732A99/TDDE01/732A68 MACHINE LEARNING

LAB 1 BLOCK 2: ENSEMBLE METHODS AND MIXTURE MODELS

JOSE M. PEÑA
IDA, LINKÖPING UNIVERSITY, SWEDEN

INSTRUCTIONS

The instructions and submission procedure from the previous labs apply to this lab as well.

RESOURCES

The assignment 1 is designed to be solved with the R package `randomForest`. No R package is required to solve the assignments 2 and 3.

1. ENSEMBLE METHODS

Your task is to learn some random forests using the function `randomForest` from the R package `randomForest`. The training data is produced by running the following R code:

```
x1<-runif(100)
x2<-runif(100)
trdata<-cbind(x1,x2)
y<-as.numeric(x1<x2)
trlabels<-as.factor(y)
```

The task is therefore classifying Y from X_1 and X_2, where Y is binary and X_1 and X_2 continuous. You should learn a random forest with 1, 10 and 100 trees, which you can do by setting the argument `ntree` to the appropriate value. Use `nodesize = 25` and `keep.forest = TRUE`. The latter saves the random forest learned. You need it because you should also compute the misclassification error in the following test dataset (use the function `predict` for this purpose):

```
set.seed(1234)

x1<-runif(1000)
x2<-runif(1000)
tedata<-cbind(x1,x2)
y<-as.numeric(x1<x2)
telabels<-as.factor(y)
plot(x1,x2,col=(y+1))
```

- Repeat the procedure above for 1000 training datasets of size 100 and report the mean and variance of the misclassification errors. In other words, create 1000 training datasets of size 100, learn a random forest from each dataset, and compute the misclassification error in the **same** test dataset of size 1000. Report results for when the random forest has 1, 10 and 100 trees.
- Repeat the exercise above but this time use the condition `(x1<0.5)` instead of `(x1<x2)` when producing the training **and** test datasets.

- Repeat the exercise above but this time use the condition `((x1<0.5 & x2<0.5) | (x1>0.5 & x2>0.5))` instead of `(x1<x2)` when producing the training **and** test datasets. Unlike above, use `nodesize = 12` for this exercise.
- Answer the following questions:
  - What happens with the mean error rate when the number of trees in the random forest grows? Why?
  - The third dataset represents a slightly more complicated classification problem than the first one. Still, you should get better performance for it when using sufficient trees in the random forest. Explain why you get better performance.

2. Mixture Models

Your task is to implement the EM algorithm for Bernoulli mixture model. Please use the R template below to solve the assignment. Then, use your implementation to show what happens when your mixture model has too few and too many clusters, i.e. set $M = 2, 3, 4$ and compare results. Please provide a short explanation as well.

A Bernoulli mixture model is

$$p(\boldsymbol{x}) = \sum_{m=1}^{M} \pi_m \, Bern(\boldsymbol{x}|\boldsymbol{\mu}_m)$$

where $\boldsymbol{x} = (x_1, \ldots, x_D)$ is a D-dimensional binary random vector, $\pi_m = p(y = m)$ and

$$Bern(\boldsymbol{x}|\boldsymbol{\mu}_m) = \prod_{d=1}^{D} \mu_{m,d}^{x_d} \, (1 - \mu_{m,d})^{(1-x_d)}$$

where $\boldsymbol{\mu}_m = (\mu_{m,1}, \ldots, \mu_{m,D})$ is a D-dimensional vector of probabilities. As usual, the log likelihood of the dataset $\{\boldsymbol{x}_i\}_{i=1}^n$ is

$$\sum_{i=1}^{n} \log p(\boldsymbol{x}_i).$$

Finally, in the EM algorithm, the parameter updates for the Bernoulli mixture model are the same as for the Gaussian mixture model (see Equations 10.16a,b in the lecture slides).

```
set.seed(1234567890)

max_it <- 100 # max number of EM iterations
min_change <- 0.1 # min change in log lik between two consecutive iterations
n=1000 # number of training points
D=10 # number of dimensions
x <- matrix(nrow=n, ncol=D) # training data

true_pi <- vector(length = 3) # true mixing coefficients
true_mu <- matrix(nrow=3, ncol=D) # true conditional distributions
true_pi=c(1/3, 1/3, 1/3)
true_mu[1,]=c(0.5,0.6,0.4,0.7,0.3,0.8,0.2,0.9,0.1,1)
true_mu[2,]=c(0.5,0.4,0.6,0.3,0.7,0.2,0.8,0.1,0.9,0)
true_mu[3,]=c(0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5)
plot(true_mu[1,], type="o", col="blue", ylim=c(0,1))
points(true_mu[2,], type="o", col="red")
points(true_mu[3,], type="o", col="green")

# Producing the training data
for(i in 1:n) {
m <- sample(1:3,1,prob=true_pi)
for(d in 1:D) {
x[i,d] <- rbinom(1,1,true_mu[m,d])
}
}
```

```
M=3 # number of clusters
w <- matrix(nrow=n, ncol=M) # weights
pi <- vector(length = M) # mixing coefficients
mu <- matrix(nrow=M, ncol=D) # conditional distributions
llik <- vector(length = max_it) # log likelihood of the EM iterations

# Random initialization of the parameters
pi <- runif(M,0.49,0.51)
pi <- pi / sum(pi)
for(m in 1:M) {
mu[m,] <- runif(D,0.49,0.51)
}
pi
mu

for(it in 1:max_it) {
plot(mu[1,], type="o", col="blue", ylim=c(0,1))
points(mu[2,], type="o", col="red")
points(mu[3,], type="o", col="green")
#points(mu[4,], type="o", col="yellow")
Sys.sleep(0.5)

# E-step: Computation of the weights
# Your code here

#Log likelihood computation.
# Your code here

cat("iteration: ", it, "log likelihood: ", llik[it], "\n")
flush.console()
# Stop if the lok likelihood has not changed significantly
# Your code here

#M-step: ML parameter estimation from the data and weights
# Your code here
}
pi
mu
plot(llik[1:it], type="o")
```

3. Theory

In this task, you are ask to find answers to some questions in the main course book (MLFC). The answer to each question should be up to five sentences long, and should also include the page numbers in the book where this information can be found.

- In an ensemble model, is it true that the larger the number B of ensemble members the more flexible the ensemble model?
- In AdaBoost, what is the loss function used to train the boosted classifier at each iteration?
- Sketch how you would use cross-validation to select the number of components (or clusters) in unsupervised learning of GMMs.