**80-Topic Model:**

topicModel=read.table("topics80.csv", header=T, row.names=1, dec=".", sep=",")

topicModel=topicModel[-1,]

head(topicModel, n=10)

JensenDist <- function(inputLocal, pseudocount=0.000001, ...) {

Kullback <- function(x,y) sum(x \*log(x/y))

jensenLocal<- function(x,y) sqrt(0.5 \* Kullback(x, (x+y)/2) + 0.5 \* Kullback(y, (x+y)/2))

dataColumn <- length(colnames(inputLocal))

matrixRowSize <- length(rownames(inputLocal))

colnames <- colnames(inputLocal)

jensenMat <- matrix(0, dataColumn, dataColumn)

inputLocal = apply(inputLocal,1:2,function(x) ifelse (x==0,pseudocount,x))

for(i in 1:dataColumn) {

for(j in 1:dataColumn) {

jensenMat[i,j]=jensenLocal(as.vector(inputLocal[,i]),

as.vector(inputLocal[,j]))

}

}

colnames -> colnames(jensenMat) -> rownames(jensenMat)

as.dist(jensenMat)->jensenMat

attr(jensenMat, "method") <- "dist"

return(jensenMat)

}

topicModel.dist=JensenDist(topicModel)

topicModel.dist

# Lets cluster the data matrix

Kmeans=function(x,k) {

require(cluster)

cluster = as.vector(pam(as.dist(x), k, diss=TRUE)$clustering)

return(cluster)

}

topicCluster=Kmeans(topicModel.dist, k=5)

require(clusterSim)

numKmeans = index.G1(t(topicModel), topicCluster, d = topicModel.dist, centrotypes = "medoids")

numKmeans=NULL

for (k in 1:20) {

if (k==1) {

numKmeans[k]=NA

} else {

topicCluster\_temp=Kmeans(topicModel.dist, k)

numKmeans[k]=index.G1(t(topicModel),topicCluster\_temp, d = data.dist,

centrotypes = "medoids")

}

}

plot(numKmeans, type="h", xlab="number of clusters[peak]", ylab="CH index",main="Optimal number of clusters")

obs.silhouette=mean(silhouette(topicCluster, topicModel.dist)[,3])

cat(obs.silhouette) #0.1899451

topicModel=noise.removal(topicModel, percent=0.01)

## plot 1

obs.pca=dudi.pca(topicModel.frame(t(topicModel)), scannf=F, nf=10)

obs.pca

obs.bet=bca(obs.pca, fac=as.factor(topicCluster), scannf=F, nf=k-1)

dev.new()

s.class(obs.bet$ls, fac=as.factor(topicCluster), grid=F,sub="Between-class analysis")

#plot 2

obs.pcoa=dudi.pco(topicModel.dist, scannf=F, nf=3)

dev.new()

s.class(obs.pcoa$li, fac=as.factor(topicCluster), grid=F,sub="Principal coordiante analysis with K=80(topics)")

s.class(obs.pcoa$li,fac=as.factor(topicCluster), grid=F, cell=0, cstar=0)

s.label(obs.pcoa$li, xax = 1, yax = 2, label = row.names(obs.pcoa$li),sub="PCoA Labeled with K=80(topics)")

**20 Topic Model:**

topicModel=read.table("topics20.csv", header=T, row.names=1, dec=".", sep=",")

topicModel=topicModel[-1,]

head(topicModel, n=10)

JensenDist <- function(inputLocal, pseudocount=0.000001, ...) {

Kullback <- function(x,y) sum(x \*log(x/y))

jensenLocal<- function(x,y) sqrt(0.5 \* Kullback(x, (x+y)/2) + 0.5 \* Kullback(y, (x+y)/2))

dataColumn <- length(colnames(inputLocal))

matrixRowSize <- length(rownames(inputLocal))

colnames <- colnames(inputLocal)

jensenMat <- matrix(0, dataColumn, dataColumn)

inputLocal = apply(inputLocal,1:2,function(x) ifelse (x==0,pseudocount,x))

for(i in 1:dataColumn) {

for(j in 1:dataColumn) {

jensenMat[i,j]=jensenLocal(as.vector(inputLocal[,i]),

as.vector(inputLocal[,j]))

}

}

colnames -> colnames(jensenMat) -> rownames(jensenMat)

as.dist(jensenMat)->jensenMat

attr(jensenMat, "method") <- "dist"

return(jensenMat)

}

topicModel.dist=JensenDist(topicModel)

topicModel.dist

# Lets cluster the data matrix

Kmeans=function(x,k) {

require(cluster)

cluster = as.vector(pam(as.dist(x), k, diss=TRUE)$clustering)

return(cluster)

}

topicCluster=Kmeans(topicModel.dist, k=5)

require(clusterSim)

numKmeans = index.G1(t(topicModel), topicCluster, d = topicModel.dist, centrotypes = "medoids")

numKmeans=NULL

for (k in 1:20) {

if (k==1) {

numKmeans[k]=NA

} else {

topicCluster\_temp=Kmeans(topicModel.dist, k)

numKmeans[k]=index.G1(t(topicModel),topicCluster\_temp, d = data.dist,

centrotypes = "medoids")

}

}

plot(numKmeans, type="h", xlab="number of clusters[peak]", ylab="CH index",main="Optimal number of clusters")

obs.silhouette=mean(silhouette(topicCluster, topicModel.dist)[,3])

cat(obs.silhouette)