Programming Assignment 2: K-means and GMM

Part 1: Group member and contribution

Bin Zhang (5660329599): GMM code, K-means code, Part2 k-means report

Yihang Chen (6338254416): K-means code, K-means package code, GMM package code, and software familiarization part of report

Hanzhi Zhang (4395561906): GMM code, Part2 GMM of report, and applications part of report

Part 2: Implementation of K-means and GMM

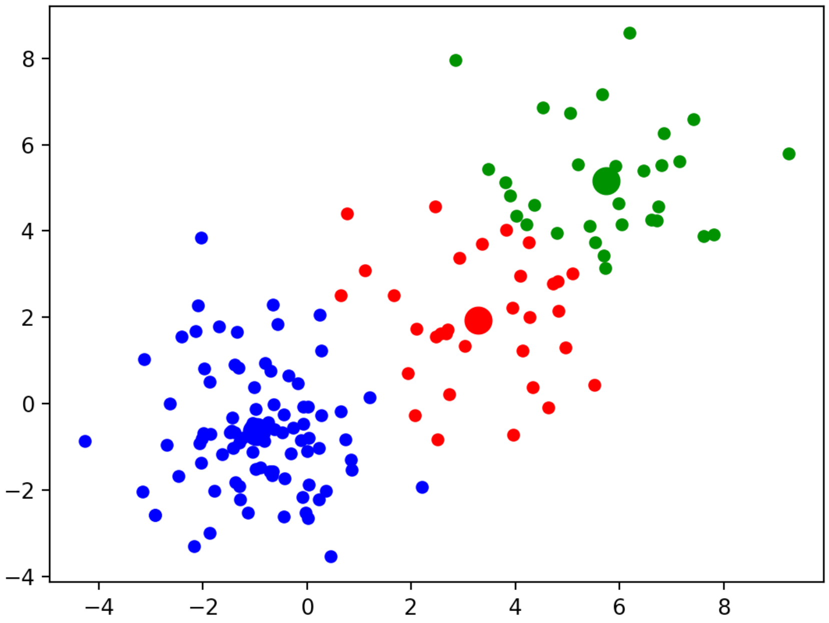
K-means:

K-means is a sample algorithm which we only need to define the input data's structure, here we first use the nested list to sore the data, but for the sake of convenience for GMM, we use matrix to store our input data. Due to the 150 sample size and the feature num is 2, the stored data is a matrix with (150,2) shape. There are three steps in k-means, first we initialize the 3 centroids and store it into a matrix with (3,2) shape, then we assigned data points to its nearest centroid ,here for the clustered groups, we use nested list to store data like [[cluster1 's data points], [cluster2 's data points], [cluster3 's data points]] for the following process. After that we recompute the centroids, the new centroids is matrix with (3,2) shape. We repeated the second and third part until convergence. Here we make rules, setting a loop constraint: when exceeding the setting times we end the loop.

After clustering, the 3 centroid are:

```
[[ 3.28884856  1.93268837]
 [-0.96065291 -0.65221841]
 [ 5.73849535  5.16483808]]
```

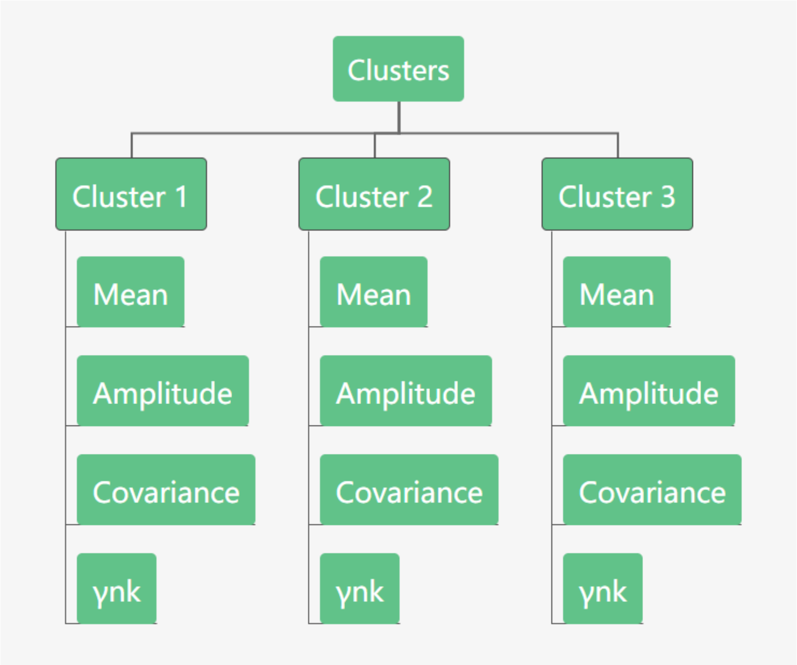
The graph of our data points distribution is also showed below:



Gaussian Mixture Model (GMM)

In GMM, most of data is stored in a NumPy array structure.

The data structure used to store the final result is a list of dictionaries. The structure diagram is showed below. Clusters is a list which contains three clusters. Each cluster is a dictionary. For every dictionary, the keys are mean (mu_k presented in Python programming) with (1,2) shape, Amplitude (pi_k presented in Python programming) whose structure is a float number, Covariances (cov_k presented in Python programming) with (2,2) shape, and γnk (gamma_nk presented in Python programming) with (150,1) shape. γnk means the probability of this datapoint in this cluster.

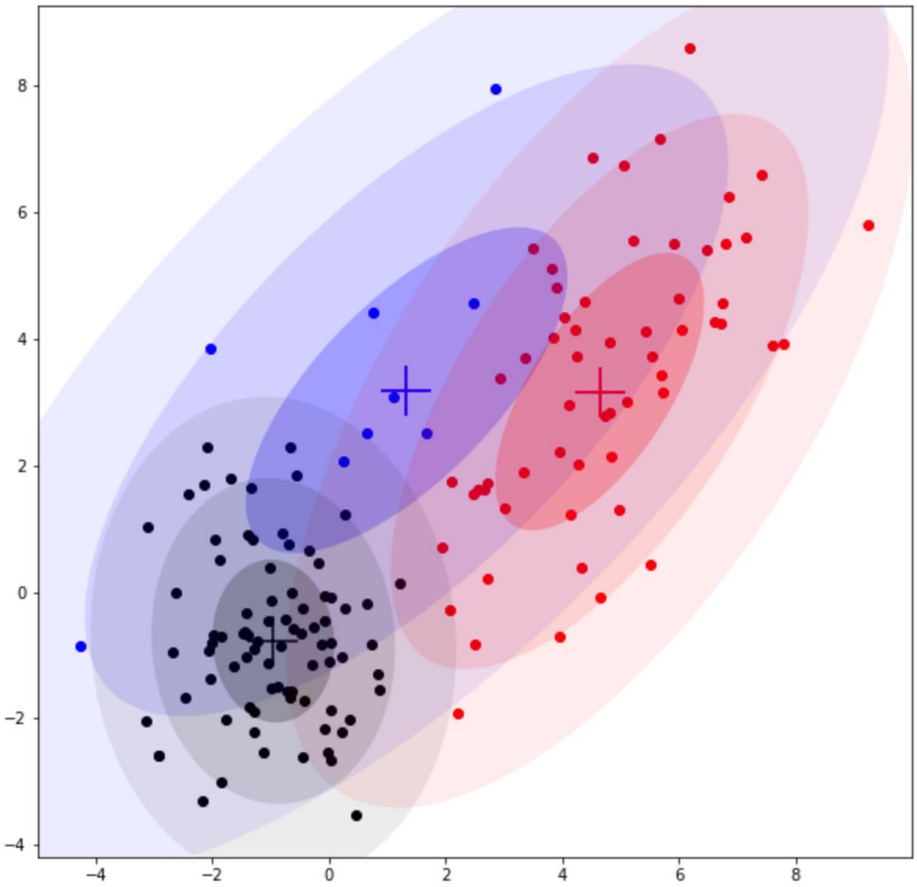


In order to optimize our code, our code is modular. Basically, our code is divided into four modular, initialization of data, expectation, maximization, and training.

The challenge of this algorithm is how to determine whether the convergence is close enough. Most of codes on the Internet use the maximum likelihood θ of the model as a standard. Once the change of maximum likelihood is close enough ($|\theta_{i+1} - \theta_i| < \varepsilon$), the code would stop. However, in our programming, we choose to compare the previous γ_{nk} with the updated value. If there is no change, the code would stop. Moreover, we also set a limitation of iterations times, 5000. Compared with online code, our algorithm is supposed to require more iterations. Fortunately, our code reached the requirement after 1278 iterations. However, if the datasets become lager, our programming will become very slow.

The output of our programming is showed below. Moreover, the graph of our final output is also showed below.

	Mean	**Amplitude**	**Covariance**
Cluster 1	$(1.3056314 \quad 3.18634189)$	0.11215244430039525	$\begin{pmatrix} 7.58964771 & 4.97065771 \\ 4.97065771 & 6.63812034 \end{pmatrix}$
Cluster 2	$(4.64236749 \quad 3.16626138)$	0.36786471919348	$\begin{pmatrix} 3.21424076 & 2.53373844 \\ 2.53373844 & 4.81041169 \end{pmatrix}$
Cluster 3	$(-0.95284859 \quad -0.77590855)$	0.5199828365061246	$\begin{pmatrix} 1.09144685 & -0.05787625 \\ -0.05787625 & 1.65569406 \end{pmatrix}$



Part 3: Software Familiarization

To implement these two algorithms, we use functions in the python library Scikit-learn: sklearn.cluster.KMeans and sklearn.mixture.GaussianMixture.

K-means

For the k-means algorithm, sklearn initialize the k points using k-means++ method as default. Compared with randomly initializing k points, k-means++ can initialize points which have better performance in terms of both accuracy and speed (proved in *"Arthur, David, and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Stanford, 2006."*). The k-means++ algorithms are shown below:

1a. Take one center c_1, chosen uniformly at random from χ.

1b. Take a new center c_i, choosing $x \in \chi$ with probability $\frac{D(x)^2}{\sum_{x \in \chi} D(x)^2}$

1c. Repeat Step 1b. until we have taken k centers altogether

2. Proceed as with the standard k-means algorithm

In general, the k-means++ initialize points based on the rules that farther points to the centroid already generated have higher possibility to be chosen as a new centroid.

Besides the default setting for the k-means algorithms is “auto”, which means that the function will choose “elkan” for dense data and classic EM-style algorithm for sparse data. *The “elkan” variation is more efficient by using the triangle inequality, but currently doesn’t support sparse data.* The “elkan” algorithm can reduce computing cost when calculating distances. But for now it cannot be used on sparse data.

In our hand-write k-means algorithm, we simply using EM algorithm and generate initial points randomly. It is usable but we can improve it on the strategy of k points initialization.

GMM

For the GMM algorithm, sklearn initialize the parameters using k-means method as default. And it can set covariance type:

‘full’ (default):each component has its own general covariance matrix

‘tied’:all components share the same general covariance matrix

‘diag:’each component has its own diagonal covariance matrix

‘spherical’:each component has its own single variance

Basically, in **sklearn**, we can set restrictions from the most general possible mixture to a very specific kind of mixture.

Part 4: Applications

K-means

K-means algorithm is typically applied on continues data. If some groups of similar issues are required to divide into several parts, k-means is suitable for these scenarios. For example, Document Classification is one of the applications. In this project, words in the document are converted to vector. Commonly used word terms in each document are considered as data to implement k-means. The result of clustering can help people label different categories on different documents. Moreover, K-means can also be used to Customer Segmentation. In Document Classification, the content of document is used to cluster. In this case, the basic information or the behavior of customer is used to cluster. A telecommunications company uses the consumer information including money spent in recharging, sending SMS, and browsing the internet to determine the type of customer. This project can help company design specific advertisements to special people group.

GMM

Different from K-means, GMM can provide a probability of one datapoint in one cluster. One of the important applications of this algorithm is safety problem. For example, GMM is used as a detection for flight operation and safety detection. The existing monitoring system can only detect some hazardous on a pre-defined list. However, with the help of GMM, the system can find some unknown risks. In this project, the flight data from different planes is shared in one system. With the help of GMM, system can easily find a unusual data.