732A95 INTRODUCTION MACHINE LEARNING

LAB 2B BLOCK 2: MIXTURE MODELS

JOSE M. PEÑA
IDA, LINKÖPING UNIVERSITY, SWEDEN

INSTRUCTIONS

Each student must submit a report with his/her solutions to the lab. Submission is done via LISAM and before the deadline. The submission file should be named Name_LastName.pdf. The report must be concise but complete. It should include (i) the code implemented or the calls made to existing functions, (ii) the results of such code or calls, and (iii) explanations for (i) and (ii).

PhD students pass the lab if their individual report is of sufficient quality. TDDE01 and 732A95 students pass the lab as follows. The students must discuss their lab solutions in a group. Each group must compile a collaborative report that will be used for presentation at the seminar. The report should clearly state the names of the students that participated in its compilation and a short description of how each student contributed to the report. This report should be submitted via LISAM and before the deadline. The file should be named Group_X.pdf where X is the group number. The collaborative reports are corrected and graded. The individual reports are also checked, but feedback on them will not be given. The students pass the lab if their group report passes the seminar and their individual reports have reasonable quality, otherwise the students must complete their individual reports by correcting the mistakes in them.

Please note that one assignment is mandatory and the other is optional, i.e. students must solve the mandatory assignment to pass the lab.

Please use `set.seed(1234567890)` at the beginning of each assignment to ensure reproducibility.

RESOURCES

No R package is needed to solve the assignment. **Please use only basic R functions in your solution, e.g. sum() and alike**.

ASSIGNMENT 1 (**Mandatory**)

Implement the EM algorithm for mixtures of multivariate Benouilli distributions. Please use the template in the next page to solve the assignment. Please use only basic R functions in your solution, e.g. sum() and alike

ASSIGNMENT 2 (**Optional**)

Use the implementation of the EM algorithm in the previous assignment to show what happens when your mixture models has too few and too many components, i.e. set $K = 2, 3, 4$ and compare the results. Please provide a short explanation.

```
set.seed(1234567890)

max_it <- 100 # max number of EM iterations
min_change <- 0.1 # min change in log likelihood between two consecutive EM 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(n in 1:N) {
  k <- sample(1:3,1,prob=true_pi)
  for(d in 1:D) {
    x[n,d] <- rbinom(1,1,true_mu[k,d])
  }
}

K=3 # number of guessed components
z <- matrix(nrow=N, ncol=K) # fractional component assignments
pi <- vector(length = K) # mixing coefficients
mu <- matrix(nrow=K, ncol=D) # conditional distributions
llik <- vector(length = max_it) # log likelihood of the EM iterations

# Random initialization of the paramters
pi <- runif(K,0.49,0.51)
pi <- pi / sum(pi)
for(k in 1:K) {
  mu[k,] <- 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 fractional component assignments
  # 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 fractional component assignments
  # Your code here
}
pi
mu
plot(llik[1:it], type="o")
```