Spectral Clustering Using Robust Similarity Measure Based on Closeness of Shared Nearest Neighbors

Xiucai Ye and Tetsuya Sakurai
Department of Computer Science
University of Tsukuba
Tsukuba, Japan

***Abstract*—Spectral clustering has become one of the main clustering methods and has a wide range of applications. Similarity measure is crucial to correct cluster separation for spectral clustering. Many existing spectral clustering algorithms typically measure similarity based on the undirected k-Nearest Neighbor (kNN) graph or Gaussian kernel function, which can not reveal the real clusters of not well-separated data sets. In this paper, we propose a novel algorithm called Spectral Clustering based on Shared Nearest Neighbors (SC-SNN) to improve the clustering quality of not well-separated data sets. Instead of using distance for the similarity measure, the proposed SC-SNN algorithm measures the similarity by considering the closeness of shared nearest neighbors in the *directed* kNN graph, which is able to explore the underlying similarity relationships between data points and is robust to the not well-separated data sets. Moreover, SC-SNN has only one parameter, k, and is less sensitive than the spectral clustering algorithms based on the undirected kNN graph. The proposed SC-SNN algorithm is evaluated by using both synthetic and real-world data sets. The experimental results demonstrate that SC-SNN not only achieves good performance, but also outperforms the traditional spectral clustering algorithms.**

I. Introduction

Clustering is an important technique for exploratory data analysis, and has the objective of grouping unlabeled data objects in the same cluster which are more similar to each other than to objects in other clusters [1]. Recently, spectral clustering has become one of the most popular clustering methods, which has superior performance compared to the traditional clustering methods such as K-means [2], [3]. Spectral clustering is efficient for clustering data with complex structure, e.g., non-convex data sets. Spectral clustering has been applied successfully in a large number of fields, including image segmentation [4], [5], load balancing [6], circuit layout [7], video retrieval [8], and bioinformatics [9].

Spectral clustering makes use of the eigenvectors derived from the *similarity matrix* to reveal the cluster structure of data. Thus, the performance of spectral clustering heavily relies on the effect of similarity measure for similarity matrix construction. Since the complex data are often high dimensions and heterogeneous without prior knowledge of the structure, defining pairwise similarity for effective spectral clustering is fundamentally challenging. The Gaussian kernel function is widely used as a method of similarity measure for spectral clustering, with which the pairwise similarity s_{ij} is calculated as $s_{ij} = \exp(-\|x_i - x_j\|^2/2\sigma^2)$. $\|x_i - x_j\|$ is the Euclidean distance between data points x_i and x_j ($i \neq j$), and σ is the kernel parameter. Gaussian kernel function is simple to calculate and results in a positive definite similarity matrix which simplifies the analysis of eigenvectors. However, the optimal value of kernel parameter σ which reflects the true neighborhood of data points is difficult to find.

Ng et al. [10] devoted to find an optimal value of parameter σ to improve spectral clustering. However, the found optimal value is a global value which is not suitable for data sets with arbitrary construction (e.g., data with clusters in different densities). Zelnik-Manor et al. [11] proposed a self-tuning spectral clustering algorithm, which improved the spectral clustering algorithm in [10] by locally scaling the parameter in similarity measure. The local parameter can be calculated by studying the local statistics of the neighborhood of each data point. Inspired by Zelnik-Manor et al. [11], a family of spectral clustering algorithms aiming at finding local scaling parameter in Gaussian kernel function were presented [12], [13], [14]. Li et al. [12] proposed a warping model to map data into a new space for more accurately similarity measure. Zhang et al. [13] used the local density between data points to scale the parameter, which had an effect of amplifying intra-cluster similarity. Cao et al. [14] measured the similarity based on maximum flow between data points to satisfy the requirements of a similarity measure to be used in spectral clustering.

Instead of focusing on the kernel parameter, many spectral clustering algorithms measure similarity based on the k-Nearest Neighbor (kNN) graph [2]. In the kNN graph, point x_i is connected with point x_j if x_i is among the k-nearest neighbors of x_j or x_j is among the k-nearest neighbors of x_i. According to the k-NN graph, the pairwise similarity is s_{ij} if point x_i is connected with point x_j, otherwise $s_{ij} = 0$. Similarity measure based on the kNN graph has the following two advantages. (1) The main parameter (i.e., the Kernel parameter σ) is replaced by the number of nearest neighbors, k, which is easier to find since it is an integer and usually takes small values. (2) The similarity matrix $S = (s_{ji})$ is sparse, which is computational efficiency for the solution of eigenvectors. Spectral clustering algorithms based on the kNN graph can be find in [2], [15], [16], [17]. In spectral clustering, the wildly used kNN graph is undirected due to the symmetric property of the similarity matrix. However, the undirected kNN graph usually introduces redundant connections which may cause incorrect clustering results.

Both the local scaling parameter and the undirected kNN graph based spectral clustering algorithms seem to be effective in some clustering tasks. However, they can not reveal the real clusters of some complex data sets, especially data sets which are not well separated. Such data sets are not well separated because some data points in different clusters are

978-1-4799-1959-8/15/$31.00

not so far apart. The existence of noises can also make the data sets not well separated. Since many real data sets are not well separated, making the correct clusters difficult to find, proposing a clustering algorithm which is robust to not well-separated data sets is important and desirable.

An interesting alternative to distance based similarity measure is the similarity measure using shared nearest neighbor information. In most cases, two data points belong to the same cluster not only because they are near in the distance, but also because they have many shared nearest neighbors which connect them in the same cluster. In shared nearest neighbor based clustering methods, two data points have higher similarity if they have more shared nearest neighbors [18]. Shared nearest neighbor based similarity measure methods are more robust to not well-separated data sets, and they have been reported to be effective in practice, as well as being less sensitive to the dimensionality than conventional distance measure [19]. Specifically, they have been applied successfully in agglomerative clustering algorithms [20], [21], [22] and in clustering methods for high-dimensional data sets [23], [24].

In this paper, we propose a novel spectral clustering algorithm based on shared nearest neighbors, referred to as SC-SNN, in order to improve the clustering quality of not well-separated data sets. In SC-SNN, we measure similarity based on the shared nearest neighbors in the directed kNN graph. As far as we are aware, there are no existing spectral clustering algorithms based on the shared nearest neighbors in the directed kNN graph. The main contributions of the proposed SC-SNN algorithm are summarized as follows. (1) SC-SNN uses the directed kNN graph to find the nearest neighbors, which do not introduce redundant connections and more accurately reflect the true neighborhood of data points than the undirected kNN graph. (2) Instead of the distance metric, SC-SNN measures the similarity by considering the closeness of the shared k nearest neighbors. Beside the shared k nearest neighbors, the interrelationships of the measured data points are also considered in the similarity measure. Thus, SC-SNN is able to explore the underlying similarity relationships between data points and is robust to not well-separated data sets. (3) SC-SNN has only one parameter, k, i.e., the number of nearest neighbors. The parameter k in SC-SNN is less sensitive than that in the spectral clustering algorithms based on the undirected kNN graph.

We demonstrate the effectiveness of the proposed SC-SNN algorithm on both synthetic data sets and real world data sets. We also compare SC-SNN with other existing spectral clustering algorithms in the experiments to further demonstrate the good performance of SC-SNN.

The rest of this paper is organized as follows. We begin with a brief overview of spectral clustering in Section II. The proposed spectral clustering algorithm based on shared nearest neighbors, SC-SNN, is presented in Section III. The experimental results are presented in Section IV. Finally we conclude the paper in Section V.

II. Overview of spectral clustering

Given a set of n data points $X = \{x_1, x_2, ..., x_n\}$ in $\mathbb{R}^d$, the objective of clustering is to divide the data points into K clusters. A general algorithm for spectral clustering consists of three basic stages: pre-processing, decomposition, grouping [2], [10].

A. Pre-processing

The main task in pre-processing is the similarity measure of data points to construct a similarity matrix S, which is important to the next two stages and crucial to the clustering results of spectral clustering. The similarity matrix S has pairwise similarities s_{ij} as its entries, i.e., $S = (s_{ij})$. If the Gaussian kernel function is being applied, the pairwise similarity is calculated as

$$s_{ij} = \begin{cases} \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2}), & \text{if } i \neq j \ , \\ 0, & \text{if } i = j, \end{cases} \tag{1}$$

where $\|x_i - x_j\|$ is the Euclidean distance between data points x_i and x_j, and σ is the kernel parameter. If the similarly based on the undirected kNN graph is measured, s_{ij} is as given by equation (1) when point x_i is connected with point x_j, otherwise $s_{ij} = 0$.

The normalized Laplacian matrix L is then computed based on the similarity matrix S, as $L = I - D^{-1/2}SD^{-1/2}$, where D is the $n \times n$ diagonal matrix with $d_i = \sum_{j=1}^{n} s_{ij}$ on the diagonal.

B. Decomposition

The main task in decomposition is the analysis of eigenvectors obtained from the normalized Laplacian matrix L. As suggested in [10], the K largest eigenvectors of L are computed and the matrix $U \in \mathbb{R}^{n \times K}$ with these eigenvectors as columns is formed. Let each row of U represent a data point in $\mathbb{R}^K$ and cluster these points by K-means.

C. Grouping

Each original data point x_i is mapped to the data point represented in row i of U. Each original data point x_i is assigned to a given cluster c if the mapped data point is assigned to the cluster c.

III. Spectral clustering based on shared nearest neighbors

In this section, we propose a novel Spectral Clustering algorithm based on Shared Nearest Neighbors (SC-SNN). SC-SNN first constructs the directed kNN graph, and then finds the shared nearest neighbors and measures similarity based on the closeness of shared nearest neighbors. Finally, clustering is performed based on the measured similarity.

A. Constructing the directed kNN graph

Since the undirected kNN graph usually introduces redundant connections, we consider to measure the similarity of data points based on the directed kNN graph. In the directed kNN graph, an edge is connected form x_i to x_j if x_j is one of the k-nearest neighbors of x_i. x_j is a direct successor of x_i, denoted as $x_i \rightarrow x_j$. Further, let $x_i \leftrightarrow x_j$ denote the case that x_i and x_j are the k-nearest neighbors of each other. In the directed kNN graph, each data point has exactly k nearest neighbors, while some data points have more than k nearest neighbors in the undirected kNN graph.

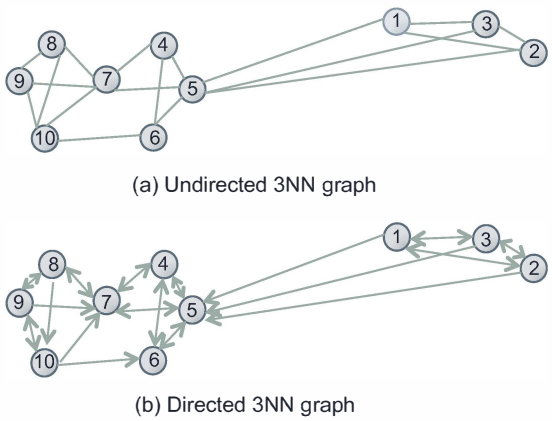


Fig. 1. Data points are clustered based on 3NN graphs.

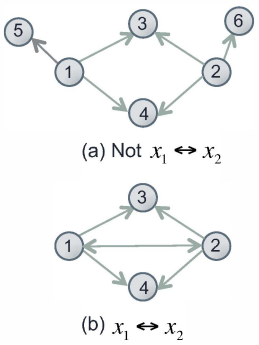


Fig. 2. Shared neighbors of points 1 and 2 in two cases.

As an example, in Fig. 1, data points are clustered based on the undirected 3NN graph (a) and the directed 3NN graph (b). Note that in the undirected 3NN graph, point 5 has more than 3 nearest neighbors. For point 5, the edges between point 5 and points 1, 2, and 3 are redundant, which leads to an incorrect clustering result. The resulting two clusters based on this undirected 3NN graph are $\{1, 2, 3, 5\}$ and $\{4, 6, 7, 8, 9, 10\}$, where point 5 is assigned to the wrong cluster.

The directed kNN graph does not suffer from redundant connections as the undirected kNN graph does. Instead, the problem with using the directed kNN graph is its asymmetry, e.g., in Fig. 1 (b), point 5 is a nearest neighbor of point 1, while point 1 is not a nearest neighbor of point 5. Since the similarity matrix in spectral clustering is symmetric, traditional similarity measurement methods (e.g., the similarity measurement method in equation (1)) cannot apply the directed kNN graph. We will show that SC-SNN introduces a novel similarity measurement method which is able to apply the directed kNN graph.

B. Founding the shared nearest neighbors

Let N_i denote the set of nearest neighbors of x_i in directed kNN graph. The set of shared nearest neighbors between x_i and x_j is $N_i \cap N_j$. The pairwise similarity s_{ij} is measured by considering the set $N_i \cap N_j$. In general, N_i dose not include x_i. Thus, $N_i \cap N_j$ dose not include the two measured points x_i and x_j. Whether $N_i \cap N_j$ should include x_i and x_j depends on the relationship of x_i and x_j, which will affect the similarity measure. We will use an example in Fig. 2 to explain the reason. We show the three nearest neighbors of points 1 and 2 in two different cases in Fig. 2 (a) and (b) respectively. Points 1 and 2 are the nearest neighbors of each other in Fig. 2 (b), while they are not in Fig. 2 (a). If we do not consider the relationship of points 1 and 2, the set of shared nearest neighbors of points 1 and 2 in the two cases are the same. However, the similarity of points 1 and 2 in Fig. 2 (b) is higher than that in Fig. 2 (a), since points 1 and 2 are the nearest neighbors of each other in Fig. 2 (b).

In SC-SNN, we redefine the set of shared nearest neighbors of two points in the directed kNN graph by considering the relationship of the two measured points. N_i is the set of nearest neighbors of x_i and does not include x_i. $N_i \cap N_j$, which is the set of shared nearest neighbors between points x_i and x_j, is redefined as

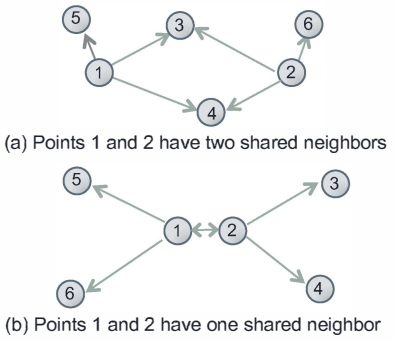
$$N_i \cap N_j = \begin{cases} N_i \cap N_j \cup \{x'_{ij}\}, & \text{if } x_i \leftrightarrow x_j, \\ N_i \cap N_j, & \text{otherwise,} \end{cases} \quad (2)$$

where x'_{ij} is a virtual data point, which represents x_i as a nearest neighbor of x_j and represents x_j as a nearest neighbor of x_i. Thus, without considering other shared neighbors, x_i and x_j have one shared nearest neighbor if $x_i \leftrightarrow x_j$, as in the example shown in Fig. 3 (b) .

C. Measuring the pairwise similarity

We measure the pairwise similarity s_{ij} based on the set of shared nearest neighbors $N_i \cap N_j$ as defined in equation (2). Many shared nearest neighbor based clustering methods consider the pairwise similarity as a function of the number of shared nearest neighbors (e.g., the clustering methods in [18], [21], [22]). According to these methods, two data points have higher pairwise similarity if they have more shared nearest neighbors. However, only considering the number of shared nearest neighbors may cause an incorrect measurement result. As the example in Fig. 3, we show the three nearest neighbors of points 1 and 2 in two different cases. Points 1 and 2 have two shared neighbors in Fig. 3 (a), while they have one shared neighbor in Fig. 3 (b). If only the number of shared nearest neighbors is considered, the pairwise similarity of points 1 and 2 in Fig. 3 (a) is higher than that in Fig. 3 (b). However, in Fig. 3 (b), points 1 and 2 are very close and they should have higher pairwise similarity than that in Fig. 3 (a). Thus, only considering the number of shared nearest neighbors may neglect the closeness of data points.

In SC-SNN, we measure the similarity by considering the closeness of shared nearest neighbors to the measured data points. The orders of the shared nearest neighbors in $N_i \cap N_j$ to the two measured data points x_i and x_j can reflect the closeness of shared nearest neighbors. For example, in Fig. 3 (a), points 3 and 4 are the shared neighbors of points 1 and 2. Point 3 is the second-nearest neighbor of point 1 and the third-nearest neighbor of point 2. Point 4 is the second-nearest neighbor of point 2 and the third-nearest neighbor of point 1. Neither of the two shared neighbors is the first-nearest neighbor of points 1 and 2. In Fig. 3 (b), although points 1 and 2 have only one shared neighbor (i.e., the virtual point which represents each of points 1 and 2 as the nearest neighbor of the other), this shared neighbor is the first-nearest neighbor of



(a) Points 1 and 2 have two shared neighbors

(b) Points 1 and 2 have one shared neighbor

Fig. 3. Three nearest neighbors of points 1 and 2 in two cases.

both points 1 and 2 (i.e., points 1 and 2 are the first-nearest neighbors of each other).

To measure the pairwise similarity s_{ij}, we first weight the shared nearest neighbors in $N_i \cap N_j$ according to their orders to the data points x_i and x_j. Let w_{ij} denote the weight of shared nearest neighbors in $N_i \cap N_j$. Let x_r be one of the shared nearest neighbors in $N_i \cap N_j$. Assume that x_r is the i_r^{th} nearest neighbor of x_i and the j_r^{th} nearest neighbor of x_j. In the directed kNN graph, the weight of shared nearest neighbors w_{ij} is calculated as

$$w_{ij} = \sum_{x_r \in N_i \cap N_j} (k - i_r^{th} + 1)(k - j_r^{th} + 1). \tag{3}$$

Note that the maximum value of w_{ij} is obtained when $|N_i \cap N_j| = k$ and the shared neighbors in $N_i \cap N_j$ have the same orders to data points x_i and x_j. According to equation (3), the maximum value of w_{ij} is $max\{w_{ij}\} = \sum_{p=1}^{k} p^2$.

For statistical analysis, we consider pairwise similarity in the range $s_{ij} \in [0, 1]$. The pairwise similarity s_{ij} is calculated as

$$s_{ij} = \frac{w_{ij}}{max\{w_{ij}\}}. \tag{4}$$

The parameter of s_{ij} in equation (4) is the number of nearest neighbors k. The similarity matrix $S = (s_{ij})_{i,j=1,...,n}$ is sparse, since k is usually small compared to n, which is similar to that considered in the spectral clustering algorithms based on undirected kNN graph. We will show that the parameter k in SC-SNN is less sensitive than that in the spectral clustering algorithms based on the undirected kNN graph.

D. Clustering based on the similarity measure

The next step is clustering performed based on the similarity matrix S. Besides the construction of similarity matrix S, the SC-SNN algorithm is summarized as

(1) Construct the directed kNN graph and find the shared nearest neighbors.

(2) Measure the pairwise similarity and construct the similarity matrix S.

(3) Compute the normalized Laplacian matrix L based on S.

(4) Compute the K largest eigenvectors of L.

(5) Cluster the data points into K clusters based on the K largest eigenvectors.

The last three steps are performed as the processes introduced in Section II. The main contributions of SC-SNN are in the first two steps, in which the similarly matrix S is constructed, since the similarity matrix S is important for the following steps and crucial to the results of spectral clustering.

IV. Experimental results

The proposed SC-SNN algorithm is evaluated on both synthetic and real data sets. We compare SC-SNN with two spectral clustering algorithms which use a local scaling parameter in the Gaussian kernel function to calculate similarity: one using density adaptive similarity, SC-DA [13], and one using self-turning, SC-ST [11]. We also compare SC-SNN with the Spectral Clustering algorithm based on the kNN graph (SC-kNN) [2]. To show the benefit of spectral clustering, we also compare these spectral clustering algorithms with K-means.

We adopt Normalized Mutual Information (NMI) as the evaluation criterion, since it is wildly used to evaluate the performance of clustering algorithms [26]. Let $C = \{c_1, c_2, ..., c_K\}$ denote the true clustering configuration, and $C' = \{c'_1, c'_2, ..., c'_K\}$ denote the predicted clustering configuration obtained by a clustering algorithm. $P(c_i) = |c_i|/n$ is the probability that data points belong to cluster c_i, where $|c_i|$ is the cardinality of cluster c_i and n is the number of total data points. $P(c_i \cap c'_j) = |c_i \cap c'_j|/n$ is the probability that data points belong to the intersection of clusters c_i and c'_j. The NMI criterion is formulated as

$$NMI(C, C') = \frac{2\varphi(C, C')}{\varphi(C) + \varphi(C')}, \tag{5}$$

where

$$\varphi(C) = -\sum_{i=1}^{K} P(c_i) \log P(c_i),$$

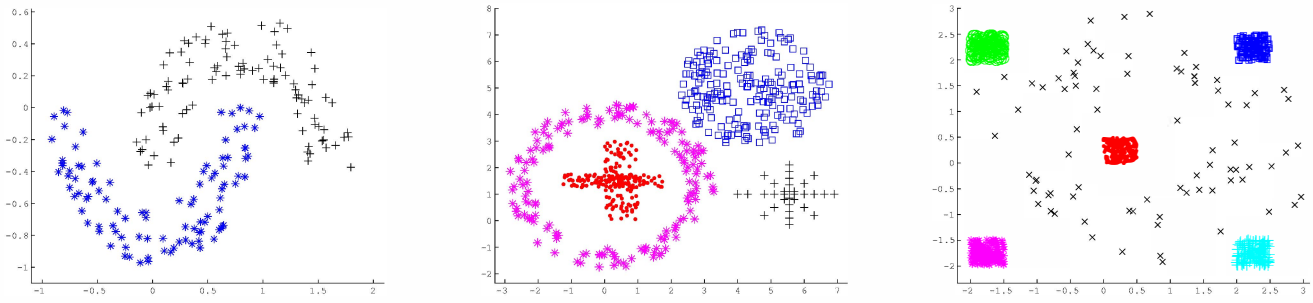
and

$$\varphi(C, C') = \sum_{i=1}^{K} \sum_{j=1}^{K} P(c_i \cap c'_j) \log \frac{P(c_i \cap c'_j)}{P(c_i)P(c'_j)}.$$

A larger value of NMI indicates a better clustering result, where NMI is at most 1, which occurs when all the data points are assigned to their correct clusters.

Note that the proposed SC-SNN algorithm measures similarity based on the shared k-nearest neighbors. The shared k-nearest neighbors can be found according to different distance metrics, such as the Euclidean distance and the cosine metric. In the experiments, we use the Euclidean distance, which is similar to the metric used in the compared clustering algorithms.

In the following experiments, we present the best results of each spectral clustering algorithm obtained after exploring their parameters as suggested in the related papers. The proposed SC-SNN algorithm has only one parameter, k. For SC-SNN and the compared SC-kNN algorithm, which have the same key parameter, k, and are both computational efficiency due to having parse similarity matrices, we will compare them on their sensitivity to parameter k.



(a) Two Moons (b) Different Shapes (c) Distributed Noises

Fig. 4. Three 2D data sets with the true clusters denoted by different markers. (a) Two clusters in two moons. (b) Four clusters in four different shapes. (c) Five clusters in five squares and one cluster (noise data) distributed among the squares.

TABLE I. PROPERTIES OF SYNTHETIC DATA SETS

Dataset	Number of instances	Dimensions	Clusters
Two Moons	200	2	2
Different Shapes	600	2	4
Distributed Noises	880	2	6
HD Data 1	600	50	5
HD Data 2	450	100	4
HD Data 3	360	150	3

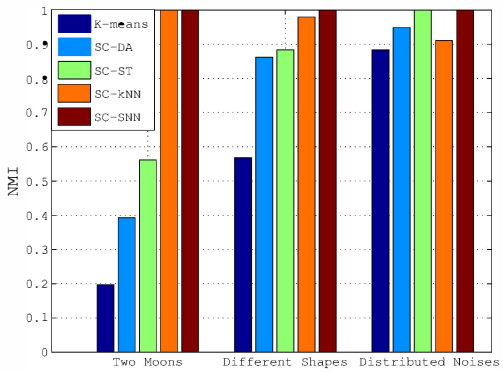


Fig. 5. Clustering results in three 2D synthetic data sets.

A. *Clustering results on synthetic data*

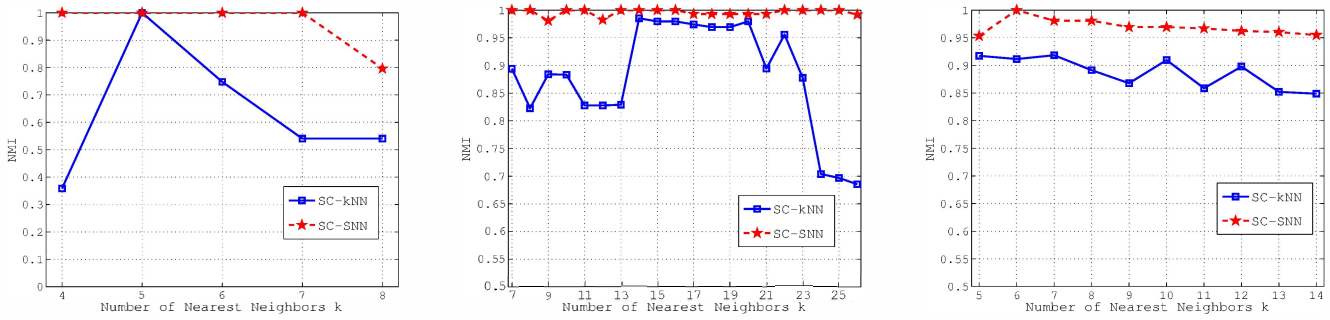
We conduct experiments on three low-dimensional (i.e., 2D) and three-high dimensional synthetic data sets. These synthetic data are not well separated. Their properties are summarized in Table I.

We first show the experimental results for the three 2D synthetic data sets shown in Fig. 4. The data set in Fig. 4 (a) (denoted as *Two Moons*) is a typical clustering example to show the benefit of spectral clustering, i.e., it can only be solved by spectral clustering. However, when two "moons" of data are not far apart, clustering become difficult even for spectral clustering algorithms. The proposed SC-SNN algorithm can accurately explore the underlying structure of such data sets. As shown in Fig. 5, SC-SNN and SC-kNN can find the two clusters correctly and they outperform other clustering algorithms significantly on *Two Moons*. K-means is poor at dealing with *Two Moons*. The data set in Fig. 4 (b) (denoted as *Different Shapes*) consists of four different shapes, which is a challenge for traditional clustering algorithms. Spectral clustering algorithms outperform K-means on *Different Shapes*. Among these, SC-SNN performs better than the other spectral clustering algorithms, and SC-kNN is second best.

To show the robustness to noises, we use the data set in Fig. 4 (c)) (denoted as *Distributed Noises*), which contains noises distributed between the other five clusters. The noises in Fig. 4 (c) are assumed to be known and constitute one cluster. The existence of noises make the data sets not well separated. We can see from Fig. 5 that SC-SNN and SC-ST are robust to noises and perform better than the other clustering algorithms. Since the data structure in the *Distirbuted Noises* is not as complex as the data structures in *Two Moons* and *Different Shapes*, the clustering result by K-means in *Distributed Noises* is better than that in *Two Moons* and *Different Shapes*.

We evaluate the sensitivity to parameter k of SC-SNN and SC-kNN for the three 2D data sets. We show the best results for SC-SNN and SC-kNN for the corresponding value k. For *Two Moons*, although SC-kNN can find the correct clusters, it is sensitive to k: it can only find the correct clusters when $k = 5$. SC-SNN performs better than SC-kNN. For *Different Shapes*, its results are more stable with respect to k, giving good clustering results for a wide range of k values. Also in *Distributed Noises*, the parameter k in SC-SNN is less sensitivity than that in SC-kNN.

Experiments were also conducted on three high-dimensional data sets, , denoted as *HD data 1*, *HD data 2* and *HD data 3*. These data sets are generated by several Gaussian distributions with random centers [9]. As shown in Fig. 7, SC-SNN outperforms the other clustering algorithms, especially for *HD data 3*, which has the highest dimension. For each HD data set, we show the best results for SC-SNN and SC-kNN for the ten corresponding values of k in Fig. 8. Note that for the high-dimensional data sets, as the data dimension increases, the k value that produces the best results also increase. In the case of *HD data 1*, these k values range from 3 to 12, whereas for *HD data 2* and *HD data 3*, best results are obtained from 12 to 21 and from 26 to 35, respectively. The parameter k in SC-SNN is less sensitive than that in SC-kNN for all the three HD data sets. Especially in *HD data 3*, The clustering results of SC-SNN is especially stable for the shown k values, which are all equal to 0.9849



(a) Two Moons (b) Different Shapes (c) Distributed Noises

Fig. 6. The value of Normalized Mutual Information (NMI) by varying the number of nearest neighbors k in three 2D synthetic data sets.

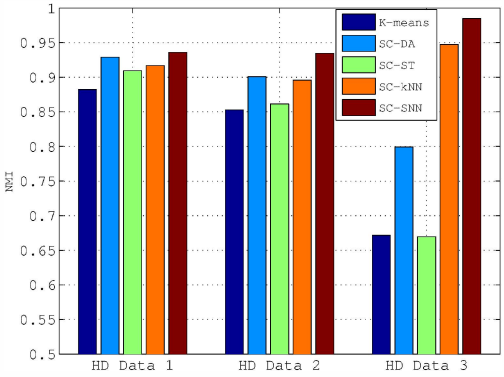


Fig. 7. Clustering results in three High Dimension (HD) synthetic data sets.

in terms of the NMI criterion.

B. Clustering results on real data

We conduct experiments on five real data sets from the UCI machine learning repository [27]. The properties of the five data sets are summarized in Table IV-B. The clustering results for these data sets are shown in Fig. 9. In terms of the NMI criterion, the proposed SC-SNN algorithm outperforms other clustering algorithms for all five UCI data sets. In the *Iris* data set, the benefit of using SC-SNN is particularly obviously.

TABLE II. PROPERTIES OF UCI DATA SETS

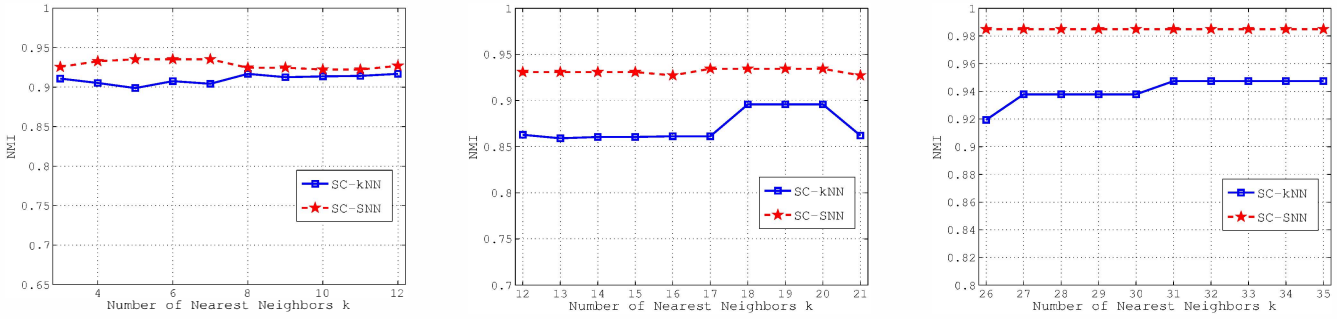
Dataset	Number of instances	Dimensions	Clusters
Iris	150	4	3
Breast	699	9	2
Wine	178	13	3
Ecoli	336	7	8
Glass	214	9	6

In the *Wine* data set, SC-SNN and SC-DA perform best, having similar clustering results, while K-means performs worst, much poorer than the other spectral clustering algorithms.

We compare the clustering results of SC-SNN with SC-kNN for variations in k for the five UCI data sets to show their sensitivity to this parameter. The five best results in terms of k for SC-SNN and SC-kNN are shown in Fig. 10. From these results, we can see that SC-SNN is less sensitivity to parameter k than is SC-kNN. Also, SC-SNN outperforms SC-kNN for all the shown k values for all five UCI data sets.

V. CONCLUSION

In this paper, we propose a novel algorithm for spectral clustering based on shared nearest neighbors, SC-SNN, to improve the clustering quality of the not well-separated data sets. In the proposed SC-SNN algorithm, the similarity measure is based on the closeness of shared nearest neighbors in the *directed* kNN graph. SC-SNN is less sensitive to its only parameter, k, than are spectral clustering algorithms based on



(a) HD Data 1 (b) HD Data 2 (c) HD Data 3

Fig. 8. The value of Normalized Mutual Information (NMI) by varying the number of nearest neighbors k in three High Dimension (HD) synthetic data sets.

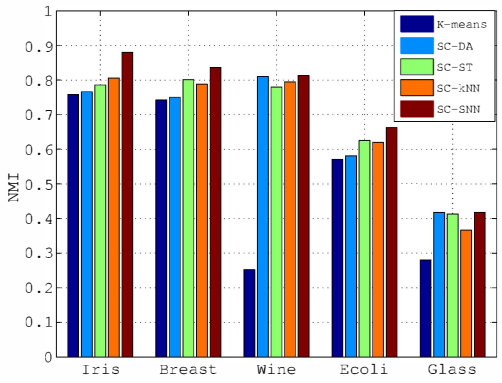


Fig. 9. Clustering results in five UCI data sets.

the undirected kNN graph. The experimental results for both synthetic and real-world data sets demonstrate that SC-SNN is not only robust to not well-separated data sets, but also outperforms the traditional spectral clustering algorithms. In the future, we will consider distributed and parallel algorithms for constructing the similarity matrix and computing the eigenvectors, which would make it possible to apply the spectral clustering algorithm to big data sets.

REFERENCES

[1] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *CM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[2] U. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[3] F. R. Bach and M. I. Jordan, "Learning spectral clustering," in *Proceeding of Advances In Neural Information Processing Systems*, 2004, pp. 305–312.

[4] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[5] J. Malik and S. Belongie, "Contour and texture analysis for image segmentation," *International Journal of Computer Vision*, vol. 43, no. 1, pp. 7–27, 2001.

[6] B. Hendrickson and R. Leland, "An improved spectral graph partitioning algorithm for mapping parallel computations," *SIAM Journal on Scientific Computing*, vol. 16, no. 2, pp. 452–469, 1995.

[7] C. J. Alpert and A. B. Kahng, "Multiway partitioning via geometric embeddings, orderings and dynamic programming," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 4, no. 11, pp. 1342–1358, 1995.

[8] W. Hu, D. Xie, Z. Fu, W. Zeng, and S. Maybank, "Semantic-based surveillance video retrieval," *IEEE Transactions on Image Processing*, vol. 16, no. 4, pp. 1168–1181, 2007.

[9] Z. Yu, L. Li, J. You, and G. Han, "Sc3: Triple spectral clustering based consensus clustering framework for class discovery from cancer gene expression profiles," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 6, pp. 1751–1765, 2012.

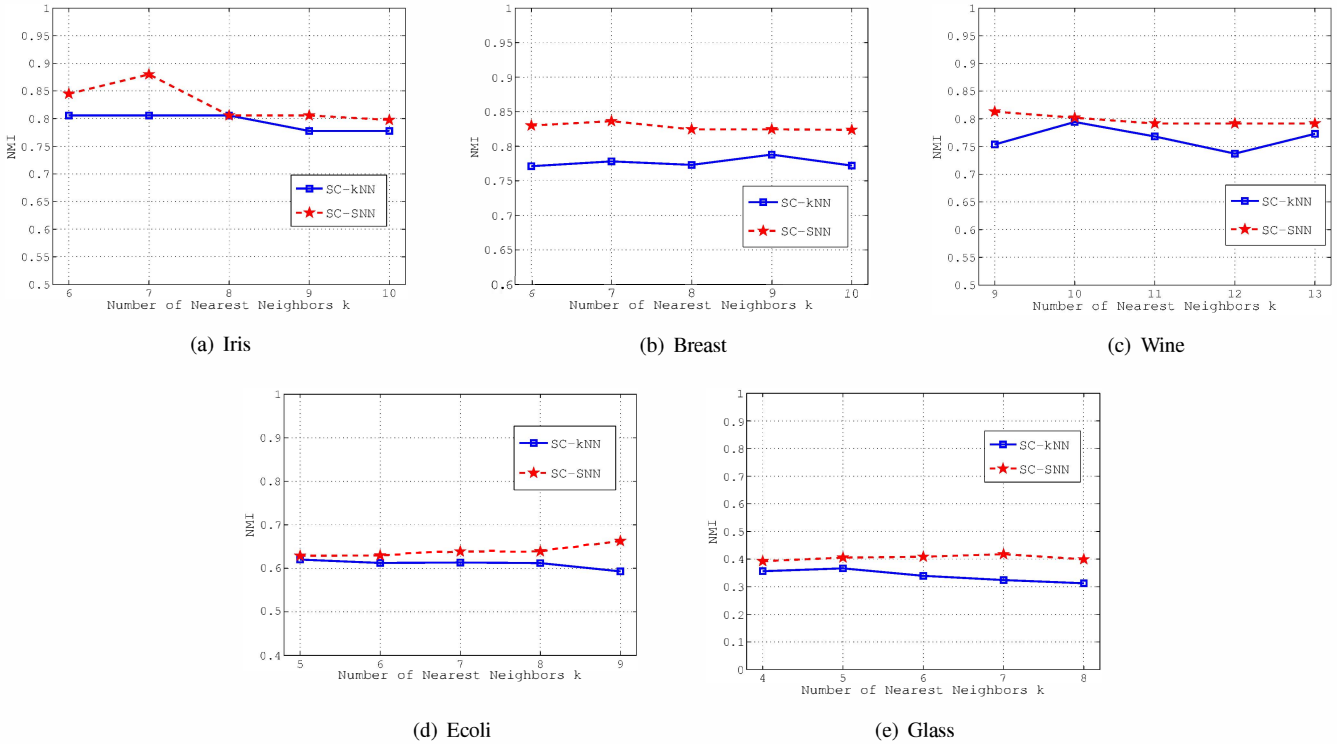
[10] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proceeding of Advances In Neural Information Processing Systems*, 2002, pp. 849–856.

[11] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proceeding of NIPS*, 2005, pp. 1601–1608.

[12] Z. Li, J. Liu, S. Chenand, and X. Tang, "Noise robust spectral clustering," in *Proceeding of ICCV*, 2007.

[13] X. Zhang, J. Li, and H. Yu, "Local density adaptive similarity measurement for spectral clustering," *Pattern Recognition Letters*, vol. 32, no. 2, pp. 352–358, 2011.

[14] J. Cao, P. Chen, and Y. Z. Q. Dai, "A max-flow-based similarity measure for spectral clustering," *ETRI Journal*, vol. 35, no. 2, pp. 311–320, 2013.



(a) Iris (b) Breast (c) Wine

(d) Ecoli (e) Glass

Fig. 10. The value of Normalized Mutual Information (NMI) by varying the number of nearest neighbors k in five UCI data sets.

[15] M. Lucińska and S. T. Wierzchoń, "Spectral clustering based on k-nearest neighbor graph," *Computer Information Systems and Industrial Management*, vol. 7564, pp. 254–265, 2012.

[16] C. Xiong, D. M. Johnson, and J. J. Corso, "spectral active clustering via purification of the k nearest neighbor graph," in *Proceedings of European Conference on Data Mining*, 2012.

[17] X. Li, W. Hu, C. Shen, A. Dick, and Z. Zhang, "Context-aware hypergraph construction for robust spectral clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 10, pp. 2588–2597, 2013.

[18] S. Guha, R. Rastogi, and S. Kyuseok, "Rock: a robust clustering algorithm for categorical attributes," in *Proceedings of 15th International Conference on Data Engineering*, 1999, pp. 512 – 521.

[19] M. E. Houle, H. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Can shared-neighbor distances defeat the curse of dimensionality?" in *Proceedings of the 22nd international conference on Scientific and statistical database management*, 2010, pp. 482–500.

[20] R. A. Jarvis and E. A. Patrick, "Clustering using a similarity measure based on shared near neighbors," *IEEE Transactions on Computers*, vol. C-22, no. 11, 1973.

[21] S. Guha, R. Rastogi, and K. Shim, "Cure: an efficient clustering algorithm for large databases," in *Proceedings of the ACM SIGMOD international conference on Management of data*, 1998, pp. 73–84.

[22] L. Ertoz, M. Steinbach, and V. Kumar, "Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data," in *Proceedings of Second SIAM International Conference on Data Mining*, 2003.

[23] M. E. Houle, "Navigating massive data sets via local clustering," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 547–552.

[24] H. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Outlier detection in axis-parallel subspaces of high dimensional data," in *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, 2009, pp. 831–838.

[25] W. Chen, Y. Song, H. Bai, C. Lin, and E. Chang, "Parallel spectral clustering in distributed systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 568 – 586, 2011.

[26] A. Strehl and J. Ghosh, "Cluster ensembles – a knowledge reuse framework for cluster ensembles – a knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, vol. 3, no. 3, pp. 583–617, 2003.

[27] "http://archive.ics.uci.edu/ml/."