**Programming assignment 2: KMeans and EM**

**Group Report**

Chenqi Liu 2082-6026-02

Che-Pai Kung 5999-9612-95

Mengyu Zhang 3364-2309-80

**Part I: Implementation**

**Part II: Software Familiarization**

The libraries using K-means and Expectations Maximization algorithm we found are “sklearn.cluster.KMeans” and “sklearn.mixture.GaussianMixture”.

1. K-means

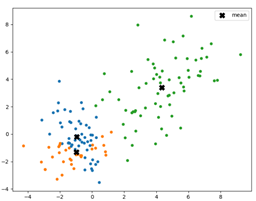
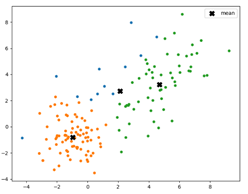
The part1 k-means algorithm generates similar outcomes from sklearn library when k equal to 3. [check if different k has different results]

There are many parameters used to create clusters through KMeans library. Among them, we found that we could work on ‘init’ and ‘n\_init’ to optimize our codes. The method for initialization is chosen through the parameter ‘init’. There are two methods, ‘k-means++’, ‘random’ and the default is set to be ‘k-means++’ which selects initial centroids in a smarter way to speed up convergence. ‘random’ means choose initial centroids randomly from data, which is also the way our kmeans algorithm uses. We might be able to decrease the number of iterations if we can utilize ‘k-means++’.

As the K-means algorithm guarantees only the local minimum solution, it is better that we run the code several times to get a chance to find a global solution. Therefore, the parameter ‘n\_init’ is introduced. By setting the parameter, the KMeans library algorithm will run with different initial centroids and return the best output. We could improve our algorithm’s accuracy by adding the parameter.

1. Expectation Maximization algorithm (GMM)

Besides parameter ‘init’ and ‘n\_init’, GaussianMixture library also includes parameter ‘init\_params’, offering the methods to initialize the program. The possible methods are ‘kmeans’ and ‘random’. While ‘random’ chooses the random points (the same as what we did in our EM algorithm), ‘kmeans’ estimate the random points using k-means algorithm. Under this data set, the results of using ‘random’ are shown as below. They are not consistent, and both seems to not have explainable clusters. (The graphs are the outcomes of sklearn library, but our EM algorithm also produces the unstable results)



Nonetheless, if we set the parameter to be ‘kmeans’, then the result is more similar to what we got from kmeans algorithm and more stable. In other words, we can get similar outcomes after each run of the program. Therefore, utilizing results from kmeans algorithm may help EM in achieving better cluster performance, and improve our code.

**Part III: Applications**

1. Kmeans:

Kmeans algorithm has been applied in many areas and one of them is image segmentation. Image segmentation classifies the image into different clusters according to its colors. The application helps recognizing the similar attributes. Take autonomous vehicles for example, the machine can detect the surroundings, recognize the objects as cars, passengers or other things and do the corresponding actions. Additionally, kmeans is applied to the healthcare industry. The algorithm makes it possible to identify cancer cells in the early stage and therefore save people’s lives.

1. Expectation Maximization algorithm:

Besides clustering, EM algorithm is used to estimate parameters of Hidden Markov Models (HMM). As there are two parameters in the maximum likelihood estimation, it is difficult to find a simple way using any iterative methods to derive the parameters. However, we can use the EM algorithm just as what we do for the clustering problems. As a result, we can analyze part-of-speech tagging (POS tagging) which is beneficial to generate sentences with more accurate grammar and words.

**Part IV: Individual Contributions**

●      Model discussion: Chenqi Liu, Che-Pai Kung, Mengyu Zhang

●      Model implementation:

●      Model optimization:

●      Software Familiarization: Che-Pai Kung

●      Applications: Che-Pai Kung