Signal Processing and Control in Neural Systems

Maryam M. Shanechi

Dept. of Electrical Engineering

University of Southern California

Project title

Clustering with Gaussian Mixture Model(GMM) and Expectation Maximization(EM)

Submitted by,

Akshay Hegde

hegdeaks@usc.edu

USC ID: 8099460970

Clustering with Gaussian Mixture Model(GMM) and Expectation Maximization(EM)

1. Abstract:

Organizing large amount of data is the biggest challenge in domains of data mining and cloud computing. One way to deal with these kind of issues is by Clustering. Clustering is a widely-used technique for discovering patterns in original data and to take care and manage datasets that consists of unconditional characteristics and attributes. EM clustering is one of the simplest unsupervised learning algorithms that solve the real-life clustering problems. This paper presents a way to implement clustering with Gaussian Mixture Model(GMM) and Expectation Maximization(EM) algorithm for solving 'Wheat strain classification problem'. We have an unlabeled dataset on wheat, 'seeds.txt', a classic multivariate dataset which consists of samples from each of three species of Wheat (Kama, Rosa and Canadian). This paper explains the use of GMM and EM Clustering Algorithm to distinguish the Wheat strains from each other.

2. Introduction:

2.1 Clustering:

Clustering is the process of partitioning a group of data points into a small number of clusters. In general, clustering uses iterative techniques to group cases in a dataset into clusters that contain similar characteristics. These groupings are useful for exploring data, identifying anomalies in the data, and eventually for making predictions. Clustering models also can help you identify relationships in a dataset that you might not logically derive by browsing or simple observation. Unlike the classification algorithm, clustering belongs to the unsupervised type of algorithms. Two representatives of the clustering algorithms, which allow model refining of an iterative process to find the best congestion,

1. K- Means – Hard clustering.
2. Expectation Maximization (EM) algorithm. – Soft clustering.

2.2 Gaussian Mixture Model(GMM):

Gaussian mixture models are a probabilistic model for representing normally distributed subpopulations within an overall population. Mixture models in general don't require knowing which subpopulation a data point belongs to, allowing the model to learn the subpopulations automatically. Since subpopulation assignment is not known, this constitutes a form of unsupervised learning.

A Gaussian mixture model is parameterized by two types of values, the mixture component weights and the component means and variances/covariances. For a Gaussian mixture model with K components, the component has a mean and variance of for the univariate case and a mean and covariance matrix for the multivariate case. The mixture component weights are defined as for component $C_k,$ ϕ_k with the constraint that so that the total $\sum_{i=1}^{K} \phi_i = 1$ probability distribution normalizes to 1.

One dimensional Gaussian Model vs K dimensional Gaussian Model,

$$p(x) = \sum_{i=1}^{K} \phi_i \mathcal{N}(x \mid \mu_i, \sigma_i)$$

$$\mathcal{N}(x \mid \mu_i, \sigma_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2} \right)$$

$$\sum_{i=1}^{K} \phi_i = 1$$

$$p(\vec{x}) = \sum_{i=1}^{K} \phi_i \mathcal{N}(\vec{x} \mid \vec{\mu}_i, \Sigma_i)$$

$$\mathcal{N}(\vec{x} \mid \vec{\mu}_i, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^K |\Sigma_i|}} \exp\left(-\frac{1}{2} (\vec{x} - \vec{\mu}_i)^{\mathrm{T}} \Sigma_i^{-1} (\vec{x} - \vec{\mu}_i) \right)$$

$$\sum_{i=1}^{K} \phi_i = 1$$

The probability given in the mixture of K Gaussians is,

$$p(x) = \sum_{j=1}^{K} w_j \cdot N(x \mid \mu_j, \Sigma_j)$$

Where wj is the prior probability (weight) of the jth Gaussian.

3. *About Data set used:*

‘Seeds.txt’, a classic multivariate dataset which consists of samples from each of three species of Wheat which are Kama, Rosa and Canadian. The dataset is taken from the ‘UCI Machine Learning Repository’ and more data is added to it for testing which are taken from ‘Awesome Public Datsets, Github.’

Links:

https://archive.ics.uci.edu/ml/datasets/seeds

https://github.com/caesar0301/awesome-public-datasets

- There are 1000 sample instances and Seven features were measured from each sample: Area A, Perimeter P, Compactness C, Length of kernel, Width of kernel, symmetry coefficient, length of kernel groove.
- Dataset characteristics – Multivariate

- Attribute characteristics – Real Continuous
- There is some redundancy in the seven input variables, so it is possible to achieve a good solution with only two of them.

Features selected for clustering:

- Perimeter of the seed
- Length of kernel

It is concluded as the dataset can be seen as drawn from Gaussian Mixture Model distribution.

4. *Algorithm:*

Problem statement: Given the dataset, X = {x1,x2…xn}, estimate the parameters θ (theta) of the GMM model that fits the data

Solution: Maximize the likelihood of the data $p(X\mid\theta)$ with regards to the model parameters

$$\theta^{*} = \arg\max_{\theta} p(X\mid\theta) = \arg\max_{\theta} \prod_{i=1}^{N} p(x_i\mid\theta)$$

One of the most popular approaches to maximize the likelihood is to use Expectation Maximization (EM) algorithm.

4.1 Expectation Maximization (EM) Algorithm:

The EM algorithm can be seen an unsupervised clustering method based on mixture models. It follows an iterative approach, sub-optimal, which tries to find the parameters of the probability distribution that has the maximum likelihood of its attributes in the presence of missing/latent data.

The concept of the EM algorithm stems from the Gaussian mixture model (GMM). The GMM method is one way to improve the density of a given set of sample data modelled as a function of the probability density of a single-density estimation method with multiple Gaussian probability density function to model the distribution of the data. In general, to obtain the estimated parameters of each Gaussian blend component if given a sample data set of the log-likelihood of the data, the maximum is determined by the EM algorithm to estimate the optimal model.

Input: Cluster number *k*, a database, stopping tolerance.

Output: A set of *k*-clusters with weight that maximize log-likelihood function.

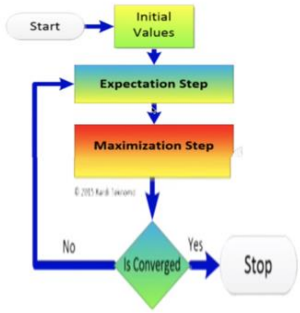
- Expectation step: For each database record *x*, compute the membership probability of *x* in each cluster *h* = 1,…, *k*.
- Maximization step: Update mixture model parameter (probability weight).

Stopping criteria: If stopping criteria are satisfied stop, else set *j* = *j* +1 and go to (1).

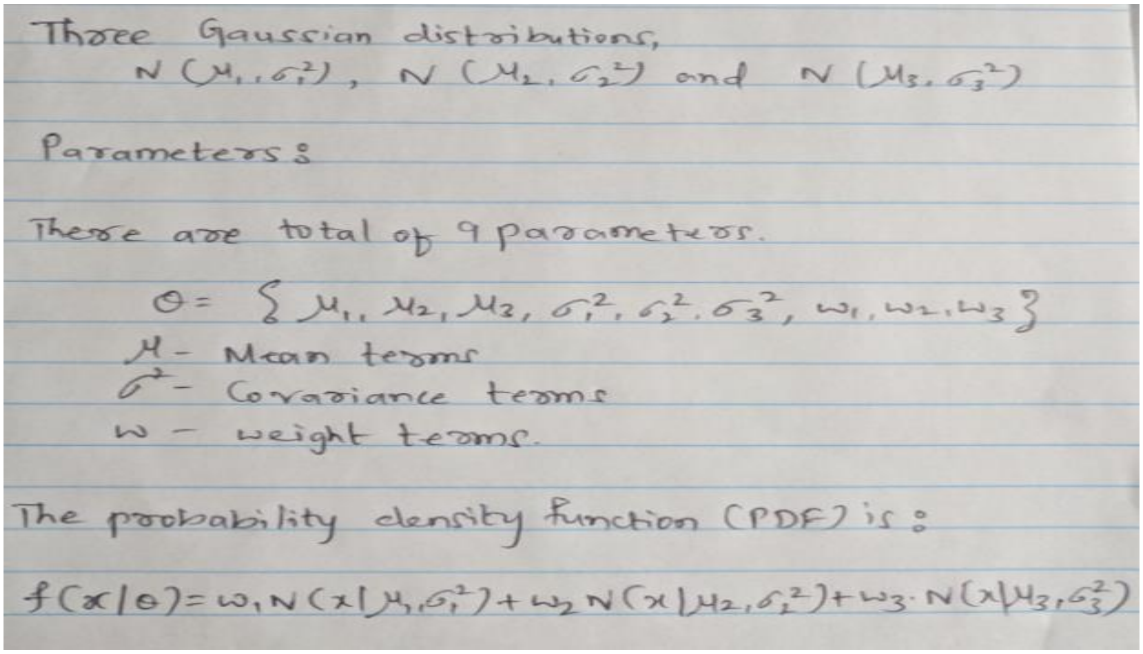
The iterative EM algorithm uses a random variable and, eventually, is a general method to find the optimal parameters of the hidden distribution function from the given data, when the data are incomplete or has missing values.

The algorithm's input are the data set X, the total number of clusters/models K, the accepted error to converge ϵ and the maximum number of iterations. For each iteration, first it is executed what's called the Expectation Step (E-step), that estimates the probability of each point belonging to each model, followed by the Maximization step (M-step), that re-estimates the parameter vector of the probability distribution of each model. The algorithm finishes when the distribution parameters converges or reach the maximum number of iterations.

Expectation–Maximization (EM) Workflow



4.2 Defining the model for this setup,



Three Gaussian distributions,

$N(\mu_1, \sigma_1^2)$, $N(\mu_2, \sigma_2^2)$ and $N(\mu_3, \sigma_3^2)$

Parameters:

There are total of 9 parameters.

$$\Theta = \{\mu_1, \mu_2, \mu_3, \sigma_1^2, \sigma_2^2, \sigma_3^2, w_1, w_2, w_3\}$$

μ - Mean terms

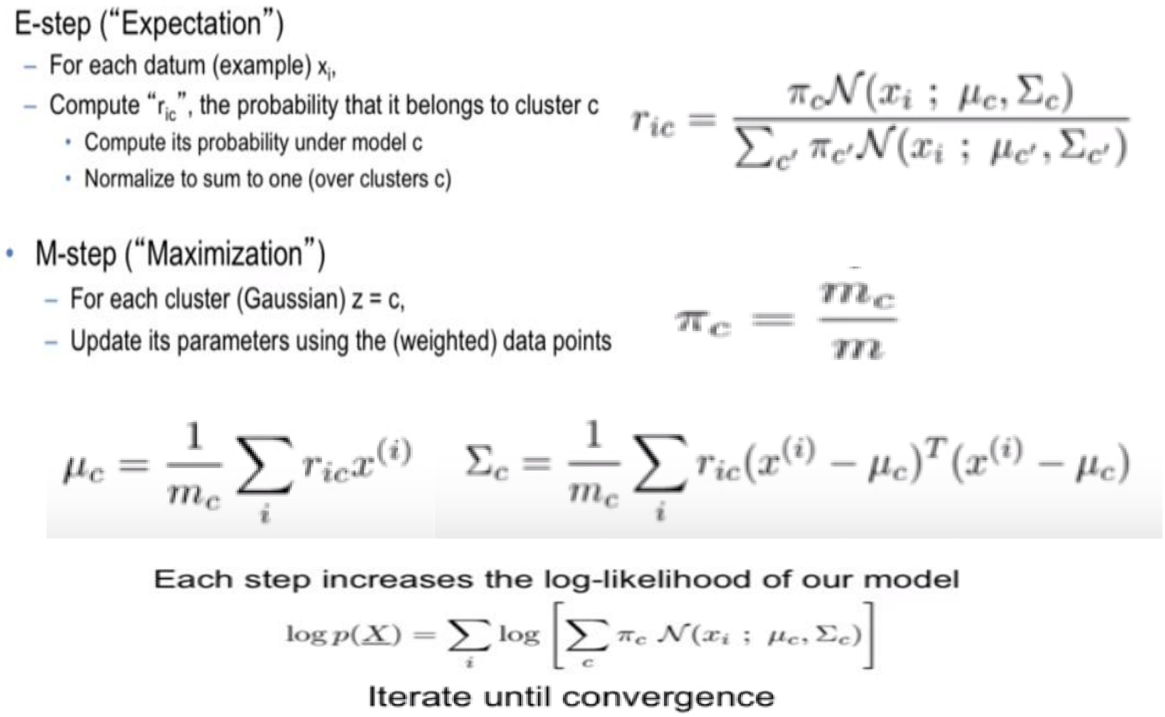
σ^2 - Covariance terms

w - weight terms.

The probability density function (PDF) is:

$$f(x|\theta) = w_1 N(x|\mu_1, \sigma_1^2) + w_2 N(x|\mu_2, \sigma_2^2) + w_3 \cdot N(x|\mu_3, \sigma_3^2)$$

4.3 EM..



E-step ("Expectation")

- For each datum (example) x_i,
- Compute "r_{ic}", the probability that it belongs to cluster c
  - Compute its probability under model c
  - Normalize to sum to one (over clusters c)

$$r_{ic} = \frac{\pi_c \mathcal{N}(x_i \; ; \; \mu_c, \Sigma_c)}{\sum_{c'} \pi_{c'} \mathcal{N}(x_i \; ; \; \mu_{c'}, \Sigma_{c'})}$$

- M-step ("Maximization")
  - For each cluster (Gaussian) z = c,
  - Update its parameters using the (weighted) data points

$$\pi_c = \frac{m_c}{m}$$

$$\mu_c = \frac{1}{m_c} \sum_i r_{ic} x^{(i)} \qquad \Sigma_c = \frac{1}{m_c} \sum_i r_{ic} (x^{(i)} - \mu_c)^T (x^{(i)} - \mu_c)$$

Each step increases the log-likelihood of our model

$$\log p(\underline{X}) = \sum_i \log \left[\sum_c \pi_c \; \mathcal{N}(x_i \; ; \; \mu_c, \Sigma_c) \right]$$

Iterate until convergence

5. Experimental Setup:

- Code Editor Used – Atom open source software.
- Unix-like environment – Cygwin system software.
- Language Used – Python

  Libraries used:

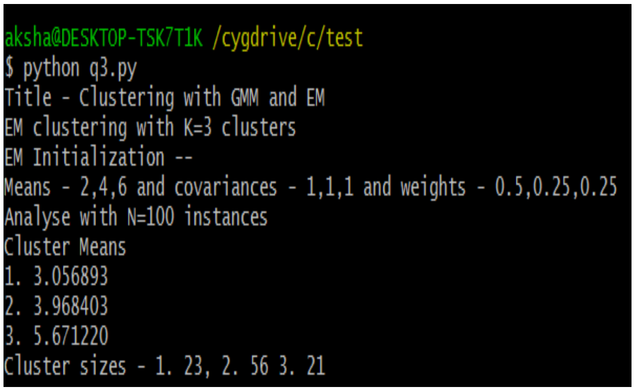
  1. Matplotlib – for plotting
  2. Numpy – for matrix math
  3. Scipy – for normalization
  4. Scipy – for probability density function computation
- Input dataset is read, EM clustering algorithm is applied on the dataset, parameters are estimated, 2-D Euclidean space diagram is plotted and results are collected.

6. Results and Analysis:

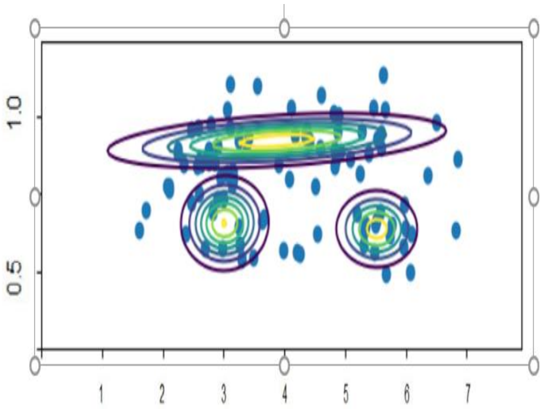
6.1 Results:

Results of applying EM Clustering on input dataset based on two features: perimeter of the seed and length of the kernel are as below, With 100 instances,

- X-axis -- Perimeter of the seed
- Y- axis – length of kernel

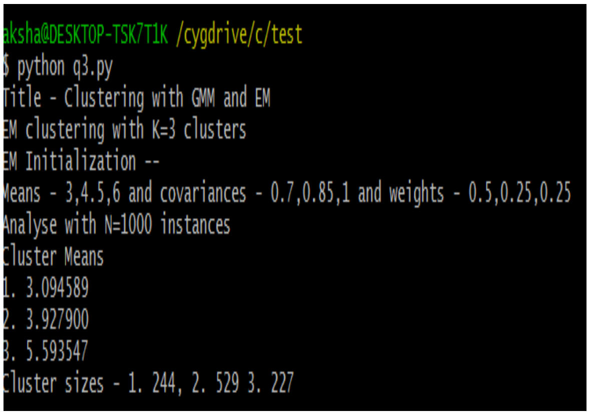


```
aksha@DESKTOP-TSK7T1K /cygdrive/c/test
$ python q3.py
Title - Clustering with GMM and EM
EM clustering with K=3 clusters
EM Initialization --
Means - 2,4,6 and covariances - 1,1,1 and weights - 0.5,0.25,0.25
Analyse with N=100 instances
Cluster Means
1. 3.056893
2. 3.968403
3. 5.671220
Cluster sizes - 1. 23, 2. 56 3. 21
```

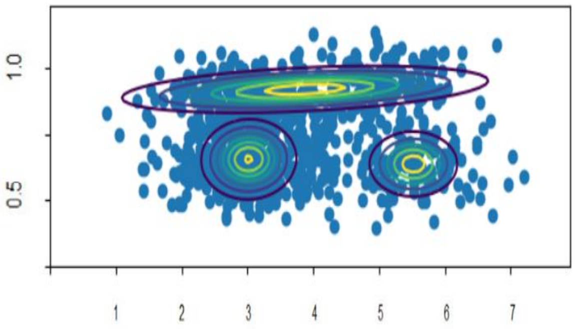


With 1000 instances,

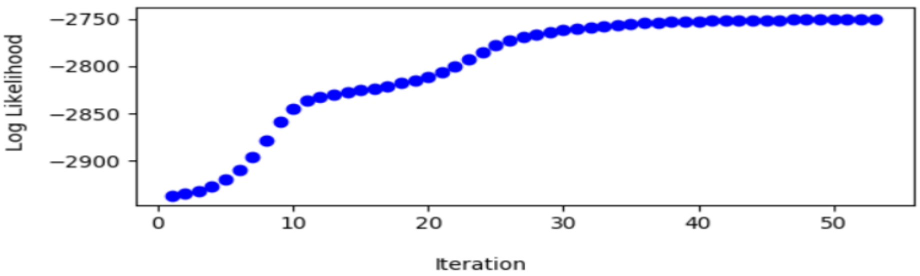
- X-axis -- Perimeter of the seed
- Y- axis – length of kernel
- Number of iterations – 50(30 for convergence)



```
aksha@DESKTOP-TSK7T1K /cygdrive/c/test
$ python q3.py
Title - Clustering with GMM and EM
EM clustering with K=3 clusters
EM Initialization --
Means - 3,4.5,6 and covariances - 0.7,0.85,1 and weights - 0.5,0.25,0.25
Analyse with N=1000 instances
Cluster Means
1. 3.094589
2. 3.927900
3. 5.593547
Cluster sizes - 1. 244, 2. 529 3. 227
```



6.2 Total Log Likelihood vs Number of EM Iterations:

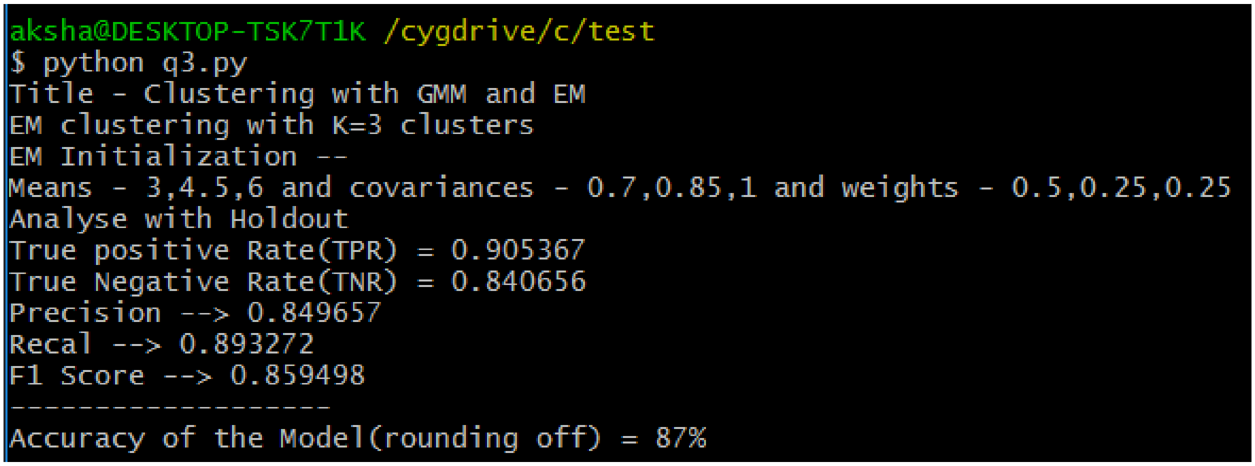


Analysis: It is evident from the graph above, how Total Log Likelihood increases abruptly in the beginning and then almost becomes constant. In the constancy region (TLL > -2750) of Total Log Likelihood, we find best set of Parameters for Gaussian Mixture Model. As we can see computing after -2750 is of no use.

6.3 Clustering Evaluation:

6.3.1 Holdout:

Out of 1000 instances we had, the model is trained on 900 instances. The remaining 100 samples are then used to test how the model is doing on Unseen data.

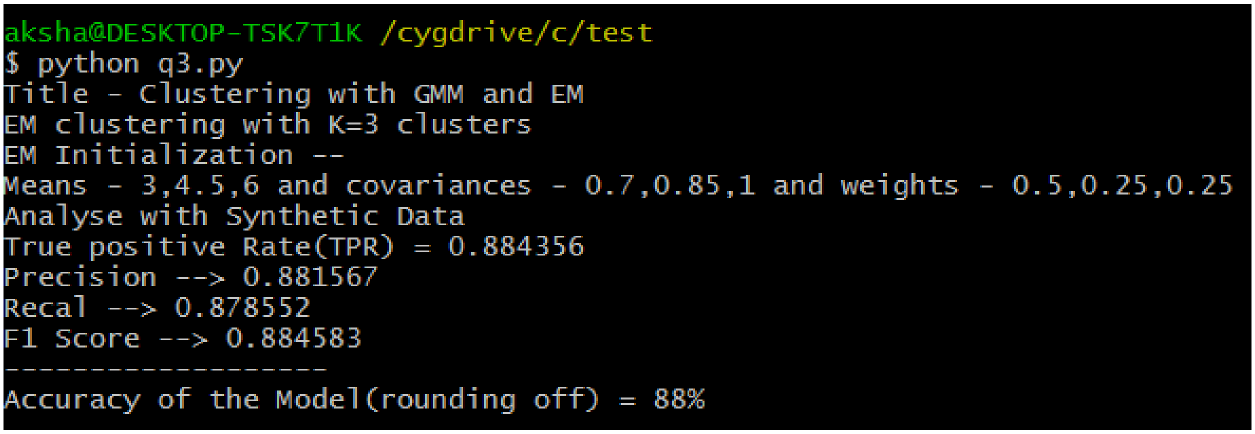


```
aksha@DESKTOP-TSK7T1K /cygdrive/c/test
$ python q3.py
Title - Clustering with GMM and EM
EM clustering with K=3 clusters
EM Initialization --
Means - 3,4.5,6 and covariances - 0.7,0.85,1 and weights - 0.5,0.25,0.25
Analyse with Holdout
True positive Rate(TPR) = 0.905367
True Negative Rate(TNR) = 0.840656
Precision --> 0.849657
Recal --> 0.893272
F1 Score --> 0.859498
-------------------
Accuracy of the Model(rounding off) = 87%
```

It suffers with high variance issues.

6.3.2 With Synthetic Data:

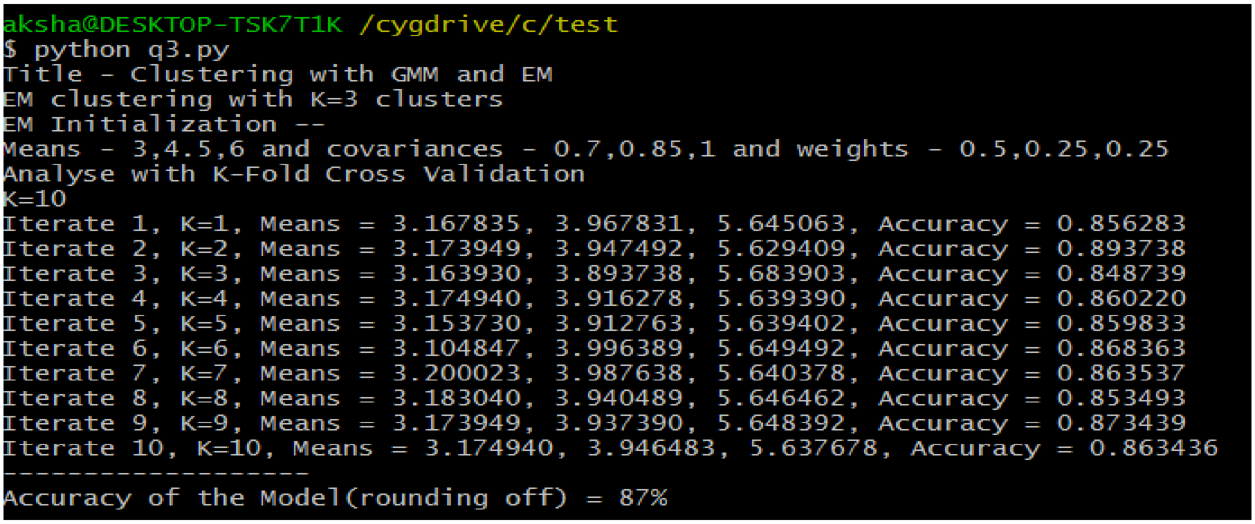
This is a method of evaluation where 100 instances of labelled data is taken and performance of the model is tested on it, the results are as follows,



```
aksha@DESKTOP-TSK7T1K /cygdrive/c/test
$ python q3.py
Title - Clustering with GMM and EM
EM clustering with K=3 clusters
EM Initialization --
Means - 3,4.5,6 and covariances - 0.7,0.85,1 and weights - 0.5,0.25,0.25
Analyse with Synthetic Data
True positive Rate(TPR) = 0.884356
Precision --> 0.881567
Recal --> 0.878552
F1 Score --> 0.884583
-------------------
Accuracy of the Model(rounding off) = 88%
```

6.3.3 K-Fold Cross Validation:

1000 instances are divided into 10 sets of 100 samples each and Holdout operation is performed k(10) times, such that each time, one of the 10 subsets is used for testing and remaining to train the model.



```
aksha@DESKTOP-TSK7T1K /cygdrive/c/test
$ python q3.py
Title - Clustering with GMM and EM
EM clustering with K=3 clusters
EM Initialization --
Means - 3,4.5,6 and covariances - 0.7,0.85,1 and weights - 0.5,0.25,0.25
Analyse with K-Fold Cross Validation
K=10
Iterate 1, K=1, Means = 3.167835, 3.967831, 5.645063, Accuracy = 0.856283
Iterate 2, K=2, Means = 3.173949, 3.947492, 5.629409, Accuracy = 0.893738
Iterate 3, K=3, Means = 3.163930, 3.893738, 5.683903, Accuracy = 0.848739
Iterate 4, K=4, Means = 3.174940, 3.916278, 5.639390, Accuracy = 0.860220
Iterate 5, K=5, Means = 3.153730, 3.912763, 5.639402, Accuracy = 0.859833
Iterate 6, K=6, Means = 3.104847, 3.996389, 5.649492, Accuracy = 0.868363
Iterate 7, K=7, Means = 3.200023, 3.987638, 5.640378, Accuracy = 0.863537
Iterate 8, K=8, Means = 3.183040, 3.940489, 5.646462, Accuracy = 0.853493
Iterate 9, K=9, Means = 3.173949, 3.937390, 5.648392, Accuracy = 0.873439
Iterate 10, K=10, Means = 3.174940, 3.946483, 5.637678, Accuracy = 0.863436
-------------------
Accuracy of the Model(rounding off) = 87%
```

Error estimation is averaged over all k trials to get total effectiveness of the model.

Few more observations:

- Getting the suitable parameters is key point.
- Initialization of the parameters affects the number of iterations.
- If you initialize the parameters with final set of parameters from previous experiment, the number of Iterations will reduce to 2 or 3 for convergence.

7. Conclusion:

A Gaussian mixture model (GMM) attempts to find a mixture of multi-dimensional Gaussian probability distributions that best model any input dataset. As it is a probabilistic model, it is also possible to find probabilistic cluster assignments, which measures the probability that any point belongs to the given cluster.

Classification of Wheat Strain can be achieved using Expectation Maximization (EM) Clustering algorithm. A model can be developed to distinguish the Wheat strains from each other using EM Clustering algorithm. The algorithm is fast, robust and easier to understand. Relatively efficient. However, the learning of algorithm requires appropriate specification of the number of cluster centers. More the number of iterations more accurate are the results. To conclude, EM Clustering technique can be efficiently used for classification of Wheat strains.

I have successfully estimated the parameters of the GMM model that fits the given data and EM is been successfully used to maximize the likelihood of the data with regards to model parameters.

8. References:

[1] http://www.cs.cmu.edu/~guestrin/Class/10701-S05/slides/EM-MixGauss4-4-2005.pdf

[2] Wikipedia pages
https://en.wikipedia.org/wiki/Mixture_model
https://en.wikipedia.org/wiki/Mixture_model#Gaussian_mixture_model
https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm

[3] Microsoft Azure Page
https://msdn.microsoft.com/en-us/library/azure/dn905944.aspx

[4] https://sites.google.com/site/dataclusteringalgorithms/em-clustering-algorithm

[5] http://scikit-learn.org/stable/datasets/index.html#toy-datasets

[6] https://www.youtube.com/watch?v=JNlEIEwe-Cg

[7] Stobak Dutta, Sabnam Sengupta: Implementation of EM Clustering in ECB Framework Of Cloud Computing Environment, Cloud Computing, Data Science & Engineering - Confluence, 2017 7th International Conference
http://ieeexplore.ieee.org.libproxy2.usc.edu/stamp/stamp.jsp?arnumber=7943165

[8] https://jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html

[9] R. Kumari, Sheetanshu, M. K. Singh, R. Jha, N.K. Singh: Anomaly Detection in Network Traffic using Kmean clustering, Recent Advances in Information Technology (RAIT), 2016 3rd International Conference on
http://ieeexplore.ieee.org.libproxy2.usc.edu/stamp/stamp.jsp?arnumber=7507933

[10] Sonali Shankar, Bishal Dey, Sai Sabitha, Deepti Mehotra: Performance Analysis of Student Learning Metric using EM Clustering Approach
http://ieeexplore.ieee.org.libproxy2.usc.edu/stamp/stamp.jsp?arnumber=7508140

[11] https://brilliant.org/wiki/gaussian-mixture-model/

[12] http://statweb.stanford.edu/~tibs/stat315a/LECTURES/em.pdf

[13] https://www.mathworks.com/help/stats/clustering-using-gaussian-mixture-models.html

[14] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4433949/

[15] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.446.7955&rep=rep1&type=pdf