types of cancers? If so, in which pairs of cancer types does the first gene express differently? (Use FDR adjustment.)

```
anova( lm( mydata[,1] ~ mylabs ) )
```

```
## Analysis of Variance Table
##
## Response: mydata[, 1]
##           Df Sum Sq Mean Sq F value  Pr(>F)
## mylabs     7 2.8931 0.41331  2.3272 0.03928 *
## Residuals 49 8.7021 0.17759
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since the p-value for the types of cancers is 0.03928 which is less than 0.05, we reject the null hypothesis and conclude that the types of cancers do affect the gene express of first gene.

```
pairwise.t.test(mydata[,1],  mylabs, p.adjust.method='fdr')

##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  mydata[, 1] and mylabs
##
##          BREAST CNS  COLON LEUKEMIA MELANOMA NSCLC OVARIAN
## CNS      0.34   -    -     -        -        -     -
## COLON    0.35   0.10 -     -        -        -     -
## LEUKEMIA 0.42   0.11 0.93  -        -        -     -
## MELANOMA 0.93   0.34 0.34  0.34     -        -     -
## NSCLC    0.34   0.93 0.10  0.10     0.34     -     -
## OVARIAN  0.34   0.93 0.10  0.12     0.37     0.99  -
## RENAL    0.93   0.34 0.34  0.35     0.93     0.34  0.34
##
## P value adjustment method: fdr
```

Since all the pairwise p-values are greater than 0.05, there is no big difference between the expression levels of every two groups.

This situation is a little bit strange. The overall means for groups are not the same while the pairwise t-test shows no big difference. The reason behind this maybe the p-value for anova is 0.03928 that is not far from 0.05, which indicates that we almost believe the overall means are the same.

(c) Check the model assumptions for analysis in part (b). Is ANOVA analysis appropriate here?

```
shapiro.test( residuals( lm( mydata[,1] ~ mylabs )))

##
##  Shapiro-Wilk normality test
##
## data:  residuals(lm(mydata[, 1] ~ mylabs))
## W = 0.97947, p-value = 0.4414

bptest( lm( mydata[,1] ~ mylabs ), studentize = FALSE)

##
##  Breusch-Pagan test
##
## data:  lm(mydata[, 1] ~ mylabs)
## BP = 8.8392, df = 7, p-value = 0.2644
```

Both of the p-values are greater than 0.05. So the model assumptions are not violated. ANOVA is appropriate here.

(d) Apply ANOVA analysis to each of the 6830 genes. At FDR level of 0.05, how many genes express differently among different types of cancer patients?

```
# using anova()
p.values <- apply(mydata,2, function(x) anova( lm( x ~ mylabs ))[["Pr(>F)"]][1])
p.fdr <-p.adjust(p=p.values, method="fdr")
sum(p.fdr < 0.05)

## [1] 2808

# using kruskal()
p.values <- apply(mydata, 2, function(x) kruskal.test(x ~ mylabs)$p.value)
p.fdr <-p.adjust(p=p.values, method="fdr")
sum(p.fdr < 0.05)

## [1] 2186
```

We can directly apply the function anova here and get the conclusion that 2808 genes express differently among different types of cancer patients.

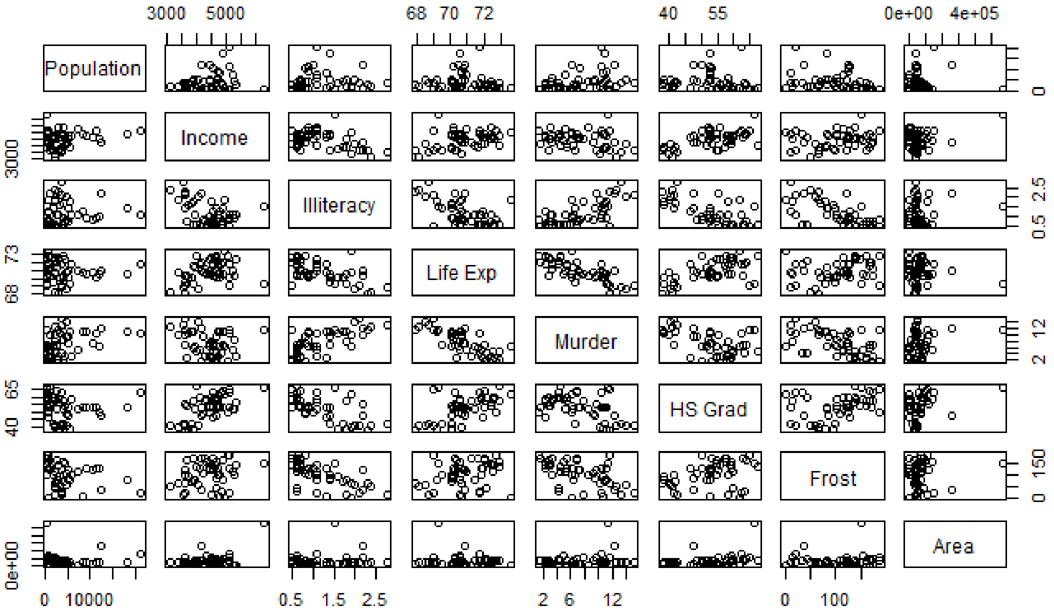
But we feel that the nonparametric version kruskal.test() is more suitable here because the model assumptions may be violated by some of the genes. By applying kruskal test, the number of genes that express differently becomes 2186.

Problem 3. (10 points) Regression

We consider the regression analysis on the state.x77 data set. In the module 9, we regressed the life expectancy on three variables: the murder rates, percentage of high-school graduates and mean number of frost days.

(a) Make pairwise scatterplots for all variables in the data set. Which variables appears to be linearly correlated with the life expectancy based on the scatterplots?

```
pairs(state.x77)
```



Based on the pairwise plot, Income, Illiteracy, Murder, HS Grad and Frost appear to be linearly correlated with the life expectancy.

(b) Conduct a regression analysis different from the example analysis in module 9. We regress the life expectancy on three variables: the per capita income (Income), the illiteracy rate (Illiteracy) and mean number of frost days (Frost). What is your regression equation? In this regression analysis, which of the three variables affect the life expectancy significantly?

```
data <- as.data.frame(state.x77[,c('Life Exp', 'Income', 'Illiteracy', 'Frost')])
names(data)<-c('Life.Exp', 'Income', 'Illiteracy', 'Frost')
lin.reg <- lm(Life.Exp~., data = data)
summary(lin.reg)

##
## Call:
## lm(formula = Life.Exp ~ ., data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.52141 -0.73588 -0.03975  0.80481  3.13642
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 72.5260080  1.6580229  43.742  < 2e-16 ***
## Income       0.0001819  0.0002825   0.644 0.522855
## Illiteracy  -1.5605544  0.3745147  -4.167 0.000135 ***
## Frost       -0.0060148  0.0040551  -1.483 0.144826
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.087 on 46 degrees of freedom
## Multiple R-squared:  0.3843, Adjusted R-squared:  0.3441
## F-statistic:  9.57 on 3 and 46 DF,  p-value: 5.038e-05
```

Life.Exp = 72.5260080 + 0.0001819Income - 1.5605544Illiteracy - 0.0060148Frost. According to the p-value, Illiteracy affects the life expectancy significantly.

(c) Find delete-one-cross-validated mean square errors $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_{(-i)})^2$ for this regression model.

```
n <- dim(state.x77)[1]
err <- 0
for (i in 1:n) {
  data.tr <- data[-i,]
  lin.reg.tr <- lm(Life.Exp~., data = data.tr)
  err <- err + as.numeric(predict(lin.reg.tr, newdata=data[i,]) - data[i,1])^2
}
err <- err/n
err

## [1] 1.345143
```

The delete-one-cross-validated mean square error is 1.345143.

Problem 4. (60 points) Predicting B-cell differentiation with gene expression.

We analyze data for the B-cell patients in the ALL data set in the textbook.

(a) Select gene expression data for only the B-cell patients. The analysis in following parts will only use these gene expression data on the B-cell patients.

```
library(ALL)
data(ALL)
Bcells <- ALL$BT %in% c('B', 'B1', 'B2', 'B3', 'B4')
ALLB <- exprs(ALL)[,Bcells]
```

(b) Select only those genes whose coefficient of variance (i.e., standard deviation divided by the mean) is greater than 0.2. How many genes are selected?

```
library(genefilter)
func <- cv(0.2)
select <- genefilter(ALLB, filterfun(func))
data.filtered <- ALLB[select,]
label.filtered <- as.character(ALL$BT[Bcells])
genes2 <- rownames(data.filtered)
dim(data.filtered)[1]

## [1] 184
```

184 genes are selected.

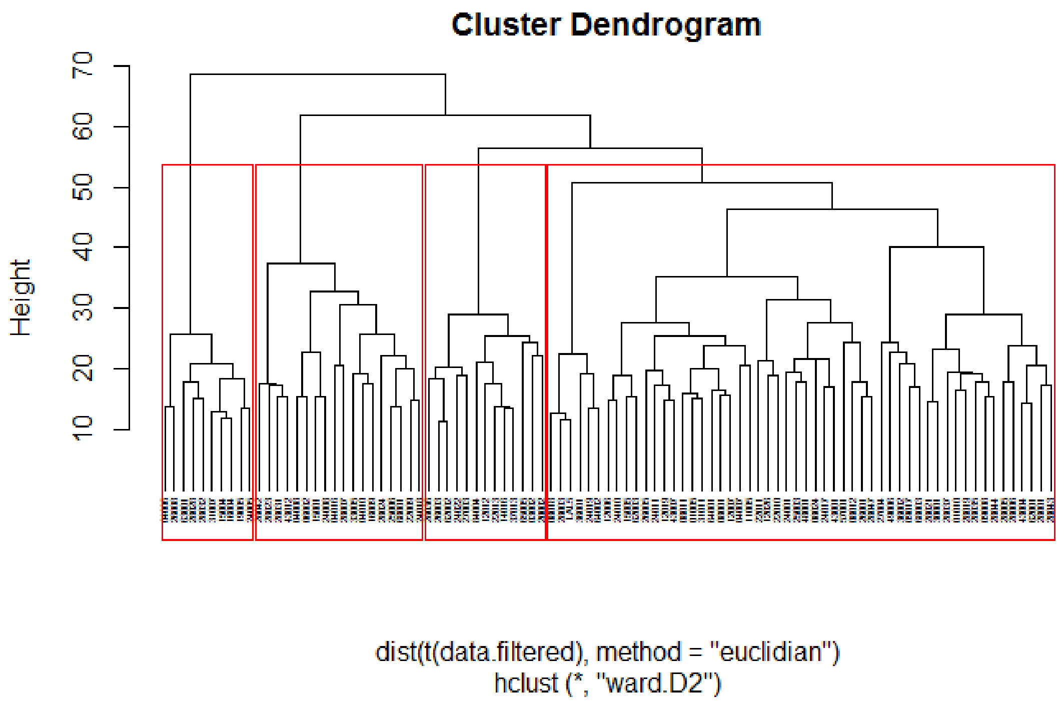
(c) We wish to conduct clustering analysis to study natural groupings of the patients predicted by the gene expression profiles. For this analysis, we first need to reduce the number of genes studied. The filter in (b) is one such choice. Please comment on what filtering methods you would use to choose genes, other than the filter in (b). What would you consider as the best gene filter in this case.

Another criteria could be that the absolute expression levels of the genes are big enough. Other methods include filtering out genes with expression levels which do not change significantly across samples, filtering out genes that violate the normality and homoscedasticity assumptios and so on.

For this task, the filter in (b) and the filter that the absolute expression levels of the genes are big enough would suffice.

(d) Conduct a hierarchical clustering analysis with filtered genes in (b). How do the clusters compare to the B-stages? How does do the clusters compare to the molecule biology types (in variable ALL$mol.biol)? Provide the confusion matrices of the comparisons, with 4 clusters.

```
hc.complete <- hclust(dist(t(data.filtered), method="euclidian"), method="ward.D2")
plot(hc.complete, hang=-1, cex=0.38)
rect.hclust(hc.complete, k=4)
```



```
groups <- cutree(hc.complete, k=4)
table(label.filtered, groups)

##                  groups
## label.filtered   1  2  3  4
##               B  3  1  0  1
##               B1  1  0 10  8
##               B2 25  6  0  5
##               B3 18  2  0  3
##               B4  7  4  0  1

MBT <- as.character(ALL$mol.biol[Bcells])
table(MBT, groups)

##              groups
## MBT           1  2  3  4
##   ALL1/AF4    0  0 10  0
##   BCR/ABL    24 13  0  0
##   E2A/PBX1    5  0  0  0
##   NEG        24  0  0 18
##   p15/p16     1  0  0  0
```

Here, I used "ward.D2" method to do clustering.

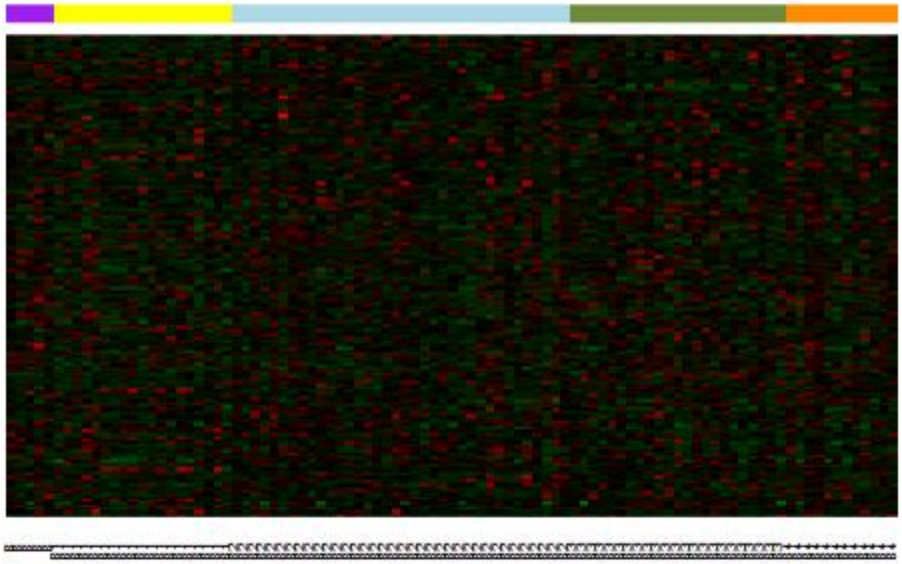
From the confusion matrices, the clusters could not reflect the information of B-stages because patients in each stage are separated in to several groups. But the clusters did reflect some information of molecule biology types because the confusion matrix here is kind of clean.

(e) Draw two heatmaps for the expression data in (d), one for each comparison. Using colorbars to show the comparison types (B-stages or molecule biology types). The clusters reflect which types better: B-stages or molecule biology types?

```
library(gplots)
```

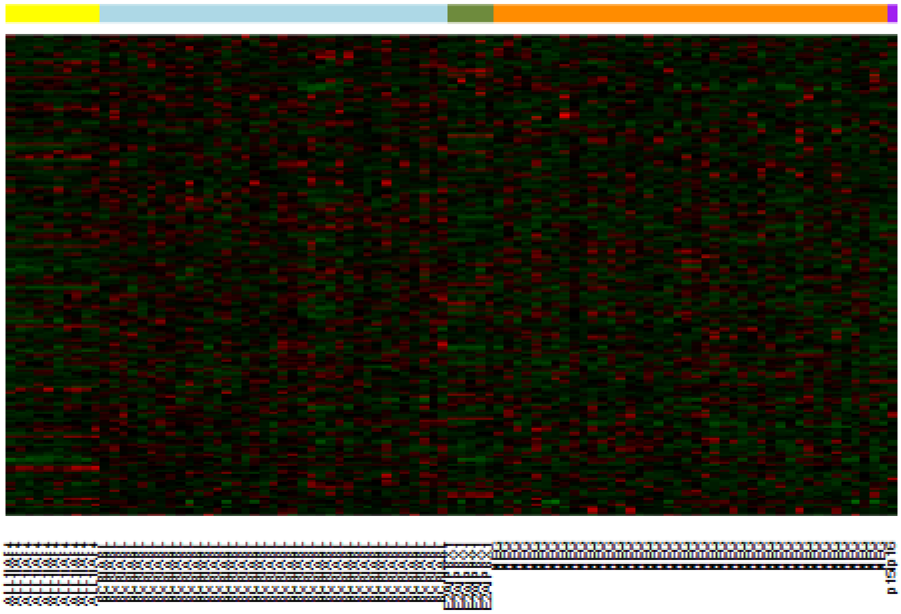
```
colnames(data.filtered) <- label.filtered
col.ord<-order(label.filtered)
dat1<-data.filtered[, col.ord]

color.map <- function(B) {
  if (B=="B1") "yellow"
  else if(B=="B2") "lightblue"
  else if(B=="B3") "darkolivegreen4"
  else if(B=="B4") "darkorange"
  else "purple"
}
patientcolors<- unlist(lapply(colnames(dat1), color.map))
heatmap.2(dat1, col=greenred(75), scale="row",
          Rowv=FALSE,
          Colv=FALSE,
          key=FALSE,
          ColSideColors=patientcolors,
          trace="none",
          dendrogram="none",
          labRow=NA)
```



```
col.ord<-order(MBT)
colnames(data.filtered) <- MBT
dat2<-data.filtered[, col.ord]

color.map <- function(MB) {
  if (MB=="ALL1/AF4") "yellow"
  else if(MB=="BCR/ABL") "lightblue"
  else if(MB=="E2A/PBX1") "darkolivegreen4"
  else if(MB=="NEG") "darkorange"
  else "purple"
}
patientcolors<- unlist(lapply(colnames(dat2), color.map))
heatmap.2(dat2, col=greenred(75), scale="row",
          Rowv=FALSE,
          Colv=FALSE,
          key=FALSE,
          ColSideColors=patientcolors,
          trace="none",
          dendrogram="none",
          labRow=NA)
```



From the two heatmaps, we see clearly that the second heatmap shows some patterns among different groups. Thus, the clusters reflect molecule biology types better.

(f) We focus on predicting the B-cell differentiation in the following analysis. We merge the last two categories "B3" and "B4", so that we are studying 3 classes: "B1", "B2" and "B34". Use linear model (limma library) to select genes that expresses differently among these three classes at FDR of 0.05. How many genes are selected?

```
library(limma)

Bcells.1 <- label.filtered %in% c('B1', 'B2', 'B3', 'B4')
label.filtered <- label.filtered[Bcells.1]
ALLB.1 <- ALLB[,Bcells.1]

mylabel <- as.factor(sapply(label.filtered, function(x)
  {if((x == 'B3') || (x == 'B4')) 'B34' else x}))

design.ma <- model.matrix(~0 + mylabel)
colnames(design.ma) <- c("B1", "B2", "B34")
fit <- lmFit(ALLB.1, design.ma)
fit <- eBayes(fit)
cont.ma <- makeContrasts(B1-B2,B2-B34, levels=mylabel)
fit1 <- contrasts.fit(fit, cont.ma)
fit1 <- eBayes(fit1)
genes1 <- rownames(topTable(fit1, number=Inf, p.value=0.05, adjust.method="fdr"))
length(genes1)
```

```
## [1] 1169
```

1169 genes are selected.

(g) Fit SVM and the classification tree on these selected genes in part (f), evaluate their performance with delete-one-cross-validated misclassification rate.

```r
library(e1071)
require(rpart)

tree.classification <- function(data.fit) {
  tree.fit <- rpart(label ~. , data= data.fit, method="class")
  pred.tr <- predict(tree.fit, data.fit, type="class")
  mcr.tr <- mean(pred.tr!=data.fit$label)
  cat("empirical mcr for classification tree :", mcr.tr, "\n")

  n <- dim(data.fit)[1]
  mcr.cv.raw <- rep(NA, n)
  for (i in 1:n) {
    fit.tr <- rpart(label ~. , data=data.fit[-i,], method="class")
    pred.tr <- predict(fit.tr, data.fit[i,], type="class")
    mcr.cv.raw[i] <- (pred.tr!=data.fit$label[i])
  }
  mcr.cv <- mean(mcr.cv.raw)
  cat("cross-validation mcr for classification tree:", mcr.cv, "\n\n")
}

svm.classification <- function(data.fit){
  svm.fit <- svm(label~., type="C-classification", kernel="linear", data=data.fit)
  pred.svm <- predict(svm.fit, data.fit)
  mcr.svm <- mean(pred.svm!=data.fit$label)
  cat("empirical mcr for SVM:", mcr.svm, "\n")

  n <- dim(data.fit)[1]
  mcr.cv.raw <- rep(NA, n)
  for (i in 1:n) {

    fit.svm <- svm(label~. , type="C-classification", kernel="linear", data=data.fit[-i,])
    pred.svm <- predict(fit.svm, data.fit[i,])
    mcr.cv.raw[i] <- (pred.svm!=data.fit$label[i])
  }
  mcr.cv <- mean(mcr.cv.raw)
  cat("cross-validation mcr for SVM:", mcr.cv, "\n\n")
}

data <- ALLB.1[genes1,]
label <- mylabel
data.fit <- data.frame(label, t(data))
svm.classification(data.fit)

## empirical mcr for SVM: 0
## cross-validation mcr for SVM: 0.2

tree.classification(data.fit)

## empirical mcr for classification tree : 0.1111111
## cross-validation mcr for classification tree: 0.3333333
```

The empirical mcr for SVM is 0% while the cross-validation mcr for SVM is 20%. The empirical mcr for classification tree is 11.11% while the cross-validation mcr for SVM is 33.33%.

(h) We select the genes passing both filters in (b) and (f). How many genes are selected? Redo part (g) on these genes passing both filters.

```
data.final <- ALLB.1[intersect(genes1,genes2),]
dim(data.final)[1]
```

```
## [1] 55
```

```
label <- mylabel
data.fit <- data.frame(label, t(data.final))
svm.classification(data.fit)
```

```
## empirical mcr for SVM: 0
## cross-validation mcr for SVM: 0.2444444
```

```
tree.classification(data.fit)
```

```
## empirical mcr for classification tree : 0.1222222
## cross-validation mcr for classification tree: 0.1777778
```

55 genes are selected.

The empirical mcr for SVM is 0% while the cross-validation mcr for SVM is 24.44%. The empirical mcr for classification tree is 12.22% while the cross-validation mcr for SVM is 17.78%.

(i) Which classifier you will consider best among the classifiers studied in part (g) and part (h)? Why?

I would like to choose classification tree in part (h) because it has the lowest cross-validation misclassification rate 17.78% and it is just a little bit overfitting.

Problem 5. Bonus question (Extra 10 points) Poisson regression.

A random sample of $(X_1, Y_1),\dots,(X_n, Y_n)$ is provided in the file "DataPoisReg.txt". The Y_i comes from the Poisson distribution $poisson(\lambda = e^{\beta_0+\beta_1 X_i})$.

(a) Find the value of MLE $(\hat{\beta}_0, \hat{\beta}_1)$ on this data set using numerical method.

```
my.dat <- read.table(file="DataPoisReg.txt", header=TRUE)
x <- my.dat[,1]
y <- my.dat[,2]

nloglik <- function(params) {
  beta0 <- params[1]
  beta1 <- params[2]
  -sum(log(dpois(y, lambda=exp(beta0+beta1*x))))
}
optim(par=c(2,2), nloglik)
```

```
## $par
## [1] 1.037357 4.300752
##
## $value
## [1] 164.428
##
## $counts
## function gradient
##      111       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

The values of MLE ($\hat{\beta}_0$,$\hat{\beta}_1$) on this data set are 1.037357, 4.300752.

(b) Is the slop parameter $\beta_1 = 2$? Use appropriate statistical test to answer this question.

```
poisson.fit <- glm(y~x, data = my.dat, family = poisson)
summary(poisson.fit)
```

```
##
## Call:
## glm(formula = y ~ x, family = poisson, data = my.dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1981  -0.8274  -0.5359   0.2275   2.7643
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.03714    0.06565   15.80   <2e-16 ***
## x            4.30043    0.28824   14.92   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 449.06  on 119  degrees of freedom
## Residual deviance: 115.66  on 118  degrees of freedom
## AIC: 332.86
##
## Number of Fisher Scoring iterations: 5
```

```
confint(poisson.fit)
```

```
## Waiting for profiling to be done...
```

```
##                 2.5 %   97.5 %
## (Intercept) 0.9050006 1.162593
## x           3.7514085 4.882484
```

Here I applied poisson regression and get the 95% confidence interval for β_1 is (3.7514085 ,4.882484). The value 2 is not in this CI, thus we reject the null hypothesis that $\beta_1 = 2$.