ENSEMBLE AND MIXTURE MODELS – 10 POINTS

(8 p) Consider the code below, which produces some training and test data from a Gaussian mixture model. The model has three components with equal priors, but different mean vectors and covariance matrices. Your first task is to use the training data to learn a generative model and, then, turn it into a discriminative model to predict the class labels of the test data. Report the error rate in the test data. Check the course slides to recall how to solve this problem.

Your next task is to apply bagging to solve the problem above. In other words, use the training data provided to learn an ensemble model consisting of B base discriminative models, each learned from a bootstrapped sample of the training data as described above, which are then combined by majority voting. Report the error rate in the test data. Check the course slides to see recall how to solve this problem.

```
#install.packages("mvtnorm")
library(mvtnorm)
set.seed(123)
N=300 # number of training points
M=3000 # number of test points
D=2 # number of dimensions
tr <- matrix(nrow=N, ncol=D+1) # training data
te <- matrix(nrow=M, ncol=D+1) # test data
B <- 10 # number of bootstrap samples
pte <- matrix(nrow=M,ncol=B) # class predictions for the test data from the individual predictors
# producing the training data
mu1<-c(0,0)
Sigma1 <- matrix(c(5,1,1,5),D,D)
dat1<-rmvnorm(n = 1100, mu1, Sigma1)
mu2<-c(4,6)
Sigma2 <- matrix(c(5,-1,-1,5),D,D)
dat2<-rmvnorm(n = 1100, mu2, Sigma2)
mu3<-c(7,2)
Sigma3 <- matrix(c(3,2,2,3),D,D)
dat3<-rmvnorm(n = 1100, mu3, Sigma3)
```

```
plot(dat1,xlim=c(-10,15),ylim=c(-10,15))
points(dat2,col="red")
points(dat3,col="blue")
tr[1:100,]<-c(dat1[1:100,],rep(1,100))
tr[101:200,]<-c(dat2[1:100,],rep(2,100))
tr[201:300,]<-c(dat3[1:100,],rep(3,100))
te[1:1000,]<-c(dat1[101:1100,],rep(1,1000))
te[1001:2000,]<-c(dat2[101:1100,],rep(2,1000))
te[2001:3000,]<-c(dat3[101:1100,],rep(3,1000))
K=3 # number of classes
pi <- vector(length=K) # mixing coefficients
mu <- matrix(nrow=K, ncol=D) # class conditional means
Sigma <- array(dim=c(D,D,K)) # class conditional covariances
p <- matrix(nrow=M, ncol=K) # posterior class probabilities
```

To solve the second task, you may want to use the following function.

```
# returns the mode/majority of a vector
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
```

(1 p) The bagging model learned above may not be much better than the single model learned above. Provide a possible explanation for this phenomenon.

(1 p) Bagging can be improved by decorrelating the ensemble's base models (recall random forest, for instance). How could you achieve this in the task above? Assume that D=10, i.e. there are 10 predictive attributes instead of two.