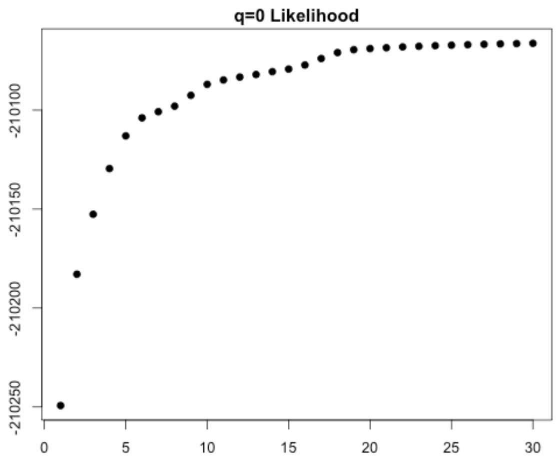
1.Initialization

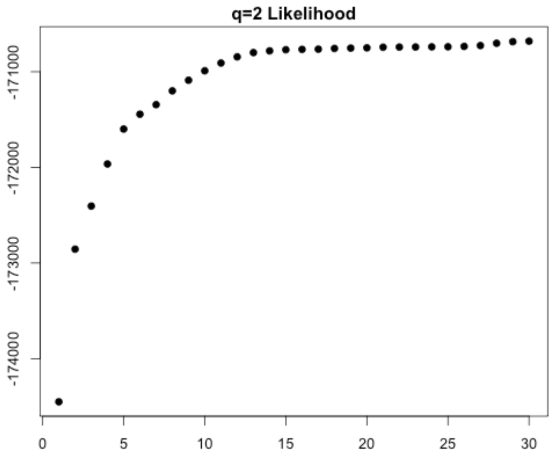
Use R's kmeans function with several random starts to build a preliminary clustering. Set γik= 1 if observation i is assigned to cluster k and γik=0 otherwise. By running the K-means algorithm 10 times, each with a maximum iteration number of 20, finally we get 10 clusters. And the clusters size show in below table

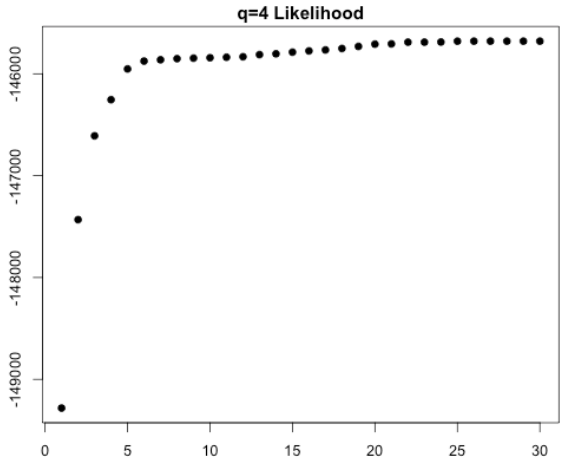
Cluster	1	2	3	4	5	6	7	8	9	10
Number	157	225	131	110	88	168	254	132	166	162

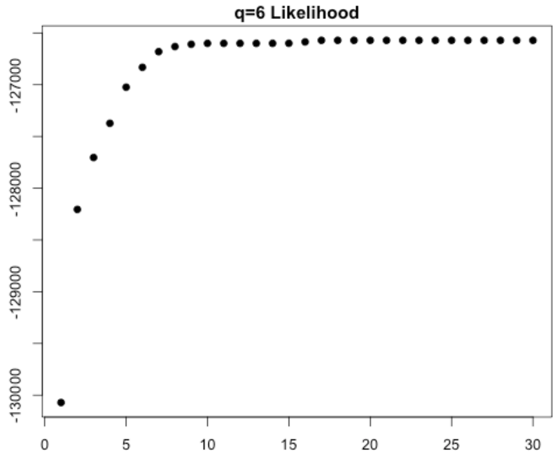
2.Convergence:

The log-likelihood vs. iteration number plots are shown as follows. We generate a plot of the observed data log-likelihood vs. iteration number (4 plots, 1 for each q, q= # principle components).









The final log-likelihood values are listed as follows:

q value	Likelihood
q=0	-210662.2
q=2	-170680.7
q=4	-145721.1
q=6	-125926.6

We can see from the table above, the log likelihood increase as the q value increase.

3.Choice of Number of Principle Component, q:

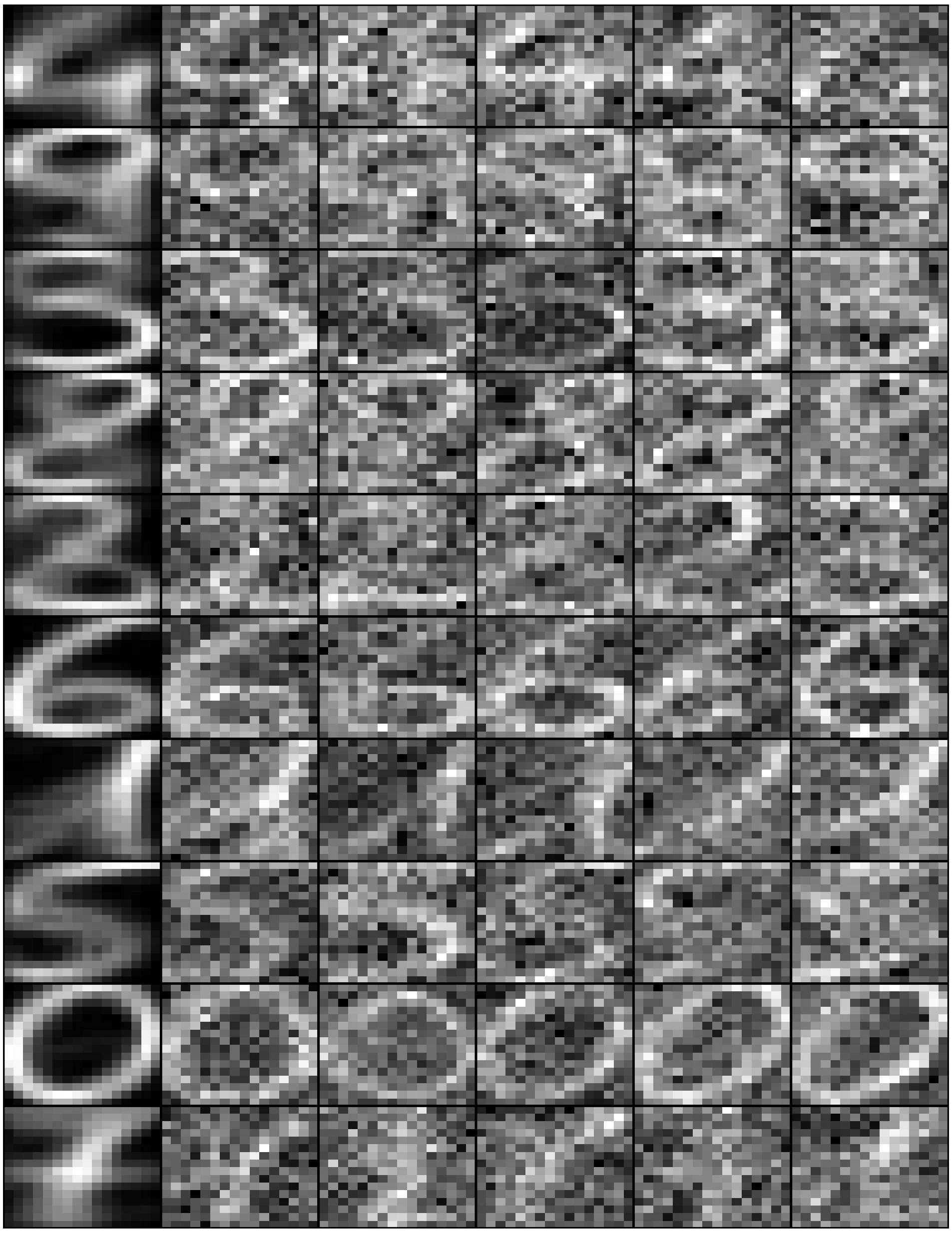
The Akaike information criterion (AIC) is a measure of the relative quality of statistical models for a given set of data. It offers a relative estimate of the information lost when a given model is used to represent the process that generates the data. Given a set of candidate models for the data, the preferred model is the one with the minimum AIC value.

q value	AIC
q=0	420134.5
q=2	342385.3
q=4	293480.0
q=6	254897.3

We can see from the table above, when q=6 the AIC value is the minimum. Thus we choose 6 principle component.

4.Visualization of Clusters:

We visualized the cluster mean and drew 5 samples from each cluster-specific distribution, where q is set to be 6. As we can see from this visualization, the digits 4, 3, 6 and 0 work pretty good for this algorithms. However, it doesn't work well for other digits. Like digits 8, it's barely recognizable from the visualization results.



5.Accuracy Assessment:

Mis-categorization rate (For q=6)

Cluster	1	2	3	4	5	6	7	8	9	10
Rate	0.0311	0.4259	0.3270	0.2579	0.1242	0.3962	0.2112	0.1646	0.3355	0.3861

Overall mis-catagorization rate

q	0	2	4	6
Rate	0.39485	0.28813	0.30138	0.26554

R Code

```
# ISyE 6740 Take Home Exam #1
library(mvtnorm)

# Read handwritten digits data
myData=read.csv("semeion.csv",header=FALSE)
myX=data.matrix(myData[,1:256])
myLabel=apply(myData[,257:266],1,function(xx){
return(which(xx=="1")-1)
})

# Number of rows
NR=dim(myX)[1]
# Number of columns
NC=dim(myX)[2]
# Number of clusters
Nclu=10
# Number of principal component
q=6

#Cluster data by using kmeans method, K=10
myCluster=kmeans(myX,10,iter.max=20,nstart=10)

#Initialization: assignments of data
gamma=matrix(0,nrow=NR,ncol=Nclu)
for(i in 1:NR) {
    Clc=myCluster$cluster[i]
    gamma[i, Clc]=1
}

#For likelihood
likeli=rep(0,30)


#Iterations
for(it in 1:30){

    N=matrix(0,1,10)
    for(i in 1:10) {
        N[i]=sum(gamma[,i])
    }

    Mu=matrix(0,nrow=Nclu,ncol=NC)
    pi=matrix(0,10,1)
```

```
#Initialization of covariance matrices
	sigma=array(dim=c(256,256,10))
	px=matrix(0,NR,Nclu)

#------------------------M-Step------------------------

	for(k in 1:Nclu){
		mu_k=rep(0,256)
		for(n in 1:NR){
			mu_k=mu_k+gamma[n,k]*myX[n, ]
		}
		Mu[k,]=mu_k/N[k]
	}

	pi=colSums(gamma)/NR

	for (k in 1:Nclu){
		Covar_k=matrix(0,256,256)

		for(n in 1:NR){
			Xi_Bar=myX[n, ]-Mu[k, ]
			Covk_temp=(Xi_Bar %*% t(Xi_Bar))*gamma[n,k]
			Covar_k=Covar_k+Covk_temp
		}
		Covar_k=Covar_k/N[k]
		myeigen=eigen(Covar_k,symmetric=TRUE)
		Vq=myeigen$vectors[,1:q]

		sigma_2=sum(myeigen$values[q+1:NC],na.rm=TRUE)/(NC-q)
		diag_q=diag(q)

		for(nq in 1:q){
			diag_q[nq,nq]=sqrt(myeigen$values[nq]-sigma_2)
		}

		Wq=Vq %*% diag_q
		sigma[ , ,k]=Wq %*% t(Wq)+(sigma_2*diag(NC))
	}
#------------------------E-Step----------------------
	for (k in 1:Nclu){
		px[ ,k]=pi[k]*dmvnorm(myX,Mu[k, ],sigma[ , ,k],log=FALSE)
	}
```

```
	for (i in 1:NR){
		for(k in 1:Nclu){
		gamma[i,k]=px[i,k]/sum(px[i, ])
		}
	}
#-----------------------Likelihood-----------------------
likeli[it]=sum(log(rowSums(px)))
print(c(it,likeli[it]))
}

#-----------------------Computer AIC---------------------
AIC= -2*likeli[30]+2*(NC*q+1-q*(q-1)/2)
likeli
AIC
# Plot likelihood VS. iter
dev.new(width=6,height=4)
par(mai=c(0.5,0.45,0.35,0.05),cex=0.8)
plot(1:30,likeli, pch=19,axes=TRUE)
title("q=6 Likelihood")

# Visulazation pictures
dev.new(width=6,height=10)
par(mai=c(0,0,0,0),cex=0.8,mfrow=c(10,6))
for(k in 1:Nclu){

image(t(matrix(Mu[k, ],byrow=TRUE,16,16)[16:1, ]),col=gray(0:256/256),axes
=FALSE)
	box( )
	for(j in 1:5){
		temp=rmvnorm(1,mean=Mu[k, ],sigma[ , ,k])

image(t(matrix(temp,byrow=TRUE,16,16)[16:1, ]),col=gray(0:256/256),axes=F
ALSE)
	box ( )
	}
}

# Accuracy Assessment
# calculate new Labels
EMLabel = matrix(0,NR,Nclu)
for(i in 1:NR){
	EMLabel[i,which.max(gamma[i,])]=1
```

```
}
# Accuracy Assessment
misRate=matrix(1,Nclu,1)
temp1=0
for(i in 1:Nclu){
temp=apply(EMLabel[myLabel==(i-1),],2,function(xx){
	return(sum(xx))
	})
misRate[i,] =1-max(temp)/sum(temp)
temp1=temp1+max(temp)
}
OverAllMisRate=1-temp1/NR

#For q=0 There is a little difference for M-step
for(k in 1:Nclu){
	mu_k=rep(0,256)
	for(n in 1:NR){
		mu_k=mu_k+gamma[n,k]*myX[n, ]
	}
	Mu[k,]=mu_k/N[k]
}

pi=colSums(gamma)/NR

for (k in 1:Nclu){
	Covar_k=matrix(0,256,256)

	for(n in 1:NR){
		Xi_Bar=myX[n, ]-Mu[k, ]
		Covk_temp=(Xi_Bar %*% t(Xi_Bar))*gamma[n,k]
		Covar_k=Covar_k+Covk_temp
	}
	Covar_k=Covar_k/N[k]
	myeigen=eigen(Covar_k,symmetric=TRUE)

	sigma_2=sum(myeigen$values[q+1:NC],na.rm=TRUE)/(NC-q)
	sigma[ , ,k]=(sigma_2*diag(NC))
}
```