Programming Assignment 2: KMeans and GMM

Kavya Sethuram(ksetura@usc.edu); Rasika Guru (rguru@usc.edu); Roshani Mangalore (rmangalo@usc.edu)

1. Implementation of KMeans and GMM Algorithm in Python v2.7

A. DataStructures used in the Implementation are:

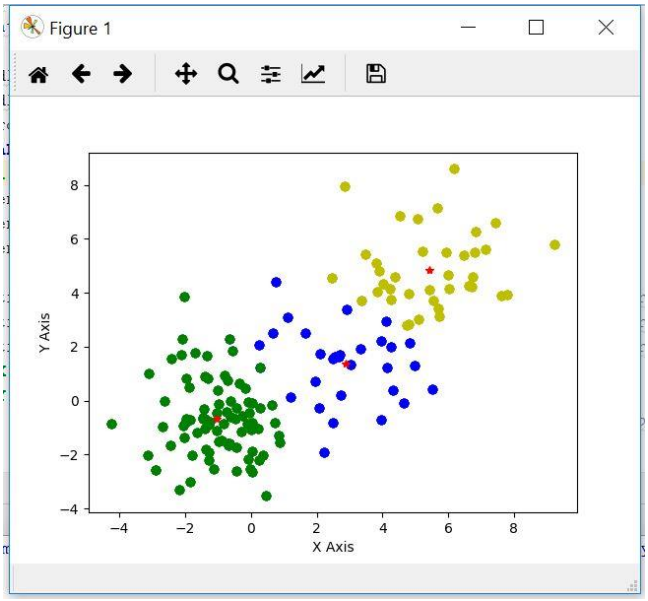
- ***Lists:*** Python has a great built-in list type named "list". List literals are written within square brackets [ ]
- ***Dataframe*:** DataFrame is a 2-dimensional labeled data structure and is the most commonly used panda object. We have generated Dataframe from the datafile using pandas.

B. Explanation:

K-means:

- Initialize k(=3) cluster centroids
- Repeat the following:
  - for each point, compute which centroid is nearest to it
  - for each centroid, move its location to the mean location of the points assigned to it
- Output of Kmeans is as follows: It displays the final centroids for each of the three Clusters and also plots a graph highlighting all the datapoints and clusters

```
Final 3 Centroids of K-means Algorithm:
[ 2.88349710907 , 1.358261948 ]
[ -1.0394086164 , -0.679196800265 ]
[ 5.43312387416 , 4.86267502661 ]
```



EM:

1. Start with data set X, initial set of K Gaussian component pdfs Nk , N (μk , Σk) and K mixture weights P(k), k = 1, . . ., K

2. **E-step:** for given parameter values we can compute the expected values of the latent variables

$$\begin{aligned} r_{ik} \equiv E(z_{ik}) &= p(z_{ik}=1|x_i,\pi,\mu,\Sigma) \\ &= \frac{p(z_{ik}=1)p(x_i|z_{ik}=1,\pi,\mu,\Sigma)}{\sum_{k=1}^{K} p(z_{ik}=1)p(x_i|z_{ik}=1,\pi,\mu,\Sigma)} \\ &= \frac{\pi_k \mathcal{N}(x_i|u_k,\Sigma_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|u_k,\Sigma_k)} \end{aligned}$$

   Using the following Probability Density Function,

$$\mathcal{N}(x|\mu,\Sigma) = \frac{1}{(2\pi|\Sigma|)^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\}$$

   mean, covariance

   Note: for the first Iteration, randomly Initialize the probabilities of Data points belonging to three clusters

3. **M-step:** maximize the expected complete log likelihood. Re-estimate the following parametres

   - For each clusters, compute Mean(μ), Co-variance matrix(Σ) and Amplitude(π) using the following formula

$$\pi_k = \frac{\sum_i r_{ik}}{n} \qquad \mu_k = \frac{\sum_i r_{ik} x_i}{\sum_i r_{ik}}$$

$$\Sigma_k = \frac{\sum_i r_{ik}(x_i-\mu_k)(x_i-\mu_k)^T}{\sum_i r_{ik}}$$

4. Iterate E-step and M-step until the log likelihood of data does not increase any more.

$$\ln p(x|\pi,\mu,\Sigma) = \sum_{i=1}^{n} \ln\{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k,\Sigma_k)\}$$

5. Convergence Criteria: When the difference between the log likelihood of the consecutive iterations is 0, we break out of the loop and display the final mean, co-variance matrix and amplitude. (In our case we are breaking out of the loop when the difference between the current and previous likelihood is 0.005)
6. Output of GMM is as follows:

```
Mean for cluster 1: [ 1.31823273  1.08924684]
Mean for cluster 2: [ 1.41561702  1.1344646 ]
Mean for cluster 3: [ 1.35028227  1.12481558]
Amplitude for cluster 1: 0.240312461136
Amplitude for cluster 2: 0.247153125546
Amplitude for cluster 3: 0.512534413318
Co variance for cluster 1: [[ 9.14993618  6.18458855]
 [ 6.18458855  7.38328478]]
Co variance for cluster 2: [[ 9.36455857  6.33654981]
 [ 6.33654981  7.38139099]]
Co variance for cluster 3: [[ 9.42725312  6.13603029]
 [ 6.13603029  7.1494122 ]]
```

C. Code Level Optimizations:

- Using NumPy comprehensions over explicit for loops in certain places, making code more compact and execution faster.
- Matrix operations are easy with NumPy arrays
- Modularizing parts of code into methods, making code more readable.

D. Challenges:

- Writing a function which involved Complex Mathematical formula was a little time consuming
- Handling the convergence condition was a little tricky

2. Software Familiarization:

We have used the following packages to implement K-means and EM-GMM

NumPy

- NumPy is a Numeric Python module. It provides fast mathematical functions.
- Numpy provides robust data structures for efficient computation of multi-dimensional arrays & matrices.
- We used numpy to read data files into numpy arrays and data manipulation.

Pandas

- Provides DataFrame Object for data manipulation
- Provides reading & writing data b/w different files.
- DataFrames can hold different types data of multidimensional arrays.

pylab*:* pylab is the library used to plot graphs. Each Cluster is represented in different colours

Existing Libraries:

1. Scikit-Learn (for K-means)
   - It's a machine learning library. It includes various machine learning algorithms.
   - It uses the following inbuilt libraries to implement KMeans Algorithm

```
from sklearn.cluster import KMeans
```

2. sklearn.mixture (for GMM)

   sklearn.mixture is a package which enables one to learn Gaussian Mixture Models (diagonal, spherical, tied and full covariance matrices supported), sample them, and estimate them from data.

   Facilities to help determine the appropriate number of components are also provided.

3. Comparison between K-means and GMM

K-means	GMM
Objective function – Minimize sum of squared Euclidean distance	Objective function – Maximize log-likelihood
K-means is based on distances	GMM is based on probabilities
Can be optimized by an EM algorithm – E-step: assign points to clusters – M-step: optimize clusters – Performs hard assignment during E-step	EM algorithm – E-step: Compute posterior probability of membership – M-step: Optimize parameters – Perform soft assignment during E-step
In K-means E-step, data points are assigned to nearest cluster ⇒ cluster membership is either 0% or 100%	In contrast, GMM responsibility P(k|xn) plays role of soft cluster membership ⇒ since P k P(k|xn) = 1, data point xn belongs P(k|xn) × 100% to cluster k, and each point contributes to some extent to each cluster
Assumes spherical clusters with equal probability of a cluster	Can be used for non-spherical clusters. Can generate clusters with different probabilities
Cannot handle when the data points are scattered	Handles scattered data points

4. Applications of Clustering Algorithms

1. **Clustering Algorithm in Search Engines**

Clustering algorithm is the backbone behind the search engines. Search engines try to group similar objects in one cluster and the dissimilar objects far from each other. It provides result for the searched data according to the nearest similar object which are clustered around the data to be searched. Better the clustering algorithm used, better are the chances of getting the required result on the front page. Hence, the definition of **similar object** play a crucial role in getting the search results, better the definition of similar object better the result is.

2. **Application of k-Means Clustering algorithm for prediction of Students' Academic Performance**

The ability to monitor the progress of students' academic performance is a critical issue to the academic community of higher learning. A system for analyzing students' results based on cluster analysis and uses standard statistical algorithms to arrange their scores data according to the level of their performance is described.

[References: *Application of k Means Clustering algorithm for prediction of Students Academic Performance (PDF Download Available)*. Available from: https://www.researchgate.net/publication/45900764_Application_of_k_Means_Clustering_algorithm_for_prediction_of_Students_Academic_Performance [accessed Sep 29, 2017].]

3. Scene Image Clustering Based on Boosting and GMM

Gaussian Mixture Model (GMM) is widely used in unsupervised learning tasks. In this paper, we propose the boost-GMM algorithm which uses GMMs to cluster real world scenes. At first, images will be extracted with gist-feature to get the data set. At each boosting iteration, a new training set is constructed by using weighted sampling from the original dataset and GMM is used to provide a new data partitioning. The final clustering solution is produced by aggregating the multiple clustering results. Experiments on real-world scene sets indicate that boost-GMM has higher result than other algorithms.

[References: https://sites.google.com/site/thanhtoando2212/p226-boost_gmm.pdf]

5. Individual Contribution

K-means	Kavya Sethuram
GMM	Rasika Guru and Roshani Mangalore