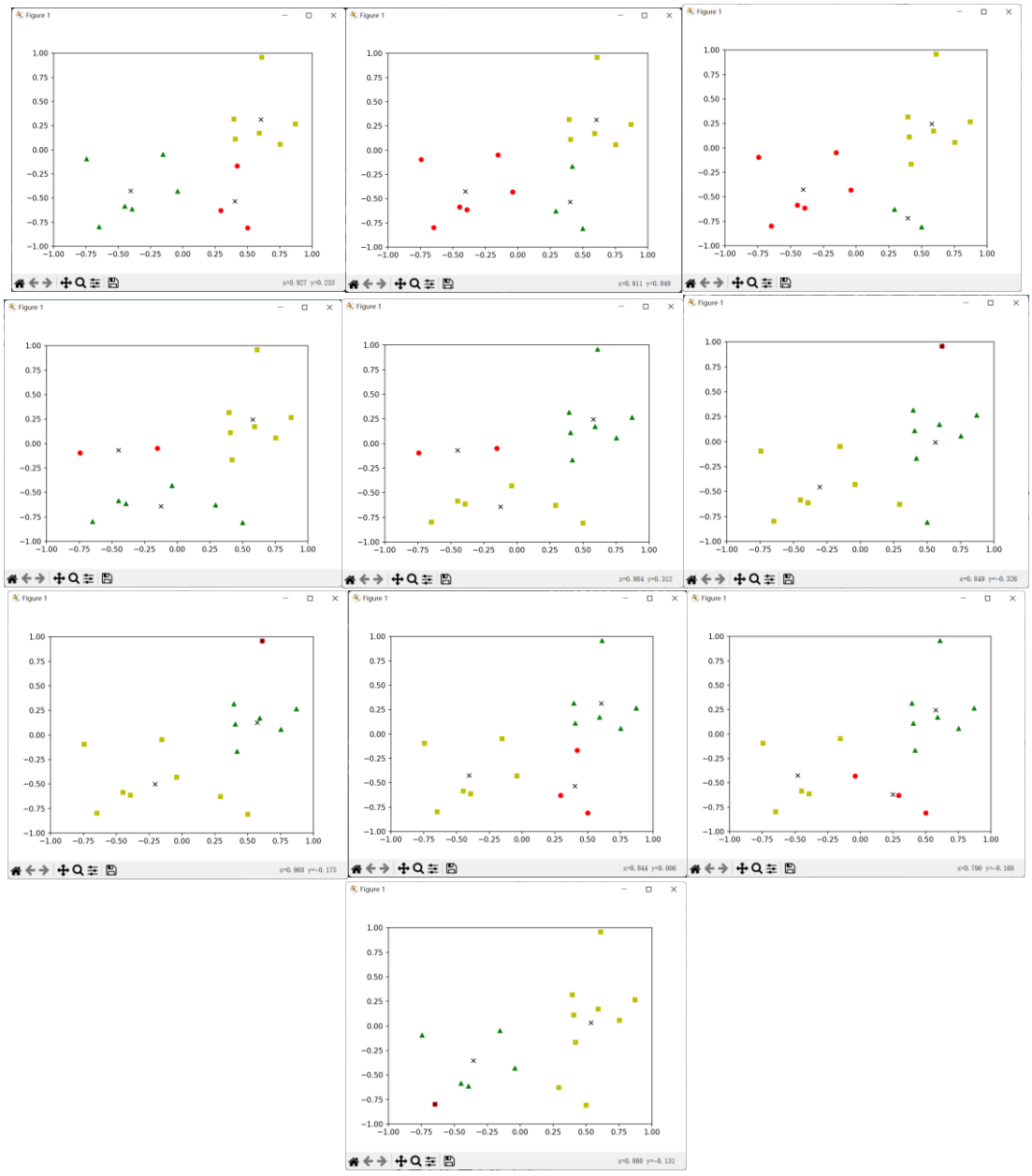
P1: K-means Random set

Using init_board() generated 15 points. The plots after ten runs are as follows.



We can observe that these points are classified into two main clusters.

1. The first cluster is concentrated in the lower central part, the second cluster is in the left central part, and the third cluster is in the right central part.

2. The first cluster is concentrated in the lower left part, the second cluster is in the lower right part, and the third cluster is in the upper right part.

In addition to classifying the rightmost top points into a cluster and the leftmost bottom points into a cluster in the above plots, The rest can be identified as both above. Since the

midpoints of the clusters are randomly assigned, the classification results can still be broadly classified into two main clusters, which are consistent with what is presented in the plots.

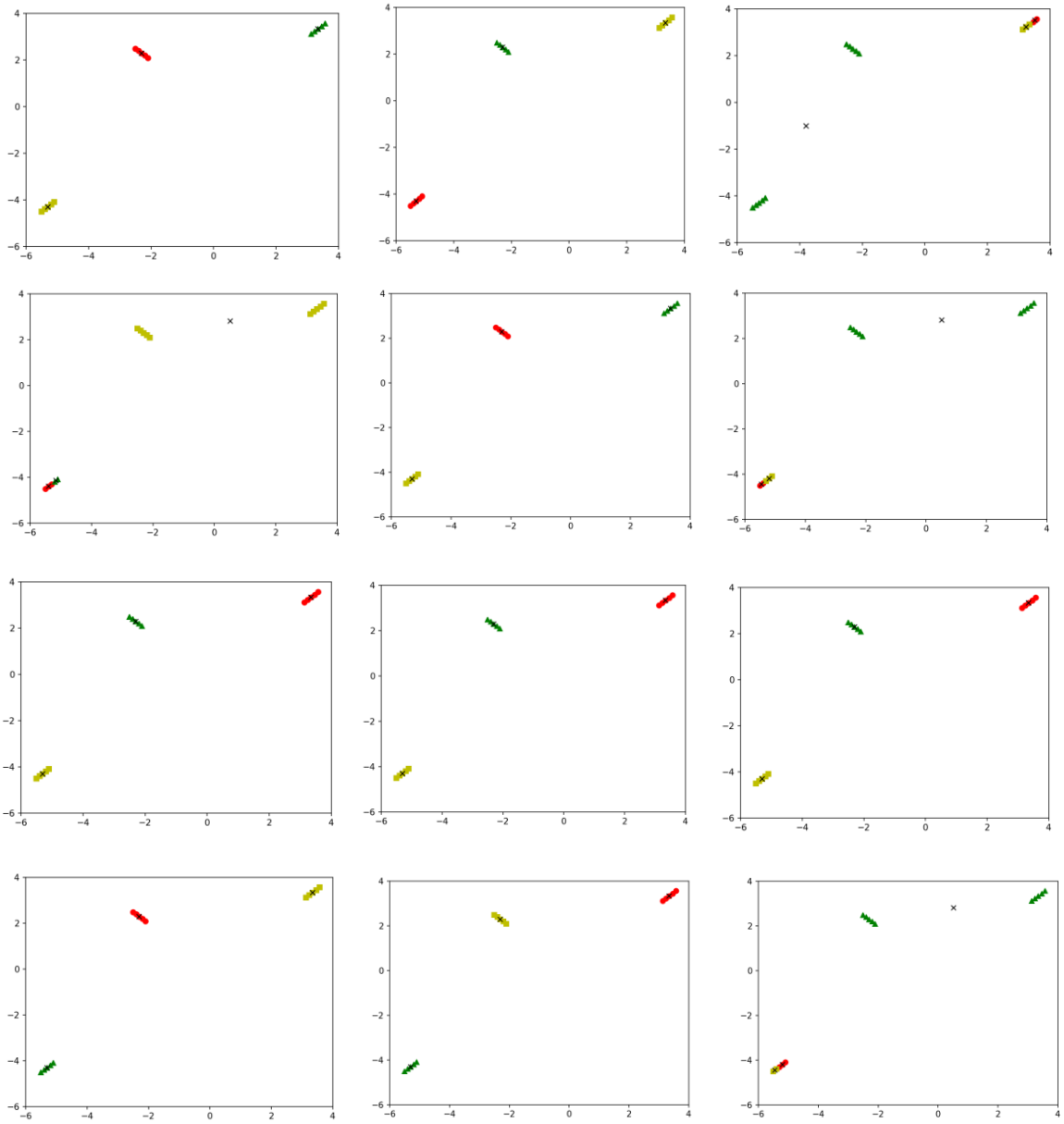
P2: K-means Synthetic set

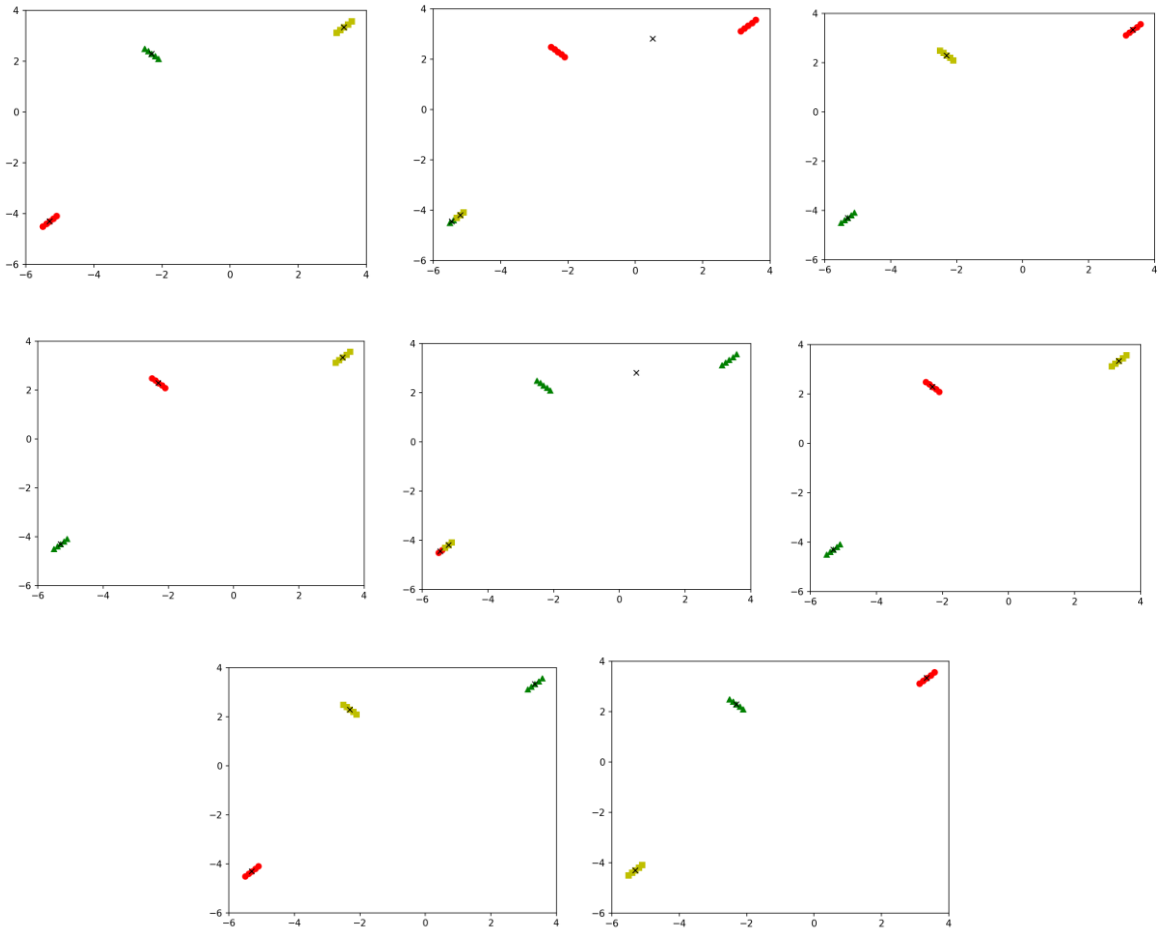
I generated 15 points with linear features. It should be evident in the classification to divide the five points into three clusters.

Data = [-5.1,-4.1],[-5.2,-4.2],[-5.3,-4.3],[-5.4,-4.4],[-5.5,-4.5],

[-2.1,2.1],[-2.2,2.2],[-2.3,2.3],[-2.4,2.4],[-2.5,2.5],

[3.123,3.123],[3.234,3.234],[3.345,3.345],[3.456,3.456],[3.567,3.567]





The overall clustering results are consistent as previously predicted. Every five points were clustered into one class. However, six out of twenty runs were misclassified. In the middle part of some data, it also clusters the data into two parts. And then, the other two groups of five data are clustered into one class. The classification results are not entirely correct.

P3: Problem and Solutions

a. Problem

For a given dataset, randomly select the initial central mass of k clusters. The dataset is divided into k clusters according to the distance between the data. Let the points within the clusters be as close as possible while letting the distance between clusters be as huge as possible. However, the inter-cluster distance is eventually calculated as a local maximum, not a global maximum which is confirmed in the above run result plots. Since the central mass is randomly generated, it has a significant impact on the results of clustering. In the data example I created above, the presence of the central mass in the interior of the data leads to the appearance of two clusters, the latter with the central mass in the middle of two different clusters, resulting in two distinctly different clusters being clustered into one.

b. Solutions

1. K-Means++

David Arthur and Sergei Vassilvitskii have improved the k-means algorithm and proposed the k-means++ algorithm[1]. Firstly, it randomly selects data as the first cluster center. Secondly, it

calculates the shortest distance between each sample and the existing cluster center, denoted by D(x). then, the more significant this value is, the higher the probability of being selected as the cluster center; then, the next cluster center is selected by the roulette wheel method. Then the above steps are repeated until k clustering centers are selected, which is different from k-means, where all clustering centers are selected randomly. Finally, after selecting the clustering centers, the k-means algorithm is used. k-means++ largely avoids the bias of k-means and is faster in handling several orders of magnitude data.

2. Bisecting K-Means

Steinbach, Karypis, and Kumar proposed bisecting k-means to weaken the effect of random center masses of k-means on the clustering results. First, it puts all the data into a queue as a cluster. Secondly, it selects a cluster from the queue for k-means algorithm division into two sub-clusters, which are then added to the queue. Then, the above operation is iterated in a loop until the abort condition is reached regarding the number of clusters, most minor square errors, or the number of iterations. Calculate the error and SSE of all clusters and select the cluster with the largest SSE for the partitioning operation. Finally, the clusters in the queue are the final set of classification clusters. As a result, the better performance of dichotomous K-means relative to ordinary K-means is because it produces clusters of relatively uniform size.

References

[1] Arthur, David and Vassilvitskii, Sergei. "k-means++: the advantages of careful seeding." Paper presented at the meeting of the SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, Philadelphia, PA, USA, 2007.

[2] Steinbach, M., Karypis, G., & Kumar, V. (2000). A comparison of document clustering techniques.