Statistical Machine Learning - Coursework II

Student Name: Harleen Gulati , Student Number: 2101550

2024-02-15

Question 1

Question 1 a

If we are fitting the model with the EM algorithm we need to maximize $\sum_{i\in[N]} \int q_i(z_i)\log(\frac{p_\theta(x_i,z_i)}{q_i(z_i)})dz_i\ (*)$ as discussed in the lecture.

The maximum is achieved by taking

$$\begin{aligned} & q_i(z_i) \\ & = p_\theta(z_i|x_i) \\ & = \frac{\pi_{z_i} N(x_i|\mu_{z_i},\sigma^2)}{\sum_{k\in[K]} \pi_k N(x_i|\mu_k,\sigma^2)}\ (**) \end{aligned}$$

as also discussed in the lectures.

Note for $z_i = k$ we can rewrite (**) as w_{ik} (i.e how responsible is cluster k for component i)

Now

$$\begin{aligned} & \log\ p_\theta(x_i,z_i) \\ & = \log\ [\pi_k N(x_i|\mu_k,\sigma^2)] \\ & = \log \pi_k - \frac{||x_i-\mu_k||^2}{2\sigma^2} \end{aligned}$$

(because $N(x_i|\mu_k,\sigma^2)$ has probability distribution $\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{||x_i-\mu_k||^2}{2\sigma^2}}\ \alpha\ e^{-\frac{||x_i-\mu_k||^2}{2\sigma^2}}$ and taking logs gives $-\frac{||x_i-\mu_k||^2}{2\sigma^2}$)

Thus maximizing (*) gives

$$\begin{aligned} & \sum_{i\in[N]}\sum_{k\in[K]} w_{ik}[\log p_\theta(x_i,z_i) - \log q_i(z_i)] \\ = & \sum_{i\in[N]}\sum_{k\in[K]} w_{ik}[\log \pi_k - \frac{||x_i-\mu_k||^2}{2\sigma^2} - \log w_{ik}] \\ = & \sum_{i\in[N]}\sum_{k\in[K]} w_{ik}[\log \pi_k - \log w_{ik} - \frac{||x_i-\mu_k||^2}{2\sigma^2}] \end{aligned}$$

as required.

Note that W has entries for w_{ik} for $i \in [N]$ and $k \in [K]$ where the ith row and kth column represent how much of datapoint i comes from cluster k. Thus W should satisfy for each row i $\sum_{k\in[K]} w_{ik} = 1.$

Question 1 b

For each $k \in [K]$ differentiate $Obj_K(W, \pi, \mu; X, \sigma)$ with respect to π_k and set the derivative equal to 0 giving us:

$\frac{d}{d\pi_k} \sum_{i\in[N]} w_{ik}[\log \pi_k - \log w_{ik} - \frac{||x_i - \mu_k||^2}{2\sigma^2}] = \sum_{i\in[N]} \frac{w_{ik}}{\pi_k}$ $(*)$ (note k is a fixed quantity).

Setting the derivative to 0 would result in $\sum_{i\in[N]} w_{ik} = 0$ (i.e, component k is not responsible for any observations => $\pi_k = 0$ but $\pi_k = 0$ would result in (*) being undefined).

Thus to maximize (*) we focus on minimizing π_k. We know $\sum_{k\in[K]} = 1 \rightarrow \sum_{i\in[N]} \sum_{k\in[K]} = N = \sum_{i\in[N]} \sum_{k\in[K]} w_{ik}$.

For each $k \in [K]$ differentiate $Obj_K(W, \pi, \mu; X, \sigma)$ with respect to μ_k and set the derivative equal to 0:

We use the fact that

$$\begin{aligned} &\frac{d}{d\mu_k}||x_i - \mu_k||^2 \\ &= \frac{d}{d\mu_k}(x_i - \mu_k)(x_i - \mu_k)^T \\ &= \frac{d}{d\mu_k}\ (x_i x_i^T - 2x_i\mu_k + \mu_k\mu_k^T) \\ &= -2x_i + 2\mu_k \end{aligned}$$

Thus

$$\begin{aligned} &\frac{d}{d\mu_k}\sum_{i\in[N]} w_{ik}[\log \pi_k - \log w_{ik} - \frac{||x_i - \mu_k||^2}{2\sigma^2}] \\ &= \sum_{i\in[N]} w_{ik}(\frac{2x_i - 2\mu_k}{2\sigma^2}) \\ &= \sum_{i\in[N]} w_{ik}(\frac{x_i - \mu_k}{\sigma^2})\ (*) \end{aligned}$$

Setting (*) = 0 gives

$$\begin{aligned} &\sum_{i\in[N]} w_{ik}x_i - \sum_{i\in[N]} w_{ik}\mu_k = 0 \\ &\implies \sum_{i\in[N]} w_{ik}\mu_k = \sum_{i\in[N]} w_{ik}x_i \\ &\implies \mu_k = \frac{\sum_{i\in[N]} w_{ik}x_i}{\sum_{i\in[N]} w_{ik}}\ (**) \end{aligned}$$

Note $\frac{d^2}{d^2\mu_k} = \sum_{i\in[N]} \frac{-w_{ik}}{\sigma^2} < 0$ (because $w_{ik} \geq 0, \sigma^2 > 0$ for all $i \in [N]$ and $k \in [K]$) hence (**) is indeed a maximiser.

Question 1 c

Differentiate $Obj_K(W, \pi, \mu; X, \sigma)$ for each $i \in [N]$ and $k \in [K]$ with respect to w_{ik} and set the derivative equal to 0.

Before however note $\frac{d}{dw_{ik}}\ w_{ik}\log w_{ik} = \log w_{ik} + \frac{w_{ik}}{w_{ik}}$ (using the chain rule - call this result I)

So, for each $i \in [N]$ and $k \in [K]$ we do the following:

$$\frac{d}{dw_{ik}}\ w_{ik}[\log \pi_k - \log w_{ik} - \frac{||x_i - \mu_k||^2}{2\sigma^2}]$$
$$= \log \pi_k - \log w_{ik} - \frac{w_{ik}}{w_{ik}} - \frac{||x_i - \mu_k||^2}{2\sigma^2}\ (*)$$

(using result I)

Setting (*) equal to 0 gives

$$\log w_{ik} = \log \pi_k - \frac{||x_i - \mu_k||^2}{2\sigma^2} - 1$$
$$\implies w_{ik} = e^{\log \pi_k - \frac{||x_i - \mu_k||^2}{2\sigma^2} - 1}$$
$$\implies w_{ik} = \frac{\pi_k\ e^{-\frac{||x_i - \mu_k||^2}{2\sigma^2}}}{e^1}$$

We know $\sum_{l\in[K]} w_{ik} = 1$ thus

$$1 = \sum_{l\in[K]} \frac{\pi_l\ e^{-\frac{||x_i - \mu_l||^2}{2\sigma^2}}}{e^1}$$
$$\implies 1 = \frac{1}{e^1} \sum_{l\in[K]} \pi_l\ e^{-\frac{||x_i - \mu_l||^2}{2\sigma^2}}$$
$$\implies e^1 = \sum_{l\in[K]} \pi_l\ e^{-\frac{||x_i - \mu_l||^2}{2\sigma^2}}$$

And thus we deduce $w_{ik} = \frac{\pi_k\ e^{-\frac{||x_i - \mu_k||^2}{2\sigma^2}}}{\sum_{l\in[K]}\ \pi_l\ e^{-\frac{||x_i - \mu_l||^2}{2\sigma^2}}} = \frac{\pi_k\ exp(-\frac{||x_i - \mu_k||^2}{2\sigma^2})}{\sum_{l\in[K]}\ \pi_l\ \exp(-\frac{||x_i - \mu_l||^2}{2\sigma^2})}$

as required.

Question 2

Preliminary Tasks

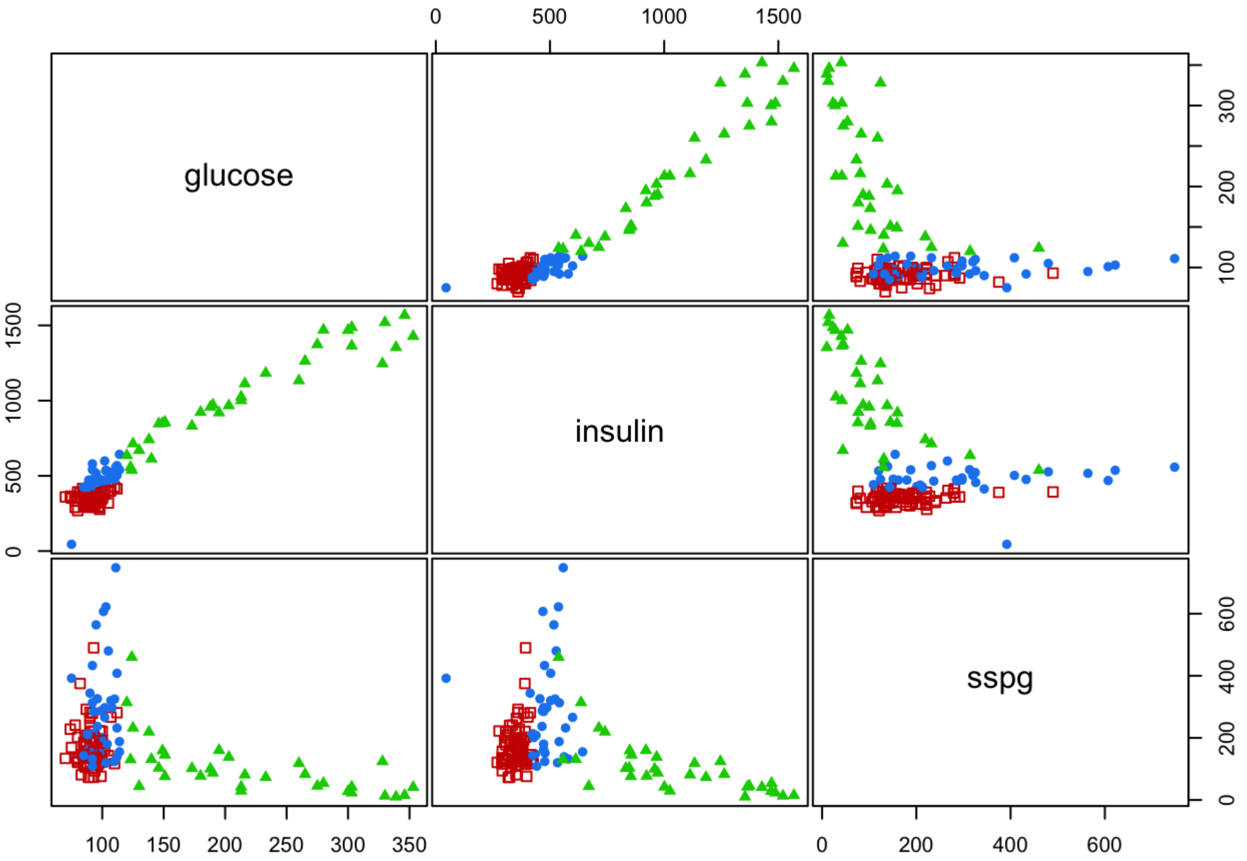
```
library(mclust)
```

```
## Package 'mclust' version 6.0.1
## Type 'citation("mclust")' for citing this R package in publications.
```

```
data("diabetes")
class <- diabetes$class
```

```
X <- diabetes[,-1]
```

```
clPairs(X, class)
```



Question 2 i

Fitting a homogeneous, spherical Guassian mixture model to the data would imply we assume each component to have the same variance and zero covariance between each dimension.

However, we observe that for example glucose and insulin have a positive correlation suggesting there is a positive covariance between them. We also observe that glucose and sspg have a negative correlation suggesting a negative covariance between them and similarly we see insulin and sspg to have a negative correlation, suggesting a negative covariance between them.

Thus, fitting a model which assumes zero covariance between each dimension may result in a poor fitting, as we observe there to be non-zero covariance between each dimension which such a model may be unable to capture.

Question 2 Model Fitting

```
sphgmm <- Mclust(X, G=3, modelNames="EII")
```

```
inhomsphgmm <- Mclust(X, G=3, modelNames="VII")
```

```
gengmm <- Mclust(X, G=3, modelNames="VVV")
```

Question 2 ii

We observe the general GMM to give the largest Rand index; the flexible spherical GMM has a similar yet smaller Rand index to the general GMM and the homogeneous spherical GMM has the smallest Rand index.

We can see that an accurate clustering may require an accurate model of the full data-generating process. This is because, an accurate model of the full data-generating process will inform us of correlations between features (e.g., a positive correlation between glucose and insulin). Having this knowledge can then allow us to choose a clustering appropriately (e.g., if we are aware of correlations between dimensions we would choose a clustering which takes this into consideration such as the general GMM). Thus, we can use the knowledge of the data-generating process to pick a clustering that will capture the patterns of the data well. If however we do not have an accurate model of the full data-generating process, we may instead choose a simpler clustering (e.g., spherical GMM) which would not be able to capture the correlations between glucose, sspg and insulin as such a model assumes zero covariance between dimensions, thus such a clustering may not be fully accurate.

```
paste("The Rand index for the spherical GMM is : " , adjustedRandIndex(class, sphgmm$classification))
```

```
## [1] "The Rand index for the spherical GMM is :  0.401870349906684"
```

```
paste("The Rand index for the inhomosphgmm GMM is : " , adjustedRandIndex(class, inhomsphgmm$classification))
```

```
## [1] "The Rand index for the inhomosphgmm GMM is :  0.635508347105214"
```

```
paste("The Rand index for the general GMM is : " , adjustedRandIndex(class, gengmm$classification))
```

```
## [1] "The Rand index for the general GMM is :  0.664018139236871"
```

Question 2 iii

For each of the three models used above, we have 3 clusters and 3 dimensions.

Consider the homogeneous spherical GMM:

The parameters required to define the model consist of:

- The mean of each cluster : in this model, each cluster has a different mean. Since the data is 3 dimensional, each cluster has a 3 dimensional mean and thus the number of parameters required to define the mean of each cluster is 3 and thus the number of parameters required to define the mean of all clusters is 3*3 = 9 (since there are three means - one corresponding to each cluster).
- The covariance matrix : This matrix has the variance of each cluster along the diagonals and the covariance between dimensions along non-diagonals. However for such a model we assume the covariance between dimensions to be 0 and we assume each cluster to have the same variance. Thus, we'd only need 3 parameters for the covariance matrix (which is the variance of all the clusters which is 3 dimensional as the data is 3 dimensional).
- The mixture coefficients : Since there are 3 clusters, there are 3 mixture coefficients however the mixture coefficients sum to 1, thus we only need 2 parameters to define the mixture coefficients.

Thus the number of parameters required to define the homogeneous spherical GMM is 9 + 3 + 2 = 14.

Now consider the flexible spherical GMM and doing similar:

The parameters required to define the model consist of:

- The mean of each cluster : like with the homogeneous spherical GMM, each cluster has a different mean thus as discussed we require 9 parameters
- The covariance matrix : here we still assume there to be 0 covariance between dimensions, however we now assume each cluster to have its own variance. Thus, we have 3 variances and each variance is 3 dimensional, hence we require 3*3 = 9 parameters to define the covariance matrix
- The mixture coefficients : same as before, we need 2 parameters to define the mixture coefficients.

Thus, the number of parameters required to define the flexible spherical GMM is : 9 + 9 + 2 = 20

Consider the general GMM:

The parameters required to define the model consist of:

- The mean of each cluster : each cluster has a different mean, thus 9 parameters are required.
- The covariance matrix : each cluster now assumes non-zero covariance between dimensions and different variances of each cluster. Thus, we require 3!/2 = 3 parameters to define the covariances between dimensions (because the covariances are scalar values and the covariances are symmetric). For the variances, we require 9 parameters as discussed previously.
- The mixture coefficients: 2 parameters.

Thus, the number of parameters required to define the general GMM is: 9 + 3 + 9 + 2 = 23.

For $p >> 3$ dimensional data and $K >> 3$ clusters:

- All three models have a different mean for each cluster, thus the number of parameters required to define the mean would be K*p (K different means, with each mean being p dimensional).
- The homogeneous spherical GMM assumes 0 covariance between dimensions and the same variance thus the number of parameters required to define the covariance matrix for this model would be p (a single variance for each cluster K which is p dimensional).
- The flexible spherical GMM also assumes 0 covariance between dimensions however assumes different variances for each cluster, thus the number of parameters required to define the variance would be K*p (K different variances, with each variance being p dimensional).
- The general GMM assumes non-zero covariance between dimensions as well as different variances for each cluster. Since there are p dimensions, we require p!/2 parameters to define the covariance between dimensions (the covariance between dimensions is a scalar value and the covariance matrix is symmetric) and K*p parameters for the variances. Thus, we require p!/2 + Kp parameters to define the covariance matrix.
- For the mixture coefficients, we require K-1 parameters to define them (since there are K clusters, so K mixture coefficients however the mixture coefficients sum to 1 so we only need to specify K-1 mixture coefficients).

Thus, the number of parameters required to define the homogeneous spherical GMM is : $Kp + p + (K - 1)$

For the flexible spherical GMM is : $Kp + Kp + (K - 1) = 2Kp + (K - 1)$

For the general GMM is : $Kp + Kp + p!/2 + (K - 1) = 2Kp + (K - 1) + p!/2$

Thus for $p >> 3$ and $K >> 3$ we see the model which requires the most parameters to specify is the general GMM.

Question 3

Question 3 i

A key issue with the approach outlined above is that the approach may always favor a larger number of clusters. This is because for a large K, the model may over fit to the training data, which is an issue as the model will then struggle to generalize to new unseen data. This over-fitting can occur since with a large K, the model is more prone to capture noise and variations in the data as distinct clusters. If the model focuses on minimizing say the following loss function: $\sum_{k\in[K]}\sum_{i\in[N]}||x_i-\mu_k||_2^2$ where μ_k is the mean of each cluster k then if K is large, we may have clusters with mean values very close to our data points and any minor change in the data points may result in the data point being assigned to a different cluster. In essence, if we have a larger K we have more μ_k parameters and thus we can be more granular in our assigning of data-points to clusters.

Another issue with the approach is that in the EM algorithm, the parameter values need to be initialized. This initialization can have a huge impact on the optimal clustering achieved in the end because this initialization is used to find the first value of w_{ik} which depends on the initial values we choose for π_k and μ_k for all $k \in [K]$ and $i \in [N]$. If for each different value of K, we choose a different initialization each time, the final clustering we achieve will depend on more than the number of clusters we chose to begin with (it will also depend on the initialization chosen).

Question 3 ii

A strategy for overcoming the over-fitting issue is to add a penalty to the loss function we use to test the performance of our clusters (e.g., this loss function could be $\sum_{k\in[K]}\sum_{i\in[N]}||x_i-\mu_k||_2^2+2K$ where μk is the mean of each cluster k). Adding such a penalty (e.g., as in AIC the penalty could be 2k) will allow us to provide an indirect estimate of the test error, by making an adjustment to the training error which accounts for the bias due to overfitting. This will then prevent overfitting to the observed data.

To counteract the second issue we mentioned of initial initialization, we could simply fix the initial parameter values μ_k and π_k for all $k \in [K]$ for each different K, so that the final clustering only depends on the number of clusters K rather than the initial initialization.