Pattern Recognition
Assignment#4

By *Professor Ying Shen*

2031566/Yang Han

December 14, 2020

1 Introduction

In this assignment, I will implement KMeans and use it to cluster Seeds Data Set ($http://archive.ics.uci.edu/ml/datasets/seeds$) and Iris Data Set($http://archive.ics.uci.edu/ml/datasets/Iris$).

1.1 Seeds Dataset

The dataset belongs to three different varieties of wheat and is consisted of seven geometric parameters, 1. area A, 2. perimeter P, 3. compactness C 4. length of kernel, 5. width of kernel, 6. asymmetry coefficient 7. length of kernel groove. All of these parameters were real-valued continuous.

1.2 Iris Dataset

Iris Dataset is perhaps the best known database to be found in the pattern recognition literature, The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.

2 Data Preprocess

As I show in the *preprocess.ipynb* file, I mainly do three things:

1. Read the raw file content, cast the str type to float type save save them to csv format.
2. Map categories to ordered numbers(eg. 0,1,2...).
3. In order to eliminate the dimensional influence between indicators, data standardization is required to solve the comparability between data indicators.

3. Program Modules

3.1 Theory

The main element of k-means algorithm works by a two-step process called expectation-maximization. The expectation step assigns each data point to its nearest centroid. Then, the maximization step computes the mean of all the points for each cluster and sets the new centroid. The main algorithm was described as below:

Algorithm 1 K-means algorithm

1: Specify the number k of clusters to assign.
2: Randomly initialize k centroids.
3: **repeat**
4: **expectation:** Assign each point to its closest centroid.
5: **maximization:** Compute the new centroid(mean) of each cluster.
6: **until** The centroid positions do not change or iteration reach the max iteration times(30).

3.2 Code

In the assignment, I follow the scikit-learn style to design my KMeans class which accepts one parameter name $n\_clusters$ with default value as 2. The two main methods name fit and $predict$, fit accepts the

samples and execute the k-means algorithm, *predict* doesn't accept parameter and return the labels of each sample.

4. Result

In Seeds Data Set, I set $n\_clusters = 3$, and get a result as below:

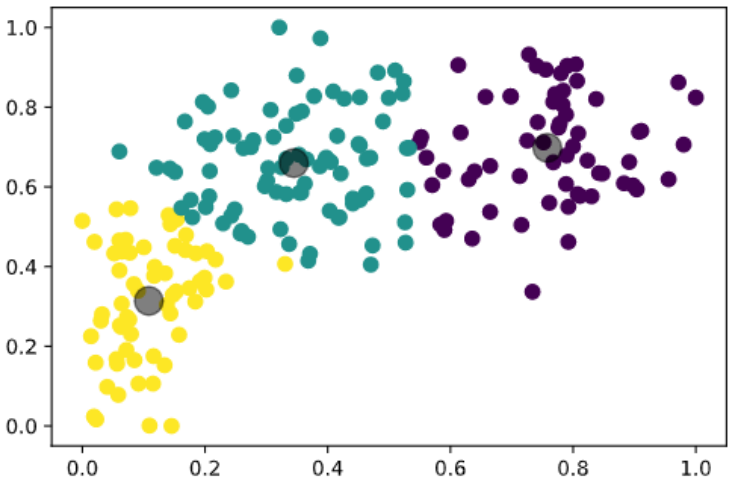

Figure 1: Seeds Data Set cluster

In Iris Data Set, I set $n\_clusters = 3$, and get a result as below:

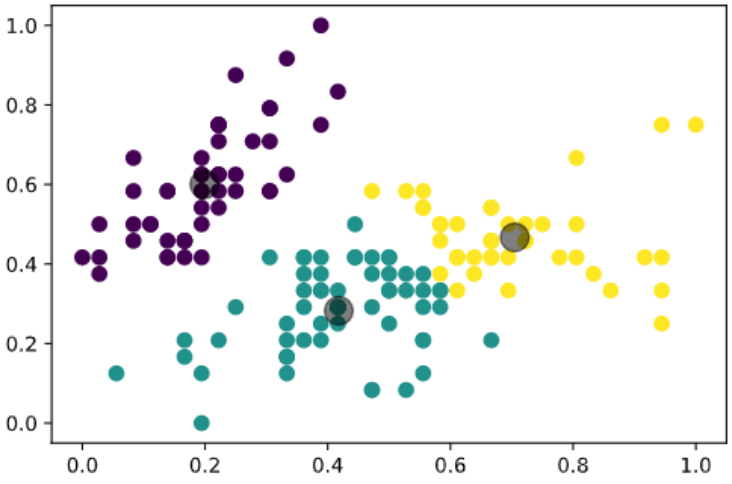

Figure 2: Iris Data Set cluster

Limitations and Inprovements

The KMeans algorithm is nondeterministic, meaning it could produce different results from two separate runs even if the runs were based on the same input. Instead we can use *DBSCAN* to get a stable result.