A New Clustering Algorithm Based on Self-Updating Process

Ting-Li Chen[1], Shang-Ying Shiu[1]
Institute of Statistical Science, Academia Sinica[1]

Abstract

Many of the popular clustering methods, such as K-means and Self-Organizing Maps, require a set of initial values to begin the iterative process. In this paper we present a simple and novel method that does not require such an initial set and can avoid the problem of local minima. The clustering strategy we propose is motivated by intuition on clustering. The algorithm stands from the viewpoint of subjects to be clustered and simulates the process of how they perform self-clustering. At the end of the process subjects belonged to the same cluster would converge to the same point, which represents the cluster location in a p-dimensional space. Our simulation study showed promising results compared to other clustering methods. An example on image segmentation will also be presented.

KEY WORDS: clustering, k-means, image segmentation.

1. Introduction

Clustering analysis is a useful technique to discover groups in the data. This technique has been widely applied to many disciplines for partitioning data into several clusters; within each cluster subjects are considered to resemble each other. For example, in image segmentation the cluster technique is used to partition an image into regions, each of which has its own color patterns. In Psychiatry the cluster technique is often used to cluster patients on the basis of their clinical and questionnaire responses. The resulting grouping structure can provide valuable information on identifying subtypes of a psychiatric disease. In biology and medicine, clustering has rapidly become a popular approach to understand and identify patterns in genome data, including in microarray gene expression data and in proteomics data.

A vast number of clustering algorithms have been developed in the literature. Among those the following two types of clustering methods are most commonly used. The first type is hierarchical clustering, which partitions data into clusters through a series of steps that operate on the proximity measure between subjects. The structure of data is revealed through the process of hierarchical clustering and is presented by a tree diagram known as dendrogram. One weakness of hierarchical clustering is the irrevocable clustering assignments: A mistake made at early steps can never be corrected at later steps.

In the second type of the commonly used clustering methods, the clustering results are obtained as an optimal solution that either maximizes or minimizes a criterion of some kind. The k-means algorithm (McQueen, 1967) that employs the square error criterion is the most frequently used clustering algorithm of this sort. However, such algorithms usually require an initial partition to start the iterative process, and the number of cluster has to be given a priori. In addition, this type of algorithms suffers from the problem of trapping into local minima (or maxima), which is a result of a poor selection of initial partitions. There exist many methods to improve the performance of the k-means algorithm, including estimation of the number of clusters (Milligan and Cooper, 1985, Tibshirani et al., 2001) and solving the local minima problem (Selim and Alsultan, 1991, Tseng and Wong, 2005).

In this paper we present a new algorithm for clustering analysis, aiming to bypass the aforementioned weaknesses of currently existing clustering algorithms. The new algorithm was first inspired by the idea of iterative generated correlation matrices (McQuitty, 1968) adopted in the Generalized Association Plots (Chen, 2002), then turned into a self-updating process (SUP) that is built upon the intuition behind clustering. By introducing a parameter that controls the degree of influence between subjects, the number of clusters is determined accordingly. Our simulation results show that the new algorithm can outperform other existing clustering methods, especially for highly noisy data.

This paper is organized as follows. Section 2 introduces the new clustering algorithm. Section 3 presents simulations that demonstrate the performance of the new algorithm and show the comparison results with other clustering methods. In Section 4 we provide a mathematical proof that guarantees the convergence of our algorithm. An illustrative example is given by an application to the problem of image segmentation in Section 5, followed by a discussion section presented in Section 6.

2. Algorithm

The central idea of our self-updating clustering algorithm can be illustrated by the following example.

Suppose there are a lot of students on the playground. A teacher asks them to form into several groups. What will the students do? Each student will probably move towards others who are closer, with respect to their locations at the playground or to any feature that can characterize the students' relationship. If everyone moves by this rule, the students will gradually form into groups.

Based on the simple and intuitive concept as described above, we propose a new clustering algorithm. Suppose there are N subjects to be clustered. For each subject, there are P observations (random variables) representing the subjects' features. We can view each subject as a data point in a P-dimensional space. Imitating the aforementioned example, we can construct the following mechanism to move the data points (subjects). The movement of each subject is determined by the between-subject proximity, which can be any measure such as the Euclidean distance or correlations.

The algorithm can be written as follows.

1. $X_1^{(0)}, X_2^{(0)}, \cdots, X_N^{(0)} \in R^p$ to be clustered.

2. At time $t+1$, every point is updated according to

$$X_i^{(t+1)} = \frac{\sum_{j=1}^{N} f(X_i^{(t)}, X_j^{(t)}) \cdot X_j^{(t)}}{\sum_{j=1}^{N} f(X_i^{(t)}, X_j^{(t)})}. \quad (1)$$

3. Repeat 2) until every point converges.

f is a statistic that measures the between-subject proximity. For the simulated examples and the application to be presented later, we propose to use

$$f(u,v) = \begin{cases} \exp[-\frac{d}{\lambda}] & d \le r \\ 0 & d > r. \end{cases} \quad (2)$$

where r and λ are fixed constants, and d is the Euclidean distance from u to v.

3. Simulation

In this section we conduct two simulation examples. The first example demonstrates the role of r in (2) to determine the number of clusters in the data. The second example simulates highly noisy data to compare the performance of our new algorithm with that of the k-means algorithms using results of hierarchical clustering as the initial partition.

Example 1.

For each $\mu_i \in \{$(0,0), (2,0), (1,1), (6,0), (8,0), (7,1), (3,3), (5,3) and (4,4)$\}$, we sample 20 points from bivariate normal distributions $BVN(\mu_i, I_2/25)$ with zero correlation. The data is shown in Figure 1. If we choose $r = 0.6$ and $\lambda = 1$, the new algorithm moved the 180 simulated data points into nine groups. If we choose $r = 2$ and $\lambda = 1$, the data points moved into three groups.

Note that, when using the k-means algorithm, you have to assign the number of clusters. In our algorithm, we do not have to estimate the number of clusters. In fact, we do not know how many clusters will be produced by this algorithm. What we control here is the r, and that represents how different you allow elements in the same cluster to be. If you have an idea on what a cluster should look like in your data, you probably know how to choose r. And we think that this is a more natural way to cluster data, instead of determining the number of clusters first.

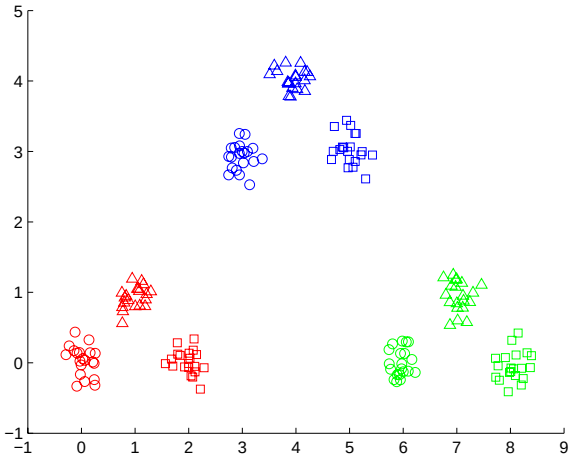

Figure 1: 3 groups, each has 3 subgroups

Example 2.

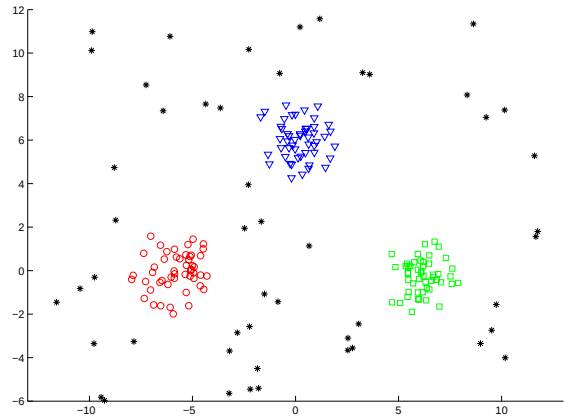

Figure 2: 3 groups with noises

The second example is proposed by Tseng and Wong (2005), in which standard normal distributions centered at (-6,0), (6,0) and (0 6) were sampled fifty times, respectively. Each sampled point is restricted to be within two standard deviations to its center. The noises were sampled uniformly from [-12, 12] $\times$ [-6, 12], but not within three standard deviations to any of the three centers. In Figure 2, points of three clusters are displayed with circles, squares and triangles. Points with symbol star represent noises.

Tseng and Wong proposed a method to overcome the local minimum problem of K-means. Suppose k is the

number of clusters. They first apply hierarchical clustering to get $k \times p$ clusters for some p. Then they choose k-largest clusters among them to determine the initial value of the K-means centers.

We compare our algorithm using $r = 4$ and $\lambda = 1$ to their method by conducting 100,0000 runs of simulations for each of the four different scenarios: 10, 50, 100 and 150 points of noises, respectively. In every run of the simulation, if any point is not clustered correctly, we label this run of simulation as a "mistake". Table 3 presents the number of mistakes in 100,000 runs of simulations. It shows that our proposed algorithm outperformed other methods by never making a single mistake.

			50*3 +10	50*3 +50	50*3 +100	50*3 +150
Km	no initial		8166	1790	2165	3723
	Single Linkage	p=1	3218	2315	1496	3183
		p=3	5	3128	1556	1532
		p=6	1	107	73	933
	Complete Linkage	p=1	285	1450	1994	3713
		p=3	34	0	0	347
		p=6	4458	154	12	216
SUP			0	0	0	0

Table 1: Mistakes in 100,000 runs of simulations

Local minimum problem is a weakness of K-means, and it is due to a poor selection of initial values. Even with some clever techniques that try to overcome this problem, it can not be fully solved. Our proposed algorithm does not need an initial value so that it can avoid the local minima problem.

4. Convergence

One may ask whether our algorithm will converge. In this section, we will prove the convergence of SUP with the proximity measure (2).

Definition 1. *The convex hull $C(X)$ for a set of points X in a vector space $\mathcal{V}$ is the minimal convex set containing X.*

Lemma 1. *Let $C_1^{(t)}$ be the convex hull of $\{X_1^{(t)}, X_2^{(t)}, \cdots, X_N^{(t)}\}$. Then*

$$C_1^{(0)} \supseteq C_1^{(1)} \supseteq \cdots \supseteq C_1^{(t)} \supseteq \cdots.$$

Proof. Since

$$X_i^{(t+1)} = \frac{\sum_{j=1}^{N} f(X_i^{(t)}, X_j^{(t)}) \cdot X_j^{(t)}}{\sum_{j=1}^{N} f(X_i^{(t)}, X_j^{(t)})},$$

$X_i^{(t+1)}$ is a weighted average of $X_j^{(t)}$ for $j = 1, \cdots, N$. Therefore,

$$X_i^{(t+1)} \in C_1^{(t)}.$$

Since the above is true for each i. We have

$$C_1^{(t)} \supseteq C(\{X_1^{(t+1)}, X_2^{(t+1)}, \cdots, X_N^{(t+1)}\}) = C_1^{(t+1)}.$$

□

The convex hull of any finite set of points in R_p is a polytope. Therefore, each $C_1^{(t)}$ is a polytope. Each vertex of $C_1^{(t)}$ must contain at least one $X_i^{(t)}$ for some i, otherwise the polytope should be smaller. Let C_1 be the limit of $C_1^{(t)}$:

$$C_1 \equiv \lim_{t\to\infty} C_1^{(t)} = \bigcap_{t=0}^{\infty} C_1^{(t)}.$$

We claim the following.

Lemma 2. *For each vertex $v_{1,i}$ of C_1, there exists at least one j, such that*

$$\lim_{t\to\infty} X_j^{(t)} = v_{1,i}. \tag{3}$$

Proof. Since

$$C_1 = \lim_{t\to\infty} C_1^{(t)},$$

there exists i (exchange vertex indices if necessary), such that

$$\lim_{t\to\infty} v_{1,i}^{(t)} = v_{1,i}.$$

Since $\forall t$,

$$v_{1,i}^{(t)} = X_k^{(t)}$$

for at least one k, there exists j, such that

$$X_j^{(t)} = v_{1,i}^{(t)}$$

for infinite many t. Therefore, there exists $t_n \to \infty$, such that

$$X_j^{(t_n)} = v_{1,i}^{(t_n)},$$

which leads to

$$\lim_{n\to\infty} X_j^{(t_n)} = v_{1,i}.$$

If $X_j^{(t)} = v_{1,i}^{(t)}$ except for any finite t, the equation (3) is established. Otherwise, there exists $j' \neq j$ and $s_n \to \infty$, such that

$$X_{j'}^{(s_n)} = v_{1,i}^{(s_n)}.$$

Without loss of generosity, assume that $v_{1,i}^{(t)} = X_j^{(t)}$ or $X_{j'}^{(t)}$ for all $t > T$. From equation (1), if $X_j^{(s)} = X_{j'}^{(s)}$ for some s, $X_j^{(t)} = X_{j'}^{(t)}$ for all $t > s$. Therefore, for any $s > 0$, there exist $t > s$, such that $v_{1,i}^{(t)} = X_j^{(t)}$ and $v_{1,i}^{(t+1)} = X_{j'}^{(t+1)}$. Furthermore, we can choose s large enough, so that $C_1^{(s)}$ is close enough to C_1. Precisely, for any ϵ, there exists s, such that

$$|v_{1,k}^{(s)} - v_{1,k}| < \epsilon \qquad \forall k.$$

From the definition of f in (2), f is smaller than 1 unless the subjects are the same, which means each subject is most similar to itself. Since $X_{j'}^{(t+1)}$ is the weighted average of $X_k^{(t)}$, $X_{j'}^{(t)}$ can not be too far from $v_{1,i}$, otherwise, $X_{j'}^{(t+1)}$ will not be at $v_{1,k}^{(t+1)}$, which is not inside the C_1. $v_{1,k}^{(t)}$ is also not inside the the C_1, and is within ϵ to $v_{1,i}$. Therefore, $X_{j'}^{(t)}$ has to be within ϵ to $v_{1,i}$ that $X_{j'}^{(t+1)}$ can be at $v_{1,k}^{(t+1)}$. Since ϵ can be chosen arbitrary small, now we let ϵ small enough that all the projections, except $k = j, j'$, from $X_k^{(t)}$ to $\overrightarrow{X_{j'}^{(t)} X_j^{(t)}}$ fall into the negative side. This means that all other subjects are closer to $X_{j'}^{(t)}$ than $X_j^{(t)}$, and they have effects to pull both towards the convex hull. Since $X_{j'}^{(t)}$ is closer to other subjects, the values of f's are larger. Recall that

$$X_j^{(t+1)} = \frac{\sum_{k=1}^{N} f(X_j^{(t)}, X_k^{(t)}) \cdot X_k^{(t)}}{\sum_{j=1}^{N} f(X_j^{(t)}, X_k^{(t)})}.$$

$f(X_j^{(t)}, X_k^{(t)}) < f(X_{j'}^{(t)}, X_k^{(t)})$ for $k \neq j, j'$. Since $f(X_{j'}^{(t)}, X_j^{(t)}) < 1$, the effect from itself is larger than that from the other subject. This means that $X_{j'}^{(t+1)}$ is closer to $X_{j'}^{(t)}$ and that $X_j^{(t+1)}$ is closer to $X_j^{(t)}$ if ignoring the effects from other subjects. Combining the fact that the effects from other subjects to pull $X_{j'}^{(t+1)}$ towards the convex hull are larger, $X_{j'}^{(t+1)}$ can not replace $X_j^{(t+1)}$ as a new vertex. This contradicts to the assumption. Therefore, $v_{1,i}^{(t)} = X_j^{(t)}$ for some j and for all t large enough. Then

$$\lim_{t\to\infty} X_j^{(t)} = \lim_{t\to\infty} v_{1,i}^{(t)} = v_{1,i}.$$

□

In Lemma 2, we have proven that some subjects converge under our algorithm. Next, we will prove that other subjects also converge in similiar arguments.

Now we consider the convex hulls of subects except those that already converge to the vertices of C_1. Let Ω_1 be the set of subjects converging to the vertices of C_1. Define $C_2^{(t)}$ be the convex hull of $\{X_i^{(t)}\}_{i\notin\Omega_1}$. Now we do not have $C_2^{(t)} \supseteq C_2^{(t+1)}$ like the result in the Lemma 1. Some subjects may move outside the currect convex hull due to the effect from subjects in Ω_1. Therefore, the volume of the convex hull may indeed increase. However, since all subjects in Ω_1 converge eventually, their effects to subjects not in Ω_1 go down to zero. Again, let C_2 be the limit of $C_2^{(t)}$:

$$C_2 \equiv \lim_{t\to\infty} C_2^{(t)}.$$

Apply the same arguments in Lemma 2, we have at least one subject converge to each vertex of C_2. Then we can run similar steps again for C_3, C_4, $\cdots$ untill all subjects converge. This proves the following theorem:

Theorem 1. *The clustering algorithm proposed in Section 2 converges, if the f in Equation (1) satisfies:*

1. *f(u,v) depends only on $\|u - v\|$, the distance from u to v.*
2. *$0 \le f(u, v) \le 1$, and f(u,v)=1 only when u=v.*
3. *f(u,v) is decreasing with respect to $\|u - v\|$.*

5. Application

We chose one test image from "The Berkeley Segmentation Dtaset". The image is displayed on Figure-3.

Figure 3: test image

Each pixel of the image is viewed as a subject, and the image will be segmented according to how pixels are clustered. For each pixel, we have information on its position and color intensity, which are important statistics for clustering. We use the YUV (Y stands for the luma component and U and V are the chrominance components) information instead of RGB for the intensity. Along with the x and y coordinates, each pixel has five variables. Since the variation on x and y are larger, we scale down both by a factor of 3.

Now we apply our proposed algorithm for clustering. Recall that the function we proposed to measure proximities are

$$f = \begin{cases} \exp[-\frac{d}{\lambda}] & d \le r \\ 0 & d > r. \end{cases}$$

In this application, we chose $\lambda = 15$ and $r = 5$. The distance d here is the sum of the difference in each dimension, instead of Euclidean distance. The result is presented in Figure-4, which shows that we can have a nice segmentation result directly from our algorithm.

6. Conlusion

We proposed a new clustering algorithm. It is very simple and intuitive, while effective. By this algorithm, subjects move gradually toward to ones similar to themselves iteratively. It works straight forward and fluently without

Figure 4: test image

the risk of trapping into local minima, which the success of most clustering algorithms like K-Means largely depends on.

Unlike algorithms like K-Means, ours do not determine a fixed number of clusters first. According to how different subjects in the same cluster are allowed, the algorithm determines the number of clusters through the processes. We do think that this is a more natural way to cluster data.

We apply our algorithm directly on image segmentation and obtain good results. Better results are expected by combining this algorithm and other segmentation techniques.

REFERENCES

Chen, C. H. (2002), *Generalized Association Plots: Information Visualization via Iteratively Generated Correlation Matrices.* Statistica Sinica **12**, 7-29.

McQueen, J. (1967), *Some methods for classification and analysis of multivariate observations.* In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 291-297

McQuitty, L. L. (1968), *Multiple clusters, types, and dimensions from iterative intercolumnar correlational analysis.* Multivariate Behavioral Research **3**, 465-477.

Milligan, G. W. and Cooper, M. .C. (1985), *An examination of procedures for determining the number of clusters in a data set.* Psychometrika, **50**, 159-179.

Selim SZ, Alsultan K. (1991), *A simulated annealing algorithm for the clustering problem.* Pattern Recognition. **24(10)**: 1003-1008.

Tibshirani, R., Walther, G., and Hastie, T. (2001), *Estimating the number of clusters in a data set via the gap statistic.* Journal of the Royal Statistical Society: Series B. **63(2)**, 411-423.

Tseng, G.C. and Wong, W.H. (2005), *Tight clustering: a resampling-based approach for identifying stable and tight patterns in data.* Biometrics, **61**, 10-16.