A Comparison Among Several Algorithms Solving Clustering Problems

GAO Yue

August 30, 2017

Date Performed:	June 21, 2017
Instructor:	Professor Anthony So

1 Objective

To find the difference among several algorithms which can solve clustering problems from several perspectives including time complexity and accuracy of running results. In this research, the objective function is :

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} \parallel x - \mu_i \parallel^2$$

Where (x_1 x_2 ... x_n) are n vectors which we want to divide into k clusters; S_1, S_2 ... S_k are the sets representing k clusters, μ_1, μ_2 ... μ_k are the centroids of each cluster.

1.1 Definitions

Clustering Clustering is a data-mining technique used to place data elements into related groups without advance knowledge of the group definitions. This research focuses on clustering points in Euclidean coordinate.

In this report, several clustering methods will be introduced and implementation will be acted on them.

2 Introduction to Several Algorithms and Their Main Properties

2.1 Kmeans Algorithm

Kmeans Algorithm is a typical unsupervised learning method, the implementation steps are as follows:

a. Randomly choose k cluster centroids μ_1, μ_2 ... μ_k

b. Repeat the following processes until converge:
For each i $\in$ {1, 2 ... k}, compute index of the cluster it belongs to by:

$$c^{(i)} := \arg\min_j \parallel x^{(i)} - \mu_j \parallel^2$$

For each cluster j, re-compute its centroid by:

$$\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)} = j\}}$$

Kmeans algorithm processes a number of advantages such as low time complexity, and an assurance of convergence. However, the clustering result may not be always good since Kmeans is sensitive to outliers, initial centroids chosen, shapes of clusters and so on.

2.2 Spectral Clustering

Spectral Clustering uses the eigenvalues of the similarity matrix of data to perform dimensionality reduction before clustering in fewer dimensions:
Given a set of points S = {s_1,...,s_n} in R^l that we want to cluster into k subsets:

a. Form the affinity matrix A $\in$ $R^{n \times n}$ defined by

$$A_{ij} = exp(- \parallel s_i - s_j \parallel^2 /2\sigma^2) \qquad if \qquad i \neq j,$$

and $A_{ii} = 0$. (Here the scaling parameter σ^2 controls how rapidly the affinity A_{ij} falls off with the distance between s_i and s_j).

b. Define D to be the diagonal matrix whose (i,i)-element is the sum of A's i-th row, and construct the matrix L = $D^{-1/2}$A$D^{-1/2}$.

c. Find x_1, x_2,..., x_k, the k largest eigenvectors of L (chosen to be orthogonal to each other in the case of repeated eigenvalues), and form the matrix X = $[x_1 x_2 ... x_k]$ $\in$ $R^{n \times k}$ by stacking the eigenvectors in columns.

d. Form the matrix Y from X by renormalizing each of X's rows to have unit length.

e. Treating each row of Y as a point in R^k, cluster them into k clusters via K-means or any other algorithm (that attempts to minimize distortion).

f. Finally, assign the original point s_i to cluster j if and only if row i of the matrix Y was assigned to cluster j.

In this algorithm, the eigenvector is seen as solving a relaxation of an NP-hard discrete graph partitioning problem. Spectral clustering is less sensitive to outliers than K-means algorithm, so the performance of spectral clustering is usually better than K-means. Besides, spectral clustering adapts to non-convex data well. When solving high-dimensional clustering problems, time complexity of spectral clustering is usually lower than that of K-means.

2.3 Mixed Integer Programming Approach

The mathematical programming model, MIP-Diameter, of the clustering problem is given below:
minimize Z=D_{max}
subject to

$$D_l \geq (\sum_{i,j=1}^{n} d_{ij} x_{il} x_{jl})/(|C_l|(|C_l - 1|)) \qquad \forall l, l = 1, ..., k,$$

$$\sum_{l=1}^{k} x_{il} = 1 \qquad \forall i, \qquad i = 1, ..., n,$$

$$D_{max} \geq \sum_{l=1}^{k} D_l$$

$$x_{il} \in \{0, 1\} \qquad \forall i, l, \quad i = 1, ..., n, \quad l = 1, ..., k,$$

$$|C_l| = \sum_{i=1}^{n} x_{il}$$

$$D_l \geq 0 \qquad \forall l, \qquad l = 1, ..., k.$$

The main disadvantage of MIP is high time complexity, a common method to reduce time complexity is to fix some points (seeds) before solving the MIP model, we now present an algorithm to fix seeds :

a. Set iteration number t=0.
Set lower and upper bounds to l(t) = min$\{d_{ij}\}$, u(t) = max$\{d_{ij}\}$.

b. Set R(t) = $\frac{lt+u(t)}{2}$.

c. Construct the graph $G_{R(t)}$ where instances correspond to vertices, and the edge (i,j) ∈ E, if d_{ij} ≤ R(t).

d. Find a maximal independent set S in $G_{R(t)}$.
If |S| = k, then go to Step 2.
If |S| < k, then set l(t + 1) = l(t), u(t + 1) = R(t), t = t + 1 and go to Step b.
If |S| > k, then set l(t + 1) = R(t), u(t + 1) = u(t), t = t + 1 and go to Step b.

After the above steps, each cluster will have a fixed seed, thus improving the efficiency of the MIP model. However, the time complexity of this algorithm is still large. Despite the accuracy of MIP, when dataset is large, this algorithm is not proper for solving this problem.

2.4 Neuron Network

In this part, SOM (Self-Organizing Map) model will be used to solve the problem. A typical SOM network is consisted of 2 layers, input layer analogs retina, while output layer (also called map) analogs Cerebral cortex. SOM is trained using unsupervised learning to produce a discretized representation of the input space of the training samples. Associated with each neuron are a weighted vector of the same dimension as the input data vectors, the idea for placing a vector from data space onto the map is to find the neuron with the closest weight vector to the data space vector. The update formula for a neuron v with weight vector W_v(s) is

$$W_v(s+1) = W_v(s) + \theta(u, v, s) \cdot (D(t) - W_v(s))$$

where t is an index into the training sample, s is the step index, u is the index of the BMU (best matching unit) for D(t), D(t) is the input vector, θ(u,v,s) is the neighborhood function which gives the distance between the neuron v and neuron u in step s, α(s) is a monotonically decreasing learning coefficient.
The algorithm is as follow:

a. Initialize iteration times, input vector, initial weight vector (generate randomly), initial position of neurons and learning rate.

b. Traverse each input vector, compute the winning neuron according to the Euclidean distance between neurons and vectors.

c. Update weight vector by the formula mentioned above:

$$W_v(s+1) = W_v(s) + \theta(u, v, s) \cdot (D(t) - W_v(s))$$

d. Updating learning rate and neighbor function.

e. When the number of iterations reaches the set times, end, otherwise update s and return to step b.

In this approach, the initialization of parameters is important, with proper learning coefficient fixed, the performance of this algorithm will be quite accurate and time complexity is much smaller than MIP model. Unlike K-means algorithm, which stops as soon as it converges, SOM posseses an important property that it will keep learning and stop only if the number of iterations has reached the bound. In summary, SOM is an appropriate model for solving clustering problems. Disadvantages of SOM is that the parameters need to be adjusted, and although in most cases it converges, actually SOM does not always converge.

2.5 Hierarchical Methods

The main difference between this method and others is that this method does not need to fix number of clusters in the first, it only need a lower bound or an upper bound for the number of clusters. Stratეties generally fall into two types: Agglomerative (”bottom-up”, each point stars in its own cluster) and Divisive (”top down”, all points start in one cluster). We only focus on agglomerative methods here.

There are different kinds of agglomerative methods, in the general case, the time complexity of agglomerative clustering is O(n^2log(n)), for some special cases, optimal efficient agglomerative methods of complexity O(n^2) are known. Here, we introduce an agglomerative algorithm (centroid clustering) of time complexity O(n^2log(n)):

Firstly, according to the objective formula, the distance/similarity parameter between two clusters is defined as follow:

$$d(C_i, C_j) = \vec{\mu}(C_i)\cdot\vec{\mu}(C_j) = (\frac{1}{N_i}\sum_{d_m \in Ci} \vec{d_m})\cdot(\frac{1}{N_j}\sum_{d_n \in Cj} \vec{d_n}) = \frac{1}{N_i N_j}\sum_{d_m \in C_i}\sum_{d_n \in C_j} \vec{d_m}\cdot\vec{d_n}$$

The algorithm is as follows:

a. Set an expected maximum of the number of clusters.

b. See each point as a cluster C_i, i = 1 to n.

c. Find the clusters C_i and C_j such that among all pairs of the clusters, d(C_i, C_j) is the smallest.

d. collapse C_i and C_j as one cluster.

e. If the number of clusters is larger than the expected maximum number, return to step c.

3 Experimental Data

Implementation for the above algorithms was did on 3000 points in 2-D. (But for MIP algorithm, since the time complexity is too large, the implementation for it is on only 81 points)

3.1 Kmeans

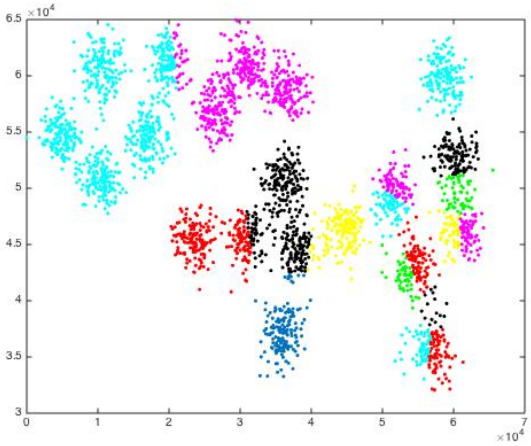

Figure 1: Result of K-means Algorithm

The elapsed time is 2.358390 seconds.
As shown in the figure, under K-means algorithm, the clustering result is not accurate since it converges too fast. The advantage is that the time cost is little.

3.2 Spectral Clustering

Performance of this algorithm under different coefficients were tested and compared:

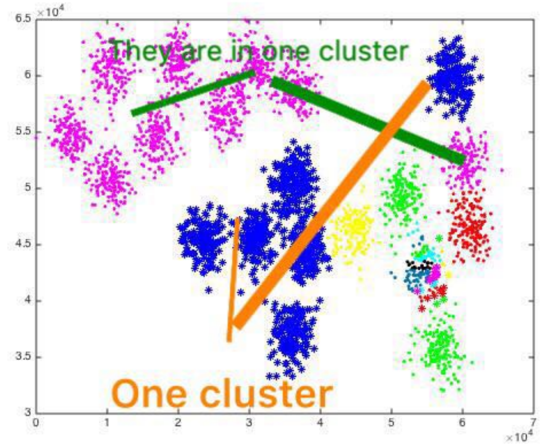

Figure 2: Result of Spectral Clustering Algorithm when coefficient is 500, the total elapsed time is 11.5325 seconds.

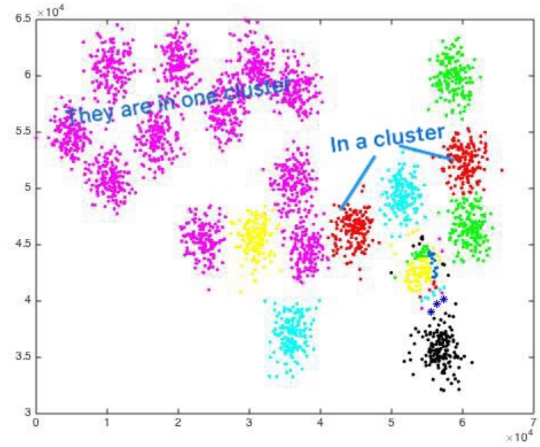

Figure 3: Result of Spectral Clustering Algorithm when coefficient is 1000, the total elapsed time is 9.5933 seconds.

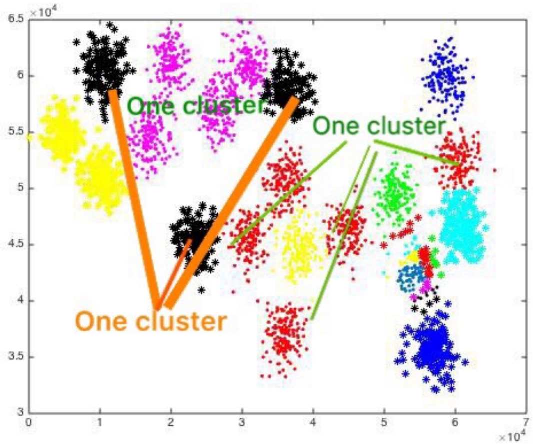

Figure 4: Result of Spectral Clustering Algorithm when coefficient is 1500, the total elapsed time is 8.9665 seconds.

From the figure 2 to 4, it can be shown that the performance of spectral clustering is better than Kmeans since the boundaries of clusters are well divided, and the time cost is little, however, the performance is not good enough since the sizes of the clusters are so different. This maybe due to that the most important advantage of Spectral Clustering is dimensionality reduction, which means that it performs better when the dimensional number is larger than k. In this dataset, the dimensional number is 2, k is 20, hence, the advantage of Spectral Clustering is not shown in this example, but in high dimensional case, it might perform better and the time cost would be even smaller than Kmeans.

3.3 MIP

Since the time complexity of MIP is too big, implementation was did only on 81 points shown in figure 5.

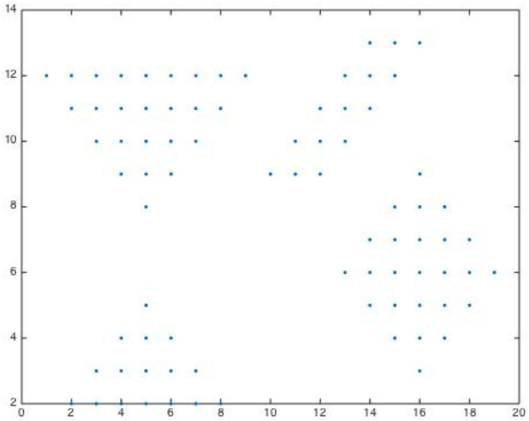

Figure 5: Data points

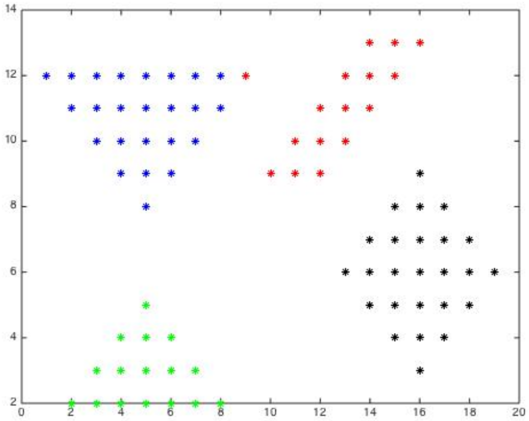

Figure 6: Result of MIP without seeds and reassignment, the elapsed time is 15.6 seconds

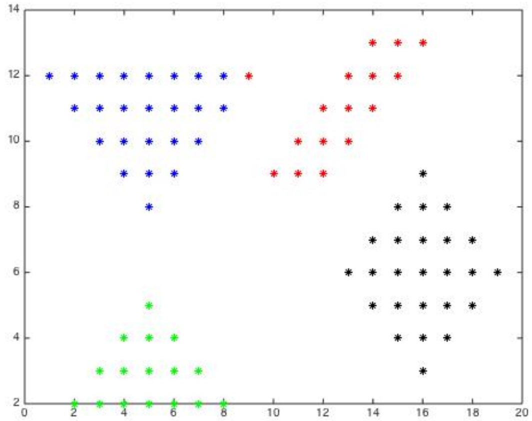

Figure 7: Result of the advanced MIP model with seeds fixed, the elapsed time is 1.4 seconds

As shown in figure 6-7, it is clear that the performance of MIP is rather accurate if the shapes of clusters are convex enough, and fixing seeds before solving the MIP can largely reduce time cost.

3.4 Neuron Network

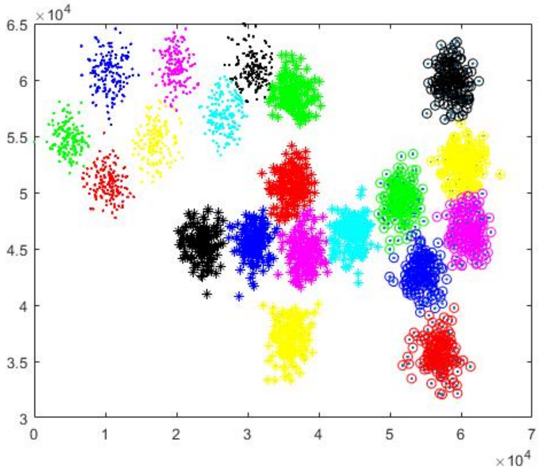

Figure 8: Result of SOM Algorithm when iteration time is 3000, the total elapsed time is 382.42285 seconds.

It can be seen from the figure that after iterating for 3000 times, the performance of SOM is very good, the time cost is larger compared to Kmeans and Spectral Clustering, but overall, SOM Algorithm is quite proper for solving this problem.

3.5 Hierarchical Methods

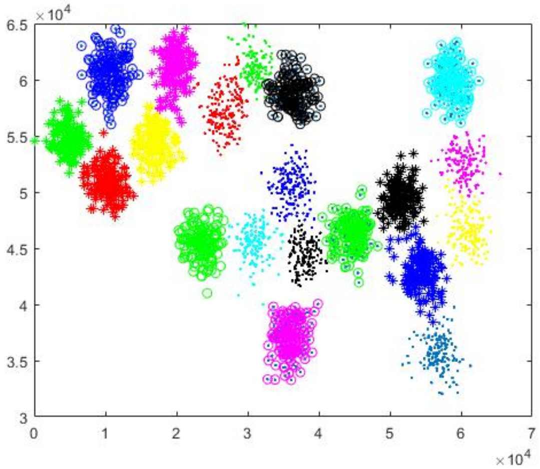

Figure 9: Result of Hierarchical Method, the total elapsed time is 3.608544 seconds.

The performance of the agglomerative algorithm is accurate, and the time cost is quite low, it could be claimed that agglomerative algorithm solves the objective problem very well.

4 Results and Conclusions

The time complexity of the five algorithms introduced are listed as follows:

Algorithm	Time Compexity
Kmeans	O(knlt) (t is the number if iterations)
Spectral Clustering	O(n^3)
MIP model	O(2^n)
SOM	O(knlt) (t is the number if iterations)
Agglomerative Method	O(n^2log(n))

Seen from the implementations results on 3000 points, for the above algorithms, the Hierarchical Method and Neuron Network (SOM) algorithm performs most accurately, and the time costs are acceptable. In summary, when the size of dataset is large and the number of clusters are known as pre-request, SOM is a good choice. When number of clusters is not known or the shapes of clusters are unpredictable, hierarchical methods would perform better. When the size of dataset is small, MIP can accurately solve the problem. When the dimensional number of the dataset is high, spectral clustering would be efficient in solving such problem. K-means algorithm is also an efficient algorithm but the choice of initial centroids is essential, fixing seeds for K-means algorithm may improve the performance of it.

5 Discussion of Experimental Uncertainty

For the objective formula, the purpose is to gather points as close to centroids of clusters as possible, which performs good when the clusters are all convex, however, in reality, there are many non-convex cases in clustering problems. Besides, other than distances between points,there are many different criterias in defining a 'good' cluster, for example, the maximum diameter of clusters, the density of clusters and so on. Under different criterias the objective formulas are different, leading to quite different results.

5.1 non-convex clustering

There are cases that the shapes of clusters are not strictly convex, for instance, the dataset shown in figure 10.

In fact, in this case, the above methods can also be used but the Euclidean distance should not be the criteria in the algorithms, other metrics such as kernel distance could be applied. For instance, as shown in figure 11, hierarchical methods are able to solve such a problem, as long as the definition of the distance/similarity parameter is changed (for example, algorithm for computing linkages of clusters is changed from 'centroid' to 'single' in Matlab).

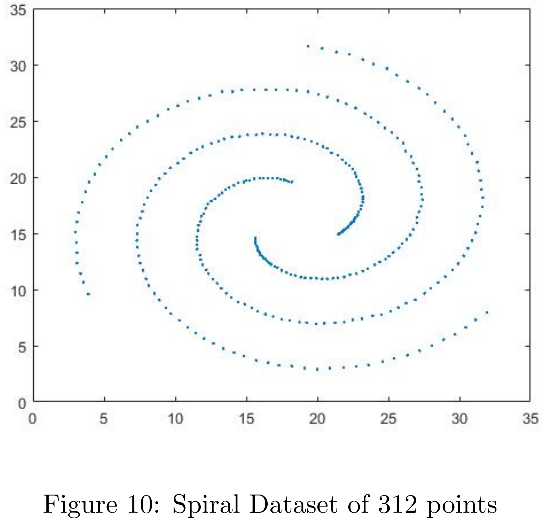

Figure 10: Spiral Dataset of 312 points

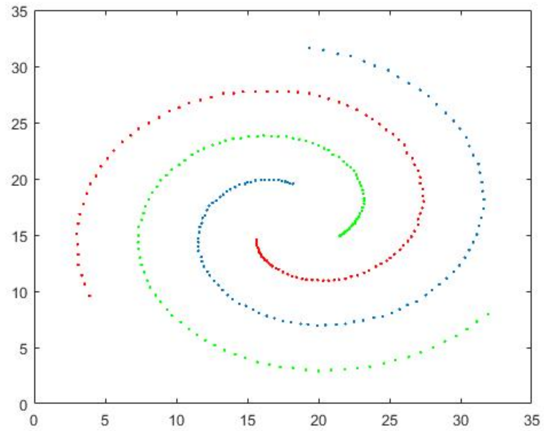

Figure 11: Result of Hierarchical Method, the total elapsed time is 0.020246 seconds.

6 References

a. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schtze, Introduction to Information Retrieval, Cambridge University Press. 2008.

b. Andrew Y.Ng, Michael I.Jordan and Yair Weiss. On Spectral Clustering: Analysis and an Algorithm. In Advances in Neural Information Processing Systems 14, pages 849-859, 2002.

c. Burcu Saglam, F. Sibel Salman, Serpil Sayin and Metin Turkay. A mixed-integer programming approach to the clustering problem with an application in customer segmentation[J]. European Journal of Operational Research, 2006, 173(3): 866-879.

d. A. Likas, N. Vlassis, J.J. Verbeek, The global K-Means clustering algorithm, Pattern Recognition 36 (2003) 451-461.

e. Vikas Chaudhary, R.S. Bhatia, and Anil K. Ahlawat, A novel Self-Organizing Map (SOM) learning algorithm with nearest and farthest neurons[J]. Alexandria Engineering Journal, 2014, 53(4): 827-831.