A Max-Flow-Based Similarity Measure for Spectral Clustering

Jiangzhong Cao, Pei Chen, Yun Zheng, and Qingyun Dai

In most spectral clustering approaches, the Gaussian kernel-based similarity measure is used to construct the affinity matrix. However, such a similarity measure does not work well on a dataset with a nonlinear and elongated structure. In this paper, we present a new similarity measure to deal with the nonlinearity issue. The maximum flow between data points is computed as the new similarity, which can satisfy the requirement for similarity in the clustering method. Additionally, the new similarity carries the global and local relations between data. We apply it to spectral clustering and compare the proposed similarity measure with other state-of-the-art methods on both synthetic and real-world data. The experiment results show the superiority of the new similarity: 1) The max-flow-based similarity measure can significantly improve the performance of spectral clustering; 2) It is robust and not sensitive to the parameters.

Keywords: Spectral clustering, maximum flow, affinity graph, similarity measure.

I. Introduction

Spectral clustering has attracted a significant amount of attention [1]-[4] due to its impressive performance on some challenging clustering datasets, with successful applications in computer vision [5], [6], VLSI design [7], and speech processing [8], [9]. It has been shown that the affinity matrix is crucial to the performance of spectral clustering [10]-[16]. Most spectral clustering methods adopted the Gaussian kernel function as a similarity measure to construct the affinity matrix [5], [11]-[13], where only the parameters are different. In [11], a fixed scaling parameter controls how fast the similarity falls off with the distance between points. In [12], a self-tuning parameter was used to adapt to the multiscale dataset. In [13], the Gaussian kernel function was scaled according to the local density between data points so that the similarity between two points is higher if there are more common points in their ε-neighborhood.

Though the Gaussian kernel-based similarity measure can describe the information of the local consistency, it does not work well on a dataset with a nonlinear elongated structure. See the example in Fig. 1(a), which reflects three spiral clusters. The grayscale of lines indicates the similarity between the points. The darker the line is, the larger the similarity is. One can easily find cases in Fig. 1(a) wherein the similarity between the points in the same manifold is smaller than that for a different manifold. This phenomenon results in unsatisfactory performance for spectral clustering algorithms.

To overcome the difficulty mentioned above, Fischer and Buhmann [17], [18] proposed a path-based similarity measure based on a connectedness criterion, which considers two objects as similar if there exists a mediating intra cluster path without any large-cost edge. Though the path-based measure

Manuscript received July 31, 2012; revised Oct. 7, 2012; accepted Oct. 22, 2012.

This work was supported by the National Natural Science Foundation of China through the program 61173083, by the Ministry of Science and Technology, China, through the 973 Program 2011CB302200 and by the Economic & Information Commission of Guangdong province through the Program GDIID2008IS007.

Jiangzhong Cao (phone: +86 135 6008 2826, cjz510@gdut.edu.cn) is with the School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China, and also with the School of Information Engineering, Guangdong University of Technology, Guangzhou, China.

Pei Chen (chenpei@mail.sysu.edu.cn) and Yun Zheng (zhengyun84@gmail.com) are with the School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China.

Qingyun Dai (daiqy@gdut.edu.cn) is with the School of Information Engineering, Guangdong University of Technology, Guangzhou, China.

http://dx.doi.org/10.4218/etrij.13.0112.0520

can partly overcome the difficulty with nonlinearity, it is sensitive to noise. In [14], a robust path-based similarity measure based on M-estimator was proposed to improve the robustness of the path-based spectral clustering. It was reported that the robust path-based measure performs well on some datasets; however, the measure favors taking the data points around the clusters as noise, as shown in [14].

In this paper, we propose a max-flow-based similarity measure for constructing the affinity matrix, originating from the fact that data points in the same cluster are more connected than data points in different clusters, as shown in Fig. 1(a). The maximum flow between data points is computed as the new similarity, in which a weighted graph is constructed by using the technique of *k*-nearest neighbor (*k*-NN), *ε*-neighborhood, or a combination of both. As opposed to the path-based similarity measure, the maximum flow takes all paths between two points into account, not just the shortest one. Thus, the proposed measure reflects the global similarity between two points through all paths: the maximum flow (similarity) is larger when there are more paths or shorter paths connecting the two points. The commute time distance (resistance distance) and its variants based on the random walk (electronic network) have been proposed to carry out a similar idea and have been widely used [19]-[21]; however, we will show that the max-flow-based similarity measure can improve the performance of the spectral clustering algorithm on most datasets in our experiments.

The rest of this paper is organized as follows. The background of spectral clustering is reviewed in section II. In section III, we propose a max-flow-based similarity measure and apply it to construct the affinity matrix in detail. Experiment results on some datasets are presented in section IV, and some concluding remarks are given in section V.

II. Background on Spectral Clustering

1. Ng-Jordan-Weiss Algorithm

Most spectral clustering algorithms follow the spirit of the Ng-Jordan-Weiss (NJW) algorithm [11]. For completeness, the NJW algorithm is briefly reviewed here.

Given a dataset $S=\{x_1,\cdots,x_n\}$ in $\Re^l$, the NJW algorithm is implemented as follows. 1) Construct an affinity matrix **A** by the Gaussian kernel function in (1):

$$A_{ij}=\begin{cases}\exp(\dfrac{-\|x_i-x_j\|^2}{\delta^2}) & \text{for } i\neq j,\\ 0 & \text{for } i=j,\end{cases}\tag{1}$$

where δ is a scale parameter to control how fast the similarity changes with the distance between the data points x_i and x_j. 2) Compute the normalized affinity matrix $\mathbf{L}$=$\mathbf{D}^{-1/2}\mathbf{A}\,\mathbf{D}^{-1/2}$, where **D** is the diagonal matrix with $\mathbf{D}_{ii}=\sum_{j=1}^{n}\mathbf{A}_{ij}$. 3) Compute the *K* eigenvectors of **L**, $\mathbf{v_1}$, $\mathbf{v_2}$,…, $\mathbf{v_k}$, which are associated with the *K* largest eigenvalues, and form the matrix $\mathbf{X}=[\mathbf{v}_1\mathbf{v}_2\cdots\mathbf{v}_\mathbf{K}]$. 4) Renormalize each row to form a new matrix $\mathbf{Y}\in\Re^{\mathrm{n\times K}}$ with $\mathbf{Y}_{ij}=\mathbf{X}_{ij}\big/(\sum_j\mathbf{X}_{ij}^2)^{1/2}$, so that each row of **Y** has a unit length. 5) Treat each row of **Y** as a point in $\Re^K$ and partition the *n* points (*n* rows) into *K* clusters via a general cluster algorithm, such as k-means algorithm. 6) Assign the original point x_i to the cluster *c* if and only if the corresponding row *i* of the matrix **Y** is assigned to the cluster *c*.

2. Similarity Graph

A weighted graph *G*=(*V*, *E*) is a convenient tool for describing the similarity between data points, where *V* is the dataset $\{x_1, x_2,\ldots, x_n\}$ and the weight for the edge between x_i and x_j is the similarity. The adjacency matrix of the graph is the affinity matrix described in subsection II.1. The Gaussian kernel similarity in (1) results in a fully connected graph.

There are other approaches to construct the graph, including the *k*-NN and ε-neighborhood. In the *k*-NN graph, one vertex is only connected to its *k*-NNs, that is, the weight is computed as

$$w_{ij}=\begin{cases}s_{ij} & \text{if } x_j \text{ is one of the } k\text{-NNs of } x_i,\\ 0 & \text{otherwise,}\end{cases}\tag{2}$$

where s_{ij} is the similarity between x_i and x_j.

In the ε-neighborhood graph, the points whose pairwise distances (similarity) are smaller (larger) than ε are connected, defined as

$$w_{ij}=\begin{cases}s_{ij} & \text{if } \|x_i-x_j\|\le\varepsilon,\\ 0 & \text{otherwise.}\end{cases}\tag{3}$$

The *k*-NN or ε-neighborhood technique produces a sparse graph, which can help our method to reduce the computation and improve the performance. Generally, the *k*-NN-based graph is recommended as the first choice [15].

III. Max-Flow-Based Similarity Measure

Gaussian kernel function is widely used as the similarity measure for its ability to reflect the homogeneity of compactness. However, it fails on a dataset with an elongated structure, as shown Fig. 1(a). The two points existing on the same manifold should be homogeneous, that is, their similarity is high even if with a large Euclidean distance. Such a fact motivates us to seek a new similarity measure.

It was observed in [22] that the density of points in each cluster is considerably higher than that within the area separating the clusters. Figure 1(b) shows the sparse graph,

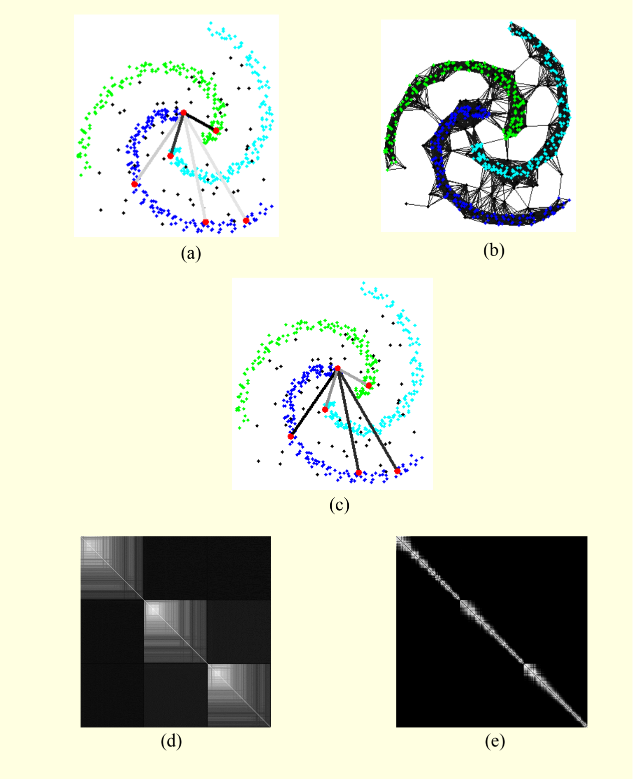

Fig. 1. Example for max-flow-based similarity measure. (a) Three spiral clusters with noises. Grayscale of lines denotes Gaussian kernel-based similarity. Darker the line is, larger the similarity is. (b) Sparse similarity graph by ε-neighborhood technique with ε=0.02. (c) Similarity based on max-flow. (d) Affinity matrix constructed with max-flow-based similarity measure. (e) Affinity matrix constructed with Gaussian kernel function as in (1).

constructed by the ε-neighborhood technique with ε=0.02. Points belonging to the same cluster are linked densely by more edges with larger weight, while the points belonging to different clusters are linked sparsely by fewer edges with smaller weight in noise area. Based on this observation, we propose to use the maximum flow between data points as the similarity measure, which is described in detail in the next subsection.

As an example, Fig. 1(c) shows the similarity between points using max-flow, which gives a good measure result.

1. Computation of *s-t* Max-Flow as Similarity Measure

For a weighted similarity graph G=(V, E), the weight of the edge connecting v_i and v_j is a positive real number known as the capacity of the edge, denoted by c_{ij}. It represents the maximum amount of flow that can pass through an edge. A feasible flow from a vertex s (source) to a vertex t (sink) is a set of numbers $\{f_{ij}\}$ associated with the edges. Each element of $\{f_{ij}\}$ satisfies the following conditions: 1) $0 \le f_{ij} \le c_{ij}$ and 2) $\sum_{e_{ik}\in E} f_{ik} - \sum_{e_{kj}\in E} f_{kj} = 0$ for each vertex $v_k \in V \backslash \{s,t\}$. The value of the flow from s to t, F_{st}, is defined as follows: $F_{st} \equiv \sum_{e_{sj}\in E} f_{sj} = \sum_{e_{it}\in E} f_{it}$.

The max-flow problem is to find the maximum F_{st}, which is related to the minimum cut. The Ford and Fulkerson theorem [23] reveals their relations: the maximum flow from the vertex s to the vertex t is equal to the value of the minimum cut separating s and t. There exist a few clustering algorithms based on the concept of "cut" [5], [24].

By regarding the edges on the graph as channels of communication, the flow on every edge can be considered as the amount of information passed through the edge, with the vertex s as the information source and the vertex t as the information receiver. The maximum flow from the vertex s to the vertex t reflects the maximum amount of information passing through all paths between the vertices s and t, not only through one path (for example, the shortest path). This way, the maximum flow carries the "global relations" between vertex s and t, and the resulting similarity is consequently more effective than similarity based only on "local relations."

There exist several algorithms for calculating the maximum flow between two points, and an experimental comparison of min-cut/max-flow algorithms is provided in [24]. However, one can easily encounter the degenerate cases in the process of computing maximum flow, wherein the minimum cut favors cutting the isolated s or t from the others in the graph.

To overcome the degeneracy difficulty, a modified maximum flow from s to t is computed. The weights of the edges connecting s (and t) with its neighbors are set as a large-enough constant (for example, the sum of the weights of all edges in our experiments) if s and t have no common neighbors and compute the maximum flow from s to t in the modified graph. Thus, the degenerate case will not occur. In computing modified maximum flow, all vertices in the graph are divided into two sets: one set contains all vertices that connect to the vertex s directly or by one intermediate vertex, denoted by V_1, and the other set V_2 contains the vertices not in V_1 ($V_1 \cup V_2 = V$ and $V_1 \cap V_2 = \Phi$). Then, the similarity among the vertices in the two sets is computed in different ways, respectively. The algorithm of computing the similarity between two vertices in a graph is described as follows.

Algorithm 1:

Input: a weighted graph G=(V, E) and vertex s.

Output: the similarity matrix between s and t.

Method:

1) Divide the vertices in the graph into two sets: V_1 and V_2, as described above.

2) For each vertex t in the V_2, construct the modified graph in which the weights of the edges connecting s (and t) with its neighbors are set as a large-enough value and calculate s-t

maximum flow F_{st} on the modified graph by the algorithm [24]. Set the similarity between s and t as the s-t maximum flow between s and t: S_{st}=F_{st}.

3) For each vertex t in the V_1, the similarity between s and t is set as $\max_{k\in V_2}(F_{sk})$.

2. Construction of Affinity Matrix with Max-Flow-Based Similarity Measure

In this subsection, we present the implementation of constructing the affinity matrix by the max-flow-based similarity measure in detail. Given a dataset with n points denoted by $D=\{x_1,x_2,\cdots,x_n\}$, the affinity matrix is constructed by the following steps.

1) Construct the weighted graph G by local similarity.

We first compute the pair similarity with a general similarity measure, such as the Gaussian kernel function. To avoid confusion, this similarity is referred to as the local similarity. There are various similarity measures for two data points. We adopt the self-tuning Gaussian kernel method [12] due to its ability to deal with multiscale data. The local similarity s_{ij} between x_i and x_j is computed as follows:

$$s_{ij}=\begin{cases}\exp(\dfrac{-\|x_i-x_j\|^2}{\delta_i\delta_j}) & \text{for } i\neq j,\\ 0 & \text{for } i=j,\end{cases} \tag{4}$$

where δ_i is the local scale parameter corresponding to x_i. Specifically, the δ_i in our method is computed as described in [12]: δ_i=$d(s_i, s_M)$, where s_M is the M-th closest neighbor of the point s_i.

Then, a similarity graph G=(V, E) is constructed as described in subsection II.2, which can be a k-NN graph or an ε-neighborhood graph. In this graph, every vertex corresponds to a point in the D and each edge is weighted by the local similarity (4) between the connected points.

2) Construct the new similarity graph by the max-flow-based similarity measure.

For each point, pair x_i and $x_{j,}$ compute the similarity between x_i and x_j by Algorithm 1, denoting F_{ij}. Then, construct the new similarity graph $G'=(V',E')$, where the edge connecting v_i and v_j in G' is weighted by F_{ij}.

The new affinity matrix $\mathbf{A}'$ is defined on the new similarity graph G' as follows:

$$\mathbf{A}'_{ij}=\begin{cases}\max(F_{ij},F_{ji}) & i\neq j,\\ 0 & i=j,\end{cases} \tag{5}$$

Since the maximum value of the elements in $\mathbf{A}'$ is often larger than 1, $\mathbf{A}'$ can be normalized into the range from 0 to 1. Figures 1(d) and 1(e) show the affinity matrices constructed with the proposed similarity measure and Gaussian kernel function-based similarity measure, respectively. Compared with the Gaussian function-based similarity measure, the similarity for long-distance pairs in the same cluster (manifold) is not negligible in the proposed measure, which helps to constructs the block-diagonal matrix. According to the theoretical results using the matrix perturbation theory [11], spectral clustering can produce satisfactory results when the matrix is block-diagonal. As mentioned in [16], although the band-diagonal matrix constructed by the Gaussian function obtains a desirable result in spectral clustering, it depends on the parameter δ. It is difficult to select an appropriate parameter to construct a good band-diagonal matrix. We will illustrate in subsection IV.6 that the matrix by the proposed method is less sensitive to the parameter than the one by the Gaussian kernel function.

IV. Experiments

To evaluate the effectiveness of the max-flow-based similarity measure, which we will hereafter refer to as FLOW, we conduct experiments on synthetic and real datasets including a comparison with other state-of-the-art similarity measures for the affinity graph, including the locally scaled Gaussian kernel function (TUNING) [12], the path-based similarity (PATH) [14], the ranking on manifolds (ROM) [10], and the amplified commute kernel (ACK) [20]. The affinity graphs constructed by different measures are used in the spectral clustering algorithm [11] to evaluate the effectiveness.

In this section, we adopt the normalized mutual information (NMI) as the performance metric, which is widely employed to evaluate the clustering result [13], [25]. Supposing **X** and **Y** denote two random variables, the NMI is defined as

$$\mathrm{NMI}(\mathbf{X},\mathbf{Y})=\frac{I(\mathbf{X},\mathbf{Y})}{\sqrt{H(\mathbf{X})H(\mathbf{Y})}}, \tag{6}$$

where I(**X**, **Y**) is the mutual information between **X** and **Y**, and H(**X**) is the entropy of **X**. When evaluating the performance of clustering algorithms, the NMI is computed by regarding the clustering result and true class label as two random variables. Specifically, given a dataset with n points, the true classes and clustered classes are represented by $\{C_1^l,C_2^l,\cdots,C_k^l\}$ and $\{C_1^h,C_2^h,\cdots,C_{k'}^h\}$, where k and k' denote the number of true classes and the number of clustered classes, respectively. The NMI is computed as

$$\mathrm{NMI}=\frac{\sum_{i=1}^{k}\sum_{j=1}^{k'}n_{ij}\log(\dfrac{nn_{ij}}{n_i^l n_j^h})}{\sqrt{(\sum_{i=1}^{k}n_i^l\log\dfrac{n_i^l}{n})(\sum_{j=1}^{k'}n_j^h\log(\dfrac{n_j^h}{n})}}, \tag{7}$$

where n_i^l and n_j^h denote the number of points in C_i^l and C_j^h, and n_{ij} denotes the number of points belonging to C_i^l and C_j^h. The range of NMI is from 0 to 1, and, the larger the NMI, the better the clustering result.

In the following experiments, the parameters of TUNING, PATH, ROM, and ACK are set as suggested in their related papers. As for FLOW, we discuss the sensitivity of parameters in subsection IV.6. Since the computations of PATH, ROM, ACK, and FLOW are all based on a weighted graph, for a fair comparison, identically weighted graphs are used in the four methods. Thus, the sigmas in PATH, ROM, and ACK are the same as in FLOW. Additionally, the parameter (α) in ROM is set to 0.99, as in [5]. The parameter for defining the neighborhoods in PATH is selected so that the neighborhood is just large enough to include at least two neighbors in each neighborhood [14]. As for TUNING, the only hyper-parameter *M* is set to 7, as suggested in [12]. As in the literature [10], [11], [13], [14], the number of clusters is an input of the spectral clustering algorithm.

1. Synthetic Dataset

To illustrate the efficacy of the proposed method, the three synthetic datasets shown in Figs. 2(a) through 2(c) are used in the experiments. The three datasets, two-circle, three-spiral, and circle-Gaussian, were used in previous work to evaluate the performance of spectral clustering [11], [13], [14].

The first two datasets are well clustered by all five methods, with a result as that in Figs. 2(a) and 2(b), so that the result of five methods is omitted. The circle-Gaussian dataset contains two clusters with Gaussian distribution and one cluster with a circle shape. Unlike the first two datasets, the circle-Gaussian dataset is not well separated because the points around the Gaussian clusters tend to connect the three clusters. The results on the circle-Gaussian dataset are shown in Fig. 2(c). The PATH and FLOW can separate the three clusters well, and fewer points are incorrectly clustered by FLOW than PATH. However, the other methods fail to find correct clusters on the circle-Gaussian dataset. As shown in the results, though TUNING separates the data with Gaussian distribution, it cannot help spectral clustering algorithms to find the circle cluster with an elongated structure. The ACK and ROM can find the elongated structure to a certain extent, with some data clustered incorrectly. The similarity measures with global information (PATH, ROM, ACK, and FLOW) are more beneficial to cluster data with a nonlinear elongated structure than the Gaussian kernel-based similarity measure (TUNING). It is also seen that the proposed measure, FLOW, can better help spectral clustering to find elongated structure and is insensitive to noise.

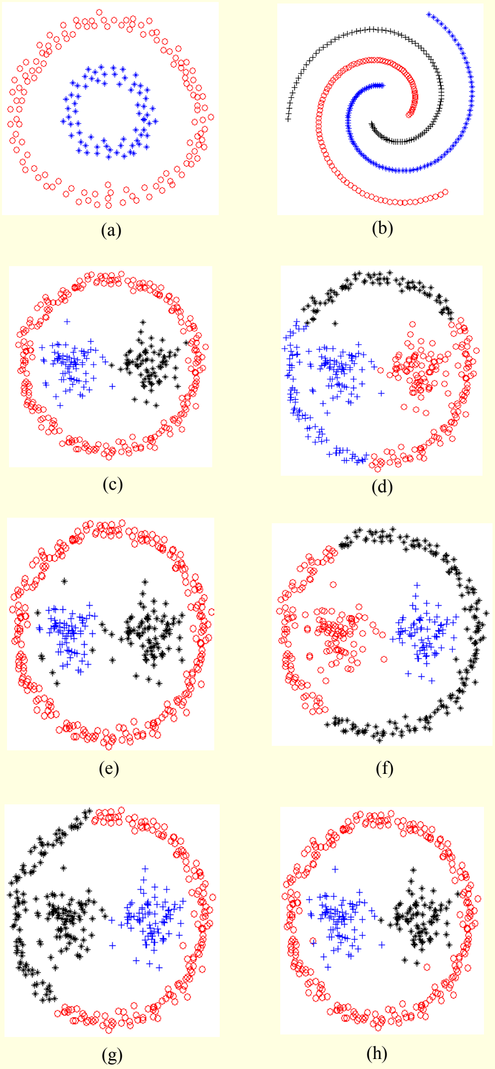

Fig. 2. Three synthetic datasets and some results on this data: (a) two-circle; (b) three-spiral; (c) circle-Gaussian; (d) TUNING, NMI=0.43; (e) PATH, NMI=0.88; (f) ROM, NMI=0.68 ; (g) ACK, NMI=0.64; and (h) FLOW, NMI=0.98.

Table 1. Five datasets from UCI.

Dataset	Iris	Wine	WDBC-1	WDBC-1	Yeast
Number of instances	150	178	569	699	1,484
Number of features	4	13	30	9	8
Number of clusters	3	3	2	2	10

2. UCI Dataset

In this subsection, we conduct the experiments on the five

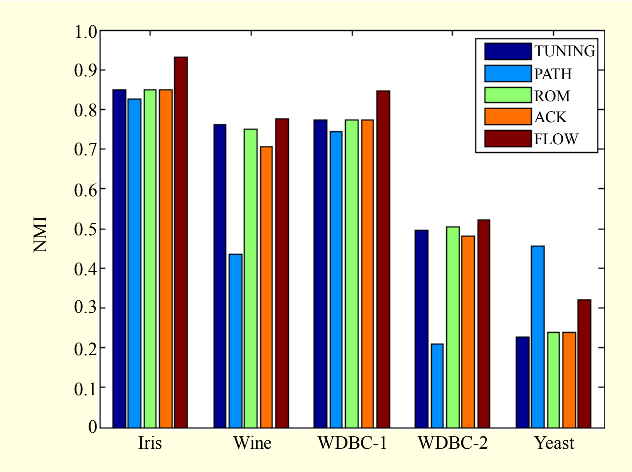

Fig. 3. Clustering results on five UCI datasets.

UCI datasets that have been on the most-downloaded list since 2007[1]. The characteristics of the datasets are shown in Table 1. As a preprocessing step, the mean of each dimensionality is normalized to zero. The experiment results on the UCI datasets are shown in Fig. 3. In terms of the NMI criteria, the proposed method, FLOW, outperforms the others on four datasets: Iris, Wine, WDBC-1, and WDBC-2. On Yeast, FLOW is the second best, inferior to PATH.

3. MNIST Dataset

The MNIST dataset of handwritten digits[2] contains 10 digits with a total of 70,000 examples (Fig. 4). Every example is a 28×28 grayscale image, and the dimensionality is thus 784. To obtain a comparable result, in our experiments, the first 200 examples from each digit are used instead of randomly sampling 200 samples, as described in [14].

Each pair of the digits is used for clustering, with a total of 45 tests. Figure 5 summarizes the results. In most cases, the proposed method outperforms other methods. Specifically, TUNING, PATH, ROM, ACK, and FLOW achieve the best result in 0, 5, 5, 9, and 26 cases, respectively. The mean value and standard deviation of NMIs of different methods on the 45 tests are shown in Table 2. The proposed method obtains the maximum mean value and the minimum standard deviation of NMIs, showing that the proposed method has the best performance and is robust for most data. To estimate the robustness of all methods, the parameters of each method are fixed in all experiments.

Additionally, to evaluate the performance of our proposed method on the data with more clusters, we also select 200 images from each digit to form one subset with a total of 2,000 examples, which contains 10 classes and is denoted by Ten-Class-Digits. The results of the five methods on the subset are shown in Fig. 6. The result of our method is still better than the others.

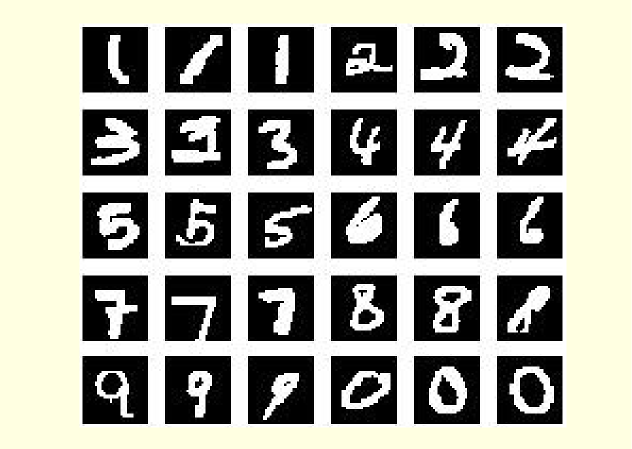

Fig. 4. Some examples from MNIST dataset.

4. UMIST and Yale Face Datasets

The UMIST face database [26] consists of 564 grayscale images from 20 subjects. The number of images per subject varies from 19 to 36, and the images of each subject contain a range of poses from the left profile to right profile. The size of each image is 220×220. The Yale faces dataset [27] consists of 165 grayscale images from 15 individuals, in which there are 11 images per subject and each image is different in facial expression or configuration. In our experiments, we down-sample each image to 20×20 pixels and represent it as a vector in $\boldsymbol{R}^{400}$. The set of the images per subject is taken as a cluster in the cluster experiments. The results on two face datasets are given in Fig. 6. Although the overall result of FLOW is not the best, compared with the other methods, FLOW still achieves good results in two cluster tasks.

5. Comparison of Affinity Matrices

To illustrate intuitively the merit of our proposed method for spectral clustering, we compare the affinity matrices constructed by different similarity measure. As described in the NJW algorithm, the result of the spectral clustering is obtained from the K largest eigenvectors. So, through the K largest eigenvectors, we can assess the clustering ability of one affinity matrix. Taking the digit pair {7, 9}, for example, the affinity matrices constructed by TUNING, ACK, and FLOW and the corresponding largest eigenvectors are shown in Fig. 7. The three leading principal components of the digit pair indicate that the two digits are not distributed with the compact shape

[1] The UCI datasets are available at http://archive.ics.uci.edu/ml/. Please note that we only consider datasets that do not have any categorical features.

[2] The MNIST dataset is available at http://yann.lecun.com/exdb/mnist/.

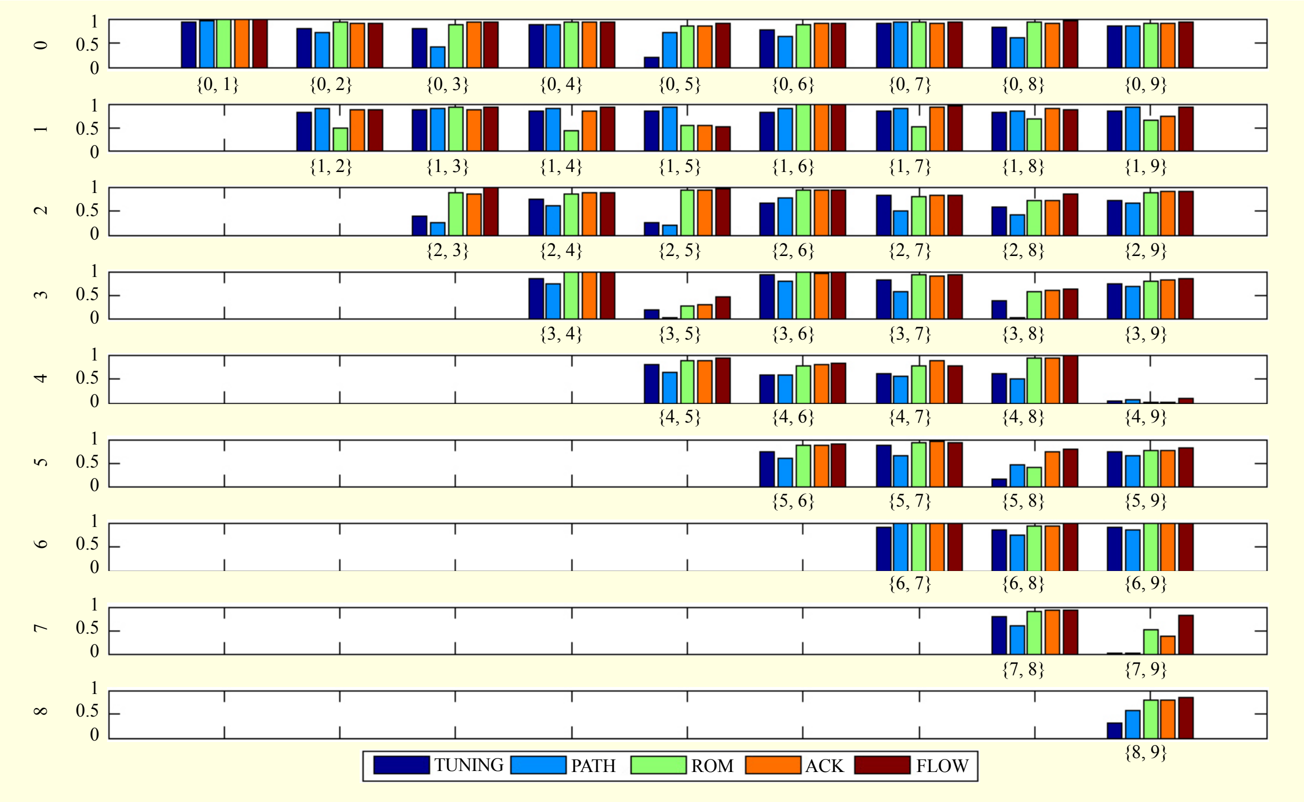

Fig. 5. Result of 45 tests on all pairs of 10 digits.

Table 2. Mean and standard deviation of NMIs of different spectral clustering methods on 45 tests.

	TUNING	PATH	ROM	ACK	FLOW
Mean	0.68	0.63	0.78	0.82	0.87
Standard deviation	0.2593	0.269	0.21	0.193	0.163

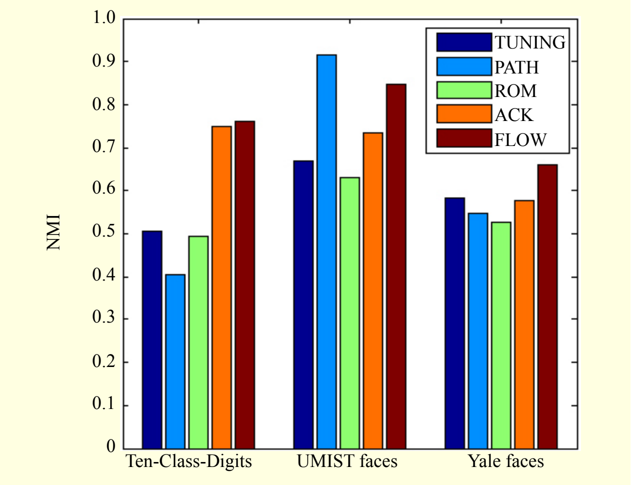

Fig. 6. Result on Ten-Class-Digits, UMIST face dataset and Yale face dataset.

and are not separated well, as shown in Fig. 7(a). In the affinity matrix by FLOW, there are less pairs from different digits with significant similarity than the ones by TUNING and ACK. Regarding perturbation [11], it means that there is a smaller perturbation to the ideal affinity matrix, which is helpful for the spectral algorithm. Through their two largest eigenvectors, we can see that the eigenvectors of the matrix constructed by TUNING have little information for distinguishing the two kinds of digits; however, the ones by the proposed method can separate the two digits well. The ACK method attains better ability than TUNING but less than ours. The results of TUNING, ACK, and FLOW on the digit pair are 0.03, 0.38, and 0.82, respectively. The results show that FLOW can significantly help the spectral clustering to improve the ability to cluster the data with a complex shape.

6. Sensitivity Analysis of Parameters

There are two parameters to set in the proposed similarity measure: the local parameters δ_i in (4) and K for constructing the k-NN graph in (2). For the parameter δ_i, we compare the proposed method with the self-tuning spectral clustering, wherein a similar local scale parameter δ_i is used. The parameter δ_i is actually computed as its distance to the M-th neighbor. Therefore, we investigate how the parameter M

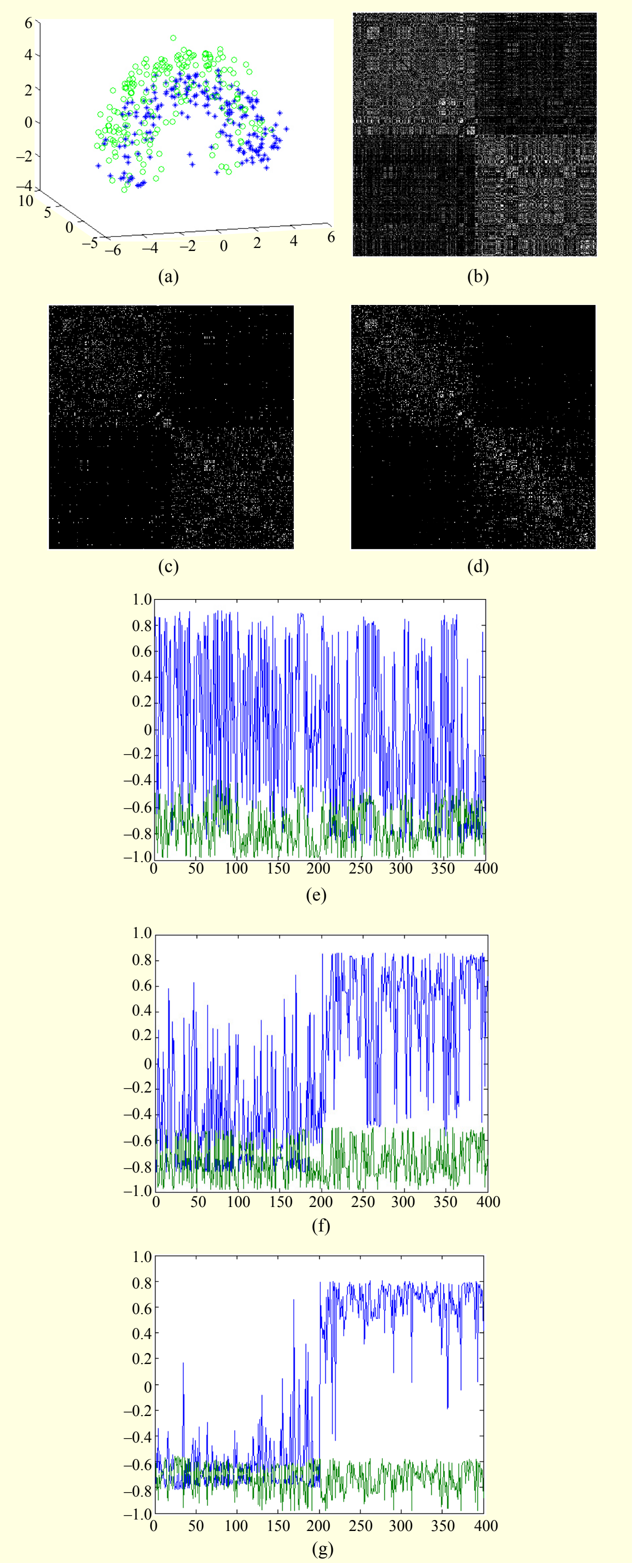

Fig. 7. Affinity matrices constructed by TUNING, ACK, and FLOW and the corresponding largest eigenvectors. (a) Three leading principal components of the digit pair. (b), (c), (d) Affinity matrices by TUNING, ACK, and FLOW, respectively. (e), (f), (g) Two largest eigenvectors corresponding to (b), (c), and (d).

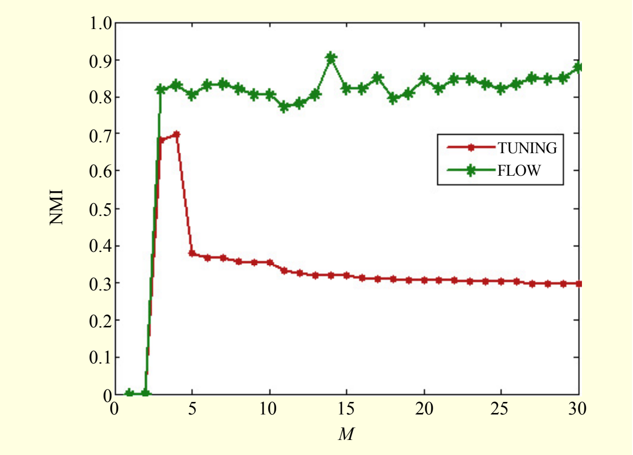

Fig. 8. Curves of NMI with varying M for parameter δ_i on digit set ("8" and "9").

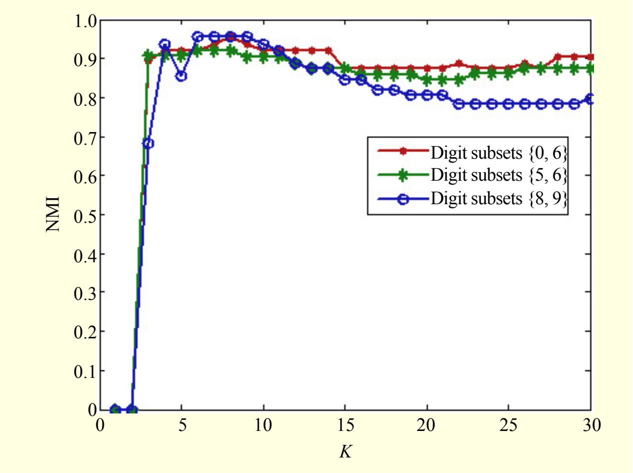

Fig. 9. Curves of NMI with varying K for t k-NN graph.

affects the performance of the spectral clustering algorithm by FLOW and TUNING, particularly on the digits "8" and "9." Figure 8 shows the change of the NMIs of two methods with M varying from 3 to 30. The FLOW achieves a stable desirable result in a large scope of M, while TUNING only achieves such in a small range. The result indicates that FLOW is much more robust to the parameter δ_i than TUNING.

We also conduct experiments to evaluate the impact of K in (2). Figure 9 shows the NMI of FLOW when varying K on the digit pairs {0, 6}, {5, 6}, and {8, 9}, showing that the proposed method is not sensitive to the parameter of K in (2).

In the above experiments, the parameters M and K are simply set as 20 and 10, respectively, for digit and face experiments and set as 30 and 20, respectively, for others.

V. Conclusion

In this paper, we proposed a new similarity measure based on the maximum flow between data points. Unlike other similarity measures based on the Gaussian kernel function, the new similarity measure can effectively reflect the global relationship between points and meanwhile satisfy the local consistency requirement. It can work on nonlinear and elongated structures and is robust to noise, as can be seen from the experiments on the synthetic and real datasets. It outperformed state-of-the-art similarity measure methods for spectral clustering on most datasets in our experiments.

References

[1] C. Alzate and J.A.K Suykens, "Multiway Spectral Clustering with Out-of-Sample Extensions through Weighted Dernel PCA," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 2, 2010, pp. 335-347.

[2] T.H. Kim, K.M. Lee, and S.U. Lee, "Learning Full Pairwise Affinities for Spectral Segmentation," *CVPR*, 2010, pp. 2101-2108.

[3] M. Li et al., "Time and Space Efficient Spectral Clustering via Column Sampling," *CVPR*, 2011, pp. 2297-2304.

[4] K. Jain, "Data Clustering: 50 Years beyond K-Means," *Pattern Recog. Lett.*, vol. 31, 2010, pp. 651-666.

[5] J.B. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.* vol. 22, no. 8, 2000, pp. 888-905.

[6] K. Ersahin, I.G. Cumming, and R.K. Ward, "Segmentation and Classification of Polarimetric SAR Data Using Spectral Graph Partitioning," *IEEE Trans. Geosci. Remote Sensing*, vol. 48, no. 1, 2010, 164-174.

[7] C.J. Alpert, A.B. Kahng, and S.Z. Yao, "Spectral Partitioning with Multiple Eigenvectors," *Discrete Appl. Mathematics*, vol. 90, no. 1-3, 1999, pp. 3-26.

[8] F.R. Bach and M.I. Jordan, "Learning Spectral Clustering, with Application to Speech Separation," *J. Mach. Learn. Res.* vol. 7, 2006, pp. 1963-2001.

[9] F.R. Bach and M.I. Jordan, "Spectral Clustering for Speech Separation," *Automatic Speech and Speaker Recognition: Large Margin and Kernel Methods*, J. Keshet and S. Bengio, Eds., New York: John Wiley & Sons, Inc., 2008, pp. 221-250.

[10] T. Xia et al., "On Defining Affinity Graph for Spectral Clustering through Ranking on Manifolds," *Neurocomputing*, vol. 72, 2009, pp. 3203-3211.

[11] A.Y. Ng, M.I. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm," *NIPS*, 2001, pp. 849-856.

[12] L. Zelnik-Manor and P. Perona, "Self-Tuning Spectral Clustering," *NIPS*, 2004, pp. 1601-1608.

[13] X. Zhang, J. Li, and H. Yu, "Local Density Adaptive Similarity Measurement for Spectral Clustering," *Pattern Recog. Lett.*, vol. 32, no. 2, 2011, pp. 352-358.

[14] H. Chang and D.Y. Yeung, "Robust Path-Based Spectral Clustering," *Pattern Recog.*, vol. 41, no. 1, 2008, pp. 191-203.

[15] U. Luxburg, "A Tutorial on Spectral Clustering, Statistics Computing," vol. 17, no. 4, 2007, pp. 395-416.

[16] I. Fischer and J. Poland, "Amplifying the Block Matrix Structure for Spectral Clustering," *Proc. 14th Annual Mach. Learning Conf. Belgium Netherlands*, pp. 21-28.

[17] B. Fischer and J.M. Buhmann, "Path-Based Clustering for Grouping of Smooth Curves and Texture Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 4, 2003, pp. 513-518.

[18] B. Fischer, T. Zöller, and J.M. Buhmann, "Path Based Pairwise Data Clustering with Application to Texture Segmentation," *Proc. 3rd Int. Workshop Energy Minimization Methods Computer Vision Pattern Recog.*, Sophia Antipolis, France, 2001, pp. 235-250.

[19] L. Yen et al., "Clustering Using a Random Walk Based Distance Measure," *ESANN*, 2005.

[20] U. Luxburg and A. Radl, "Getting Lost in Space: Large Sample Analysis of the Commute Distance," *NIPS*, 2010, pp. 153-160.

[21] F. Fouss et al., "Random-Walk Computation of Similarities between Nodes of a Graph with Application to Collaborative Recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 3, 2007, pp. 355-369.

[22] M. Ester et al., "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *KDD*, 1996, pp. 226-231.

[23] L.R. Ford, Sr., and E. Fulkerson, *Flows in Networks*, Princeton, NJ: Princeton University Press, 1962.

[24] Y. Boykov and V. Kolmogorov, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, 2004, pp. 1124-1137.

[25] Z. Li et al., "Noise Robust Spectral Clustering," *ICCV*, 2007, pp. 1-8.

[26] D.B. Graham and N.M. Allinson, "Characterizing Virtual Eigensignatures for General Purpose Face Recognition," *Face Recognition: From Theory to Applications*, H. Wechsler et al., Eds., NATO ASI Series: Series F: Computer and Systems Sciences, vol. 163, 1998, pp. 446-456.

[27] Yale Face Database. http://cvc.yale.edu/

Jiangzhong Cao received his MS in signal and information processing from Guangdong University of Technology, Guangzhou, China, in 2006. He is currently a PhD candidate in the School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China, and a lecturer in the School of Information Engineering, Guangdong University of Technology. His research interests include computer vision and pattern recognition.

Pei Chen received a PhD in wavelets from Shanghai Jiaotong University, Shanghai, China, in 2001 and a PhD in computer vision from Monash University, Melbourne, Australia, in 2004. He is currently a professor with the School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China. He was a post-doctoral researcher with Monash University, a senior research engineer with Motorola Labs, Shanghai, China, and a research professor with SIAT/CAS, Shenzhen, China. His current research interests include topics in computer vision and machine learning.

Yun Zheng received his BSc and MSc in electrical engineering from Harbin Institute of Technology, Harbin, China, in 2005 and 2008, respectively, and his PhD in computer science from Sun Yat-sen University, Guangzhou, China, in 2012. He is currently with Fujitsu Research, Beijing, China. His research interests include computer vision and pattern recognition.

Qingyun Dai received her PhD in communication and information systems from South China University of Technology, Guangzhou, China. She is currently a professor in the School of Information Engineering, Guangdong University of Technology, Guangzhou, China. She was a post-doctoral researcher with Sun Yat-sen University, Guangzhou, China. Her current research interests include topics in image processing, RFID, and the Internet of Things.