[MVA]
Probabilistic graphical models
Homework 2

Maha ELBAYAD

11 November 2015

1-Entropy and mutual information

1.Let X be a discrete random variableon a finite space $\mathcal{X}$ with $|\mathcal{X}| = k$ and p the distribution of X.

(a) $H(X) = -\sum_{x\in\mathcal{X}} p(x).\log p(x) \geq 0$ and $H(x) = 0 \iff X = cst\ a.s$

We have $\forall x \in \mathcal{X}$, $0 \leq p(x) \leq 1$ thus $\forall x \in \mathcal{X}$ $\log p(x) \leq 0$ which implies $H(X) \geq 0$.

For the equality condition, if X is deterministic i.e there exits $x^* \in \mathcal{X}$ such that $p(x^*) = 1$, consequently $p(x) = 0\ \forall x \in \mathcal{X}\{x^*\}$ then $H(X) = -p(x^*)\log p(x^*) = 0$.

Inversely, if X is not constant a.s then $\exists x^* \in \mathcal{X}\ 0 < p(x^*) < 1$ which means that $H(X) > 0$. $\square$

(b) For q the uniform distribution on $\mathcal{X}$: $q(x) = \frac{1}{k}$, the Kullback Leibler Divergence between p and q is:

$$D(p||q) = \sum_{x\in\mathcal{X}} p(x)\log\frac{p(x)}{q(x)} = \sum_{x\in\mathcal{X}} p(x)\log(kp(x)) = \log k + H(X)$$

(c) Let us prove that $D(p||q) \geq 0$ for any two distributions p and q:

$$\begin{aligned} -D(p||q) &= -\sum_{x\in\mathcal{X}} p(x)\log\frac{p(x)}{q(x)} \\ &\leq \log\sum_{x\in\mathcal{X}} p(x).\frac{q(x)}{p(x)} \\ &= \log(\sum_{x\in\mathcal{X}} q(x)) = 0 \end{aligned}$$

With equality if and only if $\frac{q(x)}{p(x)} = cst$ and since $\sum q(x) = \sum p(x) = 1$ then we have equality iff $p = q$.

Hence, $H(X) = -\log k + D(p||q) \geq -\log k$.

2. We consider a pair of r.v (X_1, X_2) on $\mathcal{X}_1 \times \mathcal{X}_2$ and we denote by p_1 and p_2 the marginal distributions of X_1 and X_2 and $p_{1,2}$ their joint distribution. We define the mutual information as follows:

$$I(X_1, X_2) = \sum_{(x_1,x_2)\in\mathcal{X}_1\times\mathcal{X}_2} p_{1,2}(x_1, x_2)\log\frac{p_{1,2}(x_1, x_2)}{p_1(x_1).p_2(x_2)}$$

(a) We can write $I(X, Y)$ as $D(p_{1,2}(x_1, x_2)||p_1(x_1)p_2(x_2))$ which is nonnegative (equal to zero if and only if $p(x_1, x_2) = p(x_1).p(x_2)$)

(b) We have:

$$\begin{aligned}
I(X_1, X_2) &= \sum_{(x_1,x_2)\in\mathcal{X}_1\times\mathcal{X}_2} p_{1,2}(x_1,x_2)\log p_{1,2}(x_1,x_2) - \sum_{(x_1,x_2)\in\mathcal{X}_1\times\mathcal{X}_2} p_{1,2}(x_1,x_2)\log p(x_1) \\
&- \sum_{(x_1,x_2)\in\mathcal{X}_1\times\mathcal{X}_2} p_{1,2}(x_1,x_2)\log p_2(x_2) \\
&= -H(X,Y) - \sum_{x_1\in\mathcal{X}_1}\log p_1(x_1)\sum_{x_2\in\mathcal{X}_2} p_{1,2}(x_1,x_2) - \sum_{x_2\in\mathcal{X}_2}\log p_2(x_2)\sum_{x_1\in\mathcal{X}_1} p_{1,2}(x_1,x_2) \\
&= -H(X,Y) - \sum_{x_1\in\mathcal{X}_1} p_1(x_1)\log p_1(x_1) - \sum_{x_2\in\mathcal{X}_2} p_2(x_2)\log p_2(x_2) \\
&= -H(X,Y) + H(X) + H(Y)
\end{aligned}$$

(c) Finding the joint distribution of maximal entropy is equivalent to finding the joint distribution that minimizes the mutual information $I(X_1, X_2)$. From 2.a we deduce that the joint distribution of maximal entropy is $p_{1,2} = p_1 p_2$ i.e X_1 and X_2 are independent.

2-Conditional independence and factorizations

1. Let X,Y and Z be three random variables. Suppose $X \perp\!\!\!\perp Y|Z$ then for (y, z) such that $p(y, z) > 0$:

$$\begin{aligned}
p(x|y,z) &= \frac{p(x,y,z)}{p(y,z)} = \frac{p(x,y|z)p(z)}{p(y,z)} \\
&= \frac{p(x|z)p(y|z)p(z)}{p(y,z)} \\
&= \frac{p(x|z)p(y|z)}{p(y|z)} = p(x|z)
\end{aligned}$$

Conversely, suppose we have $(\forall(y, z)\ s.t\ p(y, z) > 0) \quad p(x|y, z) = p(x|z)$ then for a pair (y, z) such that $p(y, z) > 0$:

$$\begin{aligned}
p(x,y|z) &= \frac{p(x,y,z)}{p(z)} = \frac{p(x|y,z)p(y,z)}{p(z)} \\
&= \frac{p(x|z)p(y,z)}{p(z)} = p(x|z)p(y|z)
\end{aligned}$$

And if $p(y, z) = 0$ then $p(x, y|z) = 0 = p(y|z)$ hence $p(x, y|z) = p(x|z)p(y|z)$.

In both cases, we end up with $X \perp\!\!\!\perp Y|Z$

2. Generally for $p \in \mathcal{L}(G)$ we would have $X \not\perp\!\!\!\perp Y|T$. In fact, the only path from X to Y in the symmetrized grpah is $X \to Z \leftarrow Y$ which is unblocked since Z is a parent of T.

3-Distributions factorizing in a graph

1. Let $(i \to j)$ be a covered edge on a DAG $G = (V, E)$ i.e $\pi_j = \pi_i \cup \{i\}$ and let us consider the grpah $G' = (V, E')$ where $E' = E\backslash\{i \to j\} \cup \{j \to i\}$.

We have $\pi_i^{G'} = \pi_i \cup \{j\}$ and $\pi_j^{G'} = \pi_i$. And if G' is a DAG then:

$$
\begin{aligned}
p \in \mathcal{L}(G) &\iff p(x) = \prod_{k \in V} p(x_k|x_{\pi_k}) = \prod_{k \in V \setminus \{i,j\}} p(x_k|x_{\pi_k}) p(x_i|x_{\pi_i}) p(x_j|x_{\pi_j}) \\
&\iff p(x) = \prod_{k \in V \setminus \{i,j\}} p(x_k|x_{\pi_k}) . \frac{p(x_i, x_{\pi_i})}{p(x_{\pi_i})} \frac{p(x_j, x_i, x_{\pi_i})}{p(x_i, x_{\pi_i})} \\
&\iff p(x) = \prod_{k \in V \setminus \{i,j\}} p(x_k|x_{\pi_k}) . \frac{p(x_j, x_i, x_{\pi_i})}{p(x_{\pi_i})} \\
&\iff p(x) = \prod_{k \in V \setminus \{i,j\}} p(x_k|x_{\pi_k}) . \frac{p(x_j, x_i, x_{\pi_i})}{p(x_{\pi_i})} \frac{p(x_j, x_{\pi_i})}{p(x_j, x_{\pi_i})} \\
&\iff p(x) = \prod_{k \in V \setminus \{i,j\}} p(x_k|x_{\pi_k}) . p(x_i|x_{\pi_i \cup j}) p(x_j|x_{\pi_i}) \\
&\iff p \in \mathcal{L}(G')
\end{aligned}
$$

2. Let G be a directed tree and G' its corresponding undirected tree. Since G is a tree then it's a DAG that doesn't contain any V-structure.

G' is equivalent to the symmetrized tree $\widetilde{G} = (V, \widetilde{E})$ where $\widetilde{E} = \{(u,v), (v,u) \,|(u,v) \in E\}$

From (Proposition 4.22 - Lecture 4) we conclude that $\mathcal{L}(G) = \mathcal{L}(\widetilde{G}) = \mathcal{L}(G')$

4-Implementation - Gaussian mixtures

(a)

```
kdata=as.matrix(read.table("data/EMGaussian.data",header=F,sep=' ',col.names=c('X','Y')));
kmeans<- function(X,K,max_iter=300,tol=1e-5){
  #X: data sample.
  #K: Number of clusters
  #max_iter: maximum number of iterations.
  #tol: Convergence tolerance.

  #Initialization of the clusers centers:
  centers=X[sample(nrow(X), K), ]
  counts=rep(0,K)
  sd=rep(0,K)
  initials=centers
  distances=matrix(rep(0,nrow(X)*K),nc=K)
  iter=1
  delta=+Inf
  distortion=rep(NaN,max_iter)
  while(iter < max_iter && delta>tol){
  #Affectation
  for (c in 1:K){
    distances[,c]=apply(X,1,function(x) sum((x-centers[c,])^2))
  }
  affect=apply( distances, 1, which.min)
  #Distortion measure:
  distortion[iter]=0
  for (c in 1:K){
```

```
   distortion[iter]=distortion[iter]+sum(distances[,c]*(affect==c))
  }
  if(iter>1) delta=distortion[iter-1]-distortion[iter]
  #update centers:
  for (c in 1:K){
    centers[c,]=colMeans(kdata[affect==c,])

  }
  iter=iter+1
  }
  for (c in 1:K) counts[c]=sum(affect==c)
  list(initials=initials,affect=affect,centers=centers,distortion=distortion,cv=iter-1,counts=counts)
}
```

Performance of k-means:

For two different runs of kmeans yielding different results i.e converging to different local minimas of the distortion, we list the initial vs the final centroids and the distortion evolution until convergence. We then plot the different clustering results.

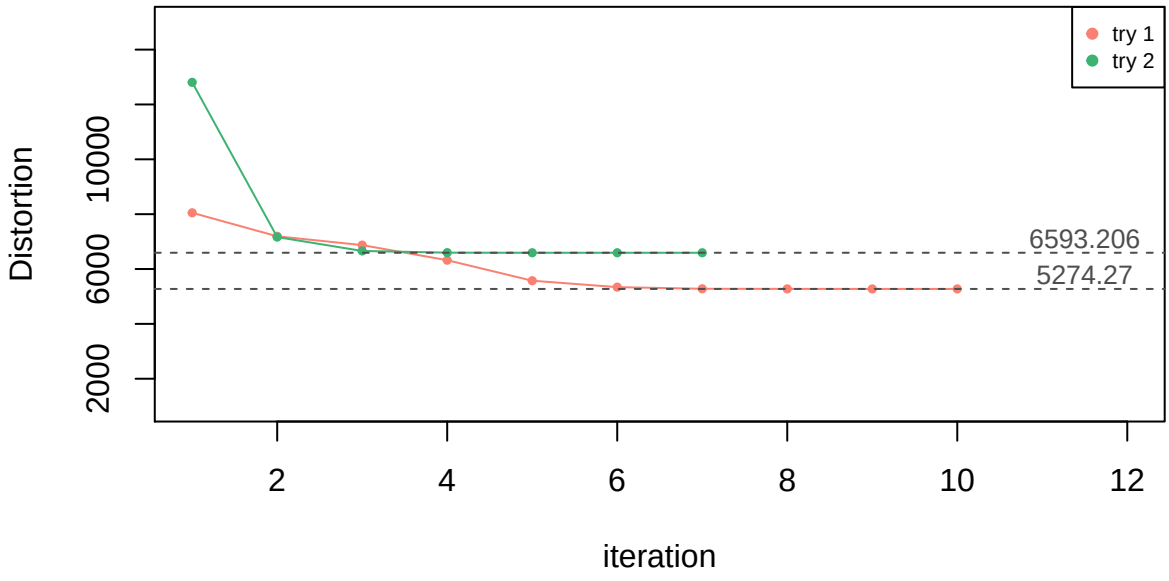
	X	Y	X	Y
1	-1.87	1.80	0.44	4.93
2	3.72	2.04	3.38	-2.16
3	-5.46	-5.43	-3.80	-4.13

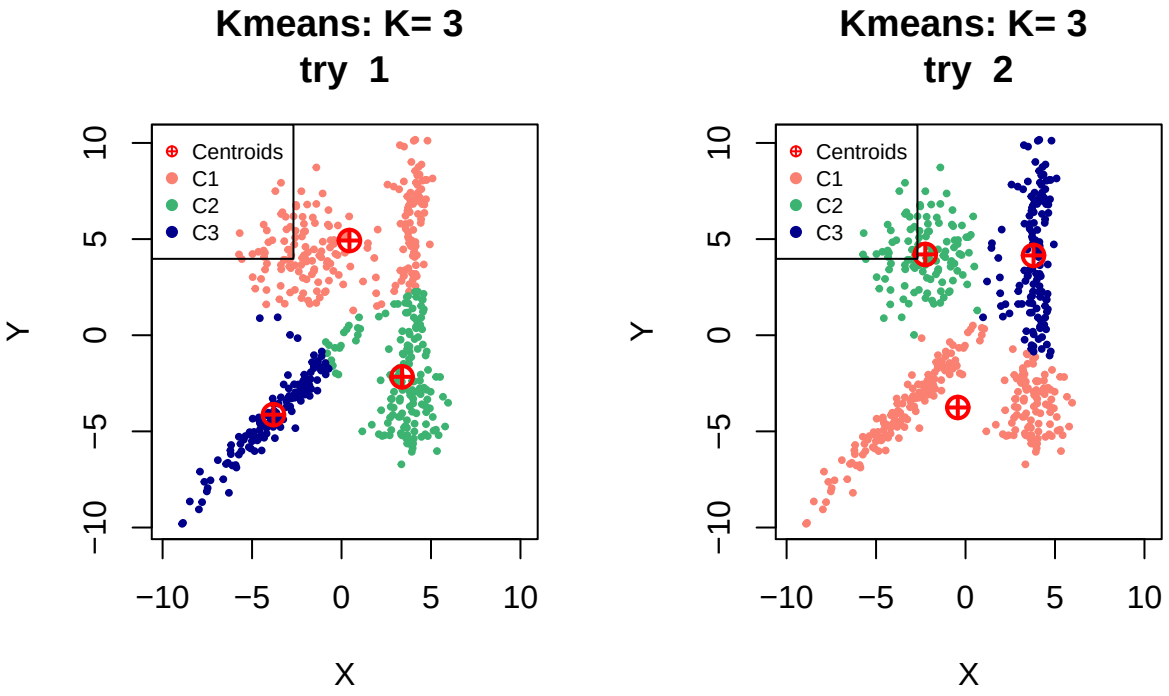
(a) Try 1

	X	Y	X	Y
1	-2.79	-3.28	-0.43	-3.76
2	-2.99	4.45	-2.25	4.20
3	-2.25	3.66	3.80	4.15

(b) Try 2

Table 1: Initial || final centroids





(b) EM-algrithm for Isotropic Gaussian mixture

We consider a Gaussian mixture model in which the covariance matrices are proportional to the identity i.e $\Sigma_k = \sigma_k^2.I_d$.

Let $(x_i, z_i)_{1\leq i\leq n}$ be a sample with $x_i \in \mathbb{R}^d$, $z_i \sim \mathcal{M}(1, p_1, ...p_K)$ and $(x_i|z_i) \sim \mathcal{N}(\mu_j, \Sigma_j = \sigma_j^2 I_p)$

Using the summation rule and Bayes forumla we can write:

$$p_\theta(z_i = k|x_i) = \frac{p_k f_k(x_i)}{\sum_{k'} p_{k'} f_{k'}(x_i)} = \tau_i^k(\theta)$$

with:

$$\begin{aligned} f_k(x) &= \frac{1}{|\Sigma_k|^{1/2}(2\pi)^{d/2}} \exp\left[-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\right] \\ &= \frac{1}{(2\pi\sigma_k^2)^{d/2}} \exp\left[-\frac{1}{2\sigma_k^2}(x-\mu_k)^T(x-\mu_k)\right] \end{aligned}$$

and $\theta = (p_1, ..., p_K, \mu_1, ..., \mu_K, \sigma_1^2, ..., \sigma_K^2)$.

Our objective is to maximize the complete likelihood at iteration t:

$$\begin{aligned} \ln p_\theta(X, Z) &= \sum_{i=1}^{n} \log p_\theta(x_i, z_i) \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K} \left(z_i^k \log(p_{k,t}) + z_i^k \log f_{k,t}(x_i)\right) \end{aligned}$$

With the latent variables $z_i^k = 1$ if $z_i = j$ and 0 otherwise.

After the E-step we end up substituting z_i^k by its expectation τ_i^k.

$$l(\theta) = \sum_{i=1}^{n}\sum_{k=1}^{K}\left(\tau_i^k \log(p_k) + \tau_i^k\left(-\frac{d}{2}\log(2\pi\sigma_k^2) - \frac{1}{2\sigma_k^2}(x_i-\mu_k)^T(x_i-\mu_k)\right)\right)$$

Thus we'll solve the optimization problem:

$$\begin{aligned} \underset{\theta}{\text{maximize}} \quad & l(\theta) \\ \text{subject to} \quad & \sum_{k=1}^{K} p_k = 1 \\ & \sum_{k=1}^{K} \tau_i^k = 1 \; \forall i \in \{1,..,n\} \end{aligned}$$

The optimiality condition w.r.t $\mathbf{p} = (p_1, ...p_K)$ gives:

$$\frac{\partial}{\partial \mathbf{p}}\left(l(\theta) + \lambda(1 - \sum_{k=1}^{K} p_k)\right) = 0 = \sum_{i=1}^{n}\frac{\tau_i^k}{p_k} - \lambda$$

Where λ is a Lagrange multiplier.

Summing over k yields $\lambda = n$ and

$$p_k = \frac{1}{n}\sum_{i=1}^{n}\tau_i^k$$

Next, we maximize with regard to μ_k for each k:

$$\frac{\partial l(\theta)}{\partial \mu_k} = 0 \propto \sum_{i=1}^{n}\tau_i^k(x_i - \mu_k)$$

Hence:

$$\mu_k = \frac{\sum_i \tau_i^k x_i}{\sum_i \tau_i^k}$$

Similarly for σ_k:

$$\frac{\partial l(\theta)}{\partial \sigma_k^2} = 0 \propto \sum_{i=1}^{n}\tau_i^k(-d\sigma_k^2 + (x_i-\mu_k)^T(x_i-\mu_k))$$

Hence:

$$\sigma_k^2 = \frac{\sum_i \tau_i^k (x_i-\mu_k)^T(x_i-\mu_k)}{d\sum_i \tau_i^k}$$

R-implementation:

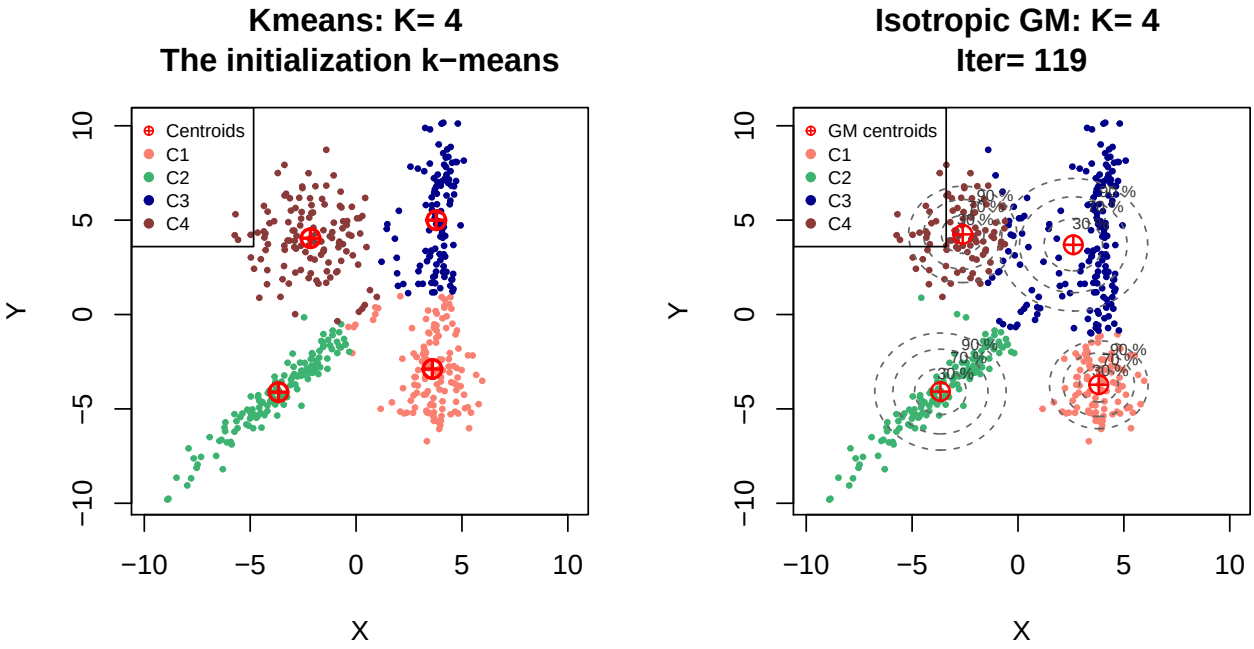
```
iso_gaussian<-function(x,mu,sigma){1/(sqrt(2*pi)*sigma)^length(x)*
    exp(-1/2/sigma^2*t(x-mu)%*%(x-mu))}
EMG_iso<-function(X,K,max_iter=300,tol=1e-5){
  #X: data sample -  K: Number of gaussians - max_iter: maximum number of EM iterations.
  #tol: Convergence tolerance.

  #Initialization with kmeans:
  km=kmeans(X,K)
  p=km$counts/nrow(X)
  mu=km$centers
```

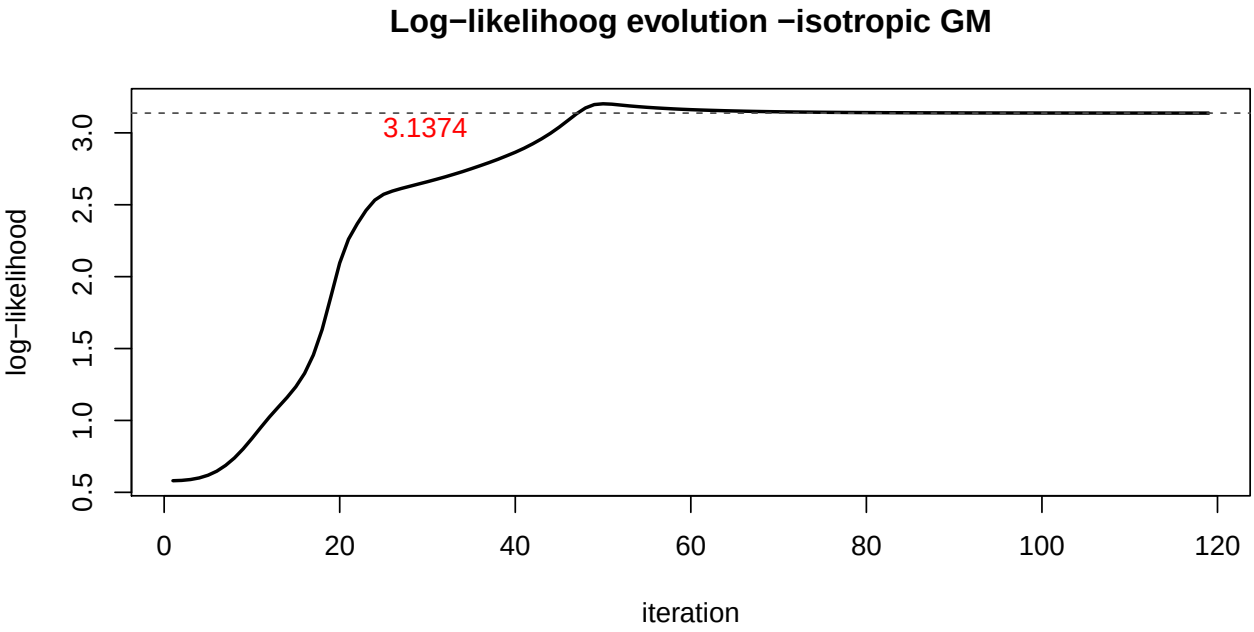
```
  sigma=rep(max(var(X)),K) #Initializing with a large variance

  n=nrow(X)
  d=ncol(X)
  tau=matrix(nrow=K, ncol=n)

  #Convergence criteria
  ll=rep(0,max_iter)
  delta=+Inf
  #EM loop
  iter=1
  while(iter<max_iter && delta>tol){
    #(E-step):
    for(k in 1:K){
      tau[k,]=as.matrix(apply(X,1,function(x) p[k]*iso_gaussian(x,mu[k,],sigma[k])))
    }
    tau=t(t(tau)/colSums(tau))
    #(M-step):
    p=rowMeans(tau)
    mu=(tau%*%X)/rowSums(tau)
    for(k in 1:K){
      temp=rowSums(t(apply(X,1,function(x) x-mu[k,]))^2)
      sigma[k]=sqrt(sum(tau[k,]*temp)/sum(tau[k,])/d)
    }

    affect=apply(tau,2,which.max)
    #Compute the log-likelihood:
    for (k in 1:K) {
      if(length(X[affect==k,])<d)
        next
      else{
        if(length(X[affect==k,])==d)
          ll[iter]=ll[iter]+p[k]*iso_gaussian(X[affect==k,],mu[k,],sigma[k])
        else
        ll[iter]=ll[iter]+sum(as.matrix(apply(X[affect==k,],1,function(x)
        p[k]*iso_gaussian(x,mu[k,],sigma[k]))))
      }
    }
    if(iter>1)
      delta=abs(ll[iter]-ll[iter-1])
    else
      delta=+Inf
    iter=iter+1
    }
  if(iter>=max_iter) warning('EM algorithm didn\'t converge')
  list(p=p,sigma=sigma,mu=mu,likelihood=ll[1:iter-1],km=km,iter=iter-1,affect=affect)
}
```

EM for Gaussian mixture results with K=4

```
## $p
## [1] 0.1952437 0.2681052 0.3675726 0.1690785
##
```

```
## $sigma
## [1] 1.177751 2.088375 2.677791 1.415462
##
## $mu
##               X         Y
## [1,]  3.815144 -3.718847
## [2,] -3.662331 -4.081966
## [3,]  2.612314  3.696034
## [4,] -2.609472  4.246807
```





(c) EM-algrithm for General Gaussian mixture

Following the same steps as in (b) we end up with similar results except for:

$$\Sigma_k = \frac{\sum_i \tau_i^k (x_i - \mu_k)(x_i - \mu_k)^T}{\sum_i \tau_i^k}$$

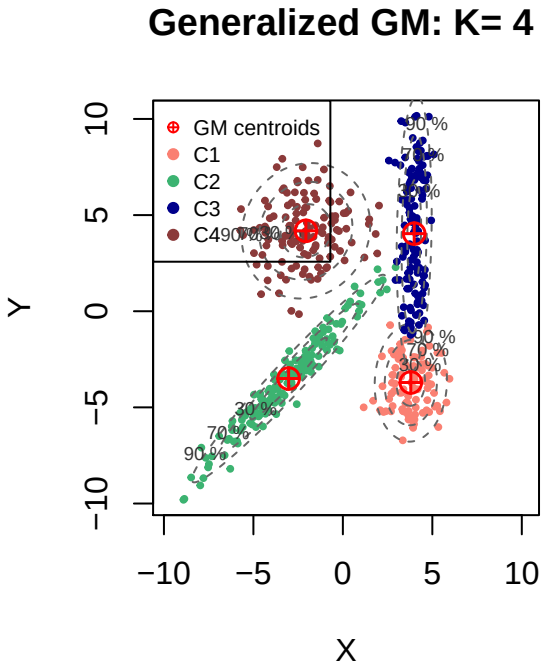
R-implementation:

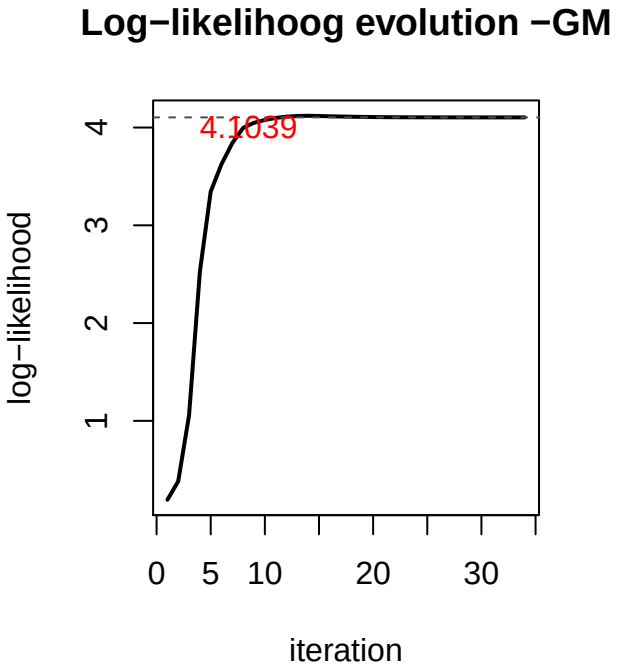
```
gaussian<-function(x,mu,sigma){1/(sqrt(2*pi*det(sigma)))^length(x)*
    exp(-1/2*t(x-mu)%*%ginv(sigma)%*%(x-mu))}
EMG<-function(X,K,max_iter=300,tol=1e-5){
  #X: data sample -  K: Number of gaussians - max_iter: maximum number of EM iterations.
  #tol: Convergence tolerance.

  #Initialization with kmeans:
  km=kmeans(X,K)
  p=km$counts/nrow(X)
  mu=km$centers
  sigma=list()
  for(k in 1:K)
    sigma[[k]]=var(X) #Initializing with a large variance

  n=nrow(X)
  d=ncol(X)
  tau=matrix(nrow=K, ncol=n)
  ll=rep(0,max_iter)

  #Convergence criteria
  delta=+Inf
  #EM loop
  iter=1
  while(iter<max_iter && delta>tol){
    #(E-step):
    for(k in 1:K){
      tau[k,]=as.matrix(apply(X,1,function(x) p[k]*gaussian(x,mu[k,],sigma[[k]])))
    }
    tau=t(t(tau)/colSums(tau))
    #(M-step):
    p=rowMeans(tau)
    mu=(tau%*%X)/rowSums(tau)
    for(k in 1:K){
      temp=t(apply(X,1,function(x) x-mu[k,]))
      sigma[[k]]=(t(temp)%*%(tau[k,]*temp))/sum(tau[k,])
    }
    affect=apply(tau,2,which.max)
    #Compute the log-likelihood:
    for (k in 1:K) {
      ll[iter]=ll[iter]+sum(as.matrix(apply(X[affect==k,],1,function(x)
        p[k]*gaussian(x,mu[k,],sigma[[k]]))))
      }
    if(iter>1)
      delta=abs(ll[iter]-ll[iter-1])
    else
```

```
      delta=+Inf
    iter=iter+1

    }
  if(iter>=max_iter) warning('EM algorithm didn\'t converge')
  list(p=p,sigma=sigma,mu=mu,likelihood=ll[1:iter-1],km=km,iter=iter-1,affect=affect)
}
```

EM for Gaussian mixture results with K=4

```
## $p
## [1] 0.1934962 0.3075661 0.2494631 0.2494746
##
## $sigma
## $sigma[[1]]
##           X         Y
## X 0.8855149 0.0539682
## Y 0.0539682 2.0524554
##
## $sigma[[2]]
##          X        Y
## X 6.360526 6.180829
## Y 6.180829 6.327390
##
## $sigma[[3]]
##           X          Y
## X 0.2081201  0.2281152
## Y 0.2281152 11.1108583
##
## $sigma[[4]]
##           X         Y
## X 2.8042065 0.2219266
## Y 0.2219266 2.6946582
##
##
## $mu
##              X         Y
## [1,]  3.798018 -3.707179
## [2,] -3.030866 -3.500587
## [3,]  3.986098  4.035388
## [4,] -2.064711  4.182521
```





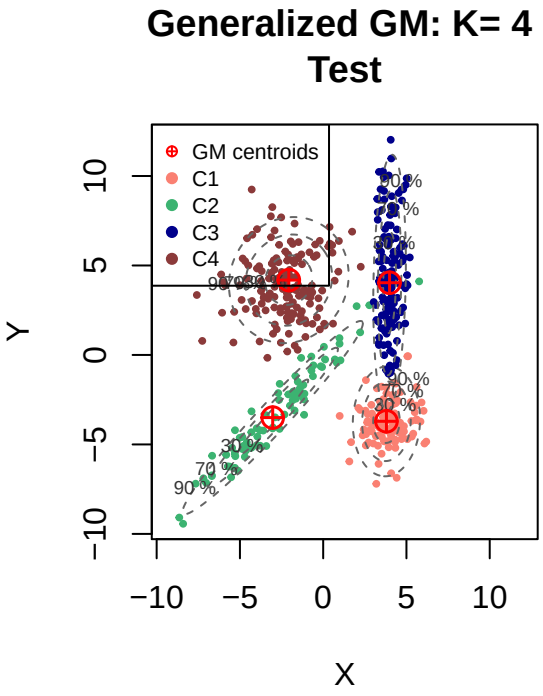
(d) Comparison of the mixture models:

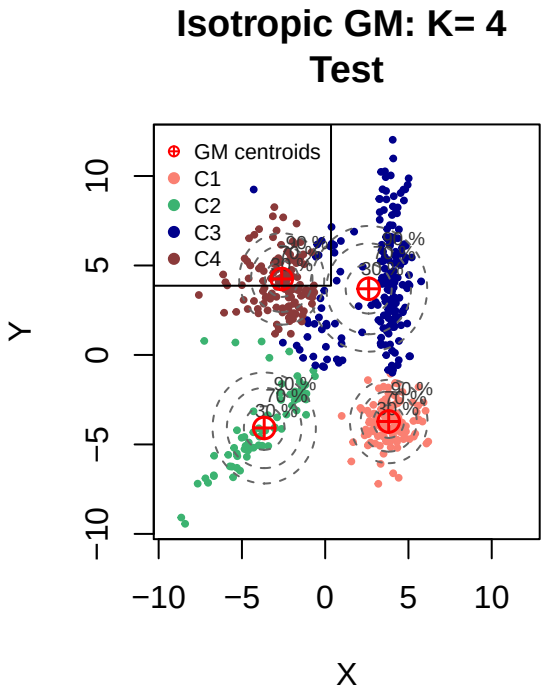
We compute the log-likelihood of the two mixture models (K=4) on both the training set and the test set:

	Train	Test
Isotropic GM	3.137	3.479
GM	4.104	3.694

Table 2: Log-likelihoods - Train & Test for the GM models

We note that the isotropic GM has low likelihood compared to the generalized GM. In fact with few degrees of freedom, the model couldn't capture the data very well. On the other hand the generalized GM is slighltly overfitting the training set especially when clustering with low K (K<5) since the ellipsoids will have eccentricities closer to 1 in order to cover the training points.


Generalized GM: K= 4
Test
GM centroids
C1
C2
C3
C4
90 %
70 %
30 %
Y
X
−10
−5
0
5
10


Isotropic GM: K= 4
Test
GM centroids
C1
C2
C3
C4
90 %
70 %
30 %
Y
X
−10
−5
0
5
10