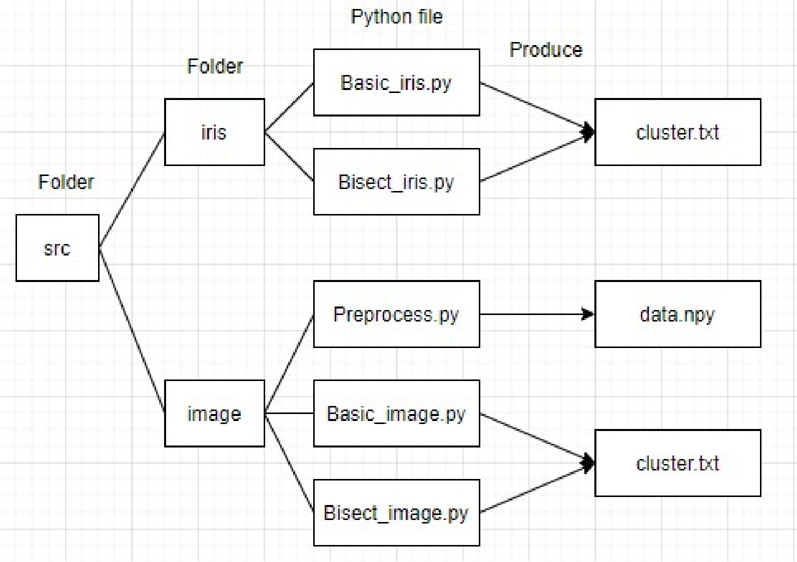
Miner2 username : Bear
Mason user id : chsiung2
Gnumber : G01272835
Best public iris score : 0.86
Best public image score : 0.53

1. Introduction

- I implemented both basic k-means and bisecting k-means.
- Basic k-means : Use most of it to implement Bisecting k-means. I handle empty cluster with putting a random node into it.
- Bisecting k-means : Each time select a cluster with the largest cluster size. Use SSE to evaluate the performance.
- The following picture is the structure of my src folder :
  iris folder : for Iris (Part 1).
  image folder : for Image Clustering (Part 2).
  Basic_iris.py、Basic_image.py : Implement basic k-means. Except for the input data, they are almost the same. Save the result as “cluster.txt”.
  Bisect_iris.py、Bisect_image.py : Implement bisecting k-means. Except for the input data, they are almost the same. Save the result as “cluster.txt”.
  Preprocess.py : Preprocess the data. Save the result as “data.npy”.



2. Approaches

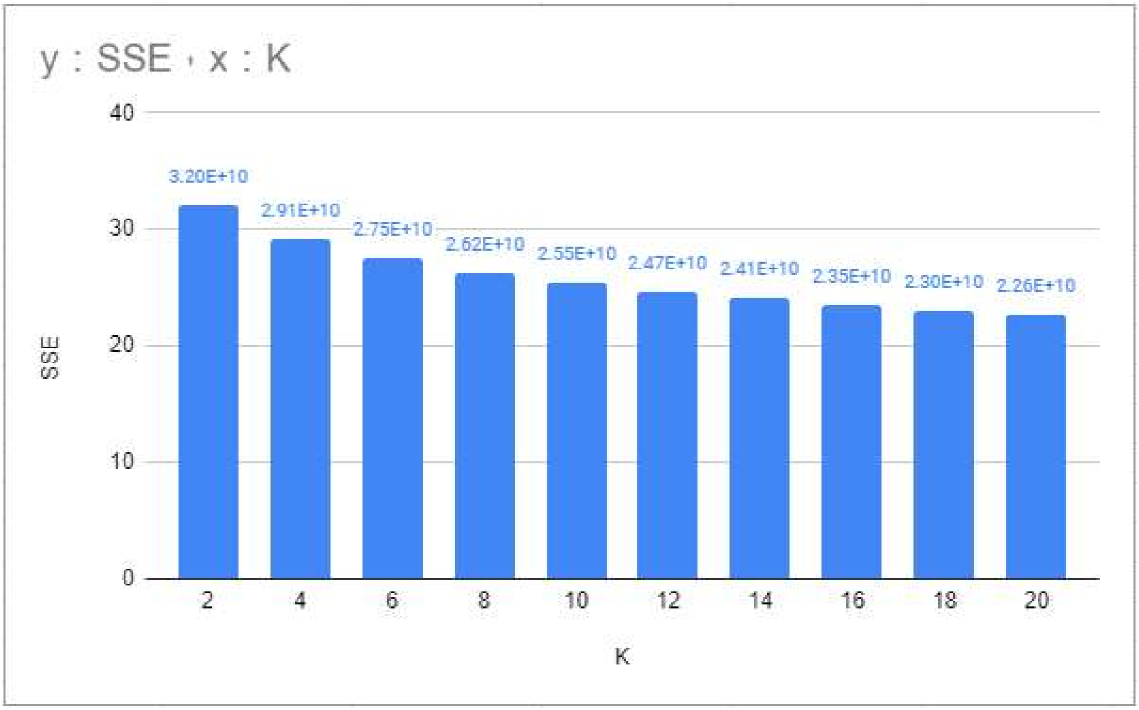
- Handling data :
  - Iris : Use “normalize()” in sklearn to preprocess data.
  - Image Cluster : Use “PCA()” in sklearn to preprocess data.
- For Basic k-means, I handle an empty cluster with adding a random node into it. It’s a bad method

but empty cluster never happen during the time I tried for k from 2 to 20 and other preprocessing approach with these set of data.

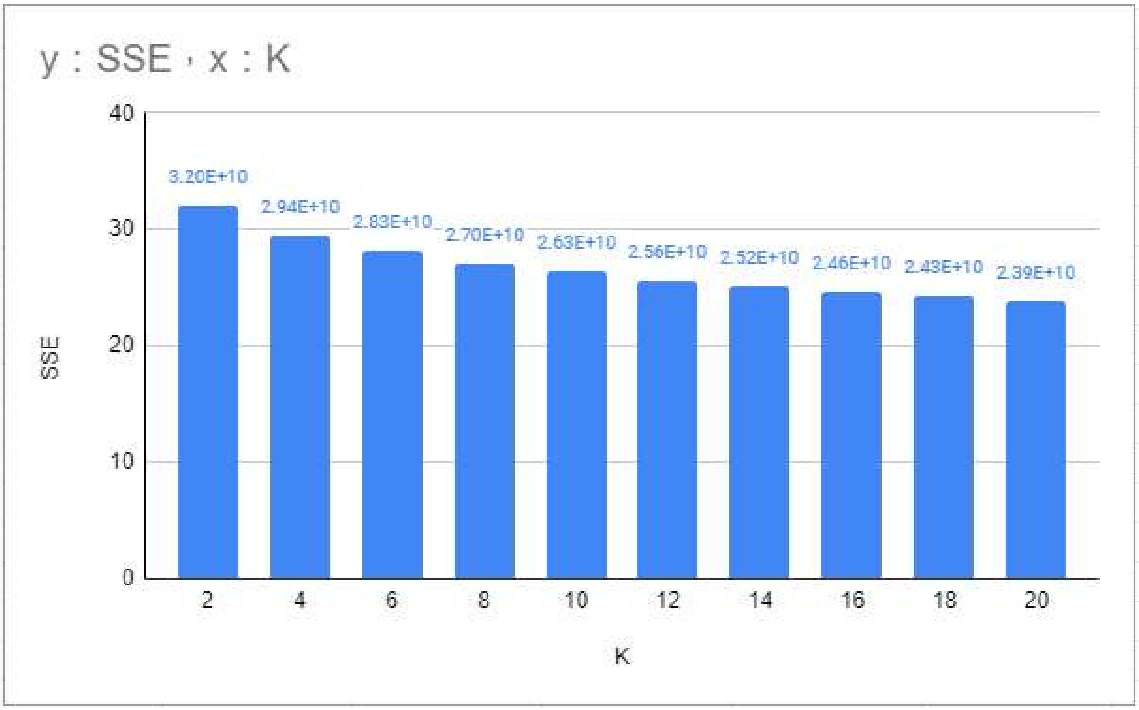
- Loop terminated condition is that the centroids are the same as the previous one.
- Basic k-means :
    - Choose k nodes from data as centroids.
    - Classify each node to one of the centroids. Form k clusters.
    - Compute new centroids. Do step ii again.
    - Stop when centroids are unchanged.
- Bisecting k-means :
    i. Select a cluster with the most nodes in it.
    ii. Run k-means “max_iterations” times with that cluster and k = 2. (Because of bisection).
    iii. Choose the best one with lowest SSE value.
    iv. Repeat i~iii until there are k clusters.
- I use one additional list to store the index of each node in the original data so that it’s easier to give them a class.

3. Experimental Results

- Iris : Bisecting got a better score (0.86) than basic k-means.
- Image Clustering : Basic k-means got my highest score(0.53). However, Bisecting k-means is about 0.38~0.50 and with quite longer runtime.
- SSE values from k = 2 to 20 with basic k-means. (Image Clustering)



- SSE values from k = 2 to 20 with bisecting k-means. Max_iterations = 10. (Image Clustering)



4. Conclusion

- Iris :
    - Basic k-means got score from 0.64 to 0.72. It’s not surprised because it just chooses k random centroids.
    - Bisecting k-means got score 0.86 with “max_iterations” = 150. I think I can get the better score if I try every possible bisection instead of a “max_iterations”. But this way can be only use here because the data size is small (only 150).
- Image Cluster :
    - It spends some hours to run bisecting k-means with “max_iterations” = 400 because the data size is larger.
    - As the graphs show, SSE of basic k-means is lower than SSE of bisecting one with low “max_iterations”. Because it can’t get the best bisection.