Pattern Recognition Letters 32 (2011) 352–358

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Local density adaptive similarity measurement for spectral clustering ☆

Xianchao Zhang *, Jingwei Li, Hong Yu

School of Software, Dalian University of Technology, Dalian 116620, China

ARTICLE INFO

Article history:
Received 6 February 2010
Available online 1 October 2010
Communicated by M.A. Girolami

Keywords:
Clustering
Spectral clustering
Similarity measure

ABSTRACT

Similarity measurement is crucial to the performance of spectral clustering. The Gaussian kernel function is usually adopted as the similarity measure. However, with a fixed kernel parameter, the similarity between two data points is only determined by their Euclidean distance, and is not adaptive to their surroundings. In this paper, a local density adaptive similarity measure is proposed, which uses the local density between two data points to scale the Gaussian kernel function. The proposed similarity measure satisfies the clustering assumption and has an effect of amplifying intra-cluster similarity, thus making the affinity matrix clearly block diagonal. Experimental results on both synthetic and real world data sets show that the spectral clustering algorithm with our local density adaptive similarity measure outperforms the traditional spectral clustering algorithm, the path-based spectral clustering algorithm and the self-tuning spectral clustering algorithm.

1. Introduction

In recent years, spectral clustering has become one of the most popular clustering algorithms (Luxburg, 2007). It gains superior performances compared to the traditional clustering methods such as *k*-means (MacQueen, 1967) and can be applied in particular in cases where traditional methods fail, e.g., non-convex data sets (Ng et al., 2001).

Spectral clustering consists of two stages: (1) build an affinity matrix (graph) with a kind of similarity measure; (2) find a good clustering of the graph. Significant progress has been made to address the latter (Hagen and Kahng, 1992; Ding et al., 2001; Ng et al., 2001; Shi and Malik, 2000), while little work have been done to study the former (Zelnik-Manor et al., 2004; Fischer and Buhmann, 2003).

The first task of building a good affinity matrix has been shown to be largely responsible for the performance of spectral clustering algorithms (Bach and Jordan, 2003). Most spectral clustering methods (Ng et al., 2001; Shi and Malik, 2000) adopt the Gaussian kernel function as the similarity since it is simple to calculate and it results in a positive definite similarity matrix which simplifies the analysis of eigenvalues (Shi and Malik, 2000). However, it has difficulties at handling complex circumstances, e.g., a multi-scale data set (Zelnik-Manor et al., 2004). Moreover, it has a scaling parameter σ to be specified manually. Generally it is non-trivial to find an optimal value for the parameter σ (Zelnik-Manor et al., 2004). Manor et al. proposed to use a local scaling parameter for each data point rather than a globally fixed one (Zelnik-Manor et al., 2004). However, this local distance aware similarity is still of little help to reveal the properties of real clusters and fails on many real world data sets (Xia et al., 2008). The similarity used in Path-based clustering (Fischer and Buhmann, 2003) stresses connectedness of data points via mediating elements rather than mutual similarity. While is seems to be effective on some clustering tasks, we will show in Section 5 that its performances are not very well on some real data sets.

Similar to the prior assumption of consistency that is often used in semi-supervised learning (Zhou et al., 2004), a good similarity measure for clustering should hold the following two kinds of assumptions of consistency: (1) local consistency: nearby points in the space should have high similarity; (2) global consistency: points in the same cluster should have high similarity. Based on these assumptions, it is desirable that, with the same Euclidean distance, two points in the same cluster should have higher similarity than two points in different clusters, this is the clustering assumption. Neither the traditional Gaussian kernel function nor the local scaled similarity in (Zelnik-Manor et al., 2004) satisfies the clustering assumption. With these similarity measures, points of small Euclidean distances in different clusters often confuse the graph clustering algorithm in the second stage (e.g., points *a* and *c* in Fig. 3 in Section 3), and lead to poor clustering results.

Note that two data points fall into the same cluster not only because they are near in the space, in most cases because they are in a same high density area, i.e., there are many data points between them which "conglutinate" them in the same cluster. With this observation, we propose a measure called Common-Near-Neighbor

☆ Supported by National Science Foundation of China (NSFC) under Grant No. 60873180.

* Corresponding author. Tel.: +86 411 87571515; fax: +86 411 87571567.
E-mail addresses: xczhang@dlut.edu.cn (X. Zhang), willa274180@yahoo.com.cn (J. Li), hongyu@dlut.edu.cn (H. Yu).

0167-8655/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2010.09.014

(CNN) that reflects the local density between two data points. The CNN measure is used to scale σ to realize the conglutinate effect. With this scale factor, the new similarity is adaptive to local density and satisfies the clustering assumption. It has an effect of amplifying intra-cluster similarity, making the affinity matrix clearly block diagonal. Experimental results both on synthetic data sets and real world data sets showed that spectral clustering with our local density adaptive similarity (SC-DA) outperforms the traditional spectral clustering (SC), the path-based spectral clustering algorithm (SC-PB) and the self-tuning spectral clustering with local scale parameter (SC-ST). Moreover, our algorithm is less sensitive to parameter σ, making it easier to apply.

The remaining of this paper is organized as follows. We begin with a brief overview of spectral clustering in Section 2. Then we analyze the shortcomings of existing similarity measures in Section 3 and propose our local density adaptive similarity measure in Section 4. Experimental results are then presented in Section 5. Finally we conclude the discussions and point out further works in Section 6.

2. Overview of spectral clustering

Given a set of n data points $X = \{x_1, x_2, \ldots, x_n\}$, the objective of clustering is to divide data points into different clusters, where data points in the same cluster are similar to each other. According to a specific similarity measure, we have the affinity matrix $S \in \Re^{n\times n}$. From S, we can construct an undirected graph $G = (V, E)$ with each vertex $v_i \in V$ corresponding to the data point x_i and each edge $e(i,j) \in E$ carries a weight S_{ij} which represents the similarity between point x_i and x_j. The clustering problem is equivalent to choosing a partition $C_1, C_2, \ldots, C_k$ of G which minimizes a specific objective function such as *RatioCut* (Hagen and Kahng, 1992), *MinmaxCut* (Ding et al., 2001), and the *Ncut* (Shi and Malik, 2000). The performances of these three objective functions are input dependent: if clusters are well separated, all the three give very similar and accurate results; when clusters are marginally separated, *NCut* and *MinMaxCut* give better results; when clusters overlap significantly, *MinMaxCut* tend to give more compact and balanced clusters (Ding, 2004). In this paper, we use the mostly often adopted *Ncut*. It was shown in (Wagner and Wagner, 1993) that the minimization of *Ncut* is NP-hard. According to Rayleigh–Ritz theory (Lütkepohl, 1997), it is possible to find an approximate solution. In solving this, we need to define the normalized Laplacian matrix $L = D^{-\frac{1}{2}}SD^{-\frac{1}{2}}$ where D is a diagonal matrix with $D_{ii} = \sum_{j=1}^{n} S_{ij}$. Then, the approximate solution could be derived from the leading eigenvectors of L. The use of Laplacian matrix eigenvector for approximating the graph minimum cut is called spectral clustering. A complete overview of spectral clustering can be found in (Luxburg, 2007).

3. Analysis of existing similarity measures

The most commonly used similarity measure, the Gaussian kernel function, is defined as $S_G(x_i, x_j) = exp(-d(x_i, x_j)^2/2\sigma^2)$, where $d(x_i, x_j)$ is the Euclidean distance between data points x_i and x_j, and σ is the kernel parameter (Shi and Malik, 2000). The obvious drawback of S_G is that the scaling parameter σ is fixed, thus the similarity between two points is only determined by their Euclidean distance, and does not vary with the change of the surroundings. An example is given in Fig. 1 (Fig. 2 in (Zelnik-Manor et al., 2004)). In Fig. 1(a), supposing $d(a,b) = d(a,c)$, then with the Gaussian kernel function we have $S_G(a,b) = S_G(a,c)$. Thus the clustering algorithm tends to cluster a, b, c together. However, the fact is that a, c are in the background cluster which is relatively sparse, while b is in the tight cluster in the center.

Zelnik-Manor et al. proposed a local scale similarity measure $S_T(x_i, x_j) = exp(-d(x_i, x_j)^2/\sigma_i\sigma_j)$, where σ_i is the distance between point x_i and its kth nearest neighbor (Zelnik-Manor et al., 2004). With S_T, in Fig. 1, we have $\sigma_c > \sigma_b$, so $\sigma_a\sigma_c > \sigma_a\sigma_b$, point a gets closer to point c than to point b. This is just the information required for separation. The effect of local scaling can be seen from the comparison of Fig. 1(b) with Fig. 1(c).

The corresponding spectral clustering based on this local scale similarity measure is called self-tuning spectral clustering (SC-ST) (Zelnik-Manor et al., 2004). Since the adaptive local scale parameter reflects the local information properly, SC-ST works well on the data with multiple scales, e.g., Fig. 1(c). It reveals that the surroundings of two data points have high impact on their similarity.

However, the local scale parameter in SC-ST, the distance to a nearby neighbor, is still a Euclidean distance factor and does not help in many cases. For example, on the toy data set in Fig. 2, consider three data points a, b, c with $d(a,b) = d(a,c)$ in the Euclidean space (see Fig. 3). Following the clustering assumption, the similar-

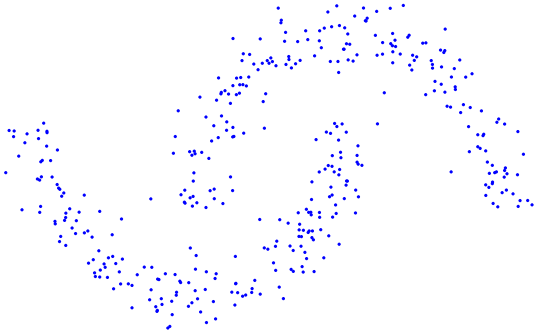

Fig. 2. The two moon dataset.

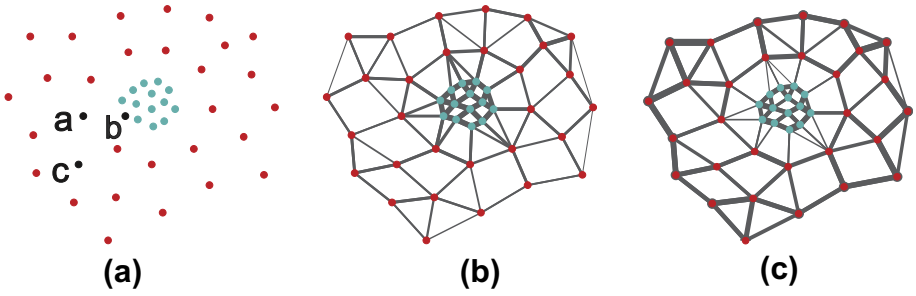

Fig. 1. The effect of local scaling. (a) The data set with multiple scales. (b) The Gaussian function affinity is indicated by the thickness of the line connecting them. With Gaussian function the affinities across clusters are larger than the affinities within the background cluster. (c) The corresponding visualization of affinities after local scaling. The affinities across clusters are significantly lower than the affinities within any single cluster.

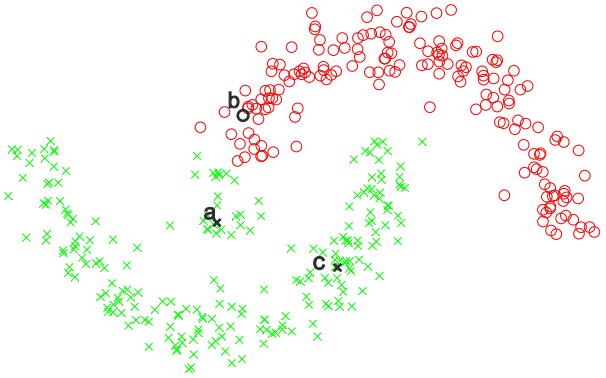

Fig. 3. Self-tuning clustering result.

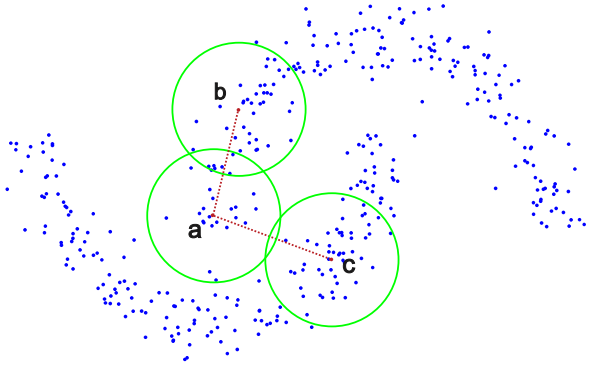

Fig. 4. Shared neighbors on the toy dataset.

ity between a and b should be higher than the similarity between a and c, since point a is in the same cluster with point b rather than point c. Unfortunately, SC-ST does not work, since the local statistics surrounding point b and c are similar, leading to $S_T(a,b) \approx S_T(a,c)$. It can not make any contribution to clustering better than using S_G, thus SC-ST fails to produce the correct clustering result (see Fig. 3).

The path-based similarity used in path based clustering (Fischer and Buhmann, 2003) is defined as follows: $S_P(x_i,x_j) = \max_{p\in\wp_{ij}}\{\min_{1\leqslant h<|p|} d(x_{p[h]},x_{p[h+1]})\}$ where $\wp_{ij}$ denotes the set of all paths from x_i to x_j, $p[h]$ denotes the hth point along the path p from x_i to x_j. This similarity reflects the idea that no matter how far the physical distance between two points, they should be considered as in one cluster if they are connected by a set of successive points in dense regions. This is intuitively reasonable. However, it is not robust enough against noise and outliers (Chang and Yeung, 2008).

4. Local density adaptive similarity measure

In Section 3 we have learned that a good similarity measure should be adaptive to the neighborhoods of the correlative points. Our proposed method is based on the following observation: if two points are distributed in the same cluster, they are in the same region which has a relatively high density. That is, two points fall into the same cluster because there are many points between them that "conglutinate" them together. To reflect the "conglutinate" effect between two data points, we define the *Common-Near-Neighbor*(*CNN*) measure.

Definition 1. $CNN(a,b)$ is the number of the points in the join region of the ε-neighborhoods around points a and b, where the ε-neighborhood of one point represents the sphere region around that point of specified radius ε.

$CNN(a,b)$ shows the local density between points a and b, which can be used to distinguish points within the same cluster from points among different clusters. For example, as shown in Fig. 4, points a and b have more shared neighbors than points a and c, so we will have $CNN(a,b) > CNN(a,c)$, which contains strong information for data partition.

The local density adaptive similarity measure between a pair of points is thus defined as:

$$S_L(x_i,x_j) = \begin{cases} exp\left(-\frac{d(x_i,x_j)^2}{2\sigma^2(CNN(x_i,x_j)+1)}\right) & i \neq j, \\ 0 & i = j. \end{cases}$$

The proposed similarity measure has the following properties:

(1) If $d(x_i,x_j) \geqslant 2\varepsilon$, then $S_L(x_i,x_j) = S_G(x_i,x_j)$, showing that the scale factor is a local one and does not affect far away data points.

(2) For two pairs of points x_i, x_j and x_m, x_n, supposing $d(x_i,x_j) = d(x_m,x_n) < 2\varepsilon$, but in fact x_i,x_j are in the same dense region while x_m,x_n are in different dense regions, then with very high probability we have $S_L(x_i,x_j) > S_L(x_m,x_n)$. For example, in Fig. 4, $S_L(a,b) > S_L(a,c)$.

Since clusters are dense regions of the data set, thus from (2) it can be seen that our defined similarity has an effect of amplifying the intra-cluster similarity. This is desirable for high quality clustering.

Till now, to the best of our knowledge, there has been no quantitative metric on to what extent a similarity measure is good. However, it is believed that a good similarity function should make the affinity matrix as block diagonal as possible (and Poland, 2005). This is usually evaluated through visualizations of the affinity matrices. For the two-moon dataset, visualizations of the affinity matrices using the three different similarity functions are given in Fig. 5. Note that with our local density adaptive similarity measure, points in the same cluster obtain the highest similarities. It is the closest to the ideal block diagonal matrix (in which the value is 1 when two points are in the same cluster and 0 otherwise).

The local density adaptive spectral clustering algorithm (SC-DA) is simply the ordinary spectral clustering algorithm using the new similarity measure S_L in place S_G. This seemingly minor modification is in fact very significant as the clustering results can be improved effectively. The clustering result on the two moon dataset

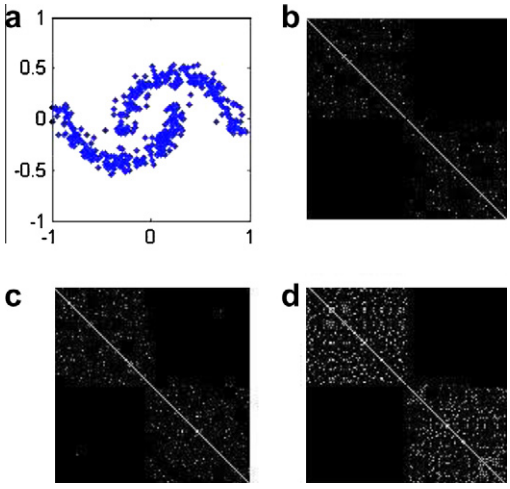

Fig. 5. Affinity matrices on the toy dataset (the brighter the element in the image of the affinity matrix, the higher similarity between the corresponding data points). (a) Scatter plot of the toy dataset, (b) affinity matrix computed with S_G, (c) affinity matrix computed with S_T and (d) affinity matrix computed with S_L.

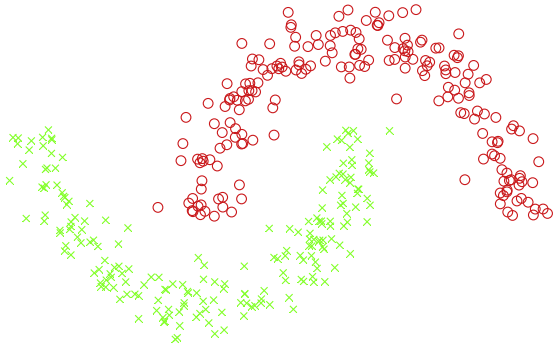

Fig. 6. Clustering results with SC-DA.

with our method is shown in Fig. 6. The groups now match the real solution well (Recall the result of SC-ST in Fig. 3. to see the superiority of our similarity for this dataset).

5. Experiments

5.1. Parameter setting

In our experiments, the value of σ is set by search over values from 10 to 20 percent of the total range of the Euclidean distances and the one that gives the tightest clusters is picked, as suggested in (Ng et al., 2001). k varies from 2 to 50 in SC-ST (including 7 suggested in Zelnik-Manor et al., 2004) and the one that gives the tightest clusters is used. Through empirical analysis with linear regression, ε in SC-DA is set as: $\varepsilon = 20mean\_d + 54min\_d + 13max\_n - 6max\_d - 65mean\_n$, where max_d is the maximum distance, $mean\_d$ is the mean of distances and min_d is minimum distance between all pairs of data points, while max_n is the maximum distance and $mean\_n$ is the mean of distances between each data point and its nearest neighbor.

5.2. Experiments on synthetic data sets

We applied SC, SC-ST, SC-PB and SC-DA to three 2D and two 3D synthetic data sets. The contrast results are shown in Figs. 7 and 8.

See the pictures on the first row of Fig. 7, when the data incorporates multiple scales, SC fails to cluster correctly. For the 2-spiral dataset on the second row of Fig. 7, SC, SC-PB and SC-DA work well, while SC-ST gives a poor result, showing that SC-ST is not good at handling datasets with complex shapes. For the 3-spiral data on the third row of Fig. 7, both SC and SC-ST perform bad, while SC-DA and SC-PB detect the three spiral arms as clusters successfully.

In Fig. 8, on the first row, SC and SC-ST both fail to find the two spirals intertwining together in the 3D space, while SC-PB and SC-DA produce the correct clustering result. On the second row the data set consists of three interlocked rings, with the existence of a bridge, the two rings on the left side are linked into the same cluster while the right one forms another cluster. Both SC and SC-ST get incorrect results for this dataset, while SC-PB and SC-DA cluster the data-set correctly.

The results on these five synthetic data sets show that with the consideration of the "conglutinate" effect of the neighbors, our SC-DA can reliably find the real clusters in multi-scale or complex-shaped data sets, in which SC and SC-ST often fail. SC-PB also performs well on these synthetic data sets.

5.3. Experiments on real world data sets

We further compare the performances of our algorithm with the other three algorithms on some UCI datasets and UPS datasets.

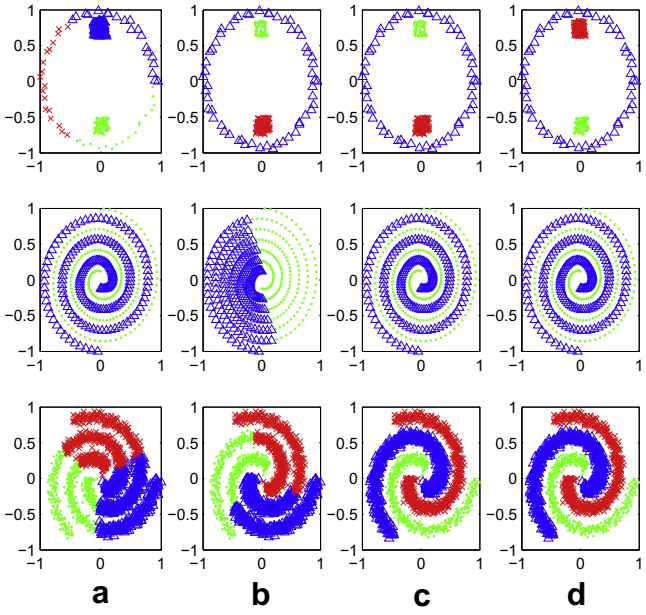

Fig. 7. Clustering results on 2D synthetic data. (a) Results of SC, (b) results of SC-ST, (c) results of SC-PB and (d) results of SC-DA.

Two metrics are used for comparison: the Clustering Error and the Normalized Mutual Information.

5.3.1. Performance measure

Clustering Error (CE) is widely used to evaluate clustering performance (Bach and Jordan, 2003). To compute it for a clustering result, we need to build a permutation mapping function that maps each cluster index to a true class label. The classification error based on this map is then computed as:

$$\text{ClassificationError} = 1 - \frac{\sum_{i=1}^{n}\delta(y_i, map(c_i))}{n},$$

where y_i and c_i are the true class label and the obtained cluster index of x_i respectively, $\delta(x,y)$ is the delta function that equals 1 if $x = y$ and equals 0 otherwise. The Clustering Error is defined as the minimum classification error among all possible permutation mappings. The smaller the Clustering Error, the better the performance.

Normalized Mutual Information (NMI) is a common measure for determining the quality of clusters (Jin et al., 2006). Given a clustering result, the NMI is estimated as (Strehl and Ghosh, 2003):

$$NMI = \frac{\sum_{i=1}^{k}\sum_{h=1}^{c} n_{l,h} log\left(\frac{n\cdot n_{l,h}}{n_l \cdot \hat{n}_h}\right)}{\sqrt{\left(\sum_{i=1}^{k} n_l \log\frac{n_l}{n}\right)\left(\sum_{h=1}^{c}\hat{n}_h \log\frac{\hat{n}_h}{n}\right)}},$$

where n_l denotes the number of data contained in the cluster $C_l (1 \leqslant l \leqslant k)$, $\hat{n}_l$ is the number of data belonging to the hth class $(1 \leqslant h \leqslant c)$ and $n_{l,h}$ denotes the number of data that are in the intersection between C_l and the hth class. The larger the NMI, the better the performance.

5.3.2. Results on UCI data sets

We conduct experiments on four real data sets which come from the UCI data repository (Asuncion and Newman, http://archive.ics.uci.edu/ml/). The data sets are listed in Table 1 together with some of their characteristics.

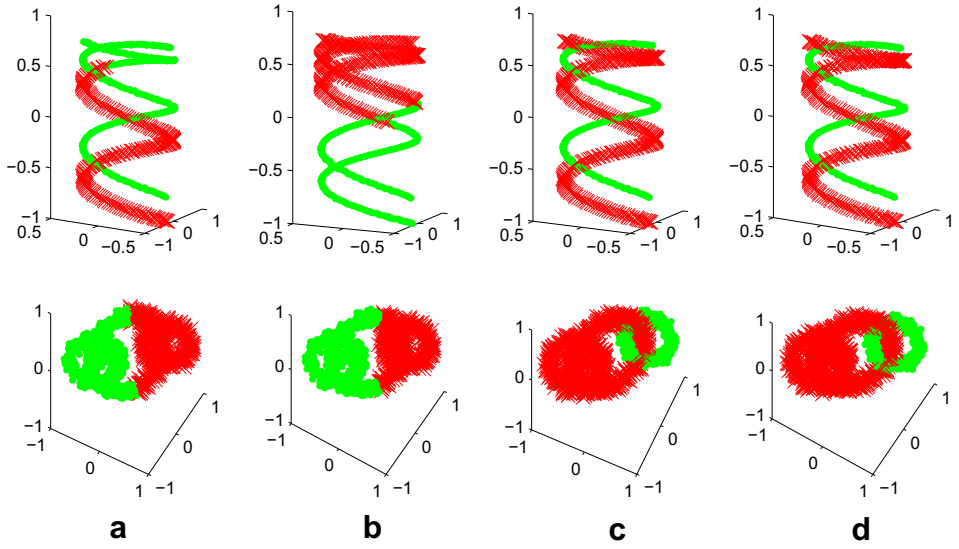

Fig. 8. Clustering results on 3D Synthetic data. (a) Results of SC, (b) results of SC-ST, (c) results of SC-PB and (d) results of SC-DA.

Table 1
Properties of UCI Datasets.

Dataset	Iris	Wine	Ionosphere	Glass	Sonar
No. of instances	150	178	351	214	208
No. of attributes	4	13	34	9	60
No. of clusters	3	3	2	6	2

Experimental results with respect to the CE criteria are presented in Fig. 9. It can be seen that SC-DA outperforms SC, SC-ST and SC-PB on all the UCI data sets in terms of CE.

Experimental results on UCI data sets by NMI are summarized in Fig. 10. With respect to the NMI criteria, the proposed SC-DA outperforms SC and SC-ST on most of the datasets. For example, in the iris dataset, one class is linearly separable from the other two; the others are nonlinearly separable. This is very challenging to SC, SC-ST and SC-PB. But with SC-DA, the two clusters are separated more accurately. On the glass dataset, the performance SC-DA is not as good as SC. One potential reason is that the incorrect

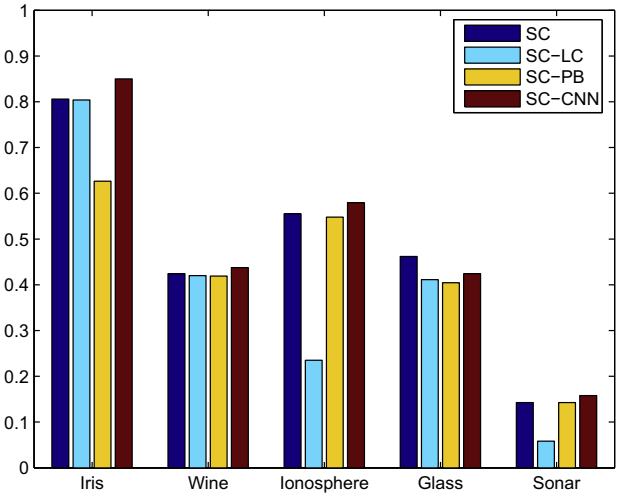

Fig. 10. Results on UCI by NMI.

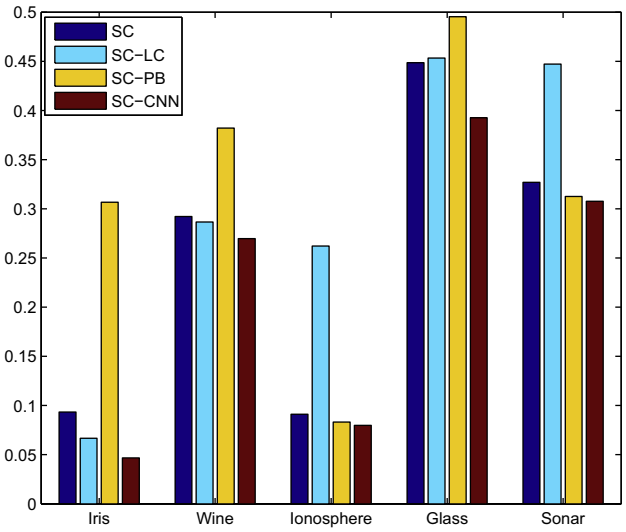

Fig. 9. Results on UCI by CE.

Fig. 11. Samples in the USPS data sets.

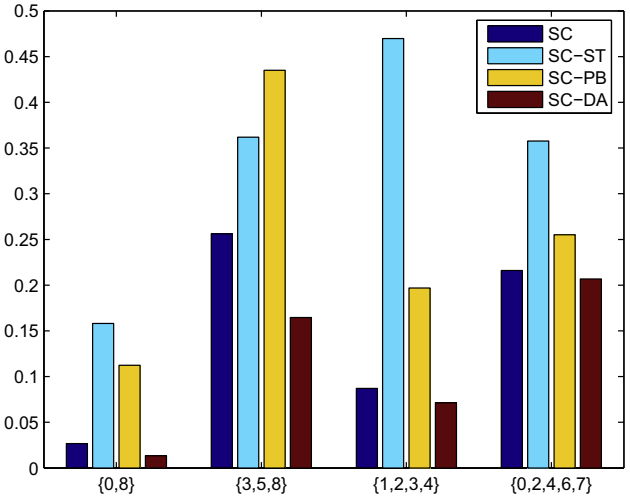

Fig. 12. Results on USPS by CE.

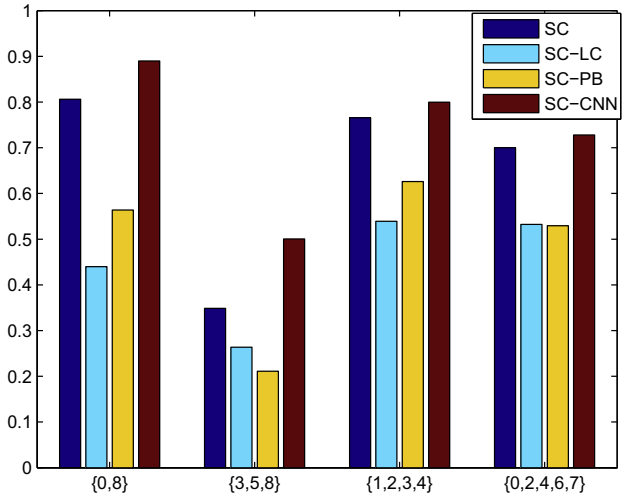

Fig. 13. Results on USPS by NMI.

assignments of only a few points will cause the NMI value declined dramatically, especially when the number of clusters is big.

5.3.3. Results on USPS data sets

We further perform experiments on handwritten digits from the well-known USPS database (LeCun et al., 1989). The digits in the database have been size normalized and centered to 16×16 gray-level images (Fig. 11), the dimensionality of the digit space is 256. It contains 7291 training instances and 2007 test instances. We choose digits $\{0,8\}$, $\{3,5,8\}$, $\{1,2,3,4\}$ and $\{0,2,4,6,7\}$ in the test instances as subsets and conduct comparison experiments separately. The clustering results by CE are summarized in Fig. 12.

From Fig. 12, it can be seen that: (1) the error rates of SC-ST and SC-PB are much higher than the other two methods, showing that they are not very good at handling this dataset. (2) Even on the most challenging USPS subset 3, 5, 8, the absolute error obtained by SC-DA is dramatically reduced by 9.35% compared to that of SC. It proves that the new similarity measure is very helpful in detecting the real structure of the digits.

The clustering results on USPS data sets by NMI are summarized in Fig. 13. From the results, it can be seen that SC-ST and SC-PB are still worse than the other two methods on all USPS data sets with respect to of NMI. By contrast, SC-DA outperforms the other three methods for all USPS data sets in terms of NMI.

5.3.4. Parameter sensitiveness

One crucial step of spectral clustering is to select an appropriate value for parameter σ. If a spectral algorithm can be less sensitive to σ, it will be easier to apply. In order to examine the parameter sensitiveness, we test the performances of SC and SC-DA on the widely used three-circle artificial dataset and the Iris dataset. The results are illustrated in Fig. 14. From the performance curves, it can be seen that SC is very sensitive to the parameter σ on these datasets, while SC-DA is less sensitive to σ.

6. Conclusion

A good similarity measure is crucial to the performance of spectral clustering algorithms. In this paper, we have introduced a local density adaptive similarity measure which exploits the local density information of the neighborhood and satisfies the clustering assumption. This measure has an effect of amplifying the intra-cluster similarity and is more helpful in detecting the intrinsic structure of the data sets. Experimental results on both synthetic data sets and real world data sets show that the proposed algorithm achieves considerable improvements over other typical spectral clustering methods such as the traditional spectral clustering algorithm, the self-tuning spectral clustering algorithm and the path-based spectral clustering algorithm. It is also less parameter

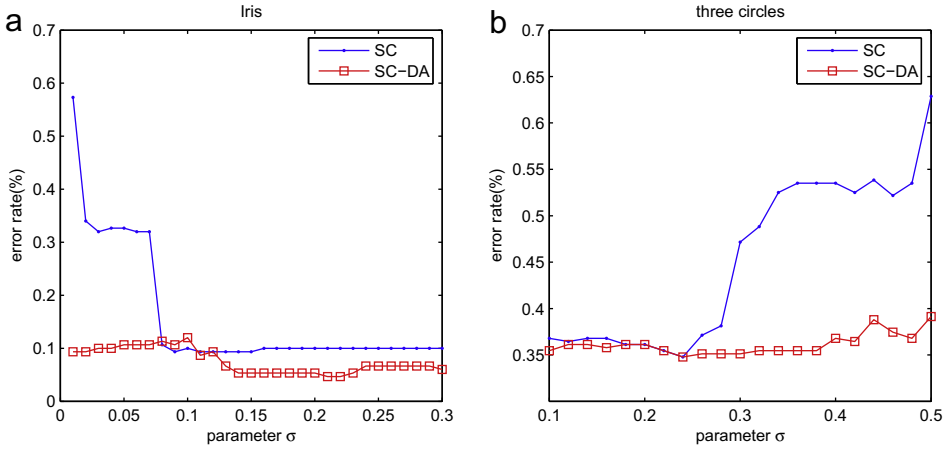

Fig. 14. Experiments about parameter sensitiveness. (a) Performances of SC and SC-DA on Iris (b) Performances of SC and SC-DA on three circles.

sensitive than the traditional spectral clustering algorithm, thus it is easier to apply. These observations show that our proposed similarity is a good candidate for implementing spectral clustering. One future work is to investigate its robustness against noises or outliers. Another interesting future work is to apply the new similarity measure to other affinity matrix based clustering algorithms.

References

Asuncion, A., Newman, D., 2007. UCI Machine Learning Repository. <http://archive.ics.uci.edu/ml/>.

Bach, F., Jordan, M., 2003. Learning spectral clustering. In: Proceedings of NIPS'03, pp. 305–312.

Chang, H., Yeung, D-Y., 2008. Robust path-based spectral clustering. Pattern Recognit. 41 (1), 191–203.

Ding, C., 2004. A tutorial on spectral clustering. Tutorial given at ICML'04, <http://ranger.uta.edu/chqding/Spectral/>.

Ding C., He X., Zha H., et al., 2001. A min–max cut algorithm for graph partitioning and data clustering. In: Proceedings of ICDM'01, pp.107–114.

Fischer, B., Buhmann, J.M., 2003. Path-based clustering for grouping of smooth curves and texture segmentation. IEEE Trans. Pattern Anal. Machine Intell. 25 (4), 513–518.

Fischer, I., Poland, J., 2005. Amplifying the block matrix structure for spectral clustering. In Proceedings of 14th Annual Machine Learning Conference of Belgium and The Netherlands, pp. 21–28.

Hagen, L., Kahng, A., 1992. New spectral methods for ratio cut partitioning and clustering. IEEE Trans. Computer-Aided Design, 11(9), pp. 1074–1085.

Jin, R., Ding, C., Kang, F., 2006. A probabilistic approach for optimizing spectral clustering. In: Proceedings of NIPS'06, pp. 571–578.

LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.J., 1989. Backpropagation applied to handwritten zip code recognition. Neural Comput. 1 (4), 541–551.

Lütkepohl, H., 1997. Handbook of Matrices. Wiley, Chichester.

Luxburg, U., 2007. A tutorial on spectral clustering. Statistics and Computing 17 (4), 395–416.

MacQueen, J.B., 1967. Some Methods for classification and Analysis of Multivariate Observations. In: Proceedings of Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297.

Ng, A., Jordan, M., Weiss, Y., 2001. On spectral clustering: Analysis and an algorithm. In: Proceedings of NIPS'01, pp. 849–856.

Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Machine Intell. 22 (8), 888–905.

Strehl, A., Ghosh, J., 2003. Cluster ensembles-a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. 3 (3), 583–617.

Wagner, D., Wagner, F., 1993. Between min cut and graph bisection. In Proceedings of MFCS'93, pp.744–750.

Xia, T., Cao, J., Zhang, Y., Li, J., 2008. On defining affinity graph for spectral clustering through ranking on manifolds. Neurocomputing 72 (13–15), 3203–3211.

Zelnik-Manor, L., Perona, P.,2004. Self-tuning spectral clustering. In Proceedings of NIPS'04, pp. 1601–1608.

Zhou, D., Bousquet, O., Lal, T., Weston, J., Schölkopf, B., 2004. Learning with local and global consistency. In Proceedings of NIPS'04, pp. 321–328.