IEICE TRANS. INF. & SYST., VOL.E93–D, NO.2 FEBRUARY 2010

LETTER

An Efficient Clustering Algorithm for Irregularly Shaped Clusters

DongMing TANG[†a)], **QingXin ZHU**[†], ***Nonmembers***, **Yong CAO**[†], ***Member***, ***and*** **Fan YANG**[†], ***Nonmember***

SUMMARY To detect the natural clusters for irregularly shaped data distribution is a difficult task in pattern recognition. In this study, we propose an efficient clustering algorithm for irregularly shaped clusters based on the advantages of spectral clustering and Affinity Propagation (AP) algorithm. We give a new similarity measure based on neighborhood dispersion analysis. The proposed algorithm is a simple but effective method. The experimental results on several data sets show that the algorithm can detect the natural clusters of input data sets, and the clustering results agree well with that of human judgment.

key words: *cluster analysis, clustering, pattern recognition*

1. Introduction

The basic concept of clustering is to divide patterns into different groups (clusters). The patterns in the same share more similarity comparing with the patterns in other clusters. Recently the spectral clustering approaches are getting more and more attention due to the works of Shi et al. [1], and Ng et al. [2]. Spectral clustering can provide good performance, and can be implemented easily. A comprehensive review can be found in the papers [3], [4]. Unlike classical partitioning clustering algorithms, spectral clustering produces better clustering results on the data sets with highly nonlinear and elongated clusters [3], [5], such as circle or stick distribution. There are some well-known spectral clustering algorithms, such as Shi and Malik algorithm [1], Ng, Jordan and Weiss algorithm (NJW) [2], and others [5]–[8].

While the spectral clustering algorithms having demonstrated good performance on many different data sets, there are still some problems to solve: (1) how to determine the right number k of clusters; (2) how to construct a function h to transform a given dataset into a graph when data points are distributed among different shaped clusters [4]. Besides, when clustering a dataset, the correct number of clusters is often unknown and hard to determine. In NJW [2] we need to set the number of clusters manually. Lihi et al. [6] proposed an alternative approach to automatically infer the number of clusters, which relies on the structure of the eigenvectors. There are several widely used functions h measuring the similarity of point pairs. NJW [2] uses a Gaussian function $h(x_i, x_j) = \exp(-\|x_i - x_j\|/2\sigma^2)$, where the norm $\|x_i - x_j\|$ measures the distance between two patterns and σ controls the rapidity of decay of h. Lihi et al. [6] suggested an improved affinity between a pair of points, that is: $h(x_i, x_j) = \exp(-\|x_i - x_j\|/\sigma_i\sigma_j)$, where $\sigma_i = d(x_i, x_K)$, x_K is the K-th neighbor of point x_i. Clearly, except for specific situations when we have complete knowledge about the data set to ensure the validity of chosen parameters, the choice of the parameters σ and K can only be determined by empirical methods.

Affinity Propagation algorithm (AP) was proposed by Frey and Dueck [9]. AP takes a collection s of similarities between data points as input, here s can be viewed as an $n \times n$ matrix, in which the similarity $s(i, k)$ indicates how well the data point k is suited to be the exemplar for data point i. In AP, the centers of clusters are selected from actual data points, they are called "exemplars". Initially, each data point is treated as a potential exemplar, and then the exemplars are selected by the message-passing procedure. There are two kinds of messages, and each corresponds to a different kind of competition. The "responsibility" $r(i, k)$, sent from data point i to candidate exemplar point k, reflects the accumulated evidence for how well-suited point k is to serve as the exemplar for point i, taking into account other potential exemplars for point i. The availability $a(i, k)$, sent from the candidate representative example point k to point i, reflects the accumulated evidence from data points as to whether each candidate exemplar would make a good exemplar.

The main advantage of AP algorithm is that it considers simultaneously all the data points as possible exemplars and partitions the points into clusters gradually. Therefore, AP does not require the number of clusters pre-specified. Another advantage of AP algorithm is that it does not require that similarities of data points are symmetric and satisfy the triangle inequality. This advantage makes it applicable to unusual measures of similarity.

AP can be viewed as a method that searches for minima of an energy function $E(c) = -\sum_{i=1}^{N} s(i, c_i)$ that depends on a set of N hidden labels, $c_1, \ldots, c_N$, corresponding to the N data points. Each label indicates the exemplar to which the point belongs, so that $s(i, c_i)$ is the similarity of data point i to its exemplar. $c_i = i$ is a special case indicating that point i is itself an exemplar, so that $s(i, c_i)$ is the input preference for point i. Essentially, the minimization of this energy function is similar to that of the objective function for k-means algorithm. Just as with the k-means algorithm, AP can't work

Manuscript received July 3, 2009.
Manuscript revised September 29, 2009.

[†]The authors are with School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China.
a) E-mail: tdm_yc@uestc.edu.cn

DOI: 10.1587/transinf.E93.D.384

Fig. 1 Clustering result of AP on irregularly shaped clusters.

well on irregularly shaped clusters, such as circle or stick distribution. Single exemplar difficulty represents this kind of distribution. AP may force to divide a single cluster into separate ones. In addition, AP forces each exemplar to point to itself. Figure 1 illustrates the obtained clustering results for an example. In this example, we use Euclidean distance to measure the similarities between data points. Clearly, this clustering result is not acceptable according to human judgment. AP may be improved for this kind data points by setting different preferences for each data point. But in practice, it is hard to tune manually each preference without prior knowledge about data distribution. From a theoretical side, AP uses exemplar to represent cluster and forces each exemplar to point to itself, in some cases, there exist several shortcomings. However, it is hard to have a further theoretical analysis to clarify the relationship between clustering quality, exemplar, and uniform effect, since this relationship is affected by many factors, such as cluster shapes and the density in the data.

In practice, we find that the combined use of spectral clustering and AP algorithm can overcome their shortcomings. In this study, we propose an efficient clustering algorithm for irregularly shaped clusters based on the combination of spectral clustering and AP. We also present a new similarity measure based on neighborhood dispersion analysis.

2. The Algorithm

First we introduce a new similarity measure. Let $X = \{x_1, x_2, \ldots, x_N\}$ be a set of data points. All coordinate values of data points are normalized to $[-1, 1]$. Adjacency between two data points is defined as follows:

$$A(x_i, x_j) = \exp(-d(x_i - x_j)\Big/\sigma_i\sigma_j) \tag{1}$$

where $d(x_i, x_j)$ denotes the Euclidean distance between data points.

Lihi et al. [6] showed that Gaussian similarity function using a single scaling parameter σ can't work well when the input data includes clusters with different local statistics. The σ_i and σ_j can be regarded as a specific scaling parameter for each point allows self-tuning of the point-to-point distances according to the local statistics of the neighborhoods surrounding points i and j [6]. The problem of selection σ_i and σ_j can be treated as a Gaussian kernel optimization problem with multiple parameters. To cluster data set with irregularly shaped clusters, we often wish that the linear separability of the mapped samples is enhanced in the kernel feature space. However, this is not always the case. Therefore, selecting a proper kernel with good group separability plays a significant role in kernel-based clustering algorithms. Along this line, we introduce a new similarity measure. The proposed computational process of σ_i in Eq. (1) is described as follows:

(1) Let $D_i = \{d_1, d_2, \ldots, d_{N-1}\}$ be the distances between the data point i and others, ordered by values.

(2) From $pos = 1$, we compute the CV (coefficient of variation) value of $c_k = \{d_{pos}, \ldots, d_{pos+2\theta-1}\}$ and CV value of $c_{k+1} = \{d_{pos+\theta}, \ldots, d_{pos+\theta+2\theta-1}\}$, if $(\mathrm{CV}(c_{k+1})/\mathrm{CV}(c_k)) > \delta$, then set $\sigma_i = mean(c_k)$. Otherwise, set $pos = pos + \theta$ and repeat until $pos + \theta + 2\theta - 1 > N$. For large scale data set, the interval value θ can be set as the integer part of $2\log(N)$. Instead, we suggest set $\theta = 8$ for small scale data set. Here, the threshold parameter $\delta > 1$ defines the maximum accepted variation between c_k and c_{k+1}. The value is selected by the user to meet the requirements of a particular domain or dataset.

(3) If there isn't a value of pos which meets the condition $(\mathrm{CV}(c_{k+1})/\mathrm{CV}(c_k)) > \delta$ until $pos + \theta + 2\theta - 1 > N$, then set $\sigma_i = mean(D_i)$.

From probability and statistics theory, the coefficient of variation (CV) is a normalized measure of dispersion of a probability distribution. It is defined as the ratio of the standard deviation σ to the mean μ: $\mathrm{CV} = \sigma/\mu$. The CV is a dimensionless number that allows comparison of the variation of populations that have significantly different mean values. In general, the larger the CV value is, the greater the variability is in the data. In this study, we use CV to measure the distribution variation of the neighborhood of data point x_i. The similarity measure proposed by Lihi et al. [6] only considers a fixed value of the k-nearest neighbor of data point, it doesn't fully reflect the distribution variation of the neighborhood of data point. In contrast, the proposed similarity measure is based on neighborhood dispersion analysis. Irregularly shaped clusters should be detected by the connectivity between points' neighborhoods and the density of cluster. Two points are in the same cluster, even though they are far apart, if there is a path of locally similar points that connects them. The CV value of $c_k = \{d_{pos}, \ldots, d_{pos+2\theta-1}\}$ represents the variance of the distances between point to it's neighborhoods. A distance between two points is considered inconsistent if its value compared to the average and standard deviation of its neighboring edges is larger than a specified factor [10]. Therefore, the terminal condition $(\mathrm{CV}(c_{k+1})/\mathrm{CV}(c_k)) > \delta$ of the proposed computational process of σ_i is consistent with this analysis.

The procedure of our clustering algorithm mainly follows that of the clustering algorithms suggested in [2], [6]. The proposed algorithm is composed of the following steps:

(1) Compute the affinity matrix $A \in \mathbb{R}^{n\times k}$ using Eq. (1).

(2) Construct a symmetric normalized matrix $L = D^{-1/2}AD^{-1/2}$ from the affinity matrix A, where D is a diagonal matrix with $D_{i,i} = -\sum_{j=1}^{N} A(i, j)$.

(3) Find the k largest eigenvectors $\{e_1, \ldots, e_k\}$ of L, the k is selected by the eigengap heuristically, form the matrix

$U = \{e_1, \ldots, e_k\} \in \mathbb{R}^{n \times k}$.

(4) Form the matrix $Y \in \mathbb{R}^{n \times k}$ from U by normalizing the rows to norm 1.

(5) Treating the rows $y_i \in \mathbb{R}^k$ $(i = 1, \ldots, n)$ of Y as points in $\mathbb{R}^{n \times k}$, cluster the points y_i $(i = 1, \ldots, n)$ with the AP algorithm into clusters.

At the last step, we use AP algorithm to perform the final partition from the matrix Y. In fact, as we have seen from the various explanations of spectral clustering, this step should be very simple if the data contain well-expressed clusters [4]. Ideally, the eigenvectors of L are piecewise constant when the clusters are fully divided. In this case, all data points in the same cluster C_j are mapped to exactly the sample point y_i, namely to the unit vector $e_s \in \mathbb{R}^k$ [4]. Therefore, we can use AP algorithm to obtain the final clustering result with the Euclidean distance between the points y_i. The idea behind the proposed algorithm is to combine spectral algorithm and AP algorithm. AP algorithm helps to overcome the shortcomings of spectral algorithm, and then the proposed algorithm can automatically determine the number of clusters. From this point of view, AP algorithm can be regarded as a post-processing of spectral algorithm. On the other hand, spectral algorithm can be regarded as a prior-processing of AP algorithm. After prior-processing for the data points, AP algorithm can work well on irregular shape and non-uniform density clusters.

The main tools for spectral clustering are graph Laplacian matrices. In this study, we use normalized graph Laplacians: $L = D^{-1/2}AD^{-1/2}$. Given a similarity adjacency matrix A, the simplest and most direct way to construct a partition of the data points is to solve the mincut problem. The proposed algorithm's objective function is normalized cut Ncut [3], [4]. The definition of Ncut is: $Ncut(Q_1, \ldots, Q_k) = \sum_{i=1}^{k} \frac{cut(Q_i, \bar{Q}_i)}{vol(Q_i)}$, where $Q_1, \ldots, Q_k$ is a partition of data set, $\bar{Q}_i$ is the complement of Q_i, $vol(Q_i)$ is the size of subset Q_i, is measured by the weights of its edges. The most important thing that differentiates clustering algorithms is the objective used for clustering. There is a close relation between spectral and kernel. This relation has been discussed in paper [3]. The results of discussion show that there is a direct equivalence between kernel and spectral clustering algorithms. Essentially, they have similar objective function. The example presented in paper [3] has illustrated the equivalence between kernel and spectral clustering algorithms from a unified view of the two approaches. Therefore, the proposed algorithm can be considered as a kernel affinity propagation algorithm or spectral affinity propagation algorithm.

3. Results and Discussion

First, we tested our algorithm using the following two-dimensional data sets: DS1-DS6 with irregularly shaped clusters inside clusters. We applied our algorithm, as well as AP [9], NJW [2] on all data sets. In order to obtain the best possible performance of NJW algorithm, we varied the parameters to obtain the most possible performance, and then recorded the best result. For NJW, the numbers of clusters were set as the right numbers. For AP, we employed the squared Euclidean distance.

Figure 2 shows the clustering results. For each data set, the clustering results of NJW, AP and our algorithm are represented in the left, middle and right respectively. The data set DS1 has three irregularly shaped clusters. This data set is generated from a black-white image with hand painting. As shown in Fig. 2 (a), we can see the NJW and our algorithm detect correctly the structure of clusters. The data sets DS2-DS3 are also generated from a black-white image with hand painting. Figure 2 (b) shows the results on the data set DS2. We can see that there are three clusters, two of them are roughly ring shape clusters, and one of them is the intersection of two stripes. It can be observed from the results that the NJW and our algorithm detect natural clusters. As shown in Fig. 2 (c), the NJW can't detect natural clusters for the data set DS3. In contrast, our algorithm correctly detects natural clusters for the data set DS3. The data set DS4 is generated by simulator program. It can be

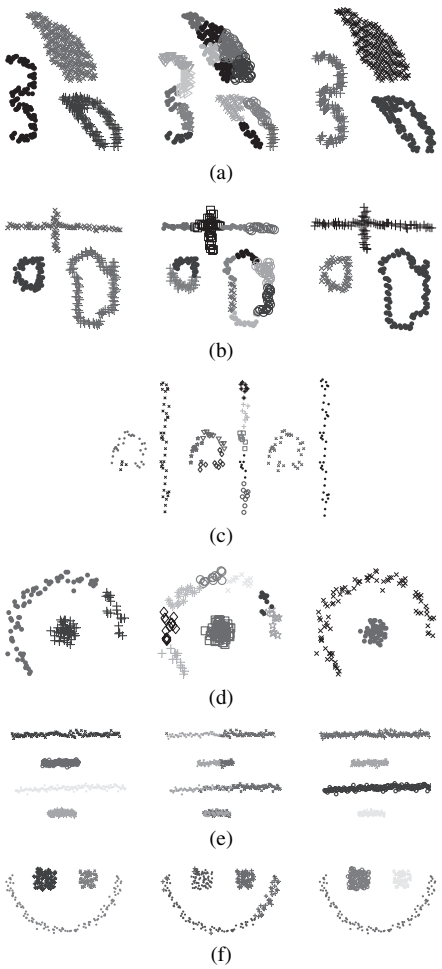

Fig. 2 Clustering results on the data sets DS1-DS6: (a) DS1; (b) DS2; (c) DS3; (d) DS4; (e) DS5; (f) DS6. For each data set, the clustering results of NJW, AP and our algorithm are represented in the left, middle and right respectively.

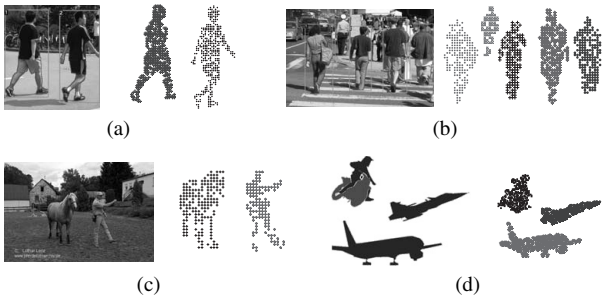

Fig. 3 Clustering results on the data sets DS7-DS10: (a) DS7; (b) DS8; (c) DS9; (d) DS10. For each data set, the original image and the clustering result are represented in the left and right respectively.

seen that our method gives the best clustering result for this data set. Indeed, we see that a small part of the half-ring is assigned to another cluster by the NJW. The data sets DS5-DS6 were proposed in the work [6]. The clustering results using different methods on the data sets DS5-DS6 are shown in Fig. 2 (e)–(f). The data set DS5 has four stripes with different lengths. For data sets DS5-DS6, the NJW and our algorithm correctly detect natural clusters. As mentioned earlier, AP can't work well on data set of this type. Here, the experimental results demonstrate the correctness of this inference.

Next, we evaluated the performance of the proposed algorithm on real world data sets. We applied it to four data sets DS7-DS10. The DS7-DS8 were proposed in the work [11]. The DS9-DS10 were proposed in the work [12]. Figure 3 shows the clustering results of the proposed algorithm on four real world data sets. Note that, in all these data sets, the data points were generated from the major components of scene of the image. For each data set, the original image and the clustering result are represented in the left and right respectively. In the data set DS7, there are two walking men. The data set DS8 has five persons. The data set DS9 has a horse and a person. The data set DS10 has a motorbike, an airliner and a fighter. As is seen from Fig. 3, the proposed algorithm can cluster these data sets in the expected way.

4. Conclusion

In this study, we introduce an efficient clustering algorithm for irregularly shaped clusters based on the combination of spectral clustering and AP. The experimental results illustrate the efficiency of the proposed algorithm. We also compared it with Ng, Jordan and Weiss algorithm and AP algorithm. The clustering results agree well with human judgment. In the further work, we will use it to cluster high dimensional data. We are also going to investigate the methods for self-tuning the threshold value.

Acknowledgments

This research was supported by the National Natural Science Foundation of China under Grant No.60671033.

References

[1] J. Shi and J. Malik, "Normalized cuts and image segmentation," IEEE Trans. Pattern Anal., vol.22, no.8, pp.888–905, 2000.
[2] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," NIPS 14, pp.849–856, 2002.
[3] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering," Pattern Recognit., vol.41, no.1, pp.176–190, 2008.
[4] U. Luxburg, "A tutorial on spectral clustering," Stat Comput, vol.17, no.4, pp.395–416, 2007.
[5] H. Chang and D.-Y. Yeung, "Robust path-based spectral clustering," Pattern Recognit., vol.41, no.1, pp.191–203, 2008.
[6] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," NIPS 16, pp.1601–1608, 2004.
[7] U. Ozertem, D. Erdogmus, and R. Jenssen, "Mean shift spectral clustering," Pattern Recognit., vol.41, no.6, pp.1924–1938, 2008.
[8] M. Wu and B. Schölkopf, "A local learning approach for clustering," NIPS 19, pp.1529–1536, 2007.
[9] B.J. Frey and D. Dueck, "Clustering by passing messages between data points," Science, vol.315, no.5814, pp.972–976, 2007.
[10] C.T. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters," IEEE Trans. Comput., vol.C-20, no.1, pp.68–86, 1971.
[11] L. Wang, J. Shi, G. Song, and I.F. Shen, "Object detection combining recognition and segmentation," Proc. ACCV 2007, pp.189–199, 2007.
[12] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge 2007," http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html