hw5

Qiuying Li UNI ql2280

4/9/2017

# 1. load histogram data  
#A matrix of histograms, denoted H, which contains one histogram vector in each row.   
#Each column corre-lation sponds to a histogram bin.  
setwd("~/Desktop/2017 spring/GR 5241/HW/hw5")  
file = ("histograms.bin")  
data = readBin(file, "double", 640000)  
W = matrix(data,4000,16)  
dire = "~/Desktop/2017 spring/GR 5241/HW/hw5/"  
file\_name = paste(dire, "histograms.bin", sep="")  
H <- matrix(readBin(file\_name, "double", 640000), 40000, 16)  
   
 # H has dimension n \* d  
#The centroids t1 , ..., tK . These are Rd vectors (just like the input features).  
 # T has dimension K \* d  
 # c has dimension K \* 1  
 # A has dimension n \* K  
 # n = 40000, d = 16, K=3,4,5  
MultinomialEM <- function(H, K, tau=0.0001){  
  
 n = dim(H)[1]  
 d = dim(H)[2]  
 index = sample(n, K, replace=F)  
 #The centroids t1 , ..., tK . These are Rd vectors  
 #T has dimension K \* d  
 #Each centroid is the parameter vector of a multinomial distribution and can be regarded as the center of a cluster.  
 T = diag(1 / rowSums(H[index, ] + 0.01)) %\*% (H[index, ]+0.01)  
 #Ck is the cluster weight  
 #Ck = (num of points in k) / n   
 c = rep(1 / K, K)  
 iter\_cut = c()  
 #The entry aik specifies the probability of feature i to be assigned to cluster k.  
 A0 = matrix(rep(0, n \* K), n, K)  
   
 # E step formula   
 theta\_1 = t(apply(tcrossprod(H, log(T)), 1, minusMax))  
 theta = exp(theta\_1) %\*% diag(c)  
 A = t(apply(theta, 1, normalize))  
  
 # M step formula  
 c = colSums(A) / n  
 bk = crossprod(A, H)  
 tk = diag(1 / rowSums(bk)) %\*% bk  
  
   
 # Compute a measure of the change of assignments during the current iteration:  
 #δ := ||A − Aold||  
 #Terminate the iteration when δ < τ  
 while (norm(A - A0, type="1") > tau){  
 A0 = A  
 # E step  
 theta\_1 = t(apply(tcrossprod(H, log(T)), 1, minusMax))  
 theta = exp(theta\_1) %\*% diag(c)  
 A = t(apply(theta, 1, normalize))  
 # M step  
 c = colSums(A) / n  
 bk = crossprod(A, H)  
 tk = diag(1 / rowSums(bk)) %\*% bk  
 iter\_cut = c(iter\_cut, norm(A-A0, type="1"))  
 }  
  
 # Turn the soft assignments into a vector m of hard assignments   
 # mi := arg max aik   
 m = apply(A, 1, which.max)  
 return(list(m=m, A=A, T=T, c=c, iter\_cut=iter\_cut))   
}  
  
normalize <- function(v) v / sum(v)  
minusMax <- function(v) v - max(v)  
  
K\_vec = c(3, 4, 5)  
figureLoc = paste(dire, "P3figure\_K", K\_vec, ".pdf", sep="")  
for (i in 1:length(K\_vec)){  
 result = MultinomialEM(H, K=K\_vec[i], tau=0.01)  
 pdf(width=8, height=8, file=figureLoc[i])  
 image(matrix(result$m, 200, 200, byrow=T), col=gray((0:16)/16), main=paste("K=", K\_vec[i], sep=""))  
}  
  
pdf(width=6, height=6, file=paste(dire, "norm\_vs\_iter.pdf", sep=""))  
plot(1:length(result$iter\_cut), result$iter\_cut, type="o", xlab="number of iterations", ylab="||A-A0||", main="K=5, tau=0.01")  
dev.off()

## quartz\_off\_screen   
## 2