Pattern Recognition
CS - 669

ASSIGNMENT 2

K - Means Clustering
&
Gaussian Mixture Models

Group Number 8

Aman Khandelwal	B16007
Amrendra Singh	B16010
Bharat Lodhi	B16015

Contents

List of Plots

List of Tables

1. Problem Description

Classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs on the basis of a training set of data containing observations (or instances) whose category membership is known.

Data-sets:

- Data-set 1: 2-dimensional artificial Non-linearly separable data of 3 classes
- Data-set 2: Real world data-set:
  - a) Two dimensional speech data-set
  - b) 3 class scene image data-set
  - c) Cell Image data-set

Classifiers to be built:

- Bayes' classifier using GMM on all data-sets for same number of mixtures for all classes. Parameters of GMM are to be initialized using K-means clustering.
- Experiments to be performed for different number of mixtures.

2. Solution Approach

To solve the given cases of classification and data analysis we used K-Means Clustering and Gaussian Mixture Model methods.

In the given data sets we don't know the incoming density distribution; hence we cannot predict the classification of data points. This is a case of incomplete data problem where we don't know the actual distribution of the incoming data and hence its parameters. Its parameters depend on each other, none of which we know beforehand.

To overcome this problem we model our data as a combination of Gaussian Mixtures, which is quite a correct solution as most of the data in nature is a mixture of Gaussians. We now use the Expectation–maximization type of algorithm.

We cluster our data into K clusters using the K-Means clustering, initialized using a set of K randomly selected points(which may or may not belong to our sample data). We take the final mean computed by this algorithm to initialize our GMM model considering K gaussian mixtures.

Then we apply the EM algorithm for the GMM to obtain the parameters for the model. We terminate when the log-likelihood function plateaus.

We obtain the mixture coefficients π_k, responsibility terms $\gamma(z_{nk})$, mean vectors and the co-variance matrices for n^{th} data point for the k^{th} cluster, for all the mixture components for all the data points.

We obtain the GMM parameters for all the classes and use those to classify our points into the classes.

For data-set 1(b) and 2(a) we classify each point (represented as a 2-dimensional feature vector).

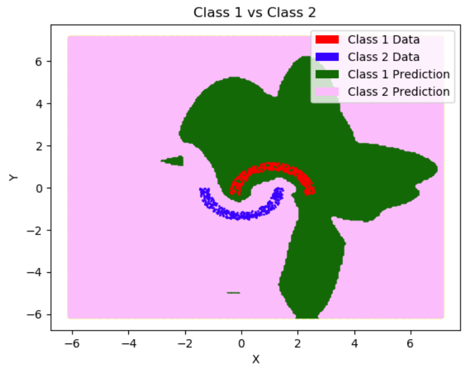
For the data-set 2(b), we have represented each image as a collection of several 24-dimensional feature vectors. To classify an image we first try to classify each of its feature vector into a class. The class to which the maximum number of feature vectors were classified is taken to be the class the image should belong to.
For this dataset, we also obtained the Bag of visual words representation(BOVW). For each image the code-vectors(feature-vectors) which we have obtained upon feature extraction from images are classified into one of the clusters(vocabulary) built from training data and the frequency of code-vectors appearing in each clusters, give us the number of clusters - dimensional BOVW.

For data-set 2(c), we have represented each image as a collection of several 2-dimensional feature vectors (mean and dispersion of the concerned patch). We cluster these feature vectors into 3 clusters using K-means. Then we further sharpen our model by using a Gaussian Mixture Model of 3 mixtures initialized by the K-means algorithm. This helps us to decide that which patch of the image belong to which cluster. This segments the given images into three regions, viz. nucleus, the cell body and the background.

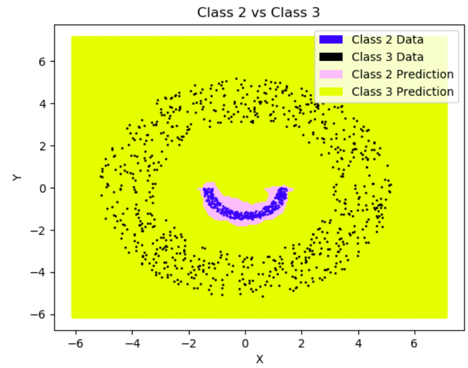
3. Results

1 Data Set 1(b)

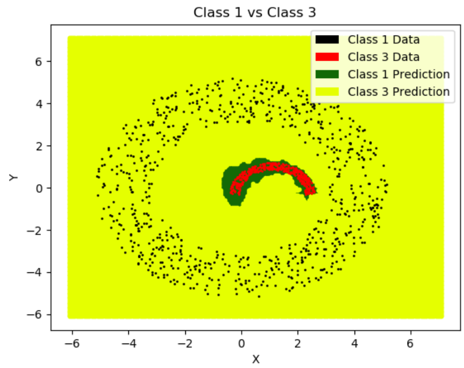
1.1 Classification Using GMM



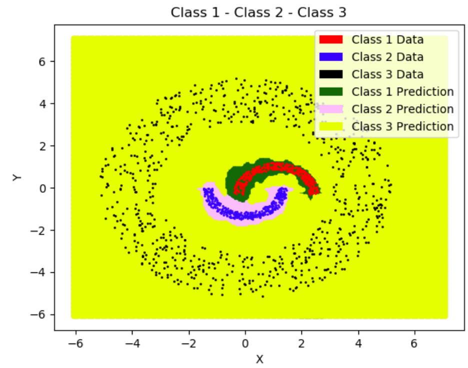
(a) Decision regions and Training data Class 1 and 2



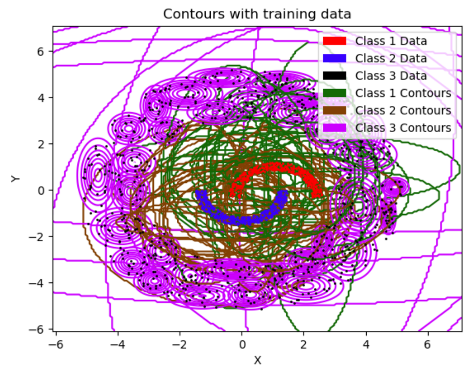
(b) Decision regions and Training data Class 2 and 3



(c) Decision regions and Training data Class 1 and 3



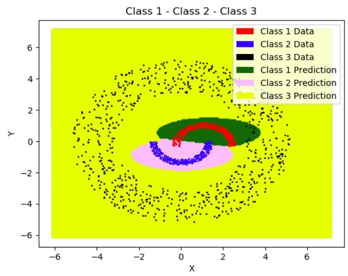
(d) Decision regions and Training data Class 1, 2, 3



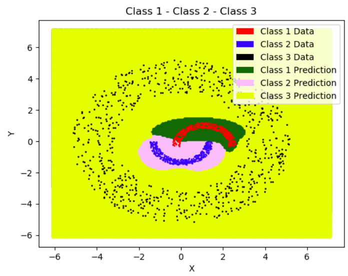
(e) Contours: Class 1, 2, 3

Figure 3..1. Non-Linear Data - Classification using GMM 32 cluster

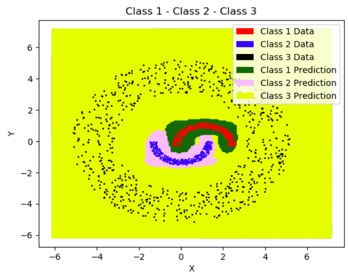
Decision Region and Contour Plots for 1, 2, 4, 8, 16 Clusters



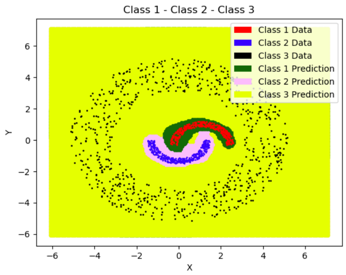
(a) Mixtures = 1



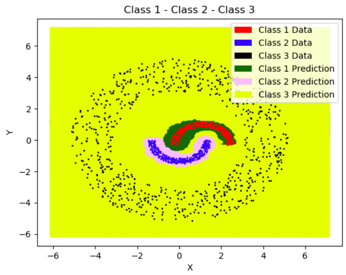
(b) Mixtures = 2



(c) Mixtures = 4

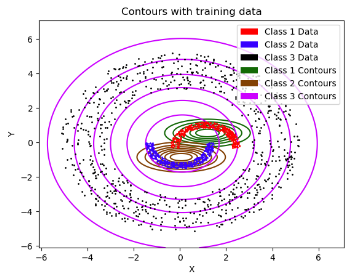


(d) Mixtures = 8

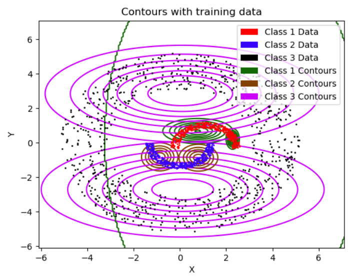


(e) Clusters = 16

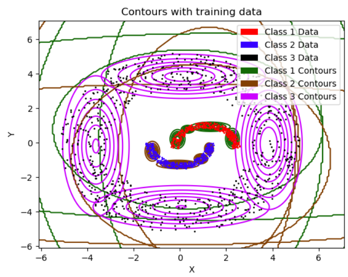
Figure 3..2. Non Linear Data(1b)- GMM Decision Regions



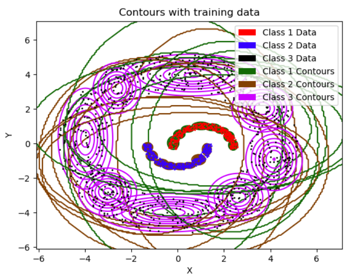
(a) Mixtures = 1



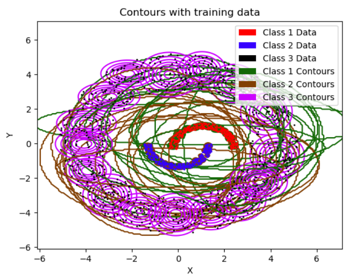
(b) Mixtures = 2



(c) Mixtures = 4



(d) Mixtures = 8



(e) Mixtures = 16

Figure 3..3. Non Linear Data(1b)- GMM Contours for different mixtures

Confusion Matrix, Precision, Recall and F-measure

	Class1	Class2	Class 3
Class1	125	0	0
Class2	0	125	0
Class3	0	0	250

(a) Confusion Matrix

	Class1	Class2	Class3
Precision	1.0	1.0	1.0
Recall	1.0	1.0	1.0
F-Measure	1.0	1.0	1.0

(b) Analysis

Accuracy	100%
Mean Precision	1.0
Mean Recall	1.0
Mean F-Measure	1.0

(c) Result

Table 3..1. Non Linear Data - GMM(32 Clusters) : Class 1 ,Class 2 and Class 3

	Class1	Class2	Class 3
Class1	125	0	0
Class2	0	125	0
Class3	0	0	250

(a) Confusion Matrix

	Class1	Class2	Class3
Precision	1.0	1.0	1.0
Recall	1.0	1.0	1.0
F-Measure	1.0	1.0	1.0

(b) Analysis

Accuracy	100%
Mean Precision	1.0
Mean Recall	1.0
Mean F-Measure	1.0

(c) Result

Table 3..2. Non Linear Data - GMM(16 Clusters) : Class 1 ,Class 2 and Class 3

	Class1	Class2	Class 3
Class1	125	0	0
Class2	0	125	0
Class3	0	0	250

(a) Confusion Matrix

	Class1	Class2	Class3
Precision	1.0	1.0	1.0
Recall	1.0	1.0	1.0
F-Measure	1.0	1.0	1.0

(b) Analysis

Accuracy	100%
Mean Precision	1.0
Mean Recall	1.0
Mean F-Measure	1.0

(c) Result

Table 3..3. Non Linear Data - GMM(8 Clusters) : Class 1 ,Class 2 and Class 3

	Class1	Class2	Class 3
Class1	125	0	0
Class2	0	125	0
Class3	0	0	250

(a) Confusion Matrix

	Class1	Class2	Class3
Precision	1.0	1.0	1.0
Recall	1.0	1.0	1.0
F-Measure	1.0	1.0	1.0

(b) Analysis

Accuracy	100%
Mean Precision	1.0
Mean Recall	1.0
Mean F-Measure	1.0

(c) Result

Table 3..4. Non Linear Data - GMM(4 Clusters) : Class 1 ,Class 2 and Class 3

	Class1	Class2	Class 3
Class1	117	8	0
Class2	2	123	0
Class3	0	0	250

(a) Confusion Matrix

	Class1	Class2	Class3
Precision	0.9831	0.9389	1.0
Recall	0.936	0.984	1.0
F-Measure	0.9590	0.9609	1.0

(b) Analysis

Accuracy	97.99%
Mean Precision	0.9740
Mean Recall	0.9733
Mean F-Measure	0.9733

(c) Result

Table 3..5. Non Linear Data - GMM(2 Clusters) : Class 1 ,Class 2 and Class 3

	Class1	Class2	Class 3
Class1	116	9	0
Class2	9	116	0
Class3	6	0	244

(a) Confusion Matrix

	Class1	Class2	Class3
Precision	0.8854	0.928	1.0
Recall	0.928	0.928	0.976
F-Measure	0.9062	0.927	0.987

(b) Analysis

Accuracy	95.19%
Mean Precision	0.9378
Mean Recall	0.9439
Mean F-Measure	0.9407

(c) Result

Table 3..6. Non Linear Data - GMM(1 Clusters) : Class 1 ,Class 2 and Class 3

	Accuracy
Classifier1	45.00%
Classifier2	45.00%
Classifier3	97.19%
Classifier4	96.99%

(a) Bayes classifier

	Accuracy
Cluster=1	95.19%
Cluster=2	97.99%
Cluster=4	100.00%
Cluster=8	100.00%
Cluster=16	100.00%
Cluster=32	100.00%

(b) GMM

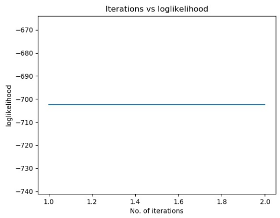
Table 3..7. Comparison between accuracy of Bayes' classifier and GMM for dataset1(b)

Inferences:

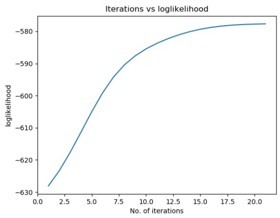
1. Here as the number of clusters increase we can observe that accuracy increases. Here, interestingly, in case of 32 clusters case we have 32(number of clusters) *(2 4(Mean-vector) + 24(diagonal co-variance matrix)) free variables, which are more than total training samples we were having, but we not observed any kind of over-fitting. This might be due to the fact that there was not much difference between training data and test data.

2. Here for the case of 1 cluster in GMM, we are observing almost same accuracy as observed for classifier 3 and 4 of Bayes', which should be the case as single cluster Gaussian means single Gaussian and also in classifier 3 we have taken diagonal matrix. As, in GMM it doesn't matter much if we take diagonal or full co-variance matrix, so our single cluster GMM case is also comparable with Bayes' classifier-4 result.

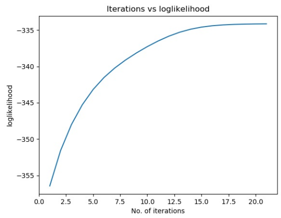
Log-likelihood vs Iterations



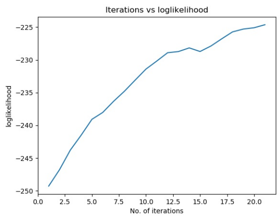
(a) Clusters = 1



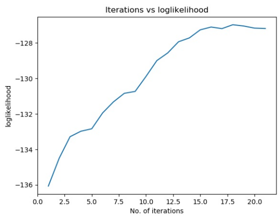
(b) Clusters = 2



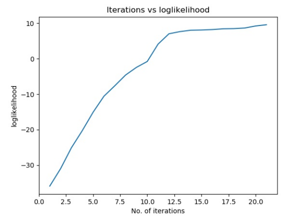
(c) Clusters = 4



(d) Clusters = 8

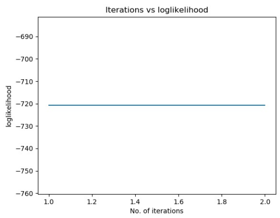


(e) Clusters = 16

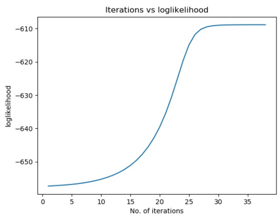


(f) Clusters = 32

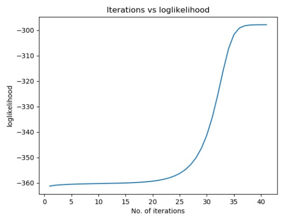
Figure 3..4. Non Linear Data(1b) Class 1 - GMM log-likelihood vs Iterations plots



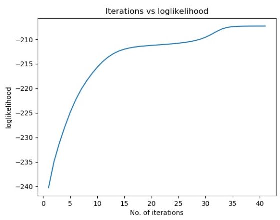
(a) Clusters = 1



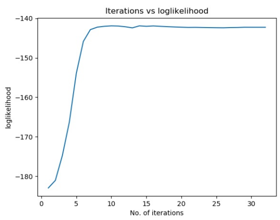
(b) Clusters = 2



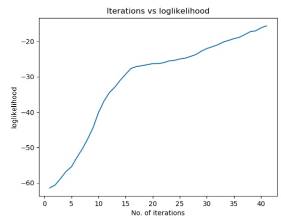
(c) Clusters = 4



(d) Clusters = 8

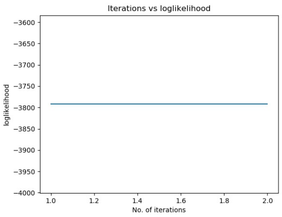


(e) Clusters = 16

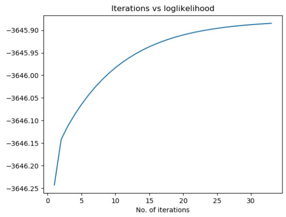


(f) Clusters = 32

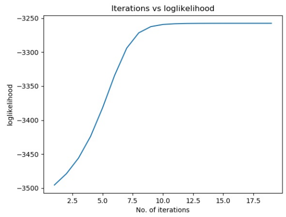
Figure 3..5. Non Linear Data(1b) Class 2 - GMM log-likelihood vs Iterations plots



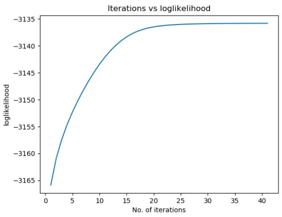
(a) Clusters = 1



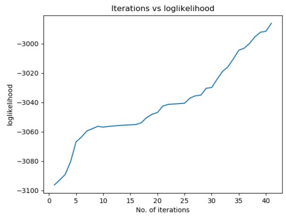
(b) Clusters = 2



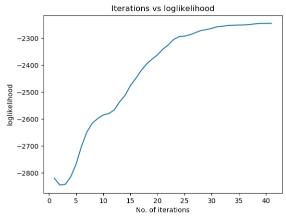
(c) Clusters = 4



(d) Clusters = 8



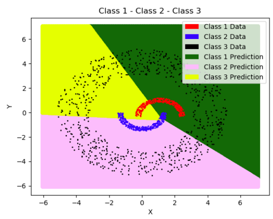
(e) Clusters = 16



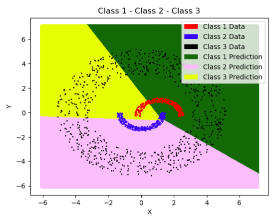
(f) Clusters = 32

Figure 3..6. Non Linear Data(1b) Class 3 - GMM log-likelihood vs Iterations plots

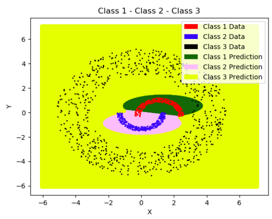
Comparison with Bayes Classifier



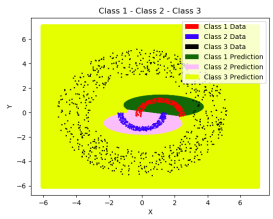
(a) classifier1



(b) classifier2



(c) classifier3



(d) classifier4

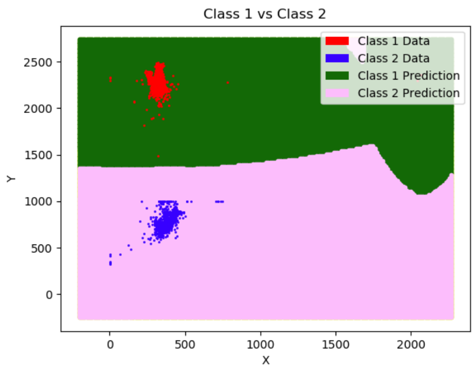
Figure 3..7. Non Linear Data - Bayes Classifier decision regions

Inferences

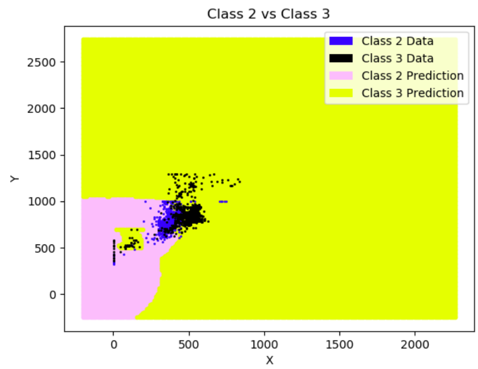
1. From figures 3..1 and 3..17 we can say that GMM gives more complex and better estimate of decision boundary than Bayes' classifier which improve classification accuracy, while using GMM.
2. In both Bayes classifier 3 and GMM we have taken co-variance matrix to be diagonal and also the nature of boundary is non-linear. But the horse-shoe type distribution of data is best fitted by multivariate Gaussian mixture model because multiple Gaussians can take the shape of horse-shoe type distribution.

2 Data Set 2(a)

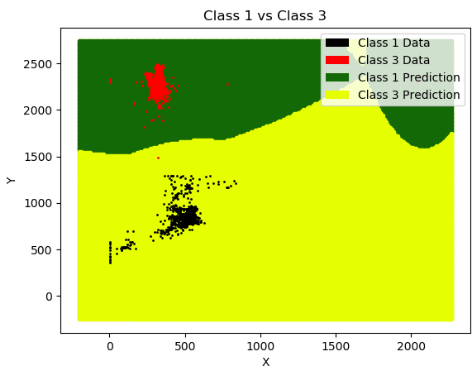
2.1 Classification Using GMM



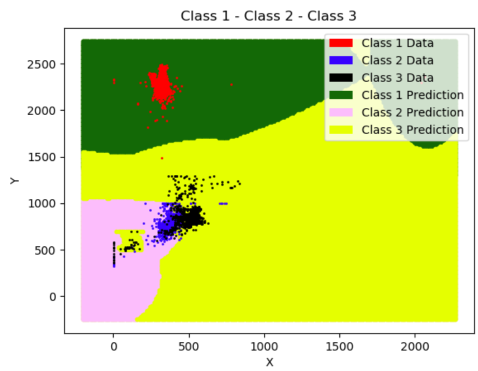
(a) Decision regions and Training data Class 1 and 2



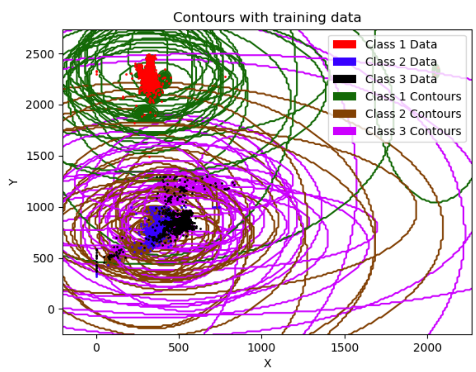
(b) Decision regions and Training data Class 2 and 3



(c) Decision regions and Training data Class 1 and 3



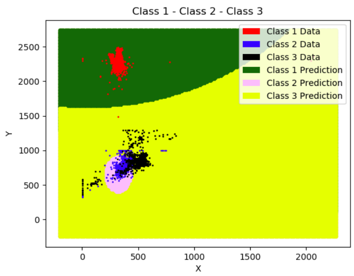
(d) Decision regions and Training data Class 1, 2, 3



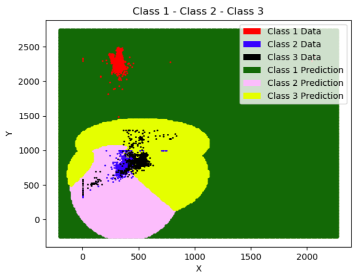
(e) Contours: Class 1, 2, 3

Figure 3..8. Real world Data - Decision regions using GMM 32 cluster

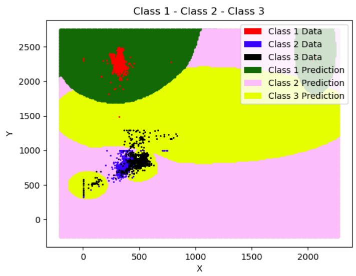
Decision Region and Contour Plots for 1, 2, 4, 8, 16 Clusters



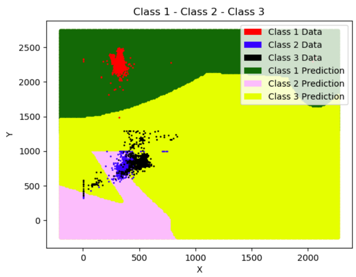
(a) Mixtures = 1



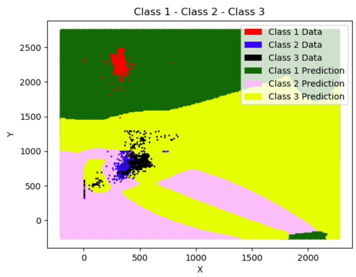
(b) Mixtures = 2



(c) Mixtures = 4

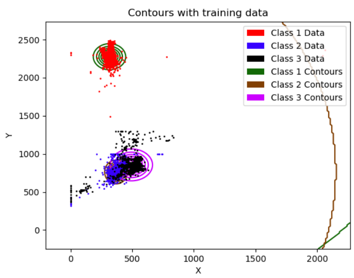


(d) Mixtures = 8

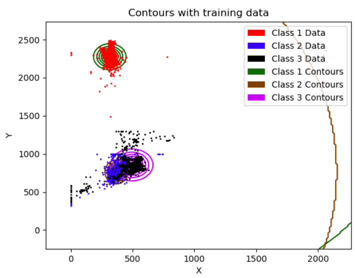


(e) Mixtures = 16

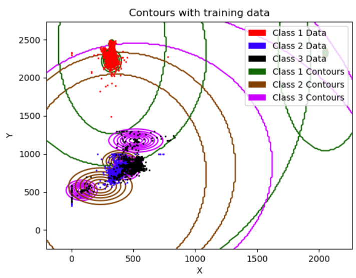
Figure 3..9. Real world Data(2a) Decision regions for GMM



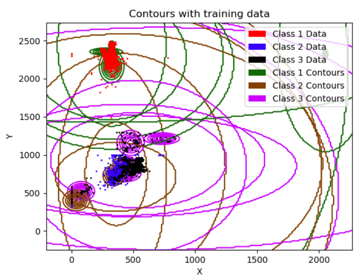
(a) Mixtures = 1



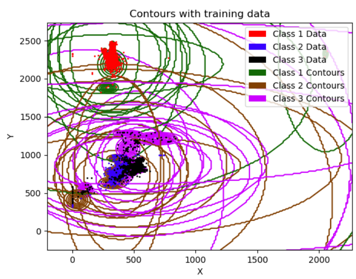
(b) Mixtures = 2



(c) Mixtures = 4



(d) Mixtures = 8



(e) Mixtures = 16

Figure 3..10. Real world Data(2a) Contours for GMM

Confusion Matrix, Precision, Recall and F-measure

	Class1	Class2	Class 3
Class1	585	3	9
Class2	0	584	38
Class3	0	185	429

(a) Confusion Matrix

	Class1	Class2	Class3
Precision	1.0	0.7564	0.9012
Recall	0.9798	0.9389	0.6986
F-Measure	0.9898	0.8378	0.7871

(b) Analysis

Accuracy	87.17%
Mean Precision	0.8859
Mean Recall	0.8725
Mean F-Measure	0.8716

(c) result

Table 3..8. Real world Data(2a) - GMM(Clusters=32) :
Class 1 ,Class 2 and Class 3

	Class1	Class2	Class 3
Class1	594	2	1
Class2	0	581	41
Class3	0	193	421

(a) Confusion Matrix

	Class1	Class2	Class3
Precision	1.0	0.7487	0.9092
Recall	0.9949	0.9340	0.6856
F-Measure	0.9674	0.8311	0.7818

(b) Analysis

Accuracy	87.07%
Mean Precision	0.8859
Mean Recall	0.8715
Mean F-Measure	0.8701

(c) result

Table 3..9. Real world Data(2a) - GMM(Clusters=16) :
Class 1 ,Class 2 and Class 3

	Class1	Class2	Class 3
Class1	587	3	7
Class2	0	596	26
Class3	0	222	392

(a) Confusion Matrix

	Class1	Class2	Class3
Precision	1.0	0.7259	0.9223
Recall	0.9832	0.9581	0.6384
F-Measure	0.9915	0.8260	0.7545

(b) Analysis

Accuracy	85.92%
Mean Precision	0.8827
Mean Recall	0.8599
Mean F-Measure	0.8573

(c) result

Table 3..10. Real world Data(2a) - GMM(Clusters=8) :
Class 1 ,Class 2 and Class 3

	Class1	Class2	Class 3
Class1	581	7	9
Class2	0	585	37
Class3	0	240	374

(a) Confusion Matrix

	Class1	Class2	Class3
Precision	1.0	0.7031	0.8904
Recall	0.9731	0.9405	0.6091
F-Measure	0.9864	0.8046	0.7234

(b) Analysis

Accuracy	84.01%
Mean Precision	0.8645
Mean Recall	0.8409
Mean F-Measure	0.8381

(c) result

Table 3..11. Real world Data(2a) - GMM(Clusters=4) :
Class 1 ,Class 2 and Class 3

	Class1	Class2	Class 3
Class1	594	3	0
Class2	0	582	40
Class3	1	76	537

(a) Confusion Matrix

	Class1	Class2	Class3
Precision	0.9983	0.8804	0.9306
Recall	0.9949	0.9356	0.8745
F-Measure	0.9966	0.9072	0.9017

(b) Analysis

Accuracy	93.45%
Mean Precision	0.9364
Mean Recall	0.9350
Mean F-Measure	0.9352

(c) result

Table 3..12. Real world Data(2a) - GMM(Clusters=2) :
Class 1 ,Class 2 and Class 3

	Class1	Class2	Class 3
Class1	580	2	15
Class2	0	546	76
Class3	0	185	429

(a) Confusion Matrix

	Class1	Class2	Class3
Precision	1.0	0.7448	0.825
Recall	0.9715	0.8778	0.6986
F-Measure	0.9855	0.8059	0.7566

(b) Analysis

Accuracy	84.83%
Mean Precision	0.8566
Mean Recall	0.8493
Mean F-Measure	0.8493

(c) result

Table 3..13. Real world Data(2a) - GMM(Clusters=1) :
Class 1 ,Class 2 and Class 3

	Accuracy
Classifier1	87.67
Classifier2	87.17
Classifier3	84.83
Classifier4	83.08

(a) Bayes classifier

	Accuracy
Cluster=1	84.83%
Cluster=2	93.45%
Cluster=4	84.01%
Cluster=8	85.92%
Cluster=16	87.07%
Cluster=32	87.17%

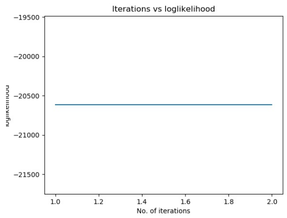
(b) GMM

Table 3..14. Comparison between accuracy of Bayes' classifier and GMM for dataset2(a)

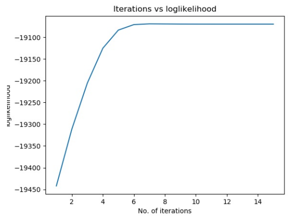
Inferences:

Here as we are increasing the number of clusters the accuracy at first is decreasing and is highest for the case when number of clusters is 2. This might be due to the fact that given random real world data and taking 32 clusters on it, means that we have 32(number of clusters) * (24(Mean-vector) + 24(diagonal co-variance matrix)) number of degrees of freedom. Given this number of freedoms, the training data was much less, as the case should be for each degree of freedom we should have 10 training examples but here this was not the case. So there might be over-fitting when the number of clusters were taken higher.

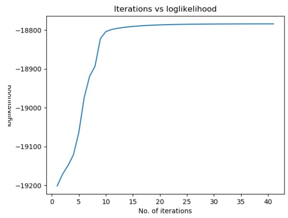
Log-likelihood vs Iterations



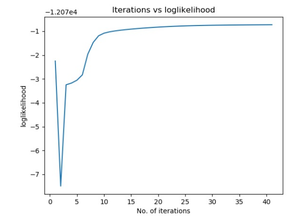
(a) Mixtures = 1



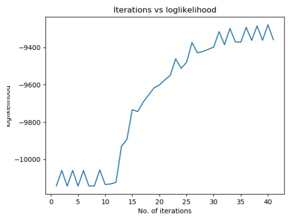
(b) Mixtures = 2



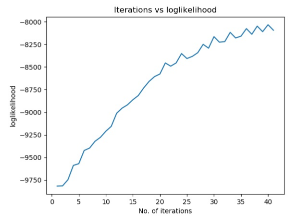
(c) Mixtures = 4



(d) Mixtures = 8

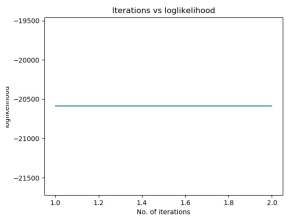


(e) Mixtures = 16

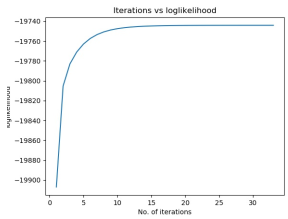


(f) Mixtures = 32

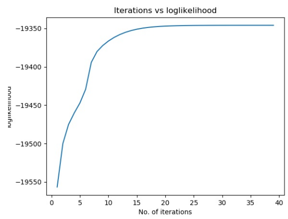
Figure 3..11. Real world Data(2a)- Class1



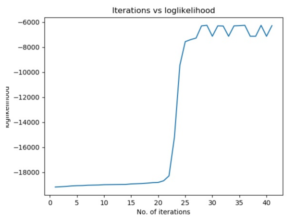
(a) Mixtures = 1



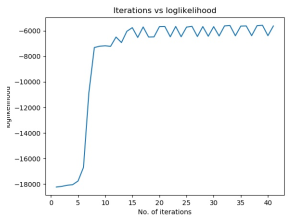
(b) Mixtures = 2



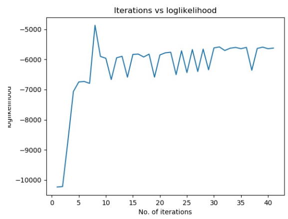
(c) Mixtures = 4



(d) Mixtures = 8

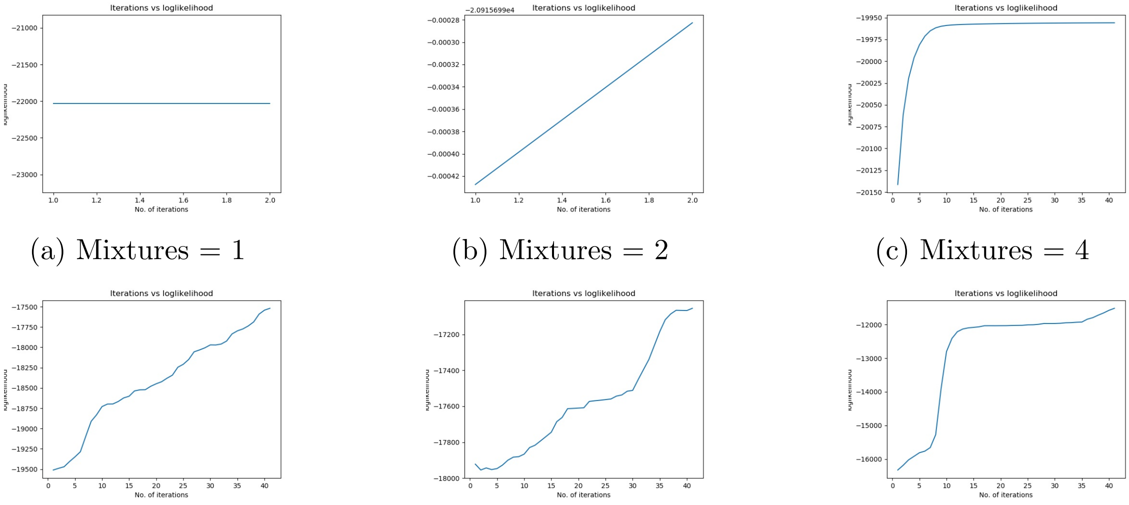


(e) Mixtures = 16



(f) Mixtures = 32

Figure 3..12. Real world Data(2a) - Class2

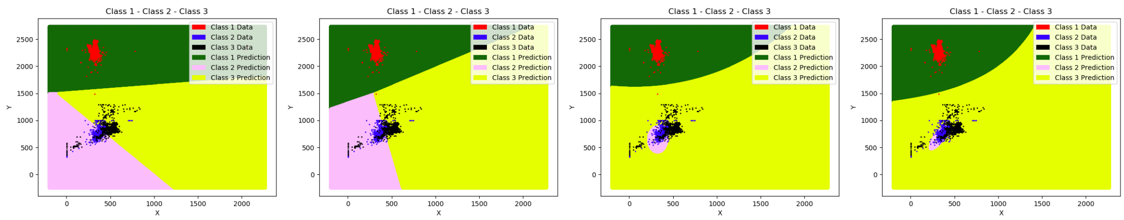


(a) Mixtures = 1 (b) Mixtures = 2 (c) Mixtures = 4

(d) Mixtures = 8 (e) Mixtures = 16 (f) Mixtures = 32

Figure 3..13. Real world Data(2a) - Class3

Comparison with Bayes Classifier



(a) classifier1 (b) classifier2 (c) classifier3 (d) classifier4

Figure 3..14. Real world Data(2a)- Bayes Classifier

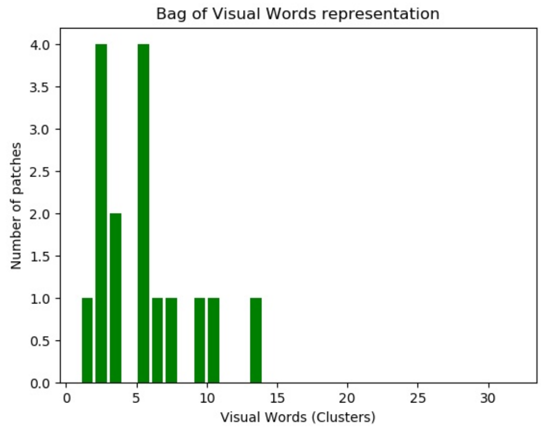
Inferences:

1. In this we can observe that we have overlapping data for classes 2 and 3, so Bayes' classifier here has done hard clustering right away, but GMM keeps some probability associated for each point to every cluster till the end so this also helps us to better estimate the overlapping points.

2. If done soft assignment too(overlapping regions) then it also gives better estimate of relation between different classes.

3 Data Set 2(b)

3.1 32 -dimensional Bag of Visual Words Representation



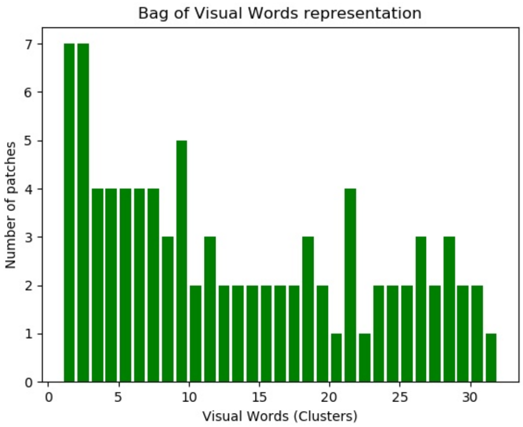


Figure 3..15. Dataset(2b) Bag of Visual Words for Class Bayou

Figure 3..16. Dataset(2b) Bag of Visual Words for Class Chalet

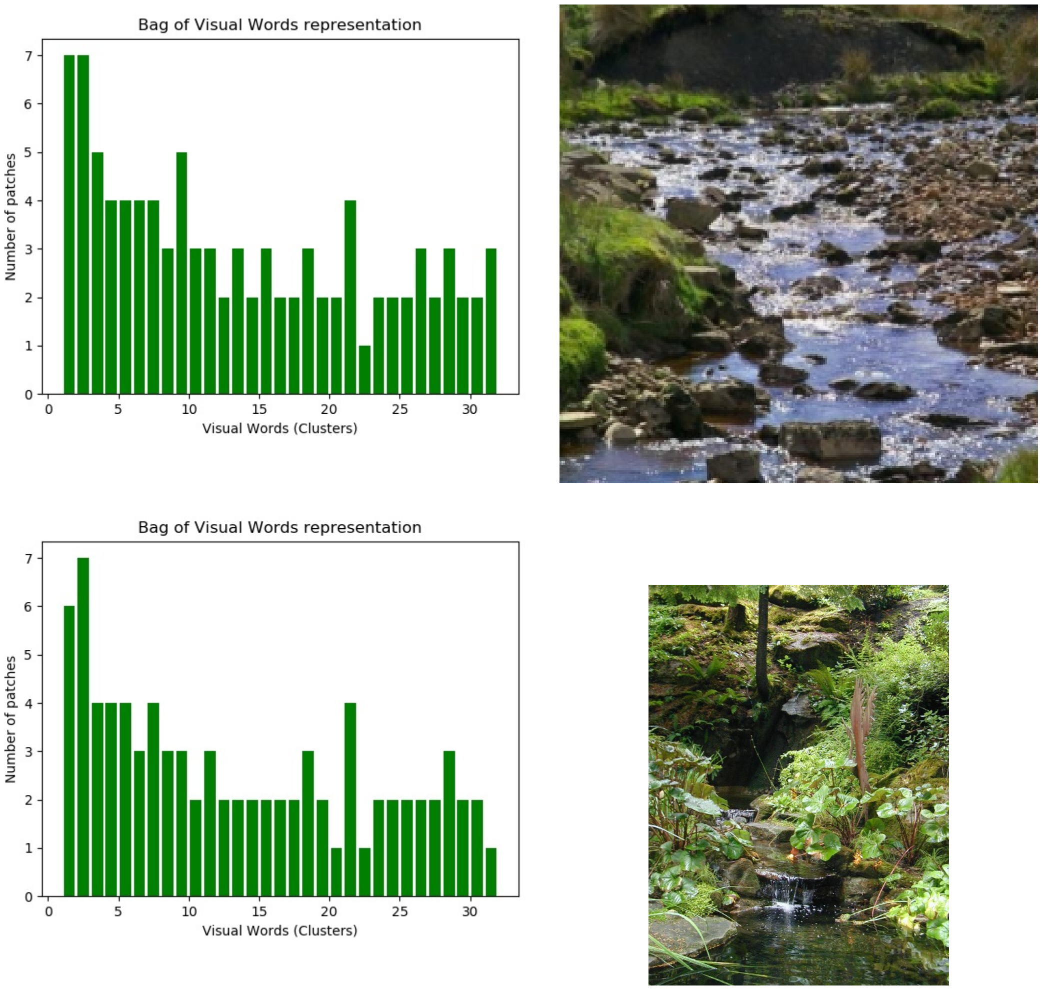


Figure 3..17. Dataset(2b) Bag of Visual Words for Class Creek

3.2 Classification Using GMM

Confusion Matrix, Precision, Recall and F-measure

	Bayou	Chalet	Creek
bayou	26	17	7
chalet	15	27	8
creek	17	14	19

(a) Confusion Matrix

	Bayou	Chalet	Creek
Precision	0.4482	0.4655	0.5588
Recall	0.52	0.54	0.38
F-Measure	0.4814	0.5	0.4523

(b) Analysis

Accuracy	47.99%
Mean Precision	0.4908
Mean Recall	0.4799
Mean F-Measure	0.4779

(c) Result

Table 3..15. GMM using 32 cluster: bayou ,chalet and creek

	Bayou	Chalet	Creek
bayou	20	17	13
chalet	18	23	9
creek	16	16	18

(a) Confusion Matrix

	Bayou	Chalet	Creek
Precision	0.3703	0.4107	0.45
Recall	0.4	0.46	0.36
F-Measure	0.3846	0.4339	0.4

(b) Analysis

Accuracy	40.66%
Mean Precision	0.4103
Mean Recall	0.4066
Mean F-Measure	0.4061

(c) Result

Table 3..16. GMM using 16 cluster: bayou ,chalet and creek

	Bayou	Chalet	Creek
bayou	16	26	8
chalet	9	35	6
creek	9	24	17

(a) Confusion Matrix

	Bayou	Chalet	Creek
Precision	0.4705	0.4117	0.5483
Recall	0.32	0.7	0.34
F-Measure	0.3809	0.5185	0.4197

(b) Analysis

Accuracy	45.33%
Mean Precision	0.4769
Mean Recall	0.4533
Mean F-Measure	0.4397

(c) Result

Table 3..17. GMM using 8 cluster: bayou ,chalet and creek

	Bayou	Chalet	Creek
bayou	10	14	26
chalet	12	25	13
creek	10	11	29

(a) Confusion Matrix

	Bayou	Chalet	Creek
Precision	0.3125	0.4166	0.4264
Recall	0.166	0.5	0.58
F-Measure	0.2173	0.4545	0.4915

(b) Analysis

Accuracy	40.00%
Mean Precision	0.3852
Mean Recall	0.4155
Mean F-Measure	0.3878 height

(c) Result

Table 3..18. GMM using 4 cluster: Bayou ,Chalet and Creek

	Bayou	Chalet	Creek
bayou	6	14	30
chalet	11	7	32
creek	6	20	24

(a) Confusion Matrix

	Bayou	Chalet	Creek
Precision	0.2608	0.1707	0.2790
Recall	0.12	0.14	0.48
F-Measure	0.1643	0.1538	0.3529

(b) Analysis

Accuracy	24.66%
Mean Precision	0.2368
Mean Recall	0.2466
Mean F-Measure	0.2237 height

(c) Result

Table 3..19. GMM using 2 cluster: Bayou ,Chalet and Creek

	Bayou	Chalet	Creek
bayou	14	4	32
chalet	7	17	26
creek	6	6	38

(a) Confusion Matrix

	Bayou	Chalet	Creek
Precision	0.5185	0.6296	0.3958
Recall	0.28	0.34	0.76
F-Measure	0.3636	0.4415	0.5205

(b) Analysis

Accuracy	46.00%
Mean Precision	0.5146
Mean Recall	0.46
Mean F-Measure	0.4419

(c) Result

Table 3..20. GMM using 1 cluster: Bayou ,Chalet and Creek

Inferences:

Here in case of GMM on images the accuracy is low, reasons might be -

1. GMM is a data-driven approach, we were having very less number of images to train on.
2. Due to high dimensional data, the data was sparse and we can't have any advantage of GMM probabilistic association of every point with every cluster.
3. Initialization of parameters also play important role in GMM, might be case that initialization through K-means is not really a good choice.

4 Data Set 2(c)

4.1 K-Means Clustering and GMM

Scatter Plots for K-Means for progressive iterations (till termination with given threshold)

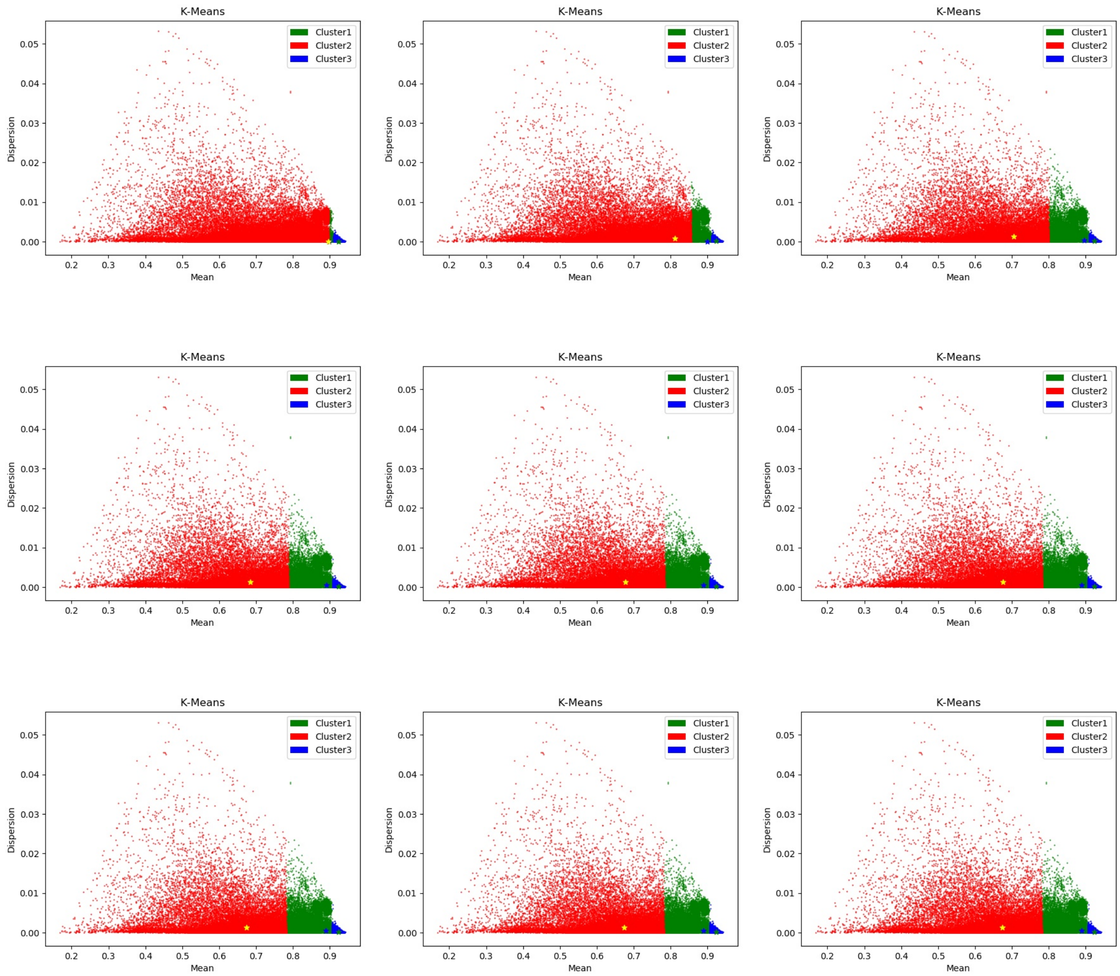
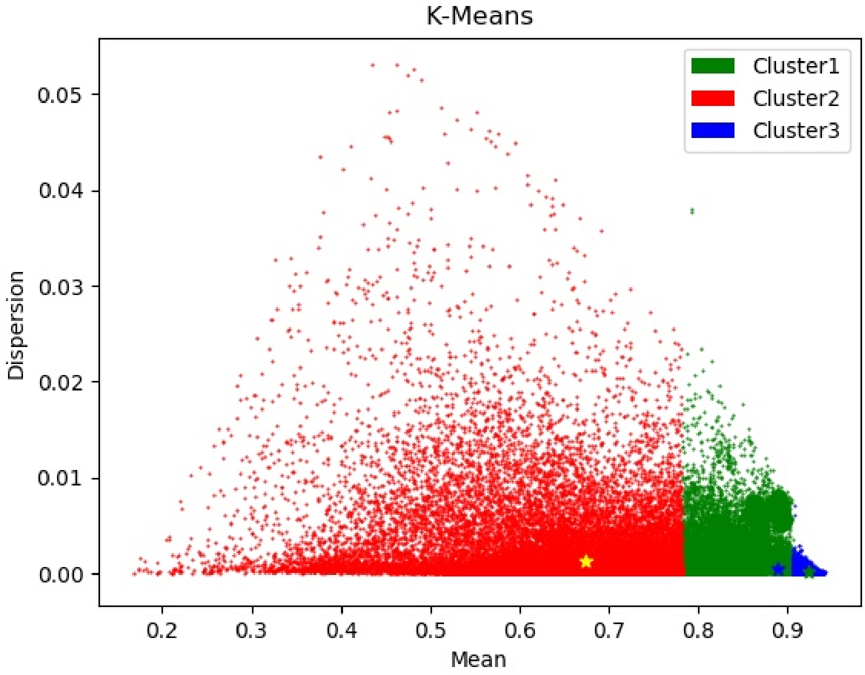


Figure 3..18. Scatter plot progress with iterations

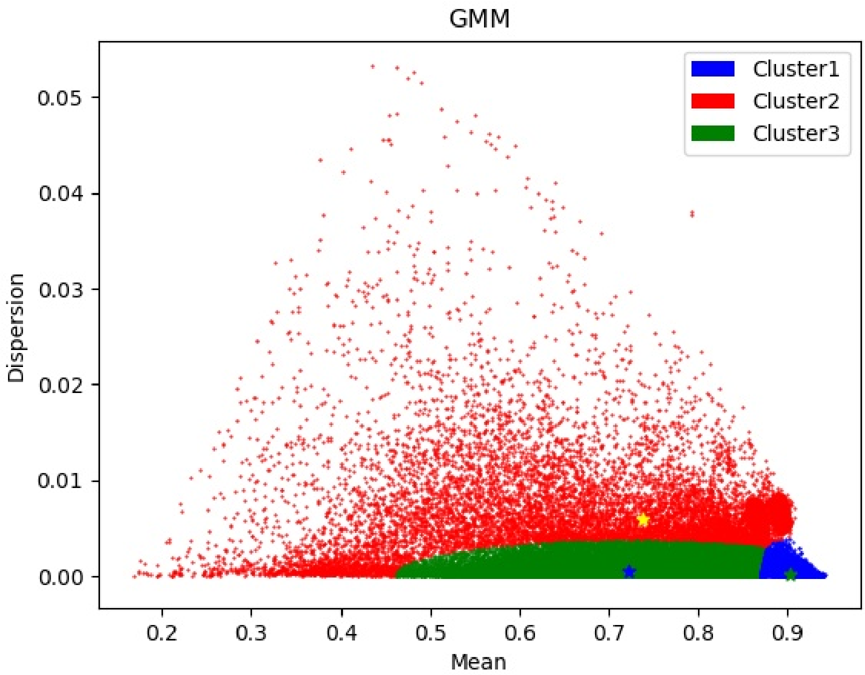
Inferences

1. Initially we have taken 3 random points as cluster centres then as iteration progresses data distribution become more accurate for 3 clusters. After some iterations data get divided in three clusters such that from now on, by increasing iteration there is no significant change in cluster distribution.

Scatter Plot K-Means & GMM

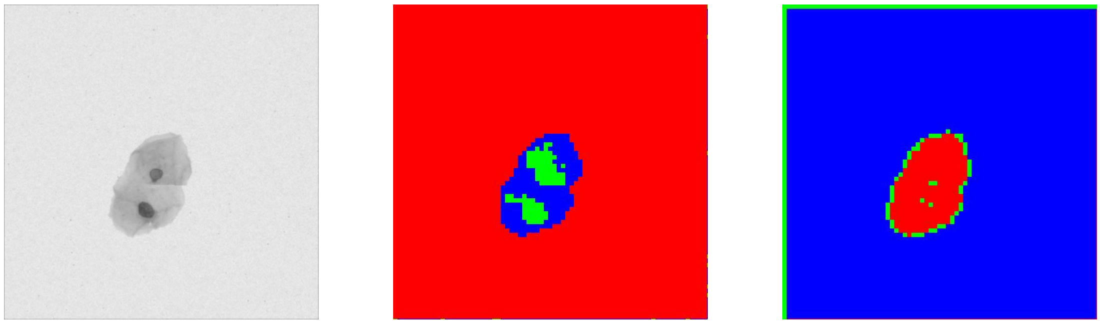


(a) K-Means



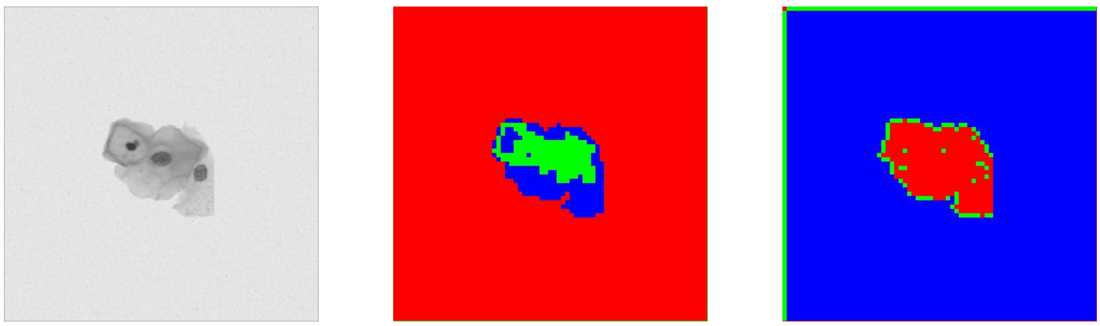
(b) GMM

Result of cluster projected on test images



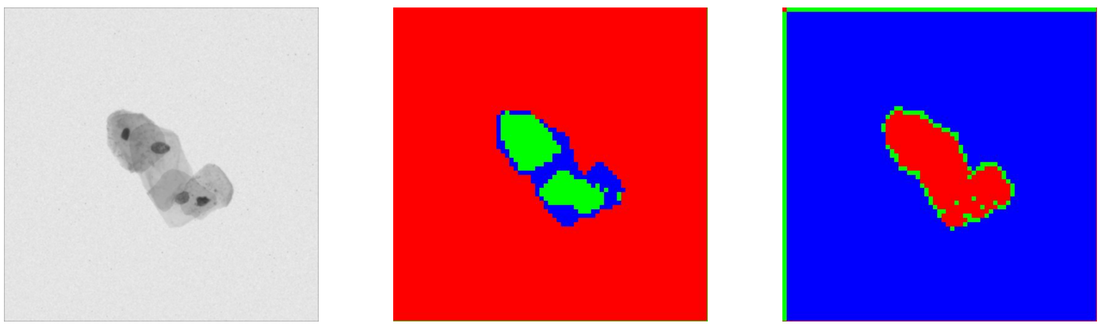
(a) Original Test Image (b) K-Means Clustering (c) GMM Segmentation

Figure 3..20. Test Image 1 Segmentation Overlay



(a) Original Test Image (b) K-Means Clustering (c) GMM Segmentation

Figure 3..21. Test Image 2 Segmentation Overlay



(a) Original Test Image (b) K-Means Clustering (c) GMM Segmentation

Figure 3..22. Test Image 3 Segmentation Overlay

Inferences

Observations:

1. K means is roughly able to segment the cell images into 3 clusters, the nucleus, cell body and the background.

2. GMM is a stronger model as compared to the K-means clustering due to the fact that we were able to distinguish nucleus in case of GMM.

The results obtained were not so accurate because of the following reasons:

1. The data used was non overlapping 7 x 7 patches. This resulted in the output image being coarse.
2. The nucleus was a small region. Many of the non-overlapping patches might have been covered in part by the nucleus region. Hence the nucleus was not so clearly segmented by any of the clustering technique.
3. We could have obtained much better results if we have used overlapping patches.

4. Conclusion & Inferences

1. GMM gives better result than simple Bayes' classifier because in GMM for a particular class we have K Gaussians which together on combining can give any complex decision boundary with other classes while in simple Bayes' classifier boundary can be only linear or hyper quadratic in nature.

2. We can not in advance say about number of Gaussians to take for each class, it is good to check accuracy by taking different number of Gaussian(clusters) then take number of clusters which give good accuracy.

3. Initialization of GMM parameters should be appropriate for better result in our case we initialized parameters of GMM from output of K-means clustering.

4. K-means try to fit the data in circular clusters in 2-D case (in higher dimensions, hyper-sphere clusters) so for eg. if there is a cluster which has to be elliptical in nature, K-means will make a circular fit for it, thus giving us a poor fit.

5. Lack of probabilistic cluster assignment in K-means. In K-means on the basis of distance we directly classify a point in cluster, this is what we call Hard Clustering. Thus for low-dimensional data-sets it might not perform well whereas in case of GMM we keep associate a certain probability of every point to be classified in any of the given cluster, so we can have overlapping decision regions, this is called Soft Clustering.

6. In iteration vs log likelihood graph as iteration increases first log likelihood value increases then become nearly constant after some iteration showing convergence.

Bibliography

[1] K-Means clustering
https://en.wikipedia.org/wiki/K-means_clustering https://www.geeksforgeeks.org/k-means-clustering-introduction/
https://towardsdatascience.com/understanding-k-means-clustering-\in-machine-learning-6a6e67336aa1

[2] GMM
https://en.wikipedia.org/wiki/Mixture_model http://www.computerrobotvision.org/2010/tutorial_day/GMM_said_crv10_tutorial.pdf
https://www.cse.iitk.ac.in/users/piyush/courses/pml_winter16/slides_lec7.pdf

[3] Bayes Classifier
https://en.wikipedia.org/wiki/Bayes_classifier

[4] Statistical Classification
https://en.wikipedia.org/wiki/Statistical_classification

[5] Naive Bayes Classifier
https://en.wikipedia.org/wiki/Naive_Bayes_classifier

[6] Matplotlib Contours
https://matplotlib.org/api/_as_gen/matplotlib.pyplot.contour.html

[7] Stack Overflow
https://stackoverflow.com