CSC311
Assignment 01 (Non-programming)
Ye Huang

Q1(e).

let $a = X - [1]X\frac{1}{N}$ (Where $[1] \in R^{N\times N}$ is the square identity matrix)

$= X - S\frac{1}{N}$ (where $S \in R^{N\times D}$ such that $S_{ij} = \sum_{j=1}^{N} X_{ij}$ for all rows in S, in other words, S is the matrix where every row is $\mu * N$)

$= X - M$ (Where $M \in R^{N\times D}$ such that $M^{(i)} = \mu$ for all row i in M)

Then,

$\frac{1}{N}aa^T$

$= \frac{1}{N}(X-M)^T(X-M)$ (construction of a)

$= \frac{1}{N}[(X-M)(X-M)]_{ij}$ (matrix multiplication)

$= \frac{1}{N}\sum_k (X-M)_{ik}(X-M)_{jk}$ (definition of matrix multiplication)

$= \frac{1}{N}\sum_{i=1}^{N}(X^{(i)}-\mu)(X^{(i)}-\mu)^T$ (by construction of M)

$= \Sigma$

To prove the covariance matrix is symmetric,
Consider, 2 features i and j in our model,

$\Sigma_{ij} = cov(i,j)$ (construction of covariance matrix)

$= \frac{\sum_{k=1}^{M}(i_k-\hat{i})(\sum_{k=1}^{M}(j_k-\hat{j}))}{N}$ (definition of covariance)

$= \frac{\sum_{k=1}^{M}(j_k-\hat{j})(\sum_{k=1}^{M}(i_k-\hat{i}))}{N}$ (multiplication is commutative)

$= cov(j,i)$ (definition of covariance)

$= \Sigma_{ji}$ (construction of covariance matrix)

Hence, the covariance matrix is symmetric.

Q2(c):

The regularization parameter overfits when n= 20 to 10 (when $reg_param = 2^{-20}$ to 2^{-10}).

This is indicated in the figure because the training accuracy is significantly higher and still rising while the validation accuracy is low and is erratic.

The regularization parameter overfits when n= 8 to 0 (when $reg_param = 2^{-80}$ to 2^{0}).

This is indicated in the figure because both the training accuracy and the validation accuracy are low and are still rising as regularization parameter takes on smaller value (as n increases).

Q2(e):

The model overfits when K = 9 to 50.

This is indicated in the figure because the training accuracy is significantly higher and still rising while the validation accuracy is low and is erratic.

The model overfits when K = 1 to 7.

This is indicated in the figure because both the training accuracy and the validation accuracy are low and are still rising as K gets larger.

The validation accuracy is somewhat low but erratic in the right half of the curve, because the QDA is extremely overfitted at that part and is highly sensitive to numerical error.

Since we are using PCA to first reduce the data, so the complexity of the reduced data heavily depends on the K, and the QDA can only "guess" the features that are reduced by PCA using the previous trained data that were fed to it.

Tiny numerical errors on the overfitted model "disrupts" the weights already built using the QDA and makes the QDA to have more wrong predictions even though it could have gotten it right.

Q3(i):

- The red diagonal line assumes the "bagged validation accuracy" and the "base validation accuracy" have a positive linear relationship. Moreover, it has a slope of 1, so it is assuming that the "bagged validation accuracy" will increase as much as the "base validation accuracy".

The blue dots are close to the red line because our bagged validation accuracy does depend on the base validation accuracy (in other words, bagged validation accuracy almost increases as much as base validation accuracy), especially when the base validation accuracy is low.

Since we have ensembled 200 distinct classifiers to make our bagged prediction, we have reduce the variance, so the values of the ensembled prediction should be closer to the mean (in other words, our bagged prediction should be more accurate than the base prediction by reducing the variance, which is the whole point of ensemble methods), so the bagged accuracy (which is on y-axis) should always be higher than the base accuracy (which is on x-axis), so the points will always stay above the red line.

With the given K and R, the base validation accuracy is not high enough for the ensemble classifiers to provide an "accurate" prediction. So, the bagged validation accuracy will vary depending on the accuracy of the base validation.

- This is possible, since the base validation accuracy is significantly higher in part(g) with the provided K and R, so our ensemble method can improve the validation accuracy exponentially, making our bagged validation accuracy "somewhat" resilient to the base classifier's accuracy.

In part(h), the lower-limit of our K is significantly larger than the upper-limit of K in part(g) (10 v.s. 50) and our upper-limit of K is much higher, this allows our classifier to have more information about the data(more dimensions to each data points), hence having a better fit to the test data.

Also, the regularization term is significantly smaller in part(h),which helps to model the classifier better, as if we picked the regularization term to be 1 in part(g), our covariance matrix will be the identity matrix(assuming all features to have 0 covariance), which can be detrimental to how we fit the data.

- I don't know

Q4(g). I don't know

Q4(h).

In K-means, we try to assign points to the cluster center with the smallest Euclidean distance.

Let X be a data point that is on the decision boundary of center i($\mu^{(i)}$) and center j$\mu^{(j)}$,
then

$||X - \mu^{(i)}||_2 = ||X - \mu^{(j)}||_2$

$||X - \mu^{(i)}||_2^2 = ||X - \mu^{(j)}||_2^2$ (squaring both sides)

$(\sqrt{\sum_{m=1}^{M}(X_m - \mu_m^{(i)})^2})^2 = (\sqrt{\sum_{m=1}^{M}(X_m - \mu_m^{(j)})^2})^2$ (definition of 2-norm)

$\sum_{m=1}^{M}(X_m - \mu_m^{(i)})^2 = \sum_{m=1}^{M}(X_m - \mu_m^{(j)})^2$ (cancelling square root)

$\sum_{m=1}^{M} X_w^2 - 2X_w\mu_w^{(i)} + (\mu_w^{(i)})^2 = \sum_{m=1}^{M} X_w^2 - 2X_w\mu_w^{(j)} + (\mu_w^{(j)})^2$ (perfect square formula)

$\sum_{m=1}^{M} X_w^2 - \sum_{m=1}^{M} 2X_w\mu_w^{(i)} + \sum_{m=1}^{M}(\mu_w^{(i)})^2 = \sum_{m=1}^{M} X_w^2 - \sum_{m=1}^{M} 2X_w\mu_w^{(i)} + \sum_{m=1}^{M}(\mu_w^{(i)})^2$ (sum of equation = sum of each term)

$||X||_2^2 + ||\mu^{(i)}||_2^2 - \sum_{m=1}^{M} 2X_m\mu_m^{(i)} = ||X||_2^2 + ||\mu^{(j)}||_2^2 - \sum_{m=1}^{M} 2X_m\mu_m^{(j)}$ (definition of 2-norm)

$||\mu^{(i)}||_2^2 - ||\mu^{(j)}||_2^2 - \sum_{m=1}^{M} 2X_m\mu_m^{(i)} + \sum_{m=1}^{M} 2X_m\mu_m^{(j)} = 0$ (re-arranging equation)

$||\mu^{(i)}||_2^2 - ||\mu^{(j)}||_2^2 - (\sum_{m=1}^{M} 2X_m\mu_m^{(i)} - 2X_m\mu_m^{(j)}) = 0$ (sum of each term = sum of equation)

$||\mu^{(i)}||_2^2 - ||\mu^{(j)}||_2^2 - 2M(\sum_{m=1}^{M} X_m\mu_m^{(i)} - X_m\mu_m^{(j)}) = 0$ (take out constant from summation)

$||\mu^{(i)}||_2^2 - ||\mu^{(j)}||_2^2 - 2M(\sum_{m=1}^{M} X_m(\mu_m^{(i)} - \mu_m^{(j)})) = 0$ (algebraic manipulation)

$||\mu^{(i)}||_2^2 - ||\mu^{(j)}||_2^2 - 2MX \cdot (\mu^{(i)} - \mu^{(j)}) = 0$ (definition of dot product)

$2M(\mu^{(i)} - \mu^{(j)}) \cdot X + (||\mu^{(i)}||_2^2 - ||\mu^{(j)}||_2^2) = 0$ (re-arranging equation)

Which imples the deciison boundary between cluster i and j is a hyperplane with $w = 2M(\mu^{(i)} - \mu^{(j)})$

and $b||\mu^{(i)}||_2^2 - ||\mu^{(j)}||_2^2 =$.

Since a hyperplane in 2d is a linear line, we have shown that the decision boundary of 2 adjacent clusters in K-means is linear.

Q4(i).

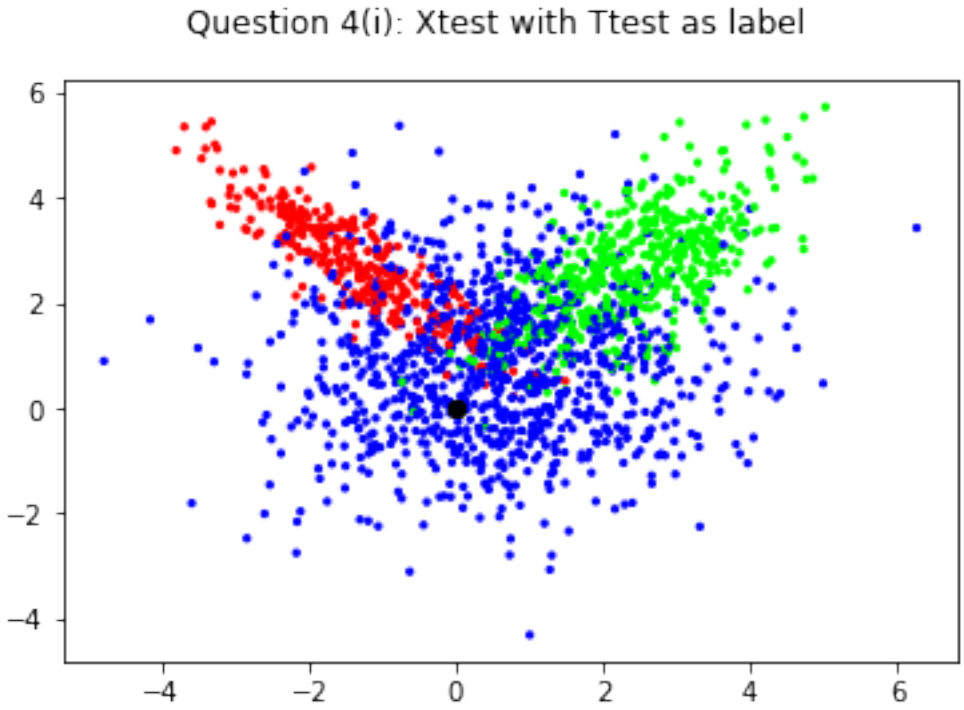
The score of part(c) is lower than part(d) because the type of covariance used for GMM are different.

In part(c), we used "spherical" covariance type, which sets the sizes in all dimension equally(in our 2d case, the mixture will look like a circle), this can cause more percentage of the weight that is supposed to be in cluster A to be assigned to cluster B due to the "shape" of our cluster.

In the following graph(the "correct" solution to our clustering problem), we can see that the red and green clusters take a more ellipse shape than a round sphere, so due to this limitation, the GMM in part(c) couldn't model the actual distribution well enough comparing to part(d).

In part(d), we used "full" covariance type, which allows our distribution to have take arbitrary "sizes" in all dimension(the covariance matrix is no longer "$\Sigma = \sigma_k^2 I$"), this makes shaping our Gaussian model to an ellipse form possible. And the ellipse form will indeed fit our test data better by observing the graph below, hence the higher accuracy in part(d).

Therefore, the difference in covariance type in part(c) and part(d) determined the difference in their accuracy.



Q4(j).

In part(f), we are not updating our covariance matrix based on the updated mean at each iteration, so we are assuming the covariance between the clusters are 0 and the variance of each cluster being 1.

Then, our Gaussian mixture will **never** change in "size", and at each iteration, we are only moving the centre(mean) closer to the points in the current cluster and hoping it can cover as much points as possible.

In part(e), the "spherical" covariance matrix only limits the covariance matrix by letting the covariance be 0, but we can still change the "size" of each Gaussian distribution by changing its variance ($\Sigma = \sigma_k^2 I$).

So, even having to remain in a "spherical" form, the "spherical" covariance type is still more versatile than our identity covariance matrix, hence having a better accuracy score than the model in part(f).

End of Assignment