CS669-Pattern Recognition

Assignment - 2

Bayes Classifier Using GMM (Gaussian Mixture Model)

Group 6

Team :

Akhilesh Devrari (B16005)

Aman Jain (B16044)

Yash Agrawal (B16120)

Table of Contents :

Objective : 3
Given datasets : 3
Dataset-1 : 2-dimensional artificial data of 3 classes: nonlinearly separable data set used in Assignment 1. 3
Real World Dataset : 3
Two dimensional speech dataset used in Assignment 1 3
3 class scene image dataset 3
Cervical cytology (cell) image dataset 3
Procedure : 4
Observations : 6
Dataset I : Non-Linearly Separable Data 6
Inferences for Dataset I Class I : 7
Inferences for Dataset I Class II : 9
Inferences for Dataset I Class III : 11
Inferences : 14
Dataset 2a : Real world Data 15
CLASS 1 FOR REAL WORLD DATA 15
CLASS 2 FOR REAL WORLD DATA : 16
CLASS 3 FOR REAL WORLD DATA : 18
Inferences for real world dataset for log-likelihood graph: 19
FOR CLASS I : 19
FOR CLASS II : 20
FOR CLASS III : 20
Inferences : 23
Dataset 2B : 3 class Scene Image Dataset 24
Dataset 2C : Cervical cytology(cell) image dataset 24
Inferences : 29
Conclusion 30
References : 31

1. Objective :

1.1. Given datasets :

1.1.1. Dataset-1 : 2-dimensional artificial data of 3 classes: nonlinearly separable data set used in Assignment 1.

1.1.2. Real World Dataset :

1.1.2.1. Two dimensional speech dataset used in Assignment 1

1.1.2.2. 3 class scene image dataset

1.1.2.3. Cervical cytology (cell) image dataset

1.2. Data of each class is given separately. For Dataset 1 and Dataset 2(a), 75% of data of a class is to be used as training data for that class, and the remaining data is to be used as test data for that class. For Dataset 2(b) and Dataset 2(c), training and test sets are given.

1.3. To build Bayes classifier using GMM on given datasets, with parameters initialized using K-means clustering.

1.4. To segment the cell images by clustering the local feature vector from cell image datasets into 3 groups using :

1.4.1. K-means clustering (Used for initialization of GMM)

1.4.2. Clustering using GMM

1.5. Finding classification accuracy, precision, mean precision, recall, mean recall and F-measure for every class and mean F-measure on test data.

1.6. Draw constant density contour plot for all the classes with the training data superposed for Dataset-1 and Dataset-2(b).

1.7. Draw decision region plot with the training data superposed.

1.8. Draw plot of 3 clusters on training data of Dataset 2(c) and the result of cluster projected on test images for Dataset-1 and Dataset-2(b).

1.9. Draw graph of iterations vs log likelihood for all the datasets with different number of components.

2. Procedure :

1. First of all Features are extracted from the image files given in train and test directories for each of the datasets.

2. Classification of all the datasets done by bayes classifier using GMM on Dataset-1, Dataset-2(a), Dataset-2(b) and Dataset-2(c). Parameters of GMM are initialized using K-means clustering.

3. The data of features extracted are taken as input from text file for all the datasets (i.e. non-linearly separable dataset , real-world dataset).

4. Data for each dataset extracted will have following dimensions :
   a. Non-linearly Separable Dataset : 2-dimension
   b. Real World Dataset :
      i. Speech Dataset used in Assignment 1 : 2-dimension
      ii. 3-class scene image dataset : 24-dimensional color histograms
      iii. Cervical cytology (cell) image dataset : 2-dimension

5. For initialization of means used for K-MEAN method are taken as randomly K data points extracted above.

6. These k data points chosen are considered to be the initial mean of k clusters.Now each clusters are associated with centers.

7. Procedure for K-MEANS Algorithm :
   a. Initialise the cluster centers using selected K data points.
   b. Now assign each data point to a cluster k*.
   c. The assigning of data points to a cluster is done using euclidean distance of a particular data point from a particular cluster.
   d. The cluster for which the euclidean distance of the data point is minimum is assigned to that cluster.
   e. Now each cluster will have some number of data points assigned to that cluster having minimum distance.
   f. The recomputation of K means is done at every iteration by taking the mean of all the data points assigned to a cluster in previous step.
   g. Now repeat step e and step f until convergence.

8. Cluster array is a 3D array having Nk number of 24 dimension data points for all the k clusters.

9. After the convergence of K-MEANS algorithm, K clusters are divided according to the mean computed.

10. Now the mean calculated in K-MEAN will be used to calculate covariance matrix and PI for each of the K clusters.

11. PI for a cluster k is defined as the ratio of number of data points reside in cluster k to the total number of data points.

12. MEAN , COVARIANCE, PI calculated in the previous steps are used as an initialized values for Gaussian Mixture Model (GMM).

13. Convergence criteria for GMM (Gaussian Mixture Model) : Difference between likelihood in successive iterations should be less than certain threshold which is taken as 0.001.

14. Now using the MEAN, COVARIANCE, PI computed from K-MEAN likelihood is calculated given by the formula :

$$-\log \Pr(x|\pi,\mu,\Sigma) = -\sum_{i=1}^{n} \log \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k,\Sigma_k) \right\}$$

- E-step: Given parameters, compute

$$r_{ik} \triangleq E(z_{ik}) = \frac{\pi_k \mathcal{N}(x_i|\mu_k,\Sigma_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k,\Sigma_k)}$$

- M-step: Maximize the expected complete log likelihood

$$E\left[\log \Pr(x,z|\pi,\mu,\Sigma)\right] = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \left\{\log \pi_k + \log \mathcal{N}(x_i|\mu_k,\Sigma_k)\right\}$$

By updating the parameters

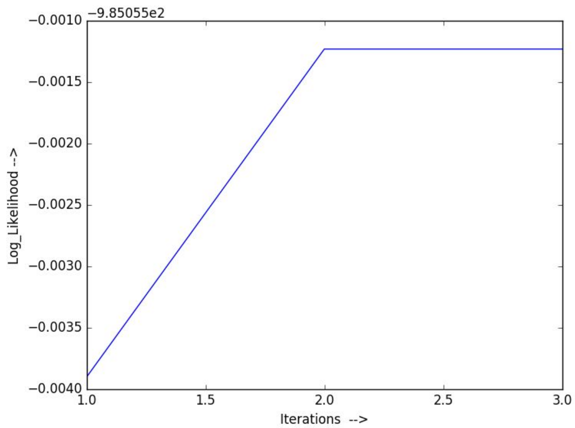
$$\pi_{k+1} = \frac{\sum_i r_{ik}}{n}, \mu_{k+1} = \frac{\sum_i r_{ik} x_i}{\sum_i r_{ik}}, \Sigma_{k+1} = \frac{\sum_i r_{ik}(x_i-\mu_k)(x_i-\mu_k)^T}{\sum_i r_{ik}}$$

- Iterate till likelihood converges.
- Converges to local optimum of the log likelihood.

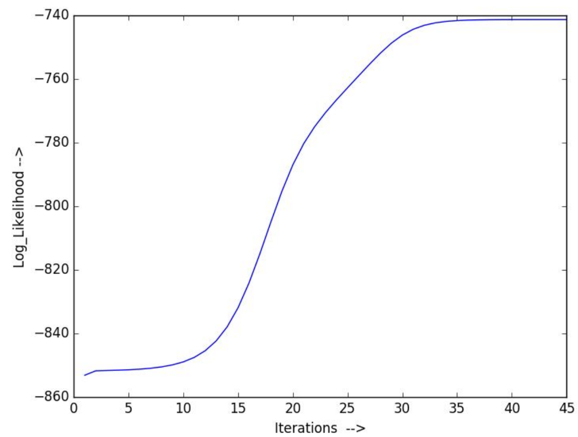
15. Above formula converges until the value of difference between the old likelihood and the new likelihood is less than some threshold value.

3. Observations :

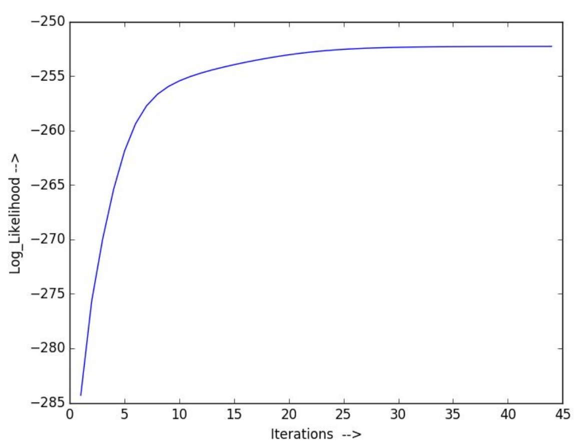
3.1. Dataset I : Non-Linearly Separable Data



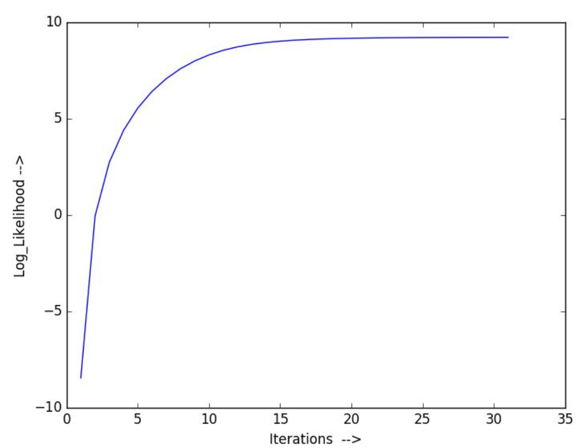
(a)



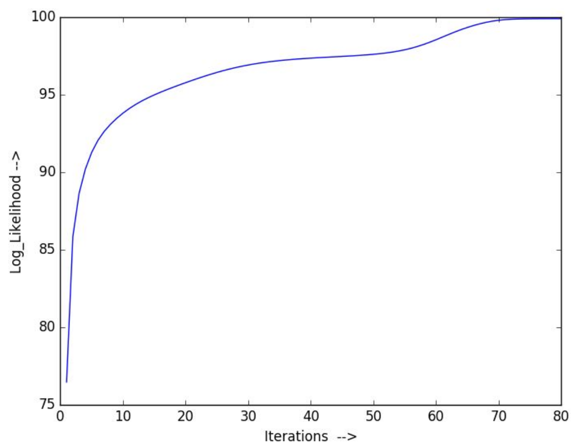
(b)



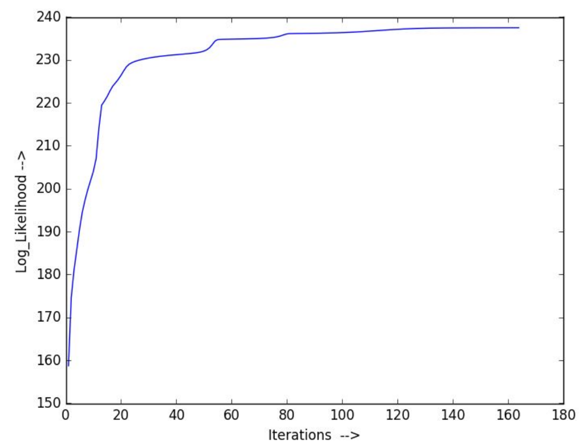
(c)



(d)



(e)



(f)

Fig 1 : Iteration vs Log Likelihood for **class 1** with different number of Components : a) k=1, b) k=2, c) k=4, d) k=8, e) k=16, e) k=32

Inferences for Dataset I Class I :

<u>For Figure(a)</u>

1. For figure (a), the value of total clusters k is taken as 1 and the graph is plotted between number of iterations and log likelihood.
2. As seen from the graph the value of likelihood is increased as the number of iterations is increasing.
3. The number of clusters is 1 so the graph gets converged in 3 iterations itself.
4. After iteration 2, the value of log likelihood doesn't change which leads to the convergence of the graph.

<u>For Figure(b)</u>

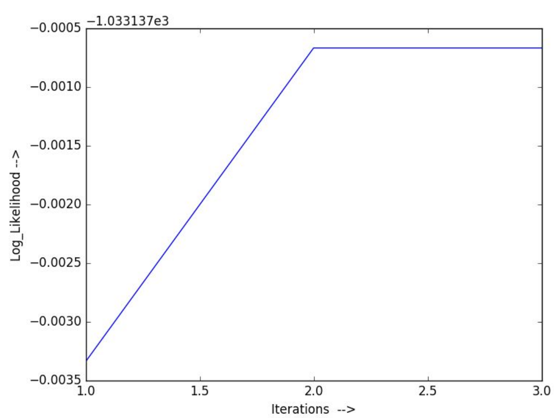
1. For figure (b), the value of total clusters k is taken as 2 and the graph is plotted between number of iterations and log likelihood.
2. The graph for log_likelihood vs iterations gets converged into 45 iterations.
3. At starting of the graph, the slope of the graph is a very low value due to small change in the value of log likelihood.
4. After certain iterations upto 20, the slope of the graph is increased suddenly which reflects the fact that the value of log likelihood changed suddenly.

For Figure(c)

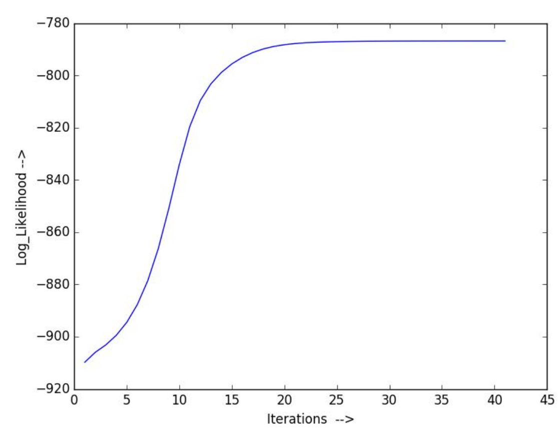
1. For figure (c), the value of total clusters k is taken as 4 and the graph is plotted between number of iterations and log likelihood.
2. Initially the difference in the value of log likelihood change but after 10 iterations the difference in the log likelihood doesn't make a significant change.

For Figure(d,e,f)

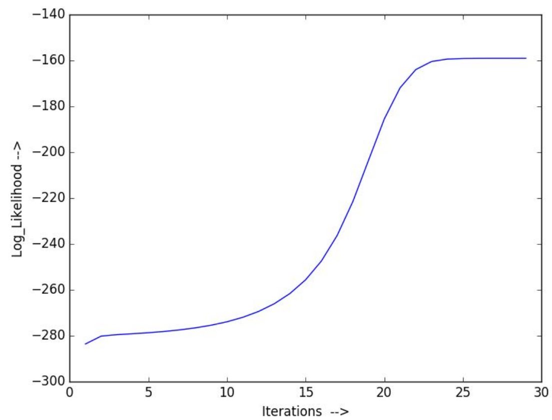
1. For figure(d),(e),(f) the graph of iterations vs likelihood is plotted for the values of cluster k as 8,16,32 which seems to be similar to the one we got in figure (c).
2. The above observation gives the fact that for changing/increasing the number of clusters doesn't reflect any major changes in the graph.



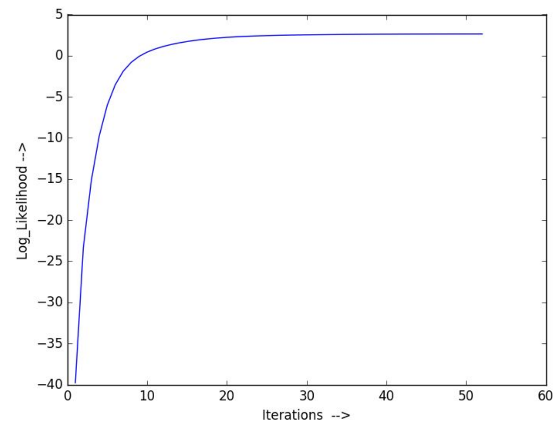
(a)



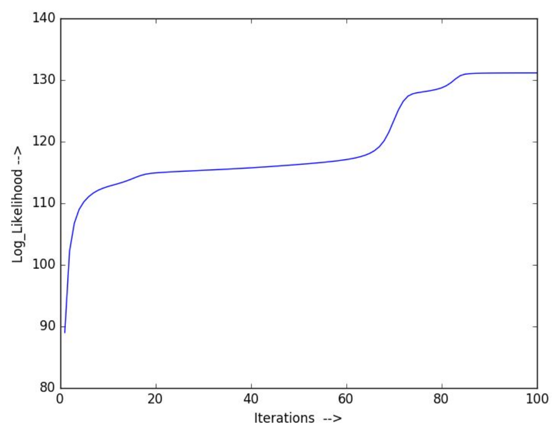
(b)



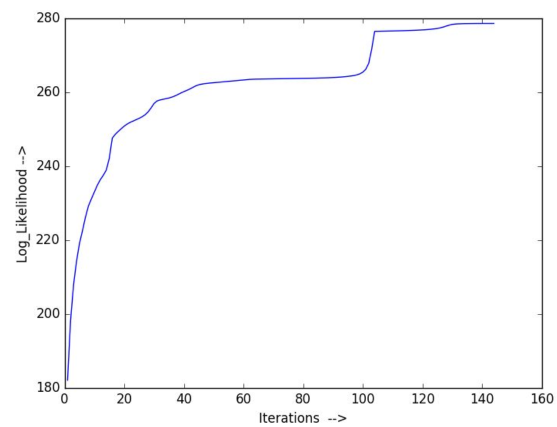
(c)



(d)



(e)

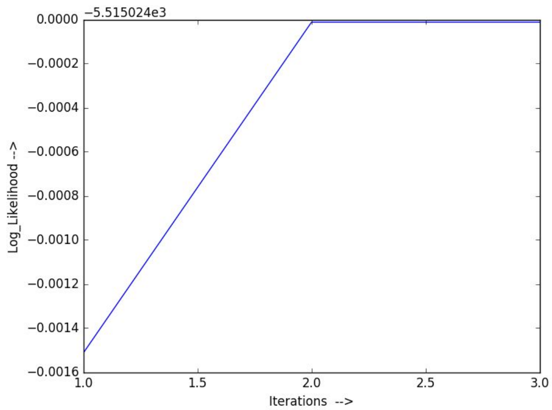


(f)

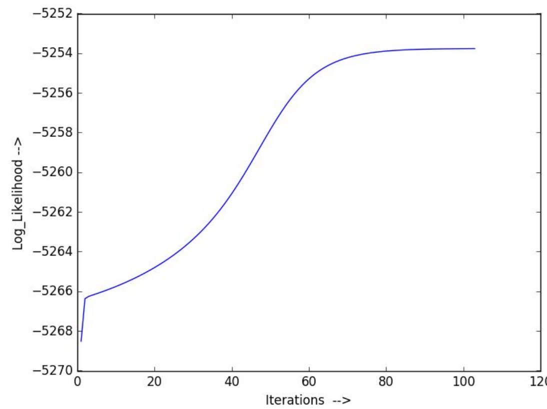
Fig 2 : Iteration vs Log Likelihood for **class 2** with different number of Components : a) k=1, b) k=2, c) k=4, d) k=8, e) k=16, e) k=32

Inferences for Dataset I Class II :

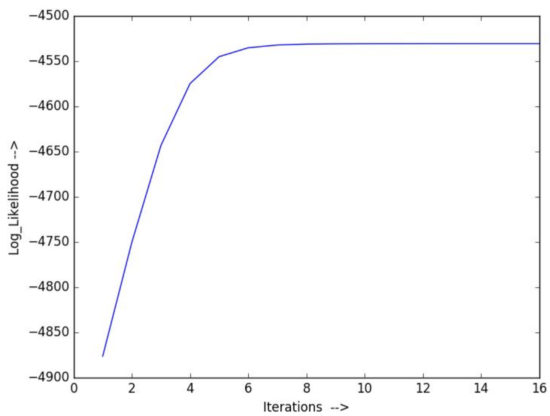
1. For figure (a),(b),(d) Initially the difference in the value of log likelihood changes rapidly and after 2nd iteration the graph converges.
2. For figure (c), Initially the value of log likelihood doesn't give significant change for iterations upto 20 and after this it starts to change rapidly which will then converge after 25 iterations.
3. For figure(e),(f), the change in the log likelihood converges after 100 iterations.


−5.515024e3
Log_Likelihood -->
Iterations -->

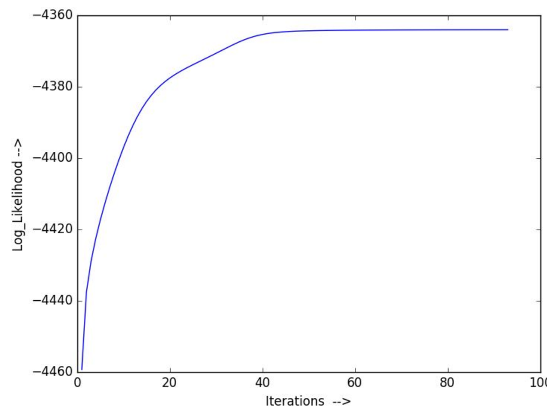
(a)


Log_Likelihood -->
Iterations -->

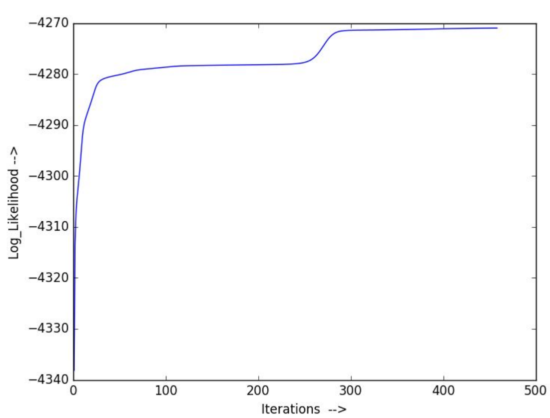
(b)


Log_Likelihood -->
Iterations -->

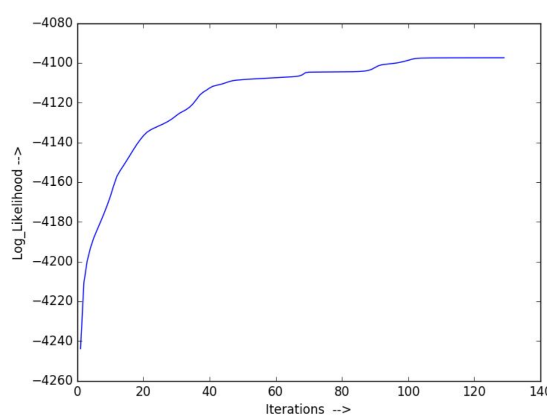
(c)


Log_Likelihood -->
Iterations -->

(d)


Log_Likelihood -->
Iterations -->

(e)


Log_Likelihood -->
Iterations -->

(f)

Fig 3 : Iteration vs Log Likelihood for **class 3** with different number of Components : a) k=1, b) k=2, c) k=4, d) k=8, e) k=16, e) k=32

Inferences for Dataset I Class III :

1. Figure (a) is similar to figure (a) obtained in above classes that is initially the change in log likelihood is very much in comparison to the change in log likelihood after 2nd iteration.
2. For Figure (b), the graph converges after 80 iterations.
3. For figure (c), the graph is plotted for 4 clusters which gets converged after 6 iterations while in figure(d),(e),(f) which is for clusters k=8,16,32 takes significant number of iterations to converge.

(a) K = 1: Accuracy : 96.5694682676 %

Confusion Matrix

	Class 1	Class 2	Class 3
Class 1	121	4	0
Class 2	6	118	0
Class 3	6	3	324

Analysis

	Class 1	Class 2	Class 3	Mean
Recall	0.968	0.944	0.9729	0.9616
Precision	0.9097	0.944	0.9969	0.9502
F-Measure	0.9379	0.944	0.9848	0.9555

(b) K = 2: Accuracy : 98.1132075472 %

Confusion Matrix

	Class 1	Class 2	Class 3
Class 1	121	4	0
Class 2	0	122	3
Class 3	0	4	329

Analysis

	Class 1	Class 2	Class 3	Mean
Recall	0.968	0.976	0.9879	0.9773
Precision	1.0000	0.9384	0.9909	0.9764
F-Measure	0.9837	0.9568	0.9894	0.9766

(c) K = 4: Accuracy : 100 %

Confusion Matrix

	Class 1	Class 2	Class 3
Class 1	125	0	0
Class 2	0	125	0
Class 3	0	0	333

Analysis

	Class 1	Class 2	Class 3	Mean
Recall	1.0000	1.0000	1.0000	1.0000
Precision	1.0000	1.0000	1.0000	1.0000
F-Measure	1.0000	1.0000	1.0000	1.0000

(d) K = 8: Accuracy : 100 %

Confusion Matrix

	Class 1	Class 2	Class 3
Class 1	125	0	0
Class 2	0	125	0
Class 3	0	0	333

Analysis

	Class 1	Class 2	Class 3	Mean
Recall	1.0000	1.0000	1.0000	1.0000
Precision	1.0000	1.0000	1.0000	1.0000
F-Measure	1.0000	1.0000	1.0000	1.0000

(e) K = 16: Accuracy : 100 %

Confusion Matrix

	Class 1	Class 2	Class 3
Class 1	125	0	0
Class 2	0	125	0
Class 3	0	0	333

Analysis

	Class 1	Class 2	Class 3	Mean
Recall	1.0000	1.0000	1.0000	1.0000
Precision	1.0000	1.0000	1.0000	1.0000
F-Measure	1.0000	1.0000	1.0000	1.0000

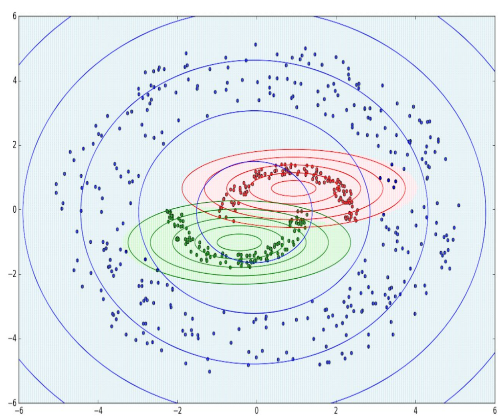
(f) K = 32: Accuracy : 100 %

Confusion Matrix

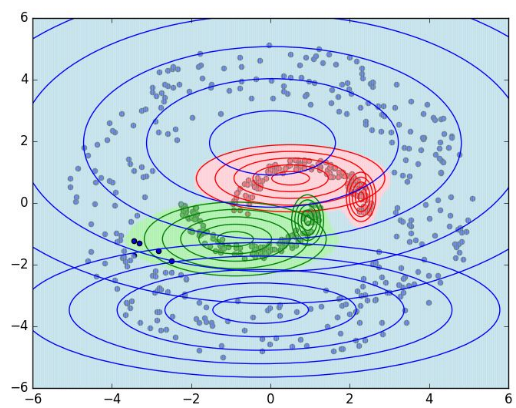
	Class 1	Class 2	Class 3
Class 1	125	0	0
Class 2	0	125	0
Class 3	0	0	333

Analysis

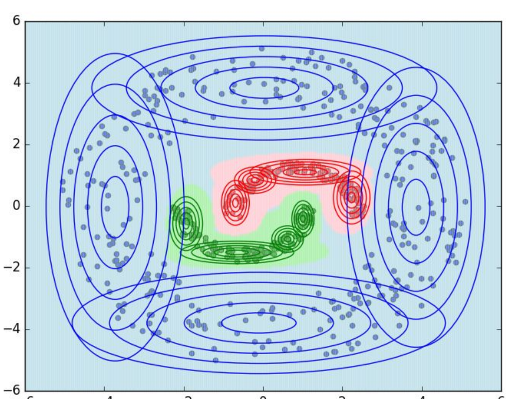
	Class 1	Class 2	Class 3	Mean
Recall	1.0000	1.0000	1.0000	1.0000
Precision	1.0000	1.0000	1.0000	1.0000
F-Measure	1.0000	1.0000	1.0000	1.0000



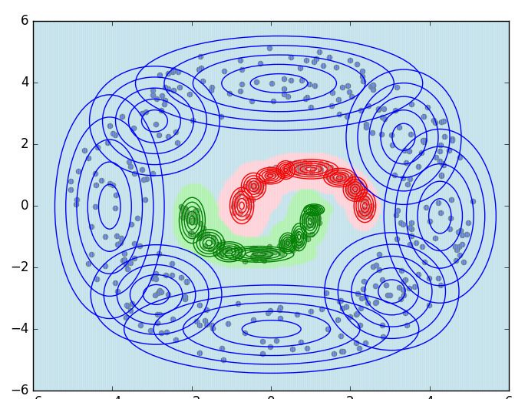
(a)



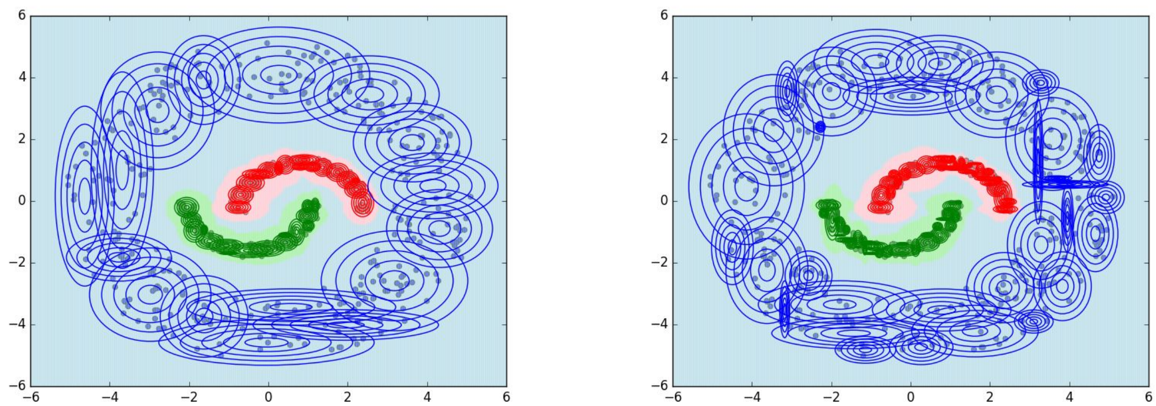
(b)



(c)



(d)



(e) (f)

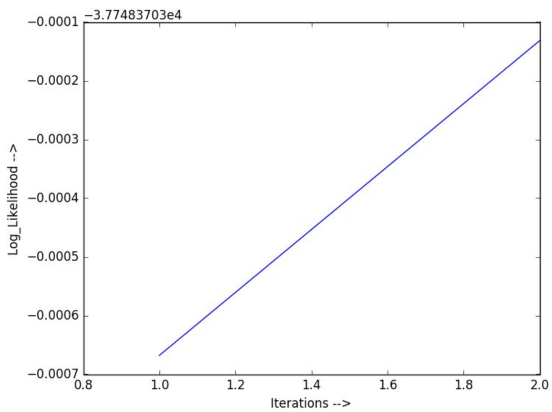
Fig 4 : Decision Region & Contour Density Plot for Dataset 1 (Non-Linearly Separable Data) with different number of mixtures : a) k=1, b) k=2, c) k=4, d) k=8, e) k=16, f) k=32

Inferences :

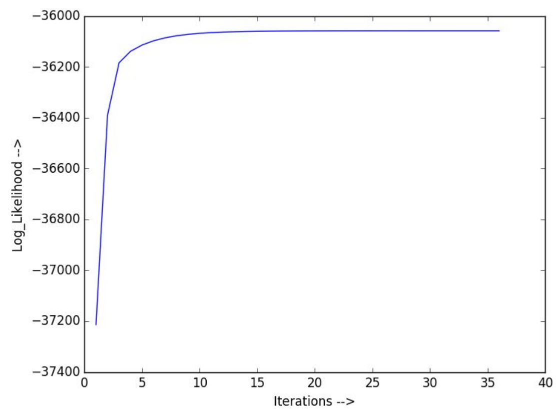
1. As clearly visible from the graph and confusion matrix itself, the accuracy and performance of classifier increases with increase in number of clusters.
2. But increasing the number of clusters reflect as increased time for classification. Therefore there is a tradeoff between accuracy and time needed for classification.
3. Number of clusters needs to be determined empirically, and for this dataset, we get 100% accuracy for k = 4. Further increasing k only leads to slowed down classification. Therefore for this particular dataset, k = 4 or k = 8 seems ideal choice.
4. Contours of different clusters have different sizes as the number of points belonging to a cluster and spatial distances between those points vary from cluster to cluster.
5. Decision boundary resembles more to actual allocation of training data points for higher value of k. This is analogous to fact that we can make any shape with large number of small shapes more effectively than with small number of larger shapes.

3.2. Dataset 2a : Real world Data

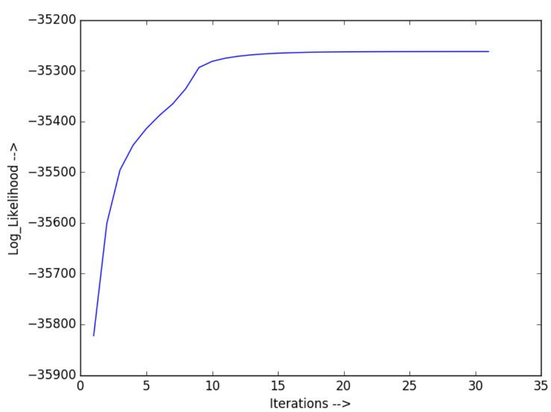
CLASS 1: FOR REAL WORLD DATA



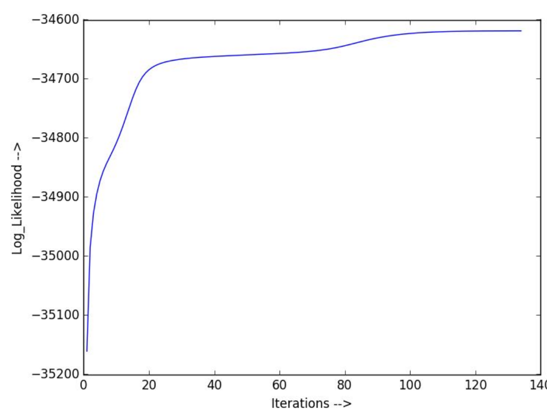
(a)



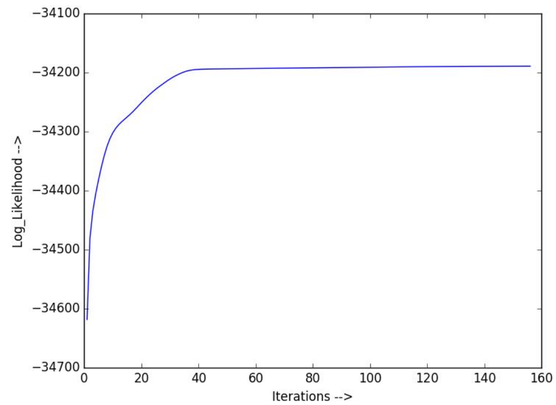
(b)



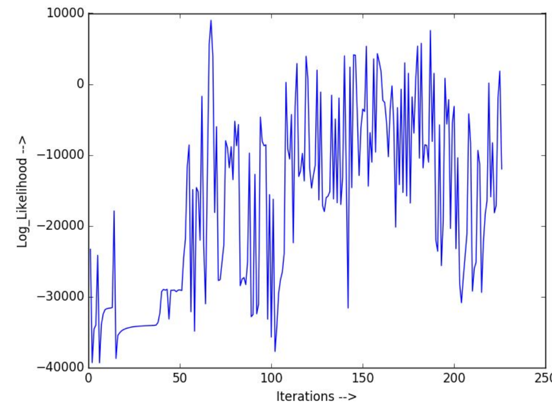
(c)



(d)



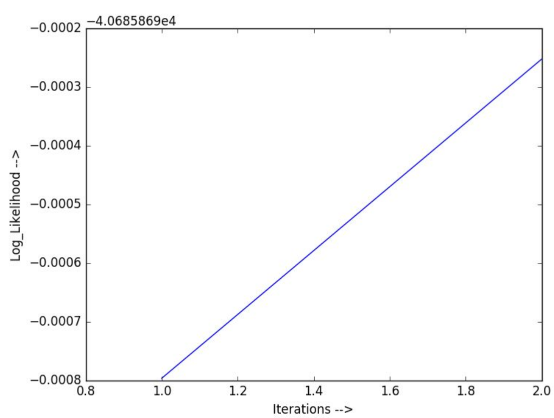
(e)



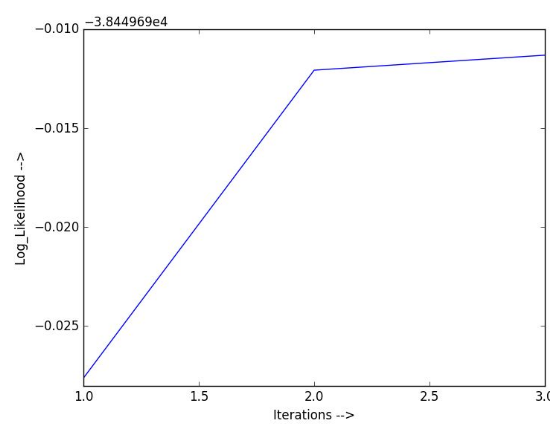
(f)

Fig 5 : Iteration vs Log Likelihood for **class 1 of dataset 2A** with different number of Components : a) k=1, b) k=2, c) k=4, d) k=8, e) k=16, e) k=32

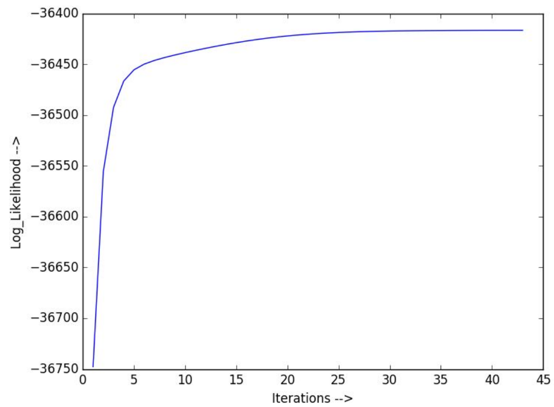
CLASS 2 FOR REAL WORLD DATA :



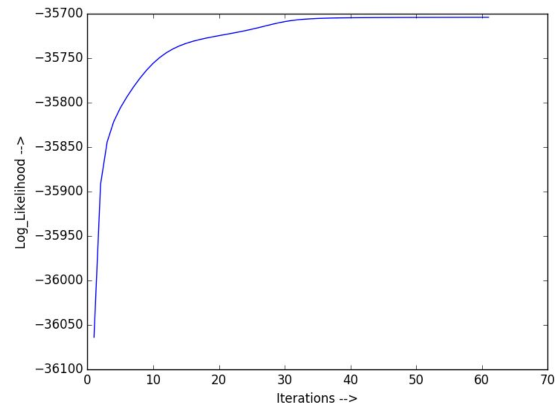
(a)



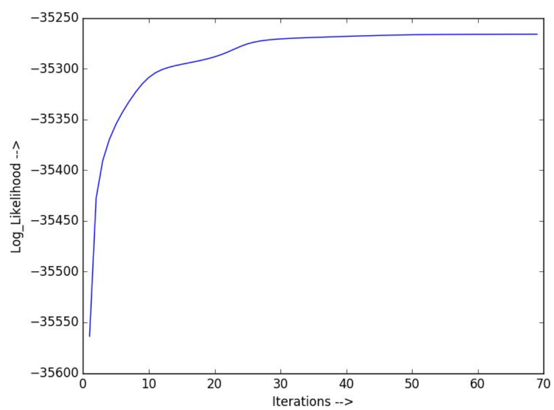
(b)



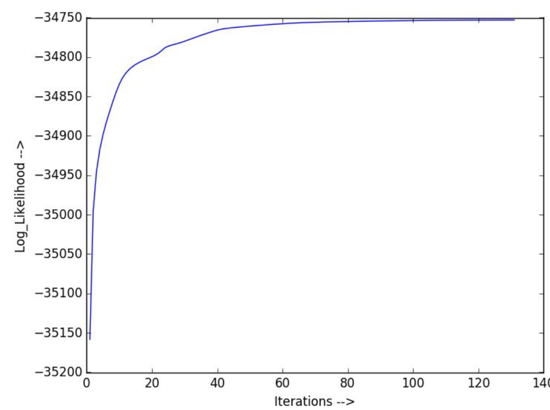
(c)



(d)



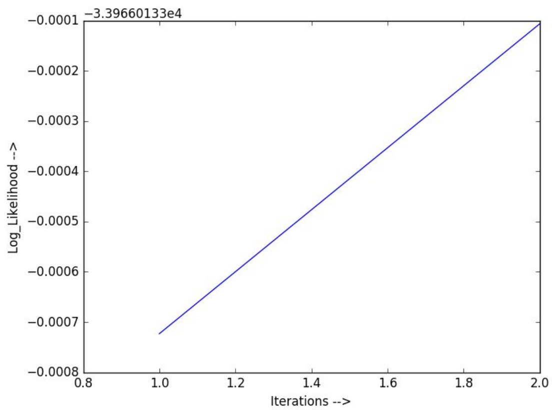
(e)



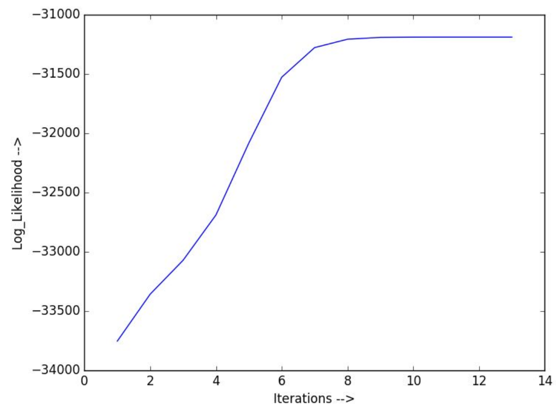
(f)

Fig 6 : Iteration vs Log Likelihood for **class 2 of dataset 2A** with different number of Components : a) k=1, b) k=2, c) k=4, d) k=8, e) k=16, e) k=32

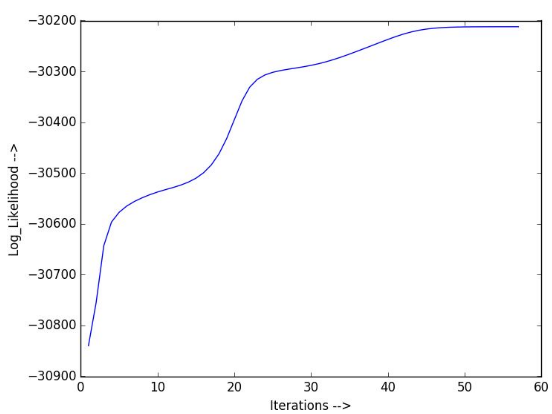
CLASS 3 :FOR REAL WORLD DATA :



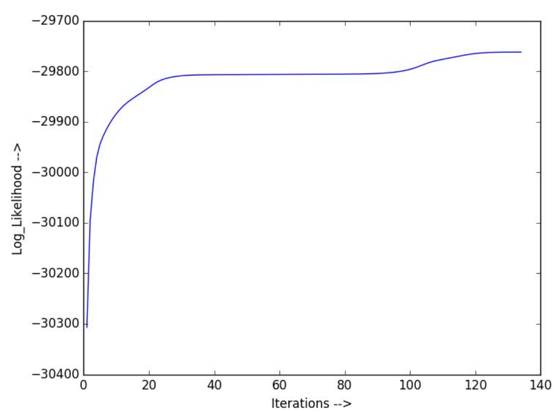
(a)



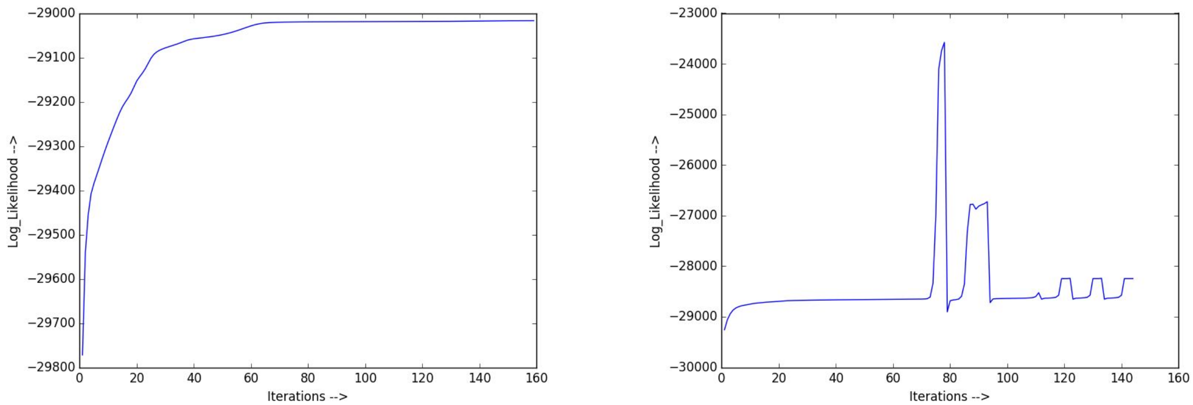
(b)



(c)



(d)



(e) **(f)**

Fig 7 : Iteration vs Log Likelihood for **class 3 of dataset 2A** with different number of Components : a) k=1, b) k=2, c) k=4, d) k=8, e) k=16, e) k=32

Inferences for real world dataset for log-likelihood graph:

1. **For Class I :**

   a. As the value of number of clusters is 1, so the graph converges in 2 iterations itself.
   b. Initially there is a drastic change in log likelihood for small change in number of iterations but after 25 iterations the graph gets converged.
   c. For figures (c),(d),(e), the graph is similar to what we get for number of cluster k = 2 i.e. the change in number of clusters doesn't affect the graph.
   d. For figure(e), the number of clusters is increased to 32 and the dataset is real world dataset so the data points are scattered in the graph at different locations. Mos of the times any of the cluster will get assigned to only one data point which leads to zero covariance matrix and that the only reason that we are getting such distortions in the graph.

2. **For Class II :**

   a. Similar to the class I, the graph of log likelihood vs iterations gets converged after 2 iterations.
   b. For figure (b), the value of log likelihood increases linearly upto 2 iterations and after 2 iterations log likelihood again grows linearly but with a different slope and suddenly converges after 3 iterations.
   c. Rest of the plots are similar to each other for all the number of clusters k = 4,6,8,16 i.e. the change in log likelihood doesn't depend on number of clusters for clusters greater than or equal to 4.

3. **For Class III :**

   a. All the graphs for class 3 are similar to the graphs obtained for class1 except for the graph (f).
   b. For graph (f), the reason of distortion in the graph is same as the reason for distortion we get for class I graph (f).
   c. The reason is because the graph has a very distributed data for class 3 and if there is only one point for a particular cluster then it gives zero covar matrix which can not be used for further processes in turn gives distortion in the graph.

(a) K = 1: Accuracy : 81.9921215532 %

Confusion Matrix

	Class 1	**Class 2**	**Class 3**
Class 1	546	53	23
Class 2	185	400	29
Class 3	15	15	511

Analysis

	Class 1	**Class 2**	**Class 3**	**Mean**
Recall	0.8778	0.6514	0.9445	0.8246
Precision	0.7319	0.8547	0.9076	0.8314
F-Measure	0.7982	0.7393	0.9257	0.8211

(b) K = 2: Accuracy : 84.0742824986 %

Confusion Matrix

	Class 1	Class 2	Class 3
Class 1	564	52	6
Class 2	165	412	37
Class 3	18	5	518

Analysis

	Class 1	Class 2	Class 3	Mean
Recall	0.9067	0.6710	0.9574	0.8450
Precision	0.7550	0.8784	0.9233	0.8522
F-Measure	0.8239	0.7608	0.9401	0.8416

(c) K = 4: Accuracy : 82.3860438942 %

Confusion Matrix

	Class 1	Class 2	Class 3
Class 1	585	30	7
Class 2	240	372	2
Class 3	25	9	507

Analysis

	Class 1	Class 2	Class 3	Mean
Recall	1.0000	1.0000	1.0000	1.0000
Precision	1.0000	1.0000	1.0000	1.0000
F-Measure	1.0000	1.0000	1.0000	1.0000

(d) K = 8: Accuracy : 83.1738885763 %

Confusion Matrix

	Class 1	Class 2	Class 3
Class 1	557	37	28
Class 2	202	410	2
Class 3	19	11	511

Analysis

	Class 1	Class 2	Class 3	Mean
Recall	0.8954	0.6677	0.9445	0.8359
Precision	0.7159	0.8951	0.9445	0.8518
F-Measure	0.7957	0.7649	0.9445	0.8350

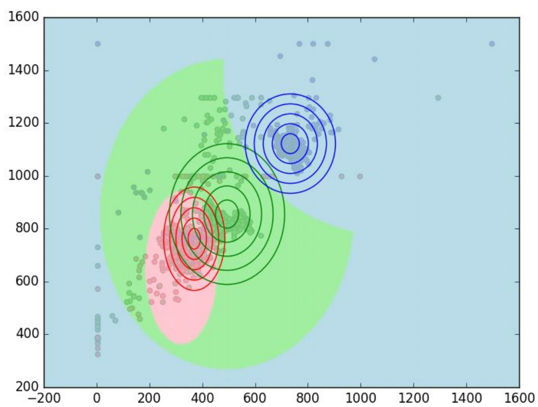
(e) K = 16: Accuracy : 83.849184018 %

Confusion Matrix

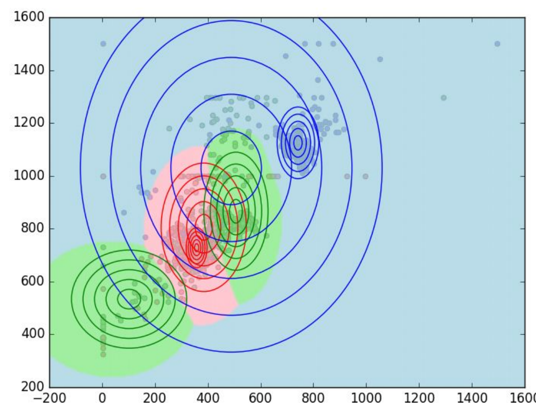
	Class 1	Class 2	Class 3
Class 1	554	37	31
Class 2	182	422	5
Class 3	19	8	514

Analysis

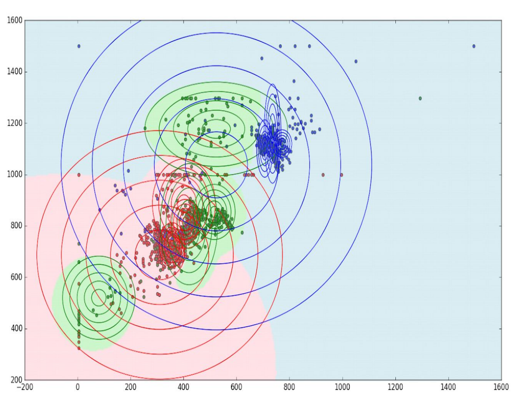
	Class 1	Class 2	Class 3	Mean
Recall	0.8906	0.6872	0.9500	0.8426
Precision	0.7289	0.9036	0.9345	0.8557
F-Measure	0.8017	0.7807	0.9422	0.8415



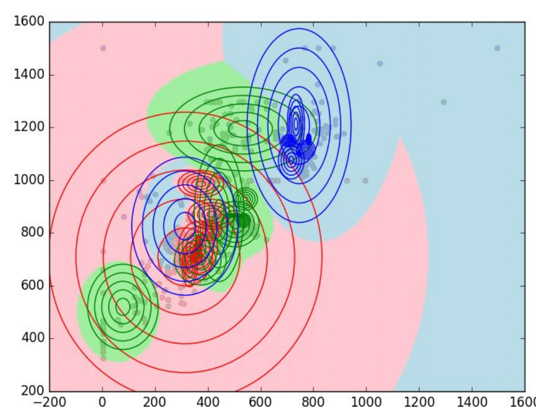
(a)



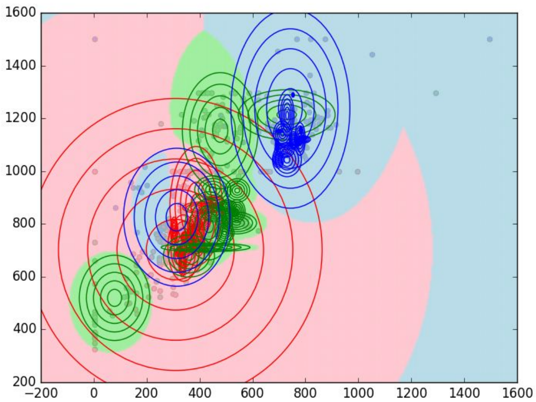
(b)



(c)



(d)



(e)

Fig 8 : Decision Region & Contour Density Plot for Dataset 2A (Real World Data) with different number of mixtures : a) k=1, b) k=2, c) k=4, d) k=8, e) k=16

Inferences :

1. In general accuracy increases with increasing number of clusters, with exception for K=2.
2. This could be due the fact that initial K-centers chosen for K-means clustering are random, and thus when we ran our code for K = 2, better cluster centers were chosen by chance. Though finally K-means clustering converges to cluster-means, but they could be different on different runs.
3. As this is real-world data, and data points are very scattered, thus is relatively less than that for dataset 1.
4. For this dataset, K = 2 seems ideal choice as accuracy doesn't increase much on further increasing value of K.

3.3. Dataset 2B : 3 Class Scene Image Dataset

For this dataset, 24 dimension colour histogram feature vector as well as Bag-of-Visual-Words(BoVW) was extracted successfully. Since there were large number of feature vectors for all of the training images in each of the class, the computation time for GMM was very large due to which we were not able to plot the log likelihood vs Iteration plot for all of the classes.

For the same, the classification could not be done from 24 Dimensional Feature vector using GMM.

From BoVW, the GMM was not converging and the results were absurd because in few cases all of the test images were being classified to only one of the class.

3.4.Dataset 2C : Cervical cytology(cell) image dataset

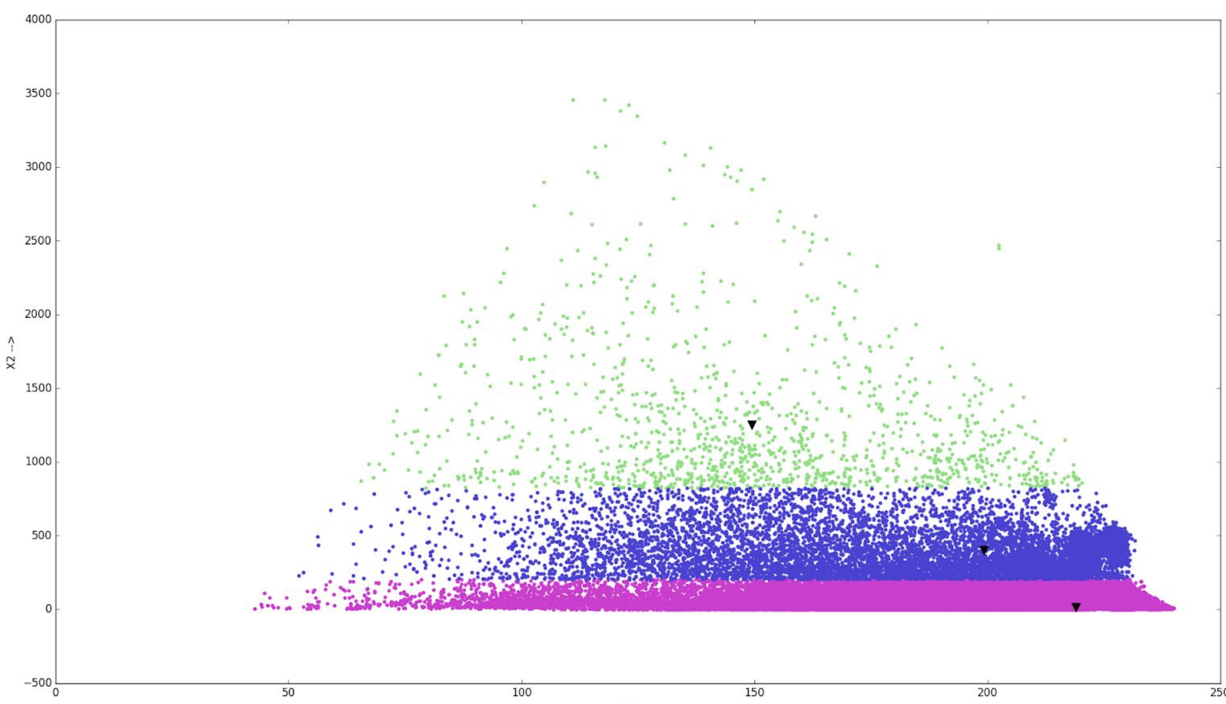


Fig 9 : Plot of 3 clusters on Training Data of Dataset 2C using **K Means** Clustering

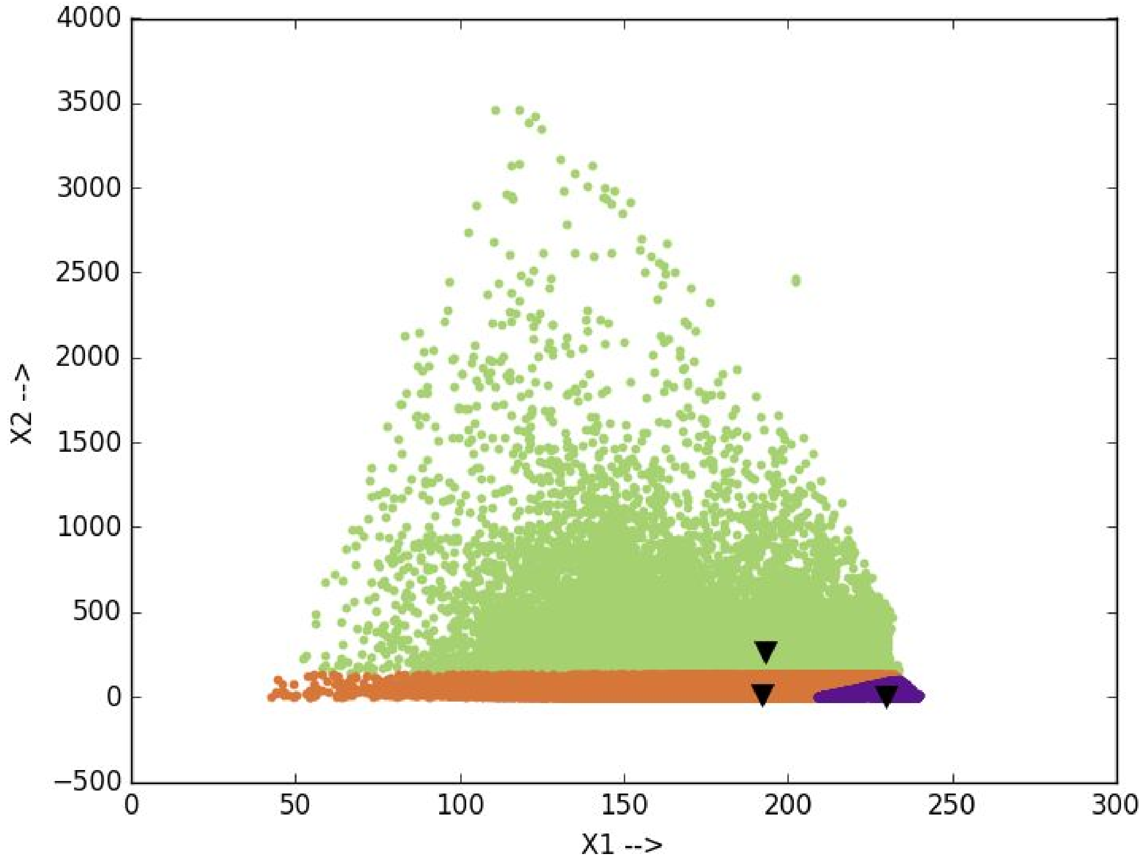
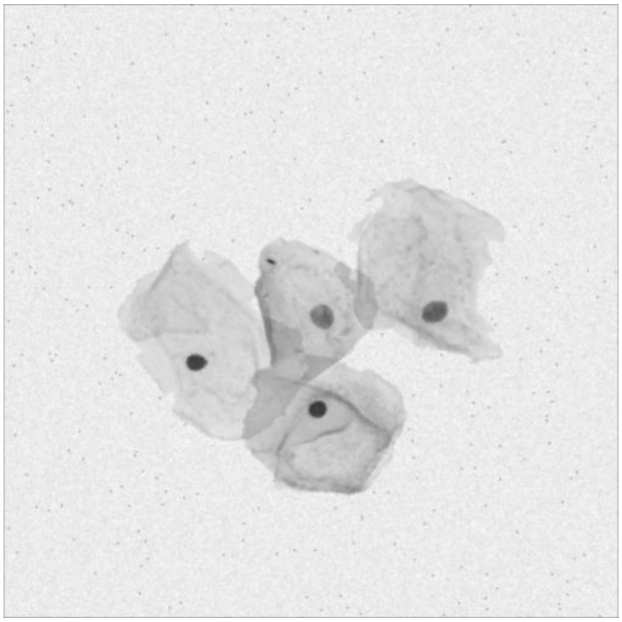
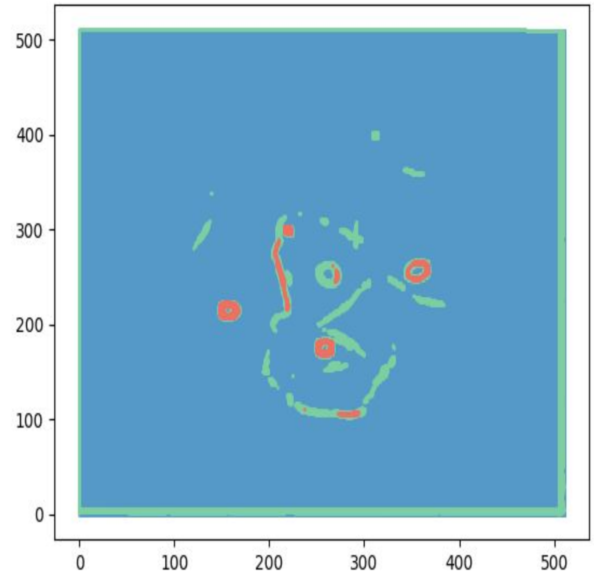
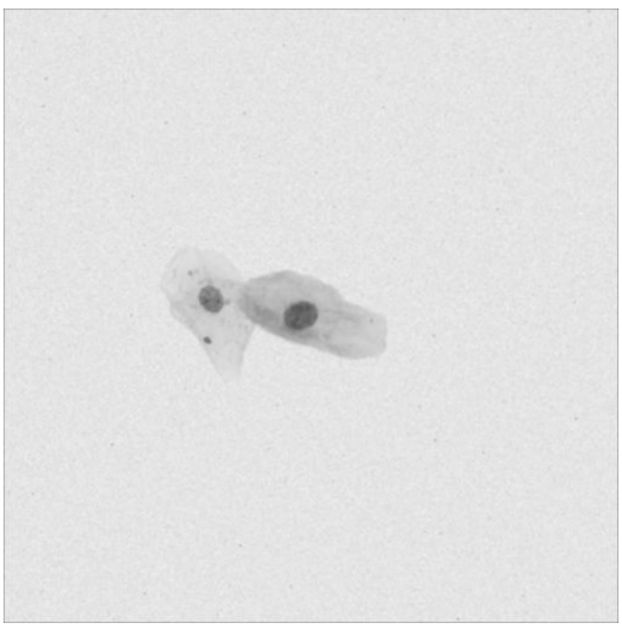
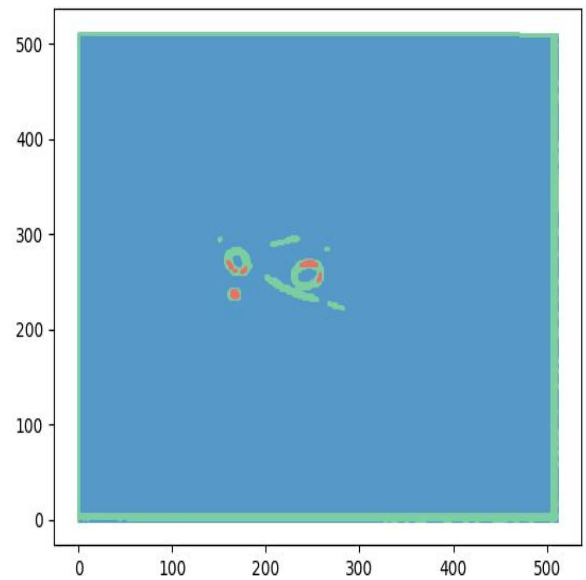


Fig 10 : Plot of 3 clusters on Training Data of Dataset 2C using **GMM** whose initial parameters are initialised using K means.




0
100
200
300
400
500
0
100
200
300
400
500




0
100
200
300
400
500
0
100
200
300
400
500

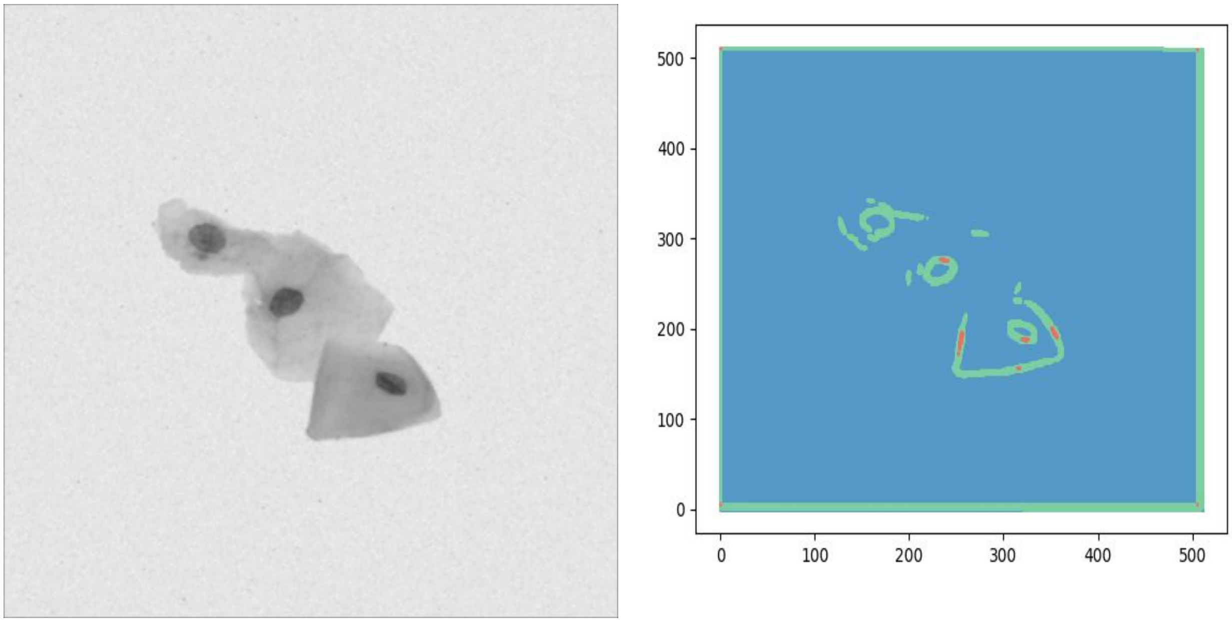
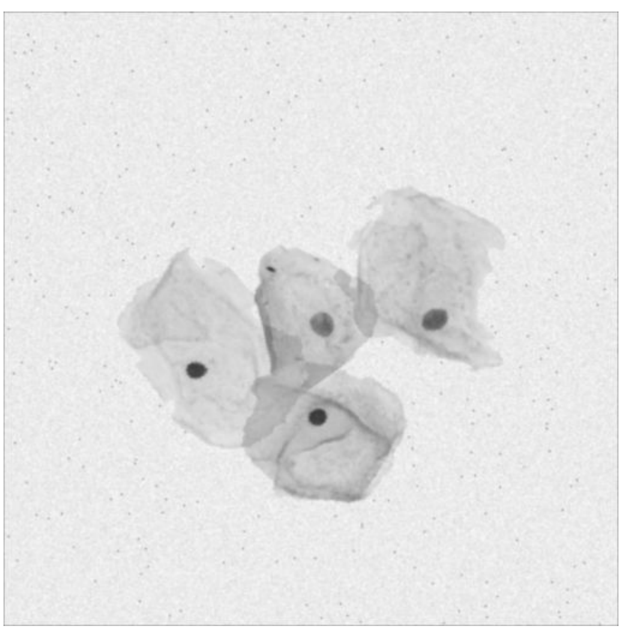
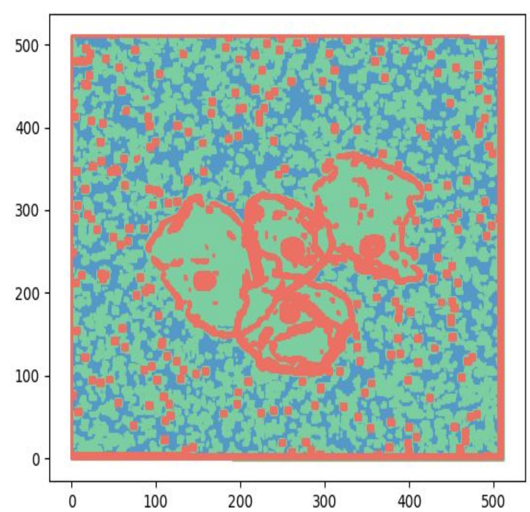
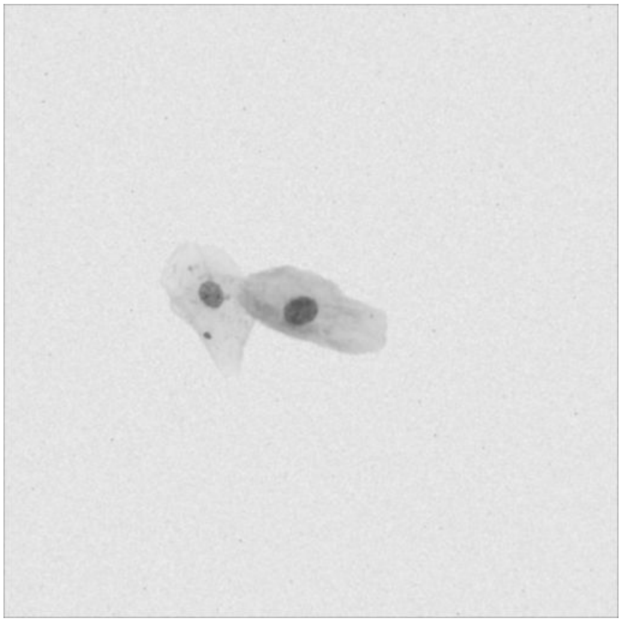


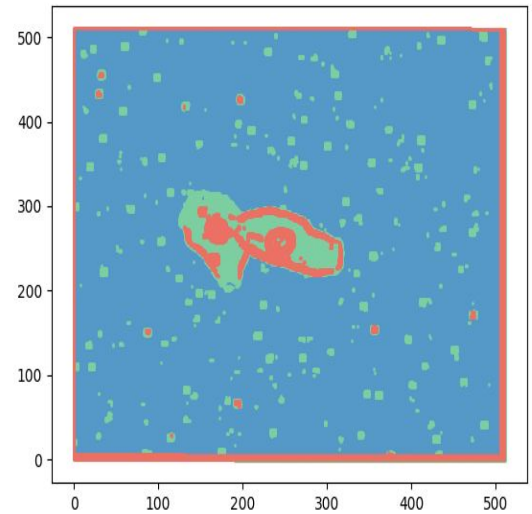
Fig _ : Result of cluster projected on test images using K-Means. Red colour represents nucleus, Green colour represent Cell, Blue colour represents background.



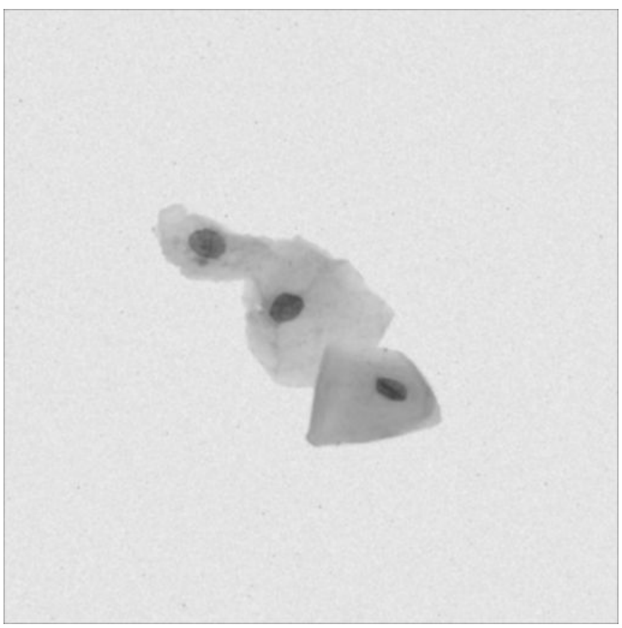
(a) Cell Image 1

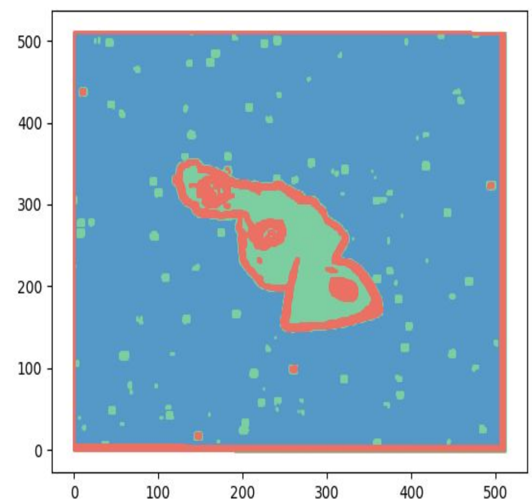






(b) Cell Image 2





(c) Cell Image 3

Inferences :

1. It can be seen from the above images that segmenting the cell images by clustering using GMM gives much better results than clustering using K-Means because K-Means is a hard clustering technique and gives absurd results if data

points are scattered widely whereas GMM is a soft clustering technique and gives and performs better even if the data points are scattered.

2. In K-Means, most of the details of nucleus is missing as the size of nucleus is small and while extracting the feature vectors from cell image, non overlapping patches were taken in order to reduce the computation time for segmentation which resulted in a loss of information.
3. There is a boundary of red colour at some places as well as at the boundary because at edges the complete patch could not be fetched, rather the average values were taken for the pixels that were going outside the size of image, that resulted in variation at the boundaries.
4. For class Images I, we were not able to figure out the reason for lot of green and red points in the background, where only blue colour should have been there.

4. Conclusion

1. Accuracy of the GMM classifier build in this assignment is 100%(K>=4) for dataset 1, while the best accuracy we could get for same dataset in previous assignment using bayesian classifier was 94%.
2. Accuracy of the GMM classifier build in this assignment and Bayesian classifier build in last assignment was similar for real world data. But the decision boundary is more refined while using GMM.
3. GMM performs better than Bayesian classifier because Bayesian classifier can only achieve hyper-quadratic decision boundary at best. While using GMM we can achieve any function as decision boundary.
4. In general accuracy of GMM increases with increasing the number of clusters to a certain extent. Further increasing value of K do not necessarily improve performance and can even decrease accuracy if number of training examples is less. This is because number of training examples for each cluster will decrease.

5. GMM with higher number of clusters take more time for classification, hence there is a tradeoff between accuracy and time taken for classification.
6. In case of segmentation, GMM performs better than K-Means clustering. Also, GMM converges in less amount of time and gives more accurate results if parameters of GMM are initialised using K-Means clustering.

5. Assumptions :

1) In some of the cases only one data point gets assigned to one cluster point which leads to zero covariance matrix. In such case the diagonal values of the matrix are taken as 10^-6 (0.0000001).
2) For datasets 2b and 2c, the patches extracted from the input images are non-overlapping patches due to big data calculation for overlapping patches.
3) Some of the plots are plotted for dataset 2b but not included in this report.

6. References :

1) **https://mas-dse.github.io/DSE210/Additional%20Materials/gmm.pdf**
2) **Class Notes, Lecture Slides**
3) **https://en.wikipedia.org/wiki/Mixture_model**
4) **https://www.geeksforgeeks.org/gaussian-mixture-model/**
5) **http://www.cse.iitm.ac.in/~vplab/courses/DVP/PDF/gmm.pdf**