Xudong LIU

7lxd@tongji.edu.cn

School of Software Engineering

Tongji University

Nov 28, 2015

Clustering & K-means Algorithm

ABSTRACT

Clustering is one of the most important tools in data mining. It can cluster data record without any prior information. K-means is one of the clustering algorithm, which is popular simple and effective. I will implement the K-means algorithm with Matlab code and compare it with the Matlab's statistic tools' K-means algorithm and try to show the different in effectiveness and efficiency aspects. The data I will use to test algorithm from Professor Zhao's website*(http://sse.tongji.edu.cn/zhaoqinpei/Datasets/)*.

Keywords: Data Mining,Clustering,K-means Algorithm, Centroids,Effectiveness, Efficiency,Local optimal, Voronoi.

INTRODUCTION

Cluster & Clustering:

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.

K-means Algorithm:

k-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the

nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells.

Given a set of observations (x1, x2, …, xn), where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k (≤ n) sets S = {S1, S2, …, Sk} so as to minimize the within-cluster sum of squares (WCSS) (sum of distance functions of each point in the cluster to the K center). In other words, its objective is to find:where μi is the mean of points in Si.

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$$

Figure 1.1

ALGORITHM & IMPLEMENT

2.1 K-means algorithm:

Given an initial set of k means m1(1),…,mk(1) (see below), the algorithm proceeds by alternating between two steps:

Assignment step: Assign each observation to the cluster whose mean yields the least within-cluster sum of squares (WCSS). Since the sum of squares is the squared Euclidean distance, this is intuitively the "nearest" mean.[8] (Mathematically, this means partitioning the observations according to the Voronoi diagram generated by the means).

$$S_i^{(t)} = \{x_p : \|x_p - m_i^{(t)}\|^2 \leq \|x_p - m_j^{(t)}\|^2 \ \forall j, 1 \leq j \leq k\},$$

where each x_p is assigned to exactly one $S^{(t)}$, even if it could be assigned to two or more of them.

Figure 2.1

Update step: Calculate the new means to be the centroids of the observations in the new clusters.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

Figure 2.2

Since the arithmetic mean is a least-squares estimator, this also minimizes the within-cluster sum of squares (WCSS) objective.

The algorithm has converged when the assignments no longer change. Since both steps optimize the WCSS objective, and there only exists a finite number of such partitionings, the algorithm must converge to a (local) optimum. There is no guarantee that the global optimum is found using this algorithm.

2.2 Matlab implementation code:

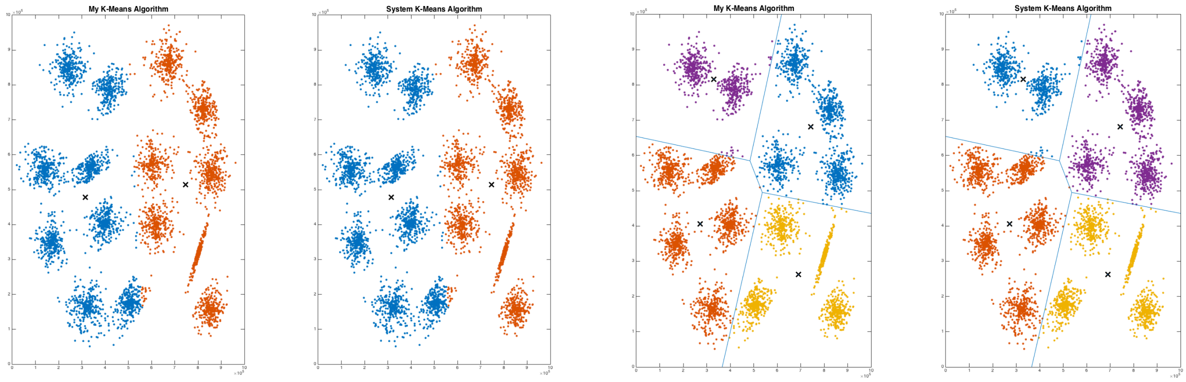
```
function [cluster,centroid] = KmeansAlgriothm(k,data)
len = length(data);
tmp=randperm(len);
index = tmp(1:k);
centroidX = data(index,1);
centroidY = data(index,2);
cluster = zeros(len,1);
% 随机选取k个点作为初始值
centroid = [centroidX,centroidY];
while 1
    % 保留原来的centroid
    old = centroid;
    % 计算每个点的distance
    for i = 1:len;
        distance = zeros(k,1);
        for j = 1:k
        distance(j) = norm(data(i,:)-centroid(j,:));
        end
        [~,ind] = min(distance);
        cluster(i,1) = ind;
    end
   % 更新centroid
   for m =1:k
       centroid(m,:)= mean(data(cluster==m,:));
   end
   if (isequal(old,centroid))
         break;
   end
end
end
```

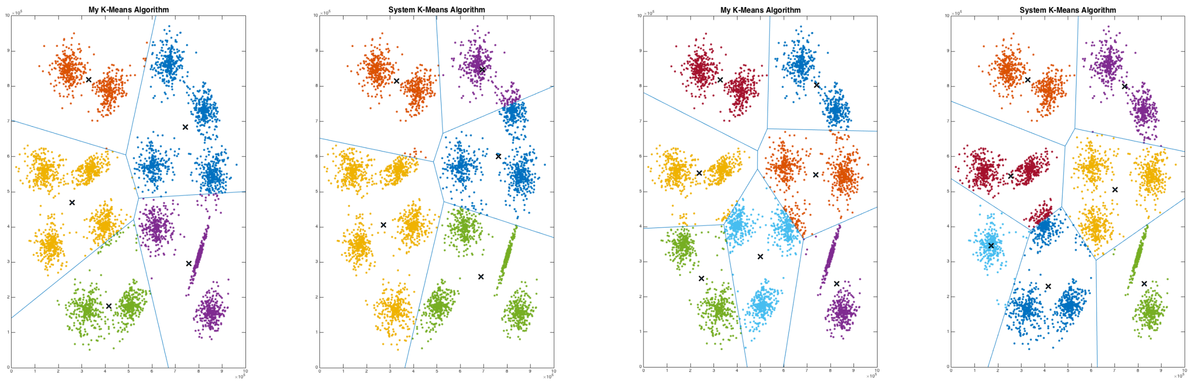
EXPERIEMENT

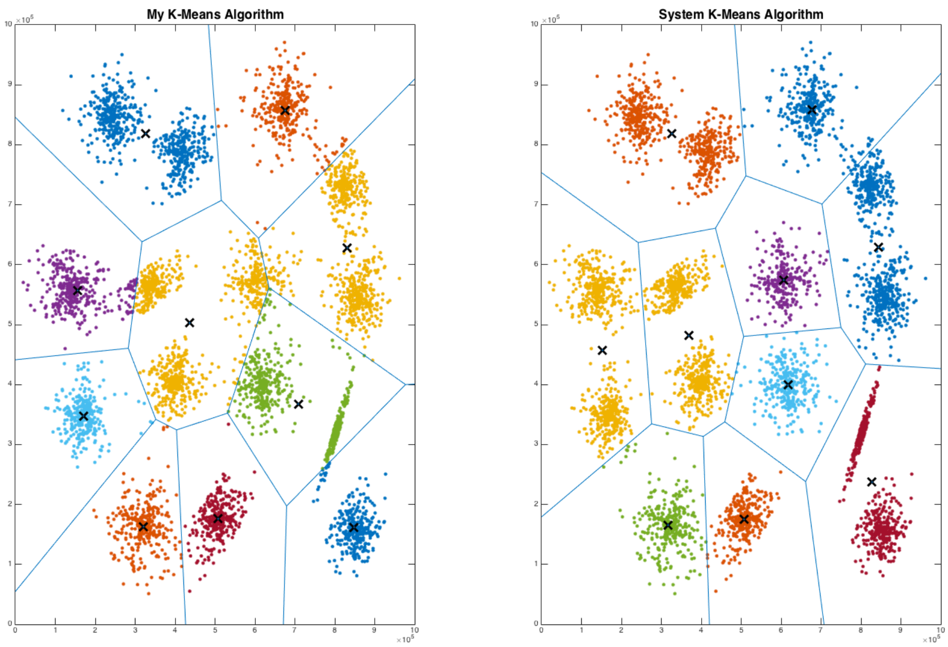
3.1 My K-means Algorithm Result & Matlab Statistic Tools' K-means Algorithm Result

Data Set 1:

k=2 4 5 7 10

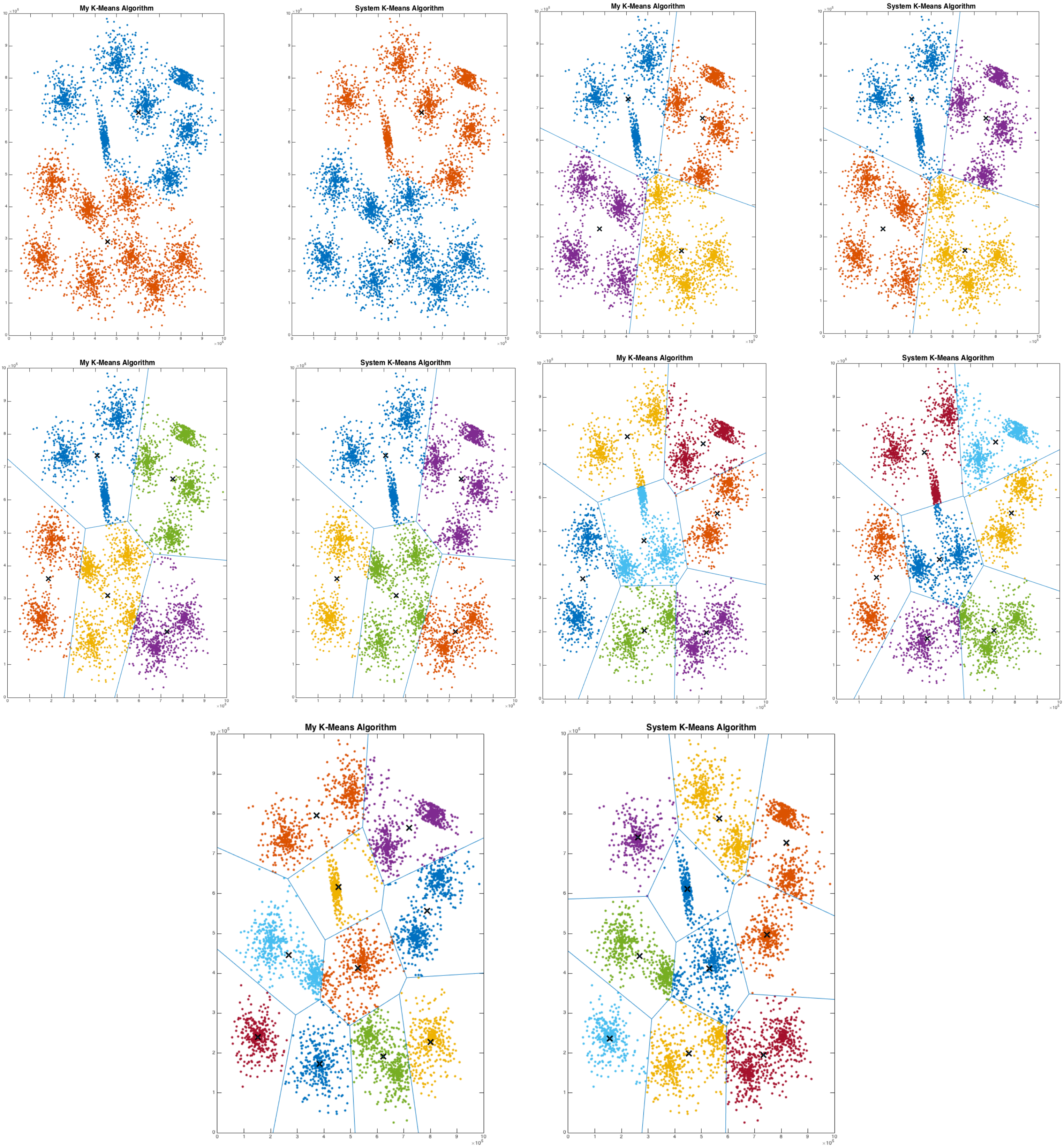

My K-Means Algorithm
System K-Means Algorithm
My K-Means Algorithm
System K-Means Algorithm


My K-Means Algorithm
System K-Means Algorithm
My K-Means Algorithm
System K-Means Algorithm


My K-Means Algorithm
System K-Means Algorithm

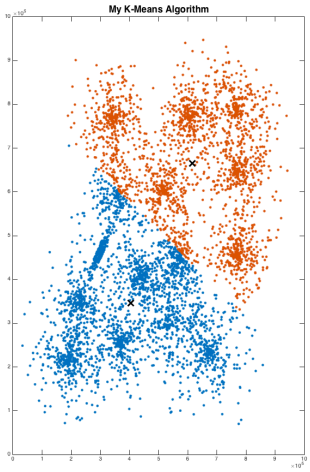
Data Set 2:

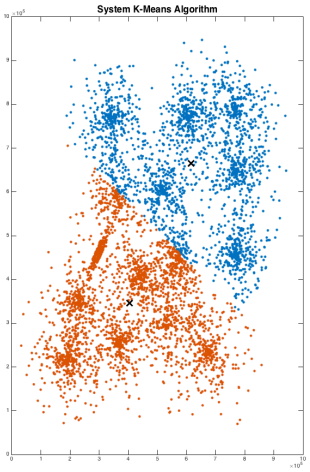
k=2 4 5 7 10

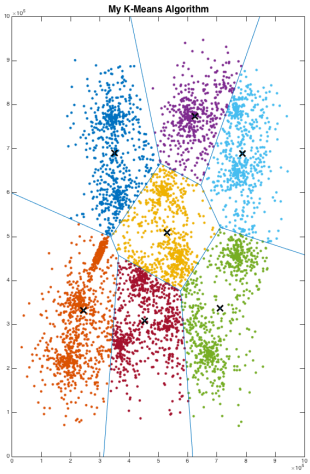


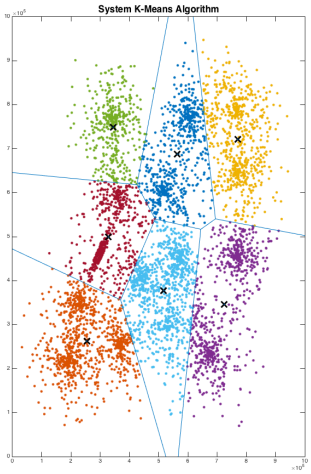
Data Set 3:

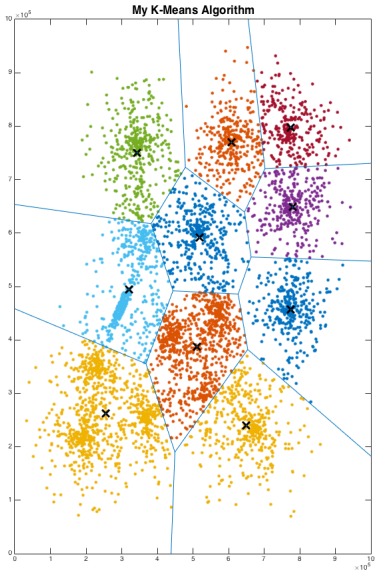
k=2 7 10

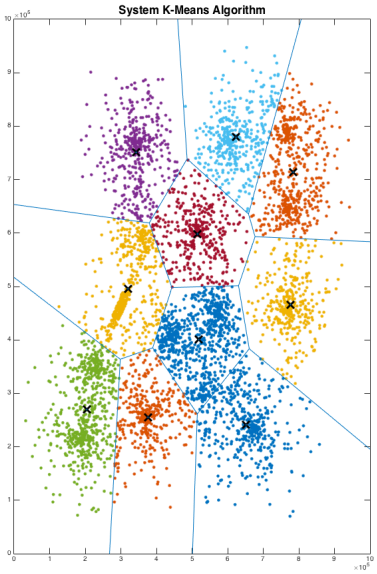






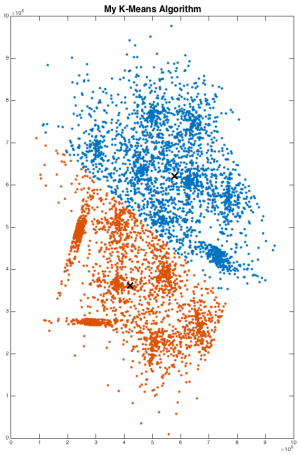


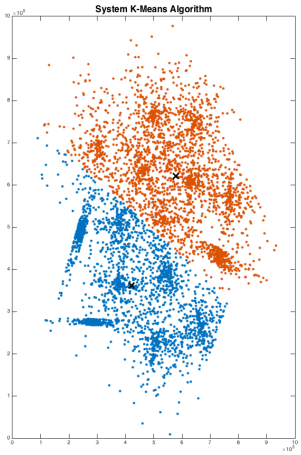


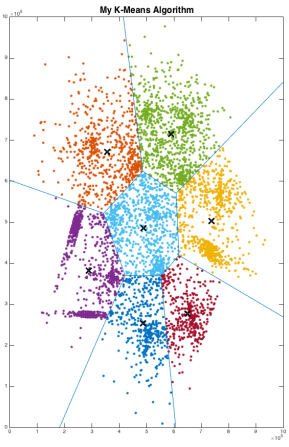


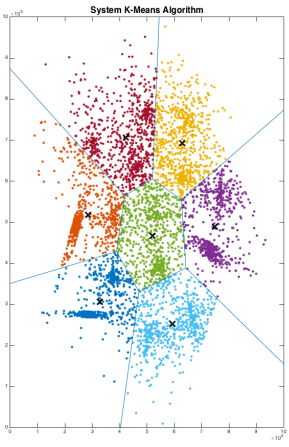
Data Set 4:

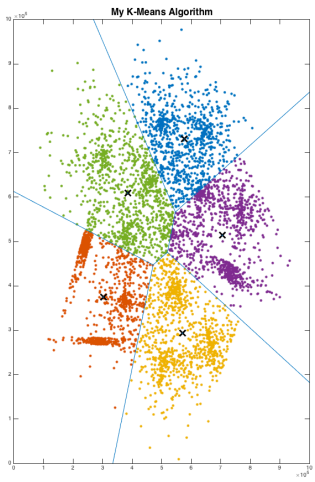
k=2 7 5

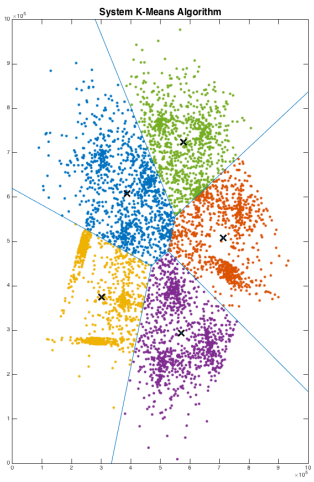








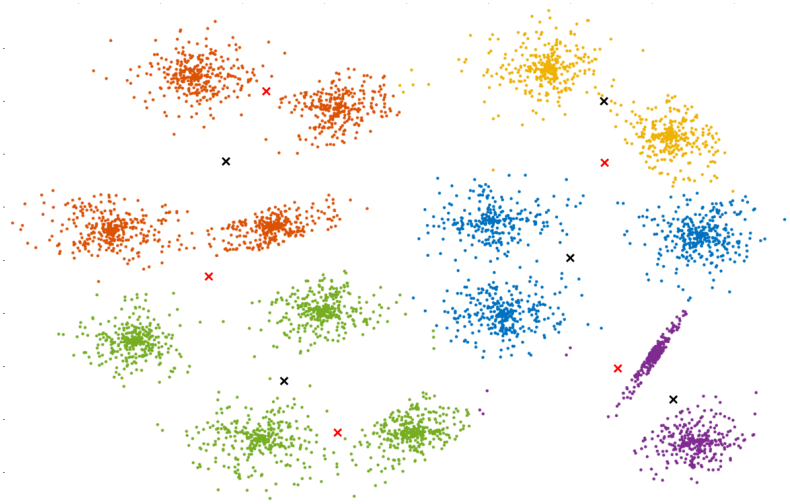


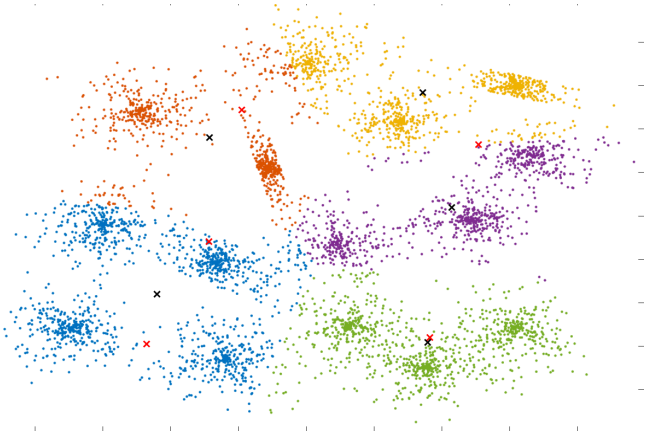


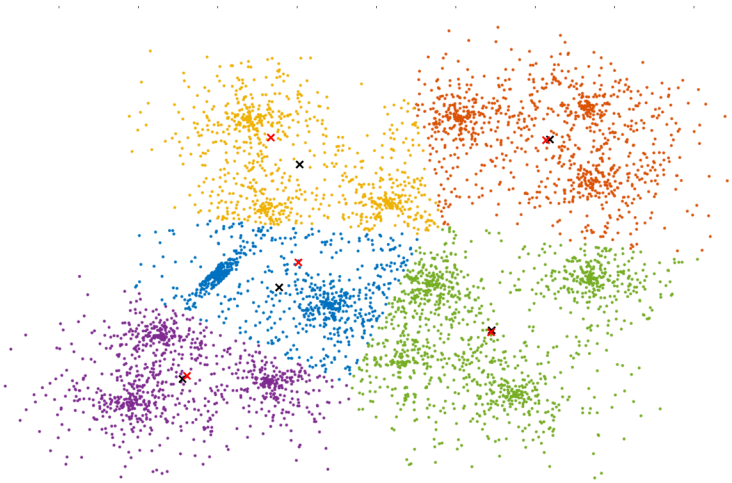
3.2 Compare (k = 5)

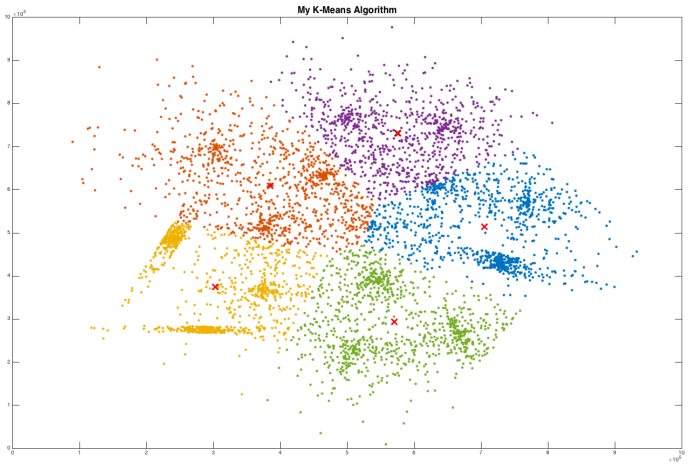
black centroids—my K-means

red centroids — statistic Tools K-means









CONCLUSION

1. K-means algorithm will finally reach a local optimal result but this local optimal may not the global optimal result.
2. K-means algorithm is a NP-hard problem with time consuming.
3. According to random initial centroids, it will show different result. Due to the fact that there may not one local optimal result.
4. Compare my K-means with statistic tools K-means algorithm.

- **Effectiveness:**
- We all reach one local optimal result finally, however, depend on the initial centroids, the result may be just the same or have some different (the different may be very huge, but both of the two result sill the local optimal result of this data). In short, it is easily to get the same effectiveness with the system's K-means algorithm as long as you write the right code.

- **Efficiency:**
- Even thought I try hard to improve my code's performance in the efficiency, I still have to admit that the system's algorithm perform better than mine.I try avoid the loop, use the system's calculator method instead of writing by myself, however, It still can't get faster than the system's algorithm. When K's value less than 10, It is hard to find that my algorithm slower than the systems'. But, with the increase of K's value, system's algorithm speed still can keep fast, and my algorithm's performance can be that good.

REFERENCE:

[1] Wikipedia, Cluster analysis, https://en.wikipedia.org/wiki/Cluster_analysis
[2] Professor ZHAO's homepage, http://sse.tongji.edu.cn/zhaoqinpei/Datasets/
[3] Matlab's official document about K-means, http://cn.mathworks.com/help/stats/kmeans.html