Pattern Recognition 45 (2012) 474–486

Contents lists available at ScienceDirect

Pattern Recognition

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

Fast affinity propagation clustering: A multilevel approach

Fanhua Shang [a,*], L.C. Jiao [a], Jiarong Shi [a], Fei Wang [b], Maoguo Gong [a]

[a] *Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Mailbox 224, No. 2 South TaiBai Road, Xi'an 710071, China*

[b] *Healthcare Transformation Group, IBM T.J. Watson Research Center at Hawthorne, NY, USA*

ARTICLE INFO

Article history:
Received 1 December 2009
Received in revised form
6 December 2010
Accepted 13 April 2011
Available online 18 May 2011

Keywords:
Clustering
Affinity propagation
Graph partitioning
Spectral clustering
Manifold structure

ABSTRACT

In this paper, we propose a novel **F**ast **A**ffinity **P**ropagation clustering approach (**FAP**). FAP simultaneously considers both local and global structure information contained in datasets, and is a high-quality multilevel graph partitioning method that can implement both vector-based and graph-based clustering. First, a new **F**ast **S**ampling algorithm (**FS**) is proposed to coarsen the input sparse graph and choose a small number of final representative exemplars. Then a density-weighted spectral clustering method is presented to partition those exemplars on the global underlying structure of data manifold. Finally, the cluster assignments of all data points can be achieved through their corresponding representative exemplars. Experimental results on two synthetic datasets and many real-world datasets show that our algorithm outperforms the state-of-the-art original affinity propagation and spectral clustering algorithms in terms of speed, memory usage, and quality on both vector-based and graph-based clustering.

1. Introduction

Clustering is one of the most important research topics in both machine learning and data mining communities. It arms at partitioning the data into groups of similar objects. From a machine learning perspective, what clustering does is to find the hidden patterns of the dataset in an unsupervised way, and the resulting system is usually referred to as a data concept. From a practical perspective, clustering plays an outstanding role in data mining applications such as scientific data exploration, information retrieval and text mining, Web analysis, computational biology, and many others [1,2].

An enormous variety of methods have been developed over the past several decades to solve clustering problems [3,4]. Recently, spectral clustering methods, which exploit pairwise similarities of data points, have been shown to be more effective than traditional methods such as K-means, and have aroused considerable interest in machine learning and data mining communities. Spectral clustering was firstly developed in the context of graph partitioning problems [5], where the problem is to partition a weighted graph into disjoint pieces, minimizing the sum of the weights of the edges linking the disjoint pieces. The final cluster assignments of the dataset can be achieved by optimizing some clustering criteria defined on the graph. The criteria of some spectral clustering methods are to optimize some cut value on an undirected graph, such as Normalized cut [6], Ratio cut [7], and min–max cut [8]. In the spectral clustering framework, one clustering problem is posed as a discrete optimization problem. This problem is generally intractable computationally, and approximate solutions are obtained by some relaxation procedures. With those relaxations, these criteria can be approximately optimized via eigen decompositions and the solutions are guaranteed to be global optimal. Unfortunately, when the number of data points (denoted as n) is large, the computational complexity of spectral decompositions can reach $O(n^3)$. Fowlkes et al. [9] proposed a Nyström approximation spectral clustering method to avoid calculating the whole affinity matrix. This approach represents a trade-off between accuracy and required computational time, where the sampling instances (also known as landmark points) are selected by a uniform random technique. Zhang and Kwok [10], Zhang et al. [11], and Yan et al. [12] apply K-means clustering algorithm to perform sampling as a pre-processing step. Thus, in both methods, only a very small informative affinity matrix is required to construct.

Affinity propagation (AP) [13] is an exemplar-based clustering method. It aims to identify data clusters and each cluster is represented by a data point called a cluster exemplar. AP algorithm is derived from a standard inference method on factor graph, and performs maximum a posteriori (MAP) inference using the max-product algorithm [14]. Although AP still does not guarantee global optimum, several experiments in [13] have shown its consistent superiority over the previous algorithms. However, AP clustering has a limitation that it is hard to determine the value of parameter 'preference', which can lead to a suboptimal clustering solution.

In this paper, we propose a novel clustering approach that simultaneously considers both local and global structure information contained in datasets. The proposed approach is a high-quality

[*] Corresponding author.
E-mail address: shangfanhua@yahoo.com.cn (F. Shang).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.04.032

multilevel graph partitioning method, and can implement both vector-based and graph-based clustering. Our approach first applies a new **F**ast **S**ampling algorithm (**FS**) to choose a small number of representative exemplars whose number is much less than all data points and larger than the number of clusters; secondly, the representative exemplars are assigned cluster labels by a proposed density-weighted spectral clustering method. In other words, the overall algorithm takes the following three phases: (1) coarsen the undirected graph by a FS algorithm to collapse the neighboring data points into a subset of representative exemplars; (2) run a density-weighted spectral clustering algorithm on the set of final representative exemplars; and (3) assign the cluster membership for each data point corresponding to its representative exemplar. So, we call our method **F**ast **A**ffinity **P**ropagation clustering algorithm (**FAP**).

To summarize, the main contributions of this work include

- We present a new global distance that can relatively elongate or shorten the distances among data points lying on different manifolds or on the same manifold, respectively. The proposed global distance is very robust against the noise and outliers, and can overcome the problem of short-circuiting in the shortest path algorithm [15].
- We propose a new FS algorithm to coarsen the input sparse graph and choose a small number of final representative exemplars. This algorithm is much faster than the original AP algorithm, and simultaneously considers both local and global structure information contained in datasets.
- Finally, we propose a novel FAP approach, which is a high-quality multilevel graph partitioning method. FAP algorithm can implement both vector-based and graph-based clustering, and can handle large-scale clustering problems.

The remainder of this paper is organized as follows: Section 2 presents a brief overview of AP and spectral clustering. A new global distance is presented in Section 3. A novel FAP algorithm is described in Section 4. Experimental results on two synthetic datasets and many real-world datasets are presented in Section 5. Section 6 is conclusions.

2. Related works

Before we go into the details of our FAP approach, first we briefly review some works that are closely related to this paper.

2.1. AP clustering

AP takes as input a collection of real-valued similarities among all data points, where the similarity $s(i,k)$ indicates how well the data point x_k is suited to be the cluster center for data point x_i. The similarity of each pariwise data points is set to a negative squared Euclidean distance: For points x_i and x_k:

$$s(i,k) = -\|x_i - x_k\|^2. \tag{1}$$

AP can be viewed by searching over valid configurations of the exemplars $Z = \{z(x_1),\ldots,z(x_n)\}$ to maximize the sum of similarities between each data point and its exemplar as follows:

$$\underset{Z}{\operatorname{argmax}} \sum\nolimits_{i=1}^{n} s\{x_i, z(x_i)\}. \tag{2}$$

The preferences P are important parameters in AP, which influence the final number of clusters. When P are larger, the number of identified exemplars is increased, otherwise, it is decreased. The values of input preferences are usually set to the median of the pairwise similarities. However, these values cannot lead to a suboptimal clustering solution in many cases, since the underlying manifold structure of the dataset is not considered.

2.2. Spectral clustering

Spectral clustering is a class of methods based on eigen decompositions of graph affinity matrices [16], and can stably detect non-convex patterns and linearly non-separable problems. Let a matrix $W \in \mathbb{R}^{n\times n}$ denote the affinity matrix for the graph $G=(V,E)$, with nodes V representing the n data points and edges E whose weights capture pairwise similarities between data points. Let D be a diagonal matrix, and the ith element on its diagonal line, d_i, denote the degree of a vertex $v_i \in V$:

$$d_i = \sum\nolimits_{j=1}^{n} w_{i,j}. \tag{3}$$

The goal of spectral clustering is to partition the data points into k disjoint clusters such that each point x_i belongs to one and only one cluster. Ng et al. [17] provided a k-way partitioning method. Those eigenvectors induce an embedding of the data points in a low-dimensional subspace. Finally, the K-means is used to assign the labels of all data points.

3. Local length and global distance

A meaningful measure of distance between pairs of data points plays an important role in clustering approaches. The idea of incorporating both local and global information into label prediction is inspired by the recent work on semi-supervised learning [18], which means: (1) nearby points are likely to have the same label; and (2) points on the same structure (usually referred to as a cluster) are likely to have the same label. Based on low density separation in semi-supervised classification [19], we present a local length and a new global distance.

Definition 1. A pairwise *local length* between each two data points of X is defined as

$$D_L(x_i,x_j) \triangleq e^{\rho\,\mathrm{dist}(x_i,x_j)} - 1, \tag{4}$$

where $\mathrm{dist}(x_i,x_j)$ is the Euclidean distance between x_i and x_j, and $\rho > 0$ is the flexing factor. In addition, the *local length* between two points can be elongated or shortened by adjusting the flexing factor ρ. According to the *local length*, we also define a new distance metric, called the *global distance*, which measures the distance between a pair of points by searching for the shortest path in the sparse graph.

Definition 2. The pairwise *global distances* among the sampling exemplars Y are defined as follows:

$$D_G(y_i,y_j) = \min_{p \in P_{i,j}} \sum_{k=1}^{|p|-1} D_L(p_k,p_{k+1}) + 1, i,j = 1,2,\ldots,m_1, \tag{5}$$

where $D_L(p_k,p_{k+1})$ is the *local length* between the node p_k and p_{k+1}, $P_{i,j}$ is the set of all paths connecting nodes y_i and y_j in the sparse graph of all data points, and m_1 is the number of the sampling exemplars Y.

The *global distance* satisfies the properties for a distance metric, i.e., $D(x_i,x_j)=D(x_j,x_i)$; $D(x_i,x_j) \geq 0$; $D(x_i,x_j) \leq D(x_i,x_k)+D(x_k,x_j)$ for all x_i,x_j,x_k, and $D(x_i,x_j)=0$ if and only if $x_i=x_j$. As a result, the *global distance* metric can measure the geodesic distance along the manifold, and achieve the aim of relatively elongating the distances among data points lying on different manifolds and simultaneously relatively shortening the distances among data points lying on the same manifold, and the *global distance* is robust against the noise and outliers, and can reflect the underlying manifold structures of datasets.

4. Fast AP algorithm

In this section, we propose a novel Fast Affinity Propagation (FAP) algorithm, which is a high-quality multilevel graph partitioning method, and can consider both local and global information contained in datasets. The framework of our FAP approach is similar to the multilevel algorithms of Karypis and Kumar [20], Dhillon et al. [21], and Wang and Zhang [22]. Fig. 1 shows a graphical overview of our multilevel framework. Below, we present our FAP algorithm in terms of its three phases: coarsening, exemplar-clustering and refinement.

4.1. Coarsening phase

We will propose a fast sampling algorithm as the coarsening phase of our FAP algorithm. The original AP algorithm takes the full similarity matrix to perform the information propagation. At each iteration step, there are generally n^2 data pairs whose responsibility and availability values need to be calculated, resulting in a computation complexity of $O(Tn^2)$, where T is the number of iterations. This greatly affects the computational cost of the algorithm especially when the number of data points is large. In our work, we first construct a sparse graph $G=(V,E)$, where the vertices V denote data points and the edges E contain parts of the pairwise edges between any two of the data points. It has been pointed out in [13] that the sparsity of the constructed graph will lead to faster calculation since the information propagation only needs to be performed on the existing edges.

4.1.1. Fast sampling algorithm

In this part, we propose a new fast sampling algorithm motivated by two factors: first, we pre-assume that whether adding or not an edge between the two data points that are far apart does not change the final result. Thus our algorithm may be boosted when it runs on the sparse graph; secondly, data points that serve as good exemplars locally may be candidates for exemplars globally [23]. So we consider a two-stage strategy to boost the original AP algorithm: in Stage I, we adopt a coarsening procedure to get local exemplars in the sparse graph using the sparse AP algorithm quickly, and empirically set $T_1=20$, where T_1 is the number of iterations; in Stage II, we only consider the exemplars from Stage I as the candidates for final representative exemplars. Here we should highlight that the pairwise distances among data points in Stage I are the *local lengths*, and only the local structure information contained in the dataset is considered. However, in Stage II the pairwise distances among the first stage exemplars Y are the *global distances*, which can reflect the underlying geometric structure of the data manifold. This complete fast sampling algorithm (FS) is listed in Algorithm 1 as follows:

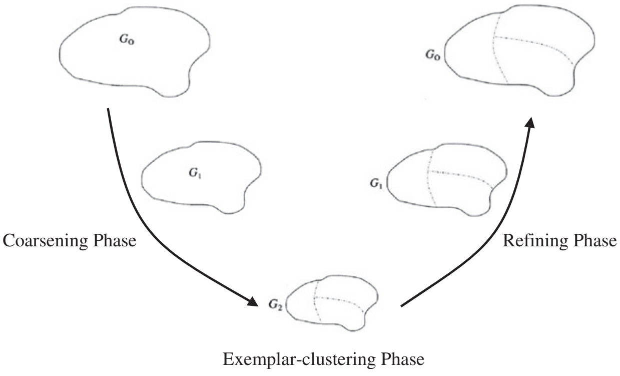

Fig. 1. Three phases of our FAP algorithm (for $k=3$).

Algorithm 1. Fast sampling algorithm (FS)

```
Input: Data points $X=\{x_1,\ldots,x_n\}$, the preferences $P_2 \in \mathbb{R}^{m_1\times 1}$
for Stage II, the size of the neighborhood $t$, and the flexing
factor $\rho$.
Output: Final representative exemplars, $Z=\{z_1,\ldots,z_{m_2}\}$.
1. Construct a $t$-nearest neighbor sparse graph.
2. In Stage I, the sparse graph is coarsened using the sparse AP
   algorithm, and $m_1$ exemplars $Y$ are identified, where the
   preferences $P_1^0$ are set to the median of the sparse pairwise
   similarities.
3. In Stage II, first compute the global distances among the $m_1$
   exemplars $Y$ by Eq. (5).
4. Refine the candidate exemplars to achieve the final $m_2$
   representative exemplars using the classical AP algorithm in
   Stage II, where the parameters $P_2$ are initialized with the
   median of the similarities among the candidate exemplars, $P_2^0$.
5. If $m_2$ is too big, let $P_2 \leftarrow P_2+P_2^0$, then run Step 4 until $m_2$
   becomes a relatively moderate value.
```

We apply the proposed FS algorithm to a simple toy dataset to illustrate its efficiency. This toy dataset consists of 3000 data points as shown in Fig. 2(a), and we aim to find 35 final representative exemplars among them. The classical AP algorithm using full similarity matrix took 261.42 s to achieve the final result as shown in Fig. 2(b). Our FS algorithm with initial neighborhood $t=50$ finds 351 candidates in Stage I in 16.09 s, and 35 final representative exemplars in Stage II in 8.12 s, as shown, respectively, in Fig. 2(c) and (d). We can see that the proposed FS algorithm is nearly 10 times faster than the classical AP algorithm. However, both the Stage II of FS and AP algorithm have a limitation that it is hard to determine the value of parameter 'preference', which can lead to a suboptimal clustering solution.

4.1.2. Determine the number of representative exemplars

The number of identified exemplars Y in Stage I, m_1, may be relatively large. How to determine the final representative

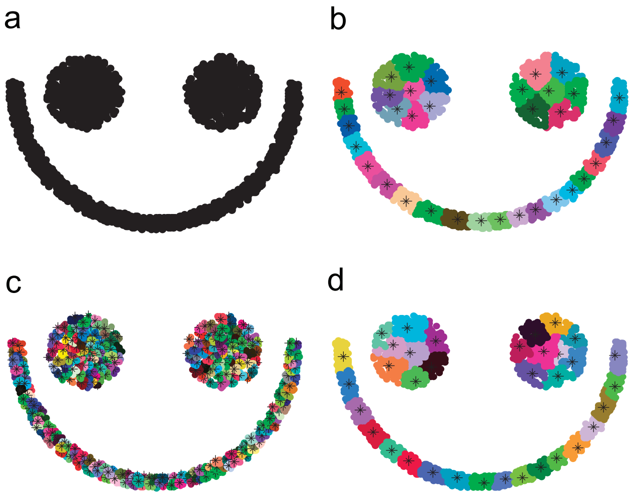

Fig. 2. A toy dataset. "*" indicates identified exemplars and colors indicate clusters. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

exemplars from those candidate exemplars Y is very important, and the number of the final representative exemplars m_2 can affect the ultimate performance of clustering. If the value m_2 is too small, the resulting FS algorithm may not be accurate. When m_2 is too large, the local structures of datasets would be hidden and the results will also be bad. Therefore, a relatively moderate value m_2 would give better results. Here, we apply a grid search style, which is to scan the search space of the parameters P_2 for finding the suboptimal number. The method only scans the search space of the parameters P_2 corresponding to a small range of the number, generally $\{9P_2^0,8P_2^0,\ldots,P_2^0\}$, where P_2^0 are the medians of the pairwise similarities among the final representative exemplars.

4.2. Exemplar-clustering phase

During the coarsening phase, we implement the proposed FS algorithm to achieve a small number of final representative exemplars, which can construct a smaller graph G_2 for the exemplar-clustering phase at the third level of FAP. In this phase, we propose a new density-weighted spectral clustering method to perform the final representative exemplars clustering [17]. Since the number of samples in each coarsened group represented by the corresponding exemplar is different, original spectral clustering methods are no longer appropriate.

Definition 3. A density-weighted affinity matrix is defined as

$$W(i,j)=\frac{|S_i|\,|S_j|}{n^2}\exp\left(-\frac{d_G^2(z_i,z_j)}{2\sigma_i\sigma_j}\right),\quad i,j=1,\ldots,m_2, \tag{6}$$

where $d_G(z_i,z_j)$ is the *global distance* between the representative exemplars z_i and z_j in the original sparse graph of all data points; the cluster sizes $|S_i|$'s, $i=1,\ldots,m_2$ for every group corresponding to the representative exemplar; σ_i is the local scale:

$$\sigma_i=d_G(z_i,s_K),\quad i=1,\ldots,m_2, \tag{7}$$

where s_K is the Kth neighbor of exemplar z_i. In this work, we use a single value of $K=[m_2/2k]$, where k is the number of clusters [24]. This complete Density-Weighted Spectral Clustering (DWSC) method is shown in the exemplar-clustering phase of Algorithm 2.

Algorithm 2. Fast AP clustering algorithm (FAP)

```
Input: Data points $X=\{x_1,\ldots,x_n\}$, the preferences $P_2\in\mathbb{R}^{m_1\times 1}$
for Stage II of FS algorithm, the size of the neighborhood $t$, and
the flexing factor $\rho$.
Output: Cluster set $C=\{C_1,\ldots,C_k\}$.
1. Coarsening phase
    • Apply Algorithm 1 to identify the final representative
      exemplars $Z=\{z_1,\ldots,z_{m_2}\}$, and count the cluster sizes
      $|S_i|$'s, $i=1,\ldots,m_2$ for each group corresponding to the
      representative exemplar $z_i$.
2. Exemplar-clustering phase (DWSC)
    • Compute the density-weight affinity matrix $W\in\mathbb{R}^{m_2\times m_2}$
      for $m_2$ representative exemplars $Z=\{z_1,\ldots,z_{m_2}\}$, given in
      Eq. (6), and the degree matrix $D$, given in Eq. (3).
    • Conduct the eigenvalue decomposition
      $D^{-1/2}WD^{-1/2}\phi_Z=\lambda_Z\phi_Z$ to find the eigenvectors
      $\phi_Z\in\mathbb{R}^{m_2\times k}$ corresponding to the $k$ largest eigenvalues $\lambda_Z$,
      and form the matrix $U\in\mathbb{R}^{m_2\times k}$ by normalizing each row
      vector of $\phi_Z$.
    • Execute $K$-means algorithm for $m_2$ row vectors of $U$, and
      assign $z_i$ to the cluster $C_l$ iff the $i$th row vector of $U$ is in
      the $l$th cluster.
3. Refinement phase
    • Achieve the assignments of all data points through the
      labels of their corresponding exemplars.
```

4.3. Refinement phase

During the final phase of the proposed FAP approach, all samples are assigned through the labels of their corresponding representative exemplars. The clustering in G_i induces a clustering in G_{i-1} as follows: if an exemplar in G_i is in cluster C_j, then all samples in G_{i-1} formed from that exemplar are in cluster C_j.

A key aspect of our FAP approach is how to select the subset of a dataset, and FAP would transfer the main computational burden from one kernel eigen-analysis to a combinatorial task of data sampling [25]. Our complete fast AP algorithm (FAP) is listed in Algorithm 2. Fig. 3 gives us an intuitive illustration of our FAP approach.

4.4. Complexity analysis of FAP

The time complexity of the proposed FAP algorithm is $O(n^2+T_1\text{tn}+T_2m_1^2+km_1^2\log n+m_2^3)$, together with the memory requirements of $O(tn)$ (t is the number of nearest neighbors, $\text{tn}\ll n^2$). In this work, we apply the naive nearest neighbors to construct the sparse graph, and its cost is $O(n^2)$. The time complexity of the proposed FS algorithm is $O(T_1\text{tn}+T_2m_1^2+km_1^2\log n)$, where T_1 and T_2 are the numbers of iterations in Stage I and Stage II, respectively, and empirically set $T_1=20$; $m_1\ll n$ is the number of identified exemplars in Stage I. The cost of computing the global distances among the m_1 identified exemplars in the sparse graph of all data points is $O(km_1^2\log n)$, then calculating the global distances among the final representative exemplars does not require any extra computational complexity. Finally, the time complexity of the density-weighted spectral clustering is $O(m_2^3)$, where m_2 is the number of final representative exemplars, and $m_2>k$, where k is the number of clusters.

5. Experiments

In this section, we present a set of clustering experiments on many datasets, including two synthetic datasets (Section 5.1), many real-world datasets (Sections 5.2–5.5), and an image object recognition dataset (Section 5.7), and discuss the sensitivity of the proposed FAP in relation to its parameters (Section 5.6). All experiments were performed with Matlab 7.1 on a Pentium- IV 3.20 GHz PC running Windows XP with 1 GB main memory.

5.1. Synthetic datasets

In this part, we considered two synthetic datasets with complex non-spherical shapes clusters, as shown in Fig. 4. These datasets represent some difficult clustering instances because they contain clusters of arbitrary shape and varying densities. First, we conducted some experiments to evaluate our FAP approach on a large-scale intertwined spirals dataset, as shown in Fig. 4(a). We illustrated the sampling results for the third levels of FAP using the proposed FS algorithm on the large-scale intertwined spirals dataset (each spiral has 20,000 points), as shown in Fig. 5(a). Clustering results of FAP on the dataset are shown in Fig. 5(b), and points that belong to the same cluster use the same color. In particular, we used $t=50$ nearest neighbors to construct the sparse graph. Our FAP approach correctly identifies the genuine clusters on the dataset. Besides, the computation

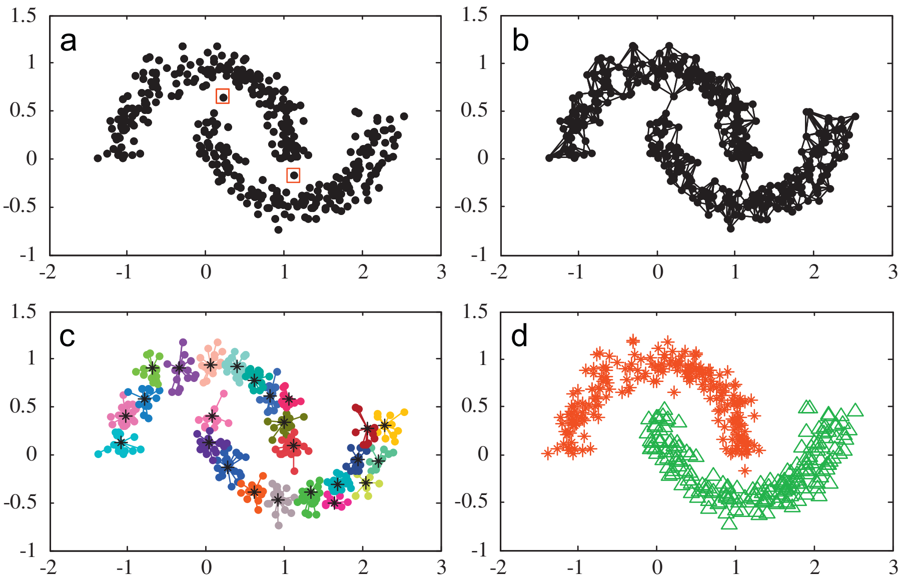

Fig. 3. Clustering results on a two-moon dataset with two bridge points using the proposed FAP. (a) A toy dataset with two bridge points, which are in red rectangles; (b) the 7-nearest-neighbor graph of the dataset; (c) the final representative exemplars are identified using fast sampling algorithm where "*" indicates identified exemplars and colors indicate clusters; (d) clustering results produced by the proposed FAP approach with the flexing factor $\rho=8$. We can see that the proposed FAP algorithm is very robust against the noise. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

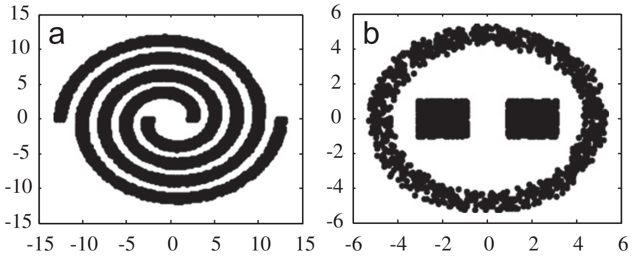

Fig. 4. Two synthetic datasets. (a) A large-scale intertwined spirals dataset has 40,000 points. (b) A toy dataset has 3000 points.

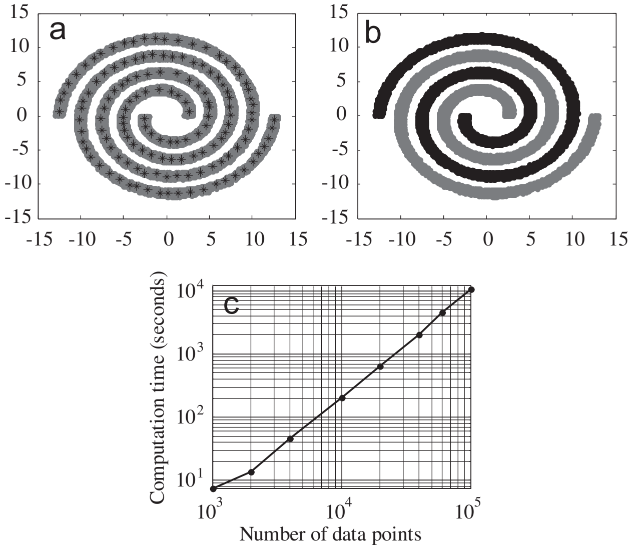

Fig. 5. The proceeding results of FAP on the large-scale intertwined spirals dataset. (a) The sampling results of the proposed FS algorithm. "*" indicates identified exemplars whose number is 143. (b) Clustering results produced by our FAP approach with the flexing factor $\rho=2$ (cluster 1 is shown in gray, cluster 2 in black). (c) Computation time of our FAP approach with respect to the number of the intertwined spirals dataset.

time of our FAP approach with respect to the number of the intertwined spirals dataset is shown in Fig. 5(c). The proposed FAP approach takes about 3 h to implement clustering on the dataset when the number of data points is 100,000. As illustrated in Fig. 5(c), the proposed FAP approach can handle large-scale clustering problems.

We also conducted some experiments to evaluate the proposed FAP on a toy dataset with additive outliers, as shown in Figs. 6(a) and 7(a). The toy dataset shown in Fig. 4(b) consists of two squared-clusters (each has 1000 points) and one circled-cluster (1000 points). In particular, 100 additive outliers are of varying distribution: they are scattered over the two squared-clusters and sparsely spread over the whole data, as shown in Figs. 6(a) and 7(a), respectively. We provided the sampling results for the third levels of our FAP approach using the proposed FS algorithm on the toy dataset with additive outliers, as shown in Figs. 6(b) and 7(b). Here, our aim is to show whether our FAP approach could be robust against the noise or outliers. As shown in Figs. 6(c) and 7(c), the three salient clusters are all satisfactorily identified. As a result, these performances confirm the robustness of our FAP approach against the noise and outliers. The affinity matrices built from the proposed global distances between every pair of the representative exemplars in Eq. (6) are shown in Figs. 6(d) and 7(d), where the identified representative exemplars are ordered such that all representative exemplars in the outer circled-cluster appear first, all representative exemplars in the left squared-cluster appear second, and all representative exemplars in the right squared-cluster appear finally. Note that this arrangement does not affect the clustering results but only for better illustration of the affinity matrices. We can see that the affinity matrices built from the proposed global distance exhibit clear block structures, meaning that each cluster becomes highly compact and the different clusters become far apart. We also illustrated the largest 10 eigenvalues of their normalized matrices, as shown in Figs. 6(e) and 7(e).

5.2. Compared algorithms and parameter settings

We compared the performance of the proposed FAP with eight other popular and related clustering methods, and the results are reported averaged over 50 independent runs.

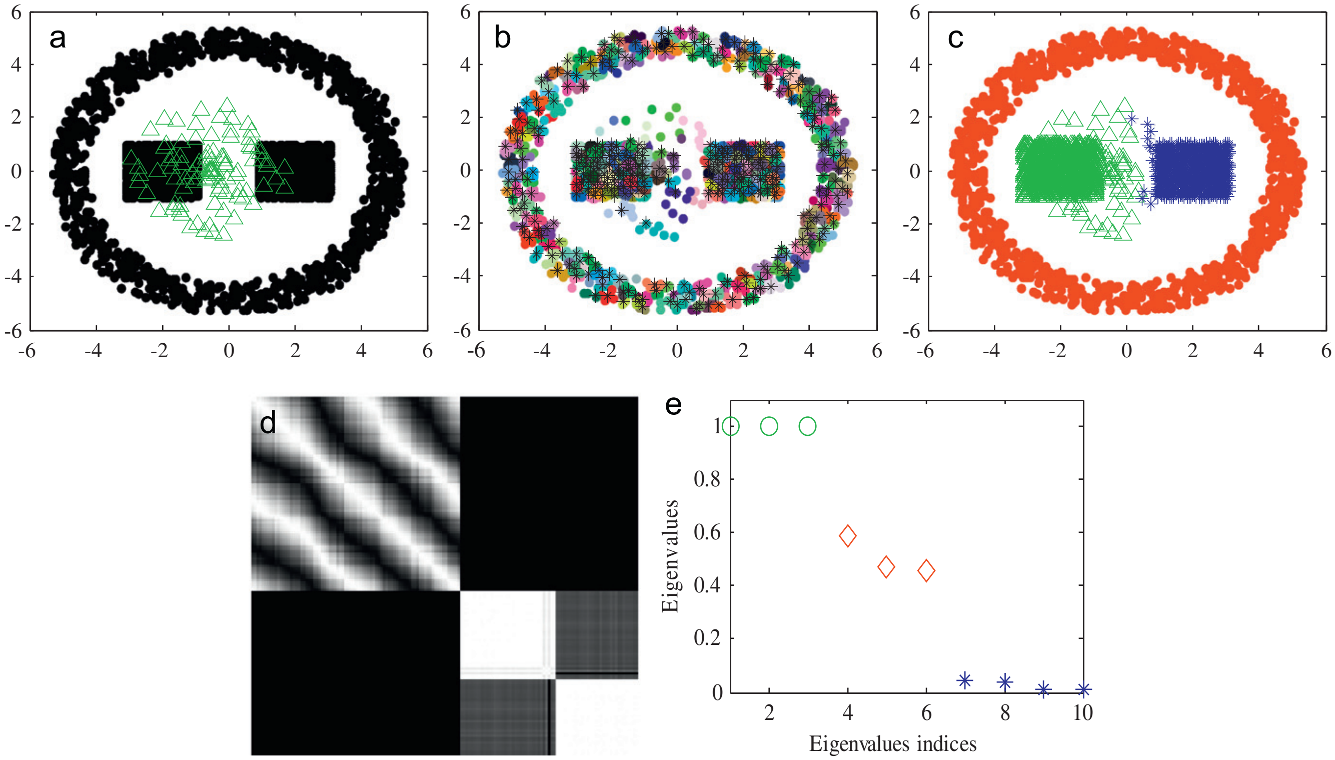

Fig. 6. Clustering results of the proposed FAP on the toy dataset with 100 additive outliers scattered over the two squared-clusters. (a) The toy dataset with 100 additive outliers scattered over the two squared-clusters. (b) The sampling results of the proposed FS algorithm. "*" indicates identified exemplars whose number is 427. (c) Clustering results produced by our FAP approach with the flexing factor $\rho=16$. (d) The affinity matrix W built from the global distances for the representative exemplars. (e) The largest 10 eigenvalues of the normalized W.

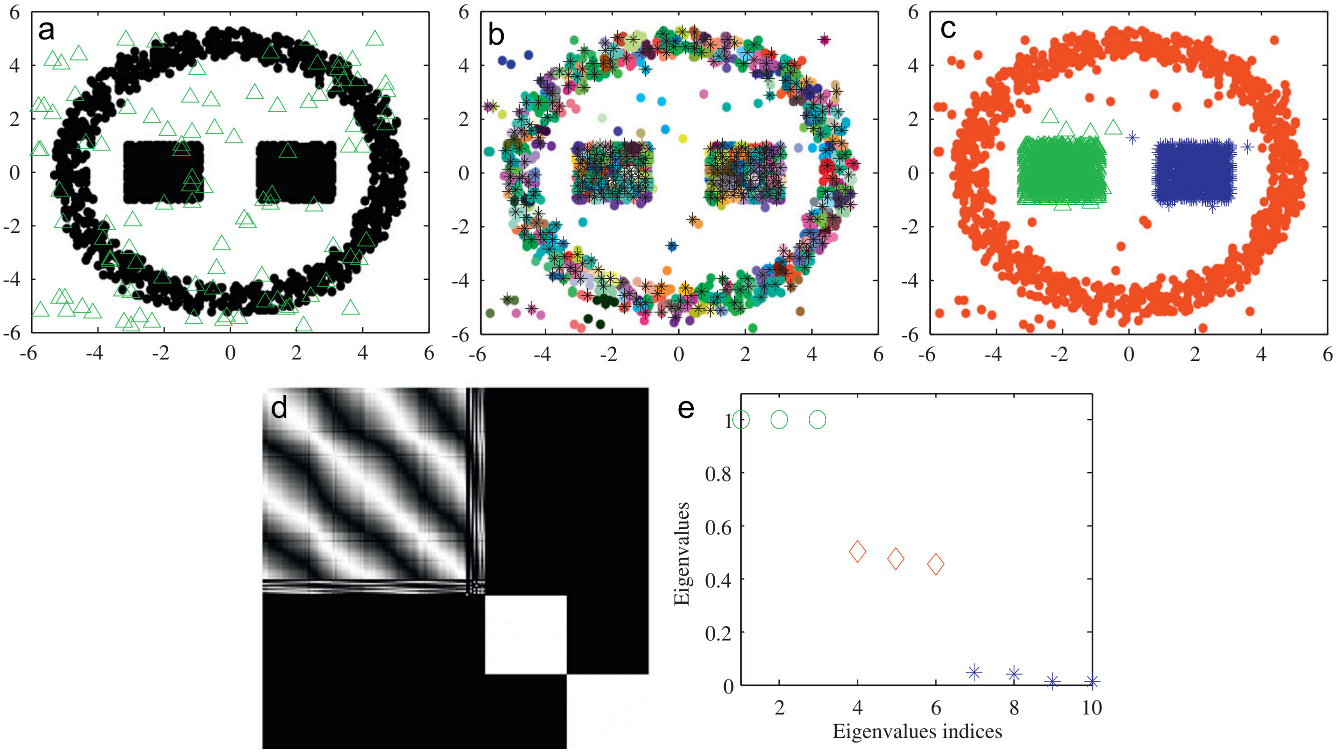

Fig. 7. Clustering results of the proposed FAP on the toy dataset with 100 additive outliers sparsely spread over the whole data. (a) The toy dataset with 100 additive outliers sparsely spread over the whole data. (b) The sampling results of the proposed FS algorithm. "*" indicates identified exemplars whose number is 409. (c) Clustering results produced by our FAP approach with the flexing factor $\rho=16$. (d) The affinity matrix W built from the global distances for the representative exemplars. (e) The largest 10 eigenvalues of the normalized W.

- *AP* [13]: In our implementations, the values of parameter P are set in the scope $[-n^2,0]$, and the appropriate value is searched using a bisection method.
- Spectral clustering (**SC**) [17] and self-tune spectral clustering (**SSC**) [26]. The similarities between pairwise points are computed using the standard Gaussian function, $W(x_i,x_j)=\exp(-\|x_i\text{-}x_j\|^2/2\sigma^2)$. The width of Gaussian function is searched from the grid of $\{4^{-3}\sigma_0,4^{-2}\sigma_0,4^{-1}\sigma_0,\sigma_0,4^1\sigma_0,4^2\sigma_0,4^3\sigma_0\}$, where σ_0 is the mean distance between any two samples in the dataset. However, the width of the Gaussian function in SSC is adaptively assigned.
- *NASC* [9]: The number of the random sampling examples is tuned from $\{2k,\ 4k,\ \ldots,20k\}$, and the width of Gaussian function is set the same as SC. To fairly compare with our approach, we also apply the proposed global distance in the implementation of NASC.
- Three fast spectral clustering methods: **KASP** [12], **KWASP** [10], and our approach **FWASC** [27]. In our implementations of KASP and KWASP, the number of partitioned groups using K-means is tuned from ,$\{2k,\ 4k,\ \ldots,20k\}$ and the Gaussian scale σ is set the same as SC.
- *The multilevel method Graclus*[1] *[21] (including three algorithms*: Kernel-K-means-NCut (**KKNC**), Kernel-K-means-RatioAssoc **(KKRA**), and Kernel-K-means-RatioCut (**KKRC**)). In our implementations, the RBF kernel is adopted, and its width is set the same as SC.
- For the proposed **FAP**, the size of the neighborhood t is tuned from $\{15,20,\ldots,100\}$, the flexing factor ρ is tuned from $\{1/64,1/32,1/16,1/8,1/4,1/2,1,2,4,8\}$, and the value of the input preference in Stage II of FS algorithm P_2 is generally searched from the grid $\{9P_2^0,8P_2^0,\ldots,P_2^0\}$.

5.3. Vector-based clustering

We used two categories of real-world datasets in our experiments. These datasets include

- *UCI Data*[2] : We performed experiments on 5 UCI datasets, including Wine, Balance, Segments, Pendigits, and Optdigits.
- *Image Data*: We performed experiments on four image datasets: MNIST digits [28], UCI Optdigits[2], YaleB3 [29], USPS digits[3] (description of these datasets can be found in the Appendix).

The basic information of those real-world datasets are summarized in Table 1.

Table 1
A summary of datasets.

Dataset	Size	Dimensions	Classes
Wine	178	13	3
Balance	625	4	3
Segment	2310	18	7
Optdigits389	1151	$8\times8=64$	3
Pendigit-test	3498	16	10
Pendigit-train	7494	16	10
YaleB3	1755	$30\times40=1200$	3
USPS0123	3588	$16\times16=256$	4
MNIST0123	24,754	$28\times28=784$	4
USPS-test	2007	$16\times16=256$	10
USPS-train	7291	$16\times16=256$	10

5.4. Evaluation metrics

In the experiments, we set the number of clusters equal to the true number of classes $\hat{C}$ for all the clustering algorithms. We use the following two popular evaluation metrics to evaluate the performance for all the clustering algorithms.

The first performance measure is the *Clustering Accuracy* (ACC), which is defined as [30]

$$\text{ACC}=\frac{\sum_{i=1}^{n}\delta(\hat{c}_i,map(c_i))}{n},\tag{6}$$

where $\hat{c}_i$ is the true class label and c_i is the obtained cluster label of x_i; $\delta(x,y)$ is the delta function, $\delta(x,y)=1$, if $x=y;\delta(x,y)=0$, otherwise, and $map(\ \bullet)$ is the best mapping function. The mapping function $map(\ \bullet)$ matches the true class label and the obtained cluster label, and the best mapping is solved by Hungarian algorithm [31]. A larger ACC indicates a better clustering performance.

Another evaluation metric that we adopt here is the *Normalized Mutual Information* (NMI) [32], which is calculated by

$$\text{NMI}=\frac{MI(\hat{C},C)}{\max(H(\hat{C}),H(C))},\tag{7}$$

where $\hat{C}$ is a set of the true labels, and C is a set of clusters obtained from the clustering algorithm; $MI(\hat{C},C)$ is the mutual information between $\hat{C}$ and C; $H(\hat{C})$ and $H(C)$ are the entropies of $\hat{C}$ and C, respectively. Given a clustering result, the NMI in Eq. (7) is estimated by

$$\text{NMI}=\frac{\sum_{i=1}^{|C|}\sum_{j=1}^{|C|}n_{i,j}\log\left(\frac{n\ n_{i,j}}{n_i\hat{n}_j}\right)}{\sqrt{\left(\sum_{i=1}^{|C|}n_i\log\frac{n_i}{n}\right)\left(\sum_{j=1}^{|C|}\hat{n}_j\log\frac{\hat{n}_j}{n}\right),}}\tag{8}$$

where n_i denotes the number of data contained in the cluster $C_i(1\le i\le|C|)$, $\hat{n}_j$ is the number of data belonging to the jth class $(1\le j\le|C|)$, and $n_{i,j}$ denotes the number of data that are in the intersection between the cluster C_i and the jth class. The higher the NMI score, the better the clustering quality.

5.5. Experimental results

The performances of those clustering algorithms on real-world datasets are shown in Tables 2 and 3, in which the best performances for each dataset are highlighted. From these tables, we can observe the following:

- In most cases, AP clustering method performs poorer than other sophisticated methods except SC and NASC. Since the distributions of most of these datasets, especially image datasets whose dimension are generally high, are commonly much more complicated than mixtures of spherical Gaussians. However, the performances of AP for the few datasets (e.g., Pendigit-test and YaleB3 datasets) are very good.
- SSC usually outperforms SC on these datasets, and SSC generally outperforms AP clustering method on the image datasets since there are clear nonlinear underlying manifolds behind those datasets, where the Euclidean distance cannot reflect the underlying geometric structure of data manifold. Note that the manifold structure has been experimentally shown to be useful in data clustering [30], and the data structural information is crucial for image clustering [33].
- Though the proposed global distance has also been used in the NASC approach, it performs generally much inferior, since this approach is an approximation method with a uniform random

[1] Software for the method is available at http://www.cs.utexas.edu/users/dml/Software/graclus.html.

[2] http://mlearn.ics.uci.edu/MLRepository.html.

[3] Available at http://www.kernel-machines.org/data.html.

Table 2
Performance comparison (mean ACC ± standard deviations) on UCI and image datasets.

Datasets	AP	SC	SSC	KKRA	NASC	KASP	KWASP	FWASC	FAP
Wine	0.7079	0.6524 ± 0.0471	0.7079 ± 0.0000	0.7135 ± 0.0012	0.6209 ± 0.0367	0.6745 ± 0.0042	0.6787 ± 0.0014	0.7022 ± 0.0320	**0.7234** ± 0.**0024**
Balance	0.5520	0.5143 ± 0.0120	**0.6410 ± 0.0106**	0.5980 ± 0.0075	0.5129 ± 0.0741	0.5169 ± 0.0158	0.5686 ± 0.0084	0.5982 ± 0.0115	0.6027 ± 0.0095
Segment	0.4944	0.4948 ± 0.0365	0.6347 ± 0.0503	0.6383 ± 0.0380	0.5785 ± 0.0360	0.6309 ± 0.0410	0.5295 ± 0.0446	0.6414 ± 0.0356	**0.6421 ± 0.0545**
Digits389	0.8836	0.8044 ± 0.0535	0.8810 ± 0.0000	0.9473 ± 0.0011	0.8609 ± 0.0332	0.9401 ± 0.0002	0.9389 ± 0.0006	0.9783 ± 0.0000	**0.9790 ± 0.0033**
Pendigit-test	0.7293	0.6952 ± 0.0357	0.6877 ± 0.0317	0.7213 ± 0.0294	0.6678 ± 0.0129	0.6959 ± 0.0195	0.6805 ± 0.0598	0.7427 ± 0.0273	**0.7456 ± 0.0318**
Pendigit-train	–	–	–	**0.7667 ± 0.0076**	0.7017 ± 0.0293	0.7179 ± 0.0589	0.7079 ± 0.0472	0.7381 ± 0.0384	0.7466 ± 0.0611
YaleB3	0.9869	0.9358 ± 0.1700	0.9634 ± 0.1155	**0.9962 ± 0.0000**	0.9522 ± 0.0427	0.9726 ± 0.0526	0.9733 ± 0.0057	0.9812 ± 0.0000	0.9928 ± 0.0046
USPS0123	0.5435	0.8107 ± 0.0548	0.7842 ± 0.0000	0.8691 ± 0.0162	0.8562 ± 0.1056	0.8436 ± 0.0432	0.8456 ± 0.0575	0.8618 ± 0.0172	**0.9024 ± 0.0068**
MNIST0123	–	–	–	0.8687 ± 0.0733	0.8531 ± 0.0866	0.8557 ± 0.0163	0.8618 ± 0.0241	0.8640 ± 0.0297	**0.8942 ± 0.0229**
USPS-test	0.5102	0.6327 ± 0.0426	0.6125 ± 0.0374	0.6338 ± 0.0429	0.6166 ± 0.0172	0.6287 ± 0.0348	0.6202 ± 0.0358	0.6645 ± 0.0320	**0.6942 ± 0.0574**
USPS-train	–	–	–	0.6884 ± 0.0262	0.6565 ± 0.0169	0.6417 ± 0.0315	0.6751 ± 0.0282	0.6747 ± 0.0419	**0.7045 ± 0.0486**

No result (‘–’) is reported for those algorithms that do not work.

Table 3
Performance comparison (mean NMI ± standard deviations) on UCI and image datasets.

Datasets	AP	SC	SSC	KKRA	NASC	KASP	KWASP	FWASC	FAP
Wine	0.4315	0.4158 ± 0.0265	0.4315 ± 0.0000	0.3802 ± 0.0026	0.2821 ± 0.0545	0.4050 ± 0.0001	0.4344 ± 0.0018	0.4426 ± 0.0083	**0.4457 ± 0.0037**
Balance	0.1554	0.1464 ± 0.0125	**0.2626 ± 0.0279**	0.2161 ± 0.0086	0.1336 ± 0.0949	0.1305 ± 0.0089	0.1894 ± 0.0098	0.1978 ± 0.0114	0.2046 ± 0.0109
Segment	0.4785	0.4940 ± 0.0268	0.5680 ± 0.0185	0.5711 ± 0.0159	0.5426 ± 0.0325	0.5658 ± 0.0137	0.5311 ± 0.0263	0.5727 ± 0.0139	**0.6408 ± 0.0538**
Digits389	0.6150	0.5264 ± 0.0359	0.6095 ± 0.0000	0.7935 ± 0.0082	0.6191 ± 0.0496	0.7601 ± 0.0001	0.7541 ± 0.0014	0.8940 ± 0.0000	**0.9023 ± 0.0120**
Pendigit-test	0.6950	0.6861 ± 0.0144	0.6799 ± 0.0126	0.6986 ± 0.0243	0.7121 ± 0.0017	0.6896 ± 0.0467	0.6807 ± 0.0147	0.7257 ± 0.0187	**0.7362 ± 0.0325**
Pendigit-train	–	–	–	0.7035 ± 0.0170	0.7139 ± 0.0125	0.7051 ± 0.0233	0.6816 ± 0.0105	0.7214 ± 0.0190	**0.7549 ± 0.0355**
YaleB3	0.9495	0.9193 ± 0.1189	0.9505 ± 0.0802	**0.9894 ± 0.0000**	0.9311 ± 0.0282	0.9191 ± 0.1266	0.9547 ± 0.1240	0.9485 ± 0.0000	0.9699 ± 0.0169
USPS0123	0.3456	0.5821 ± 0.0705	0.5473 ± 0.0000	0.7411 ± 0.0237	0.7335 ± 0.0741	0.6510 ± 0.0151	0.6542 ± 0.0308	0.6738 ± 0.0467	**0.7797 ± 0.0114**
MNIST0123	–	–	–	0.6852 ± 0.0201	0.6167 ± 0.0843	0.6277 ± 0.0086	0.6428 ± 0.0105	0.6598 ± 0.0246	**0.7122 ± 0.0150**
USPS-test	0.4396	0.5695 ± 0.0197	0.5650 ± 0.0207	0.6034 ± 0.0106	0.5988 ± 0.0150	0.5785 ± 0.0183	0.5730 ± 0.0150	0.5983 ± 0.0183	**0.7064 ± 0.0313**
USPS-train	–	–	–	0.6551 ± 0.0121	0.6200 ± 0.0091	0.6419 ± 0.0195	0.6373 ± 0.0120	0.6846 ± 0.0131	**0.6879 ± 0.0269**

No result (‘–’) is reported for those algorithms that do not work.

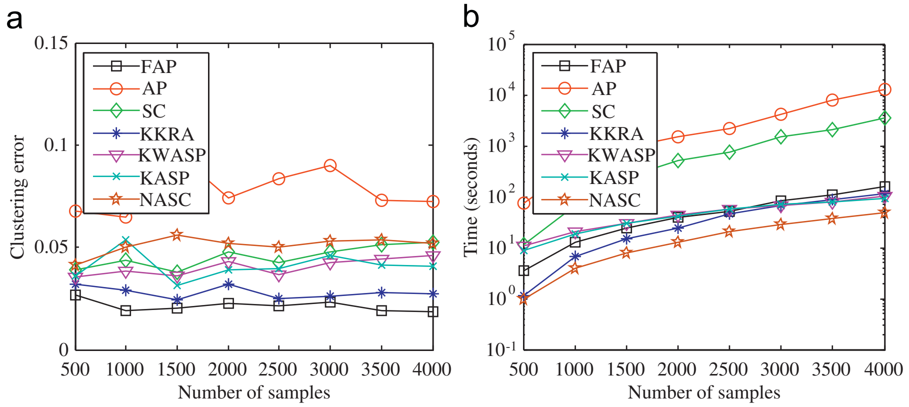

Fig. 8. Comparison of the clustering performances for SC, AP, KKRA, NASC, KASP, KWASP, and FAP on MNIST digits 6 and 8.

sampling technique. This observation confirms that the identified representative exemplars using the proposed FS algorithm have more information.

- KASP, KWASP, and FWASC algorithms are three fast algorithms for approximate spectral clustering, and they commonly outperform AP clustering method. KASP has similar performances as the SC approach. Furthermore, KWASP and FWASC sometimes outperform the SC and SSC approaches.
- As shown in Tables 2 and 3, two multilevel approaches (i.e., KKRA (we adopted KKRA as the representative) and our FAP) are generally better than the other approaches. It demonstrates that KKRA and our FAP can effectively make use of the local structure information contained in datasets. Besides, our approach FAP also considers the underlying global structure information from the global distances among the identified representative exemplars. As a result, our FAP approach generally outperforms another multilevel approach KKRA in terms of ACC and NMI.
- From Tables 2 and 3, we observe that the proposed FAP approach usually performs at least as good as the best of the other eight algorithms in terms of ACC and NMI. As expected, FAP usually performs better than the AP and SC methods because FAP considers both local and global underlying structure information contained in datasets. When the number of final representative exemplars is a relatively moderate value, the proposed FS algorithm has very good accuracy. Furthermore, the proposed global distances among the final representative exemplars can reflect the underlying manifold structures of datasets. As a result, the inter-cluster connections are relatively reduced and the within-cluster connections become relatively stronger. The proposed density-weighted spectral clustering algorithm can efficiently solve linearly non-separable problems.

Below, we used a specific example to demonstrate the efficiency of the proposed FAP. We chose digits 6 and 8 from the MNIST dataset, gradually increased the sample size of both digits, and examined the performance (here, clustering error[4] and time consumption) of our approach. For comparison, we also reported the performances of AP, SC, KKRA, NASC, KASP, and KWASP methods. As can be seen from Fig. 8(a), with the increase of the sample size, the performance of our FAP is very close to or even better than AP, SC, KKRA, NASC, KASP, and KWASP methods. From Fig. 8(b), our FAP and other four approaches including KKRA, NASC, KASP, and KWASP are much faster than SC and classical AP clustering methods. Particularly, the larger the sample size, the more obvious the improvement.

5.6. *Sensitivity in relation to parameters*

There are mainly three parameters in our FAP approach: the size of the neighborhood t, the value of the input preference in Stage II of the proposed FS algorithm P_2, and the flexing factor ρ. We conducted four experiments on the UCI Wine and Digits389, YaleB3, and USPS0123 datasets to test the sensitivity of the proposed FAP to the selection of these parameters, and the results are shown in Figs. 9 and 10. From these figures, we can clearly see that

- The proposed FAP approach is very robust to the size of neighborhood t. This observation confirms that the proposed global distance can overcome the problem of short-circuiting in the shortest path algorithm [15], and can also reflect the underlying manifold structures of datasets.
- The proposed FAP approach also is very robust to the value of the input preference in Stage II of FS algorithm P_2. Generally a too small or too large P_2 will lead to poor results. This is because when P_2 is too large, the resulting FS algorithm may not be accurate. When P_2 is too small, the local structures of datasets would be hidden and the results will also be bad.
- There is a similar situation for the flexing factor ρ to the algorithm parameter P_2. When the flexing factor ρ is too small or too large, the underlying global manifold structures of datasets would be destroyed.

In real applications, ground truth is generally not available for tuning the parameters of clustering approaches. Thus, the proposed FAP approach is very suitable for real clustering applications because FAP is not only accurate but also stable to the algorithmic parameters.

5.7. *Graph-based clustering*

In this part, we performed an experiment on an image object recognition dataset. We experimented with the Caltech-4 dataset,[5] which contains 1155 automobile, 1074 airplane, 450 face, and 800 motorcycle images [34]. Fig. 11 shows examples from this dataset. Here, we used uniform grid cells and SIFT descriptors extracted from a uniformly spaced grid on the image. Then, the pyramid match kernel [35], a kernel function between images, is used to generate a kernel matrix on the input sets of image features from each two images (the sets may have different

[4] Here, clustering error is defined as **1**−**ACC**.

[5] http://www.robots.ox.ac.uk/~vgg/data3.html.

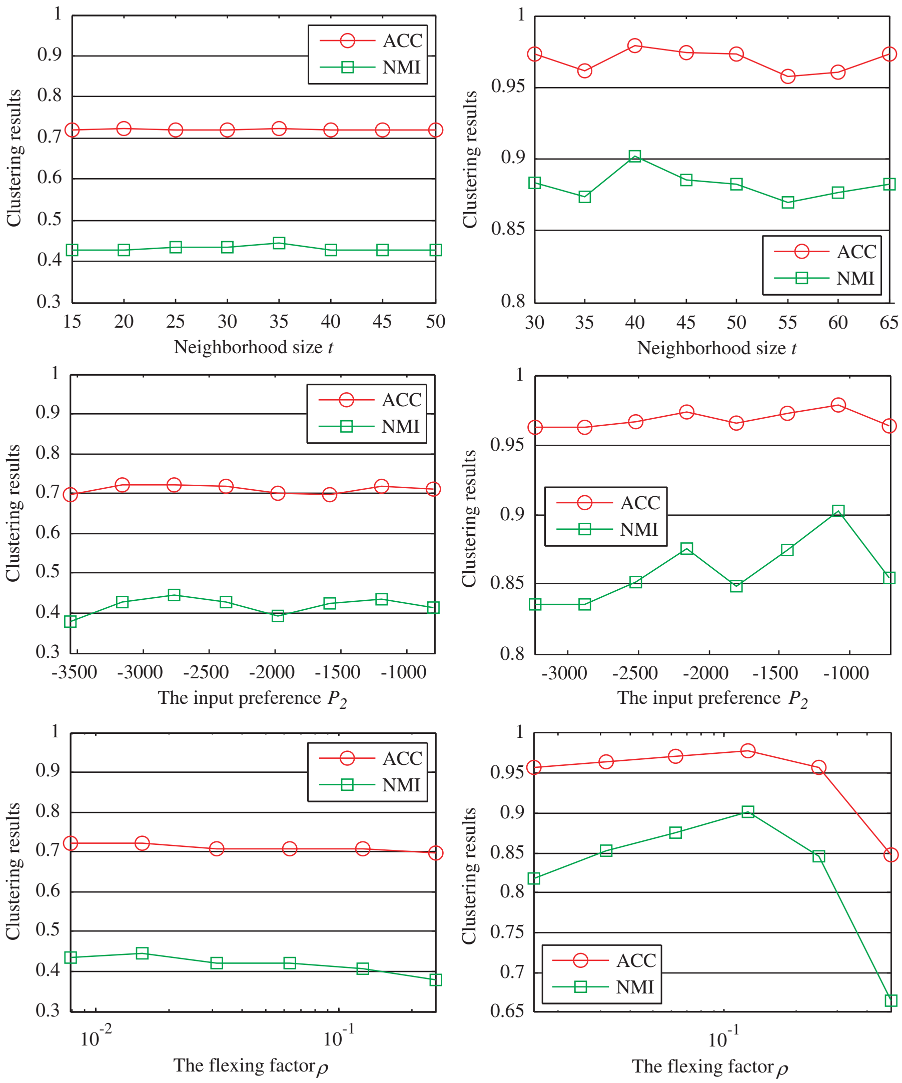

Fig. 9. Parameter sensitively testing results on **Wine** and **Digits389** datasets. First row: clustering results versus t plot. The parameter settings for Wine: $P_2=6P_2^0=-2767.33$, $\rho=1/64$ (left); for Digits389: $P_2=3P_2^0=-1079.50$, $\rho=1/8$ (right). Second row: clustering results versus P_2 plot. The parameter settings for **Wine**: $t=35$, $\rho=1/64$ (left); for Digits389: $t=40$, $\rho=1/8$ (right). Third row: clustering results versus ρ plot. The parameter settings for Wine: $t=35,P_2=6P_2^0=-2767.33$ (left); for Digits389: $t=40,P_2=3P_2^0=-1079.50$ (right).

cardinality). There is no explicit vector representation of this dataset—the kernel function defines a matrix of kernel function evaluations, which is then viewed as a dense graph.

We compared the graph-based clustering performance of the proposed FAP against various existing state-of-the-art methods: Kernel K-means (with random initialization), AP, SC [17], and the multilevel method Graclus [21]: KKNC, KKRA, and KKRC.

The results of graph-based clustering on the Caltech-4 dataset are shown in Table 4, in which the best performances are highlighted. Both FAP and Graclus outperform the spectral clustering, AP, and Kernel K-means methods (except the KKRC) in terms of ACC and NMI. The former two methods are multilevel methods, and can effectively make use of the local structure information contained in the dataset.

6. Concluding remarks

As two powerful clustering methods with various applications, AP and spectral clustering methods have received considerable attention in the last few years. However, spectral clustering methods have relied on eigenvector computation, which is very expensive for large datasets. AP has a limitation that it is hard to determine the value of parameter 'preference', which can lead to a suboptimal clustering solution. In this paper, we have designed a novel FAP approach that outperforms both spectral clustering and AP methods in terms of quality, speed, and memory usage. The proposed FAP approach is a high-quality multilevel graph partitioning method that can implement both vector-based and graph-based clustering, and has efficiently considered both local and global underlying structure information contained in datasets. Extensive experiments on two synthetic datasets and many real-world datasets show that the proposed FAP can obtain comparable performance with other existing clustering methods, and is very robust to its algorithm parameters.

There are a number of interesting potential avenues for future research in methods of selecting exemplars for semi-supervised clustering and classification. We would also like to incorporate the work of Basu et al. [36] to explore the exemplars selecting techniques of active learning. In addition, we would like to study how to determine the different preference values for each point in Stage I of the proposed FS algorithm according to the corresponding local distribution information contained in datasets.

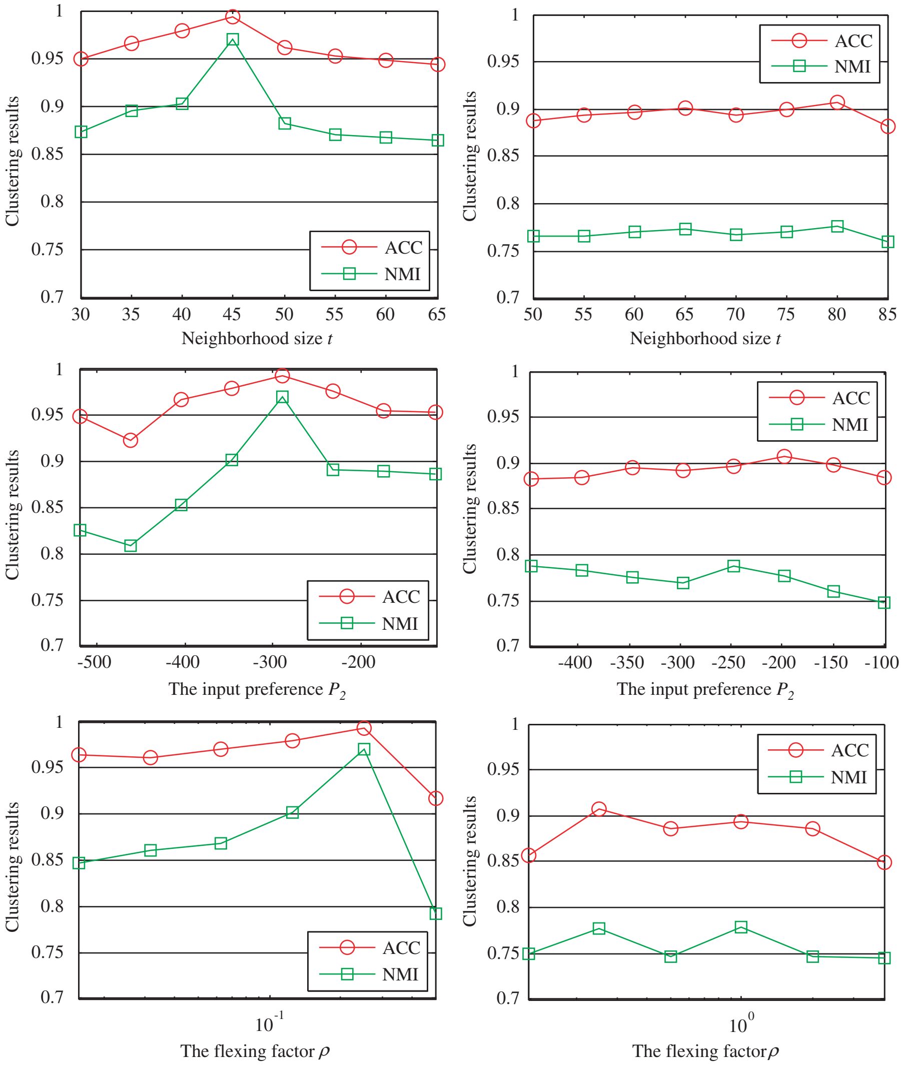

Fig. 10. Parameter sensitively testing results on **YaleB3** and **USPS0123** datasets. First row: clustering results versus t plot. The parameter settings for YaleB3: $P_2=5P_2^0=-288.68$, $\rho=1/4$ (left); for USPS0123: $P_2=4P_2^0=-198.32$, $\rho=1/4$ (right). Second row: clustering results versus P_2 plot. The parameter settings for YaleB3: t=45, $\rho=1/4$ (left); for USPS0123: t=80, $\rho=1/4$ (right). Third row: clustering results versus ρ plot. The parameter settings for YaleB3: t=45,$P_2=5P_2^0=-288.68$ (left); for USPS0123: t=80,$P_2=4P_2^0=-198.32$ (right).

Fig. 11. Example images from Caltech-4 dataset.

Table 4
Graph-based clustering results and the corresponding standard deviations on Caltech-4 dataset.

	Kernel *K*-means	AP	SC	Graclus			FAP
				KKNC	KKRA	KKRC	
ACC	0.5315 ± 0.0034	0.6183	0.6233 ± 0.0490	0.6915 ± 0.0904	0.7024 ± 0.0873	0.3475 ± 0.0541	**0.7155 ± 0.0217**
NMI	0.5908 ± 0.0071	0.5855	0.6263 ± 0.0430	0.6763 ± 0.0885	0.6827 ± 0.0762	0.0371 ± 0.0237	**0.6947 ± 0.0295**

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments and suggestions to significantly improve the quality of this paper. We also thank Dr. Liefeng Bo of University of Washington for numerous discussions and suggestions. This work is supported by the National Natural Science Foundation of China Nos. 60803097, 61003198, 61001202, 60970067; the

National Science and Technology Ministry of China Nos. 9140A07011810DZ0107, 9140A07021010DZ0131; the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) No. B07048; the Fundamental Research Funds for the Central Universities Nos. JY10000902001, K50510020001, JY10000902045; the National Research Foundation for the Doctoral Program of Higher Education of China No. 20100203120008.

Appendix

The **MNIST** database has a training set of 60,000 examples and a test set of 10,000 ones with the images as 28×28. **MNIST0123** is a subset from the training set, and consists of digits 0, 1, 2, and 3 with 5923, 6742, 5958, and 6131 examples, respectively.

Digits389 is a subset of the three classes {3, 8, 9} of the UCI handwritten digit recognition dataset from the UCI Machine—these three classes were chosen since distinguishing between sample handwritten digits from these classes visually is a difficult task.

YaleB3 used in [37] is a subset of the Yale Face Database B [29]. We use images of individuals 2, 5, and 10 and down-sample each image to 30×40 pixels. This gives us 1755 images with 1200 dimensions to work with.

USPS0123 is a subset from the USPS handwritten 16×16 digits dataset. The images of digits 0, 1, 2, and 3 are used in these experiments as four classes, and there are 1194, 1005, 731, and 658 examples in each class, with a total of 3588. And the USPS dataset consists of a training set with 7291 images and a test set with 2007 images.

References

[1] J. Han, M. Kamber, Data Mining, Morgan Kaufmann, 2001.

[2] P. Berkhin, Survey of Clustering Data Mining Techniques, Technical Report, Accrue Software, 2002. 〈http://www.ee.ucr.edu/~barth/EE242/clustering_survey.pdf〉.

[3] A. Jain, M. Murty, P. Flynn, Data clustering: a review, ACM Computing Surveys 31 (3) (1999) 264–323.

[4] R. Xu, D. Wunsch, Survey of clustering algorithms, IEEE Transactions on Neural Networks 16 (3) (2005) 645–678.

[5] W.E. Donath, A.J. Hofmann, Lower bounds for the partitioning of graphs, IBM Journal of Research and Development 17 (1973) 420–425.

[6] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905.

[7] L. Hagen, A. Kahng, New spectral methods for ratio cut partitioning and clustering, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 11 (9) (1992) 1074–1085.

[8] C.H.Q. Ding, X. He, H. Zha, M. Gu, H.D. Simon, A min–max cut algorithm for graph partitioning and data clustering, in: Proceedings of the IEEE International Conference on Data Mining (ICDM), 2001, pp. 107–114.

[9] C. Fowlkes, S. Belongie, F. Chung, J. Malik, Spectral grouping using the Nyström method, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2) (2004) 214–225.

[10] K. Zhang, J.T. Kwok, Density-weighted Nyström method for computing large kernel Eigen-systems, Neural Computation 21 (2009) 121–146.

[11] K. Zhang, I.W. Tsang, J.T. Kwok, Improved Nyström low-rank approximation and error analysis, in: Proceedings of the 25th International Conference on Machine Learning (ICML), 2008, pp. 273–297.

[12] D. Yan, L. Huang, M. Jordan, Fast approximate spectral clustering, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2009, pp. 907–916.

[13] B.J. Frey, D. Dueck, Clustering by passing messages between data points, Science 305 (5814) (2007) 972–976.

[14] F. Kschischang, B.J. Frey, H.-A. Loeliger, Factor graphs and the sum–product algorithm, IEEE Transactions on Information Theory 47 (2) (2001) 498–519.

[15] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.

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

[17] A. Ng, M. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, Advances in Neural Information Processing Systems (NIPS), vol. 14, 2002, pp. 849–856.

[18] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, B. Schölkopf, Learning with local and global consistency, in: Proceedings of the Advances in Neural Information Processing Systems (NIPS), vol. 16, 2004, pp. 321-328.

[19] O. Chapelle, A. Zien, Semi-supervised classification by low density separation, in: Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTATS), vol. 10, 2005, pp. 57–64.

[20] G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, SIAM Journal of Scientific Computing 20 (1) (1999) 359–392.

[21] I. Dhillon, Y. Guan, B. Kulis, Weighted graph cuts without eigenvectors: a multilevel approach, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (11) (2007) 1944–1957.

[22] F. Wang, C. Zhang, Fast multilevel transduction on graphs, in: Proceedings of the 7th SIAM Conference on Data Mining (SDM), Minneapolis, Minnesota, USA, 2007, pp. 157–168.

[23] Y. Jia, J. Wang, C. Zhang, X. Hua, Finding image exemplars using fast sparse affinity propagation, in: Proceedings of ACM Multimedia, 2008, pp. 639–642.

[24] Y. Song, W.-Y. Chen, H. Bai, C.-J. Lin, E.Y. Chang, Parallel spectral clustering, in: Proceedings of Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD), 2008, pp. 374–389.

[25] M.-A. Belabbas, P.J. Wolfe, Spectral methods in machine learning and new strategies for very large datasets, Proceedings of the National Academy of Sciences of the United States of America (PNAS) 106 (2009) 369–374.

[26] L. Zelnik-Manor, P. Perona, Self-tuning spectral clustering, in: Proceedings of the Advances in Neural Information Processing Systems (NIPS), vol. 17, 2005, pp. 1601–1608.

[27] F. Shang, L.C. Jiao, J. Shi, M. Gong, R.H. Shang, Fast density-weighted low-rank approximation spectral clustering, Data Mining and Knowledge Discovery 23 (2) (2011) 345–378.

[28] Y. LeCun, C. Cortes, The MNIST database of handwritten digits, 2009. 〈http://yann.lecun.com/exdb/mnist/〉.

[29] A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (6) (2001) 643–660.

[30] M. Wu, B. Schölkopf, A local learning approach for clustering, in: Proceedings of the Advances in Neural Information Processing Systems (NIPS), vol. 19 2007, pp. 1529–1536.

[31] C.H. Papadimitriou, K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Dover, New York, 1998.

[32] A. Strehl, J. Ghosh, Cluster ensembles-A knowledge reuse framework for combining multiple partitions, Journal of Machine Learning Research 3 (2002) 583–617.

[33] Y. Yang, D. Xu, F. Nie, S. Yan, Y. Zhuang, Image clustering using local discriminant models and global integration, IEEE Transactions on Image Processing 19 (10) (2010) 2761–2773.

[34] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Workshop on Generative Model Based Vision, 2004.

[35] K. Grauman, T. Darrell, The pyramid match kernel: discriminative classification with sets of image features, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2005, pp. 1458-1465.

[36] S. Basu, A. Banerjee, R.J. Mooney, Active semi-supervision for pairwise constrained clustering, in: Proceedings of the 4th SIAM International Conference on Data Mining (SDM), 2004, pp. 333-344.

[37] R. Vidal, Y. Ma, J. Piazzi, A new GPCA algorithm for clustering subspaces by fitting, differentiating and dividing polynomials, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2004, pp. 510-517.

Fanhua Shang is currently pursuing Ph.D. degree in circuits and systems from the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an, China. His current research interests include pattern recognition, machine learning, data mining, and computer vision.

L.C. Jiao was born in Shaanxi, China, on October 15, 1959. He received the B.S. degree from Shanghai Jiaotong University, China, in 1982 and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively. From 1984 to 1986, he was an Assistant Professor with the Civil Aviation Institute of China, Tianjing, China. During 1990 and 1991, he was a Postdoctoral Fellow with the Key Lab for Radar Signal Processing, Xidian University, Xi'an, China. Currently, he is the Dean of the School of Electronic Engineering and the Director of the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China. His current research interests include signal and image processing, nonlinear circuits and systems theory, learning theory and algorithms, optimization problems, wavelet theory, machine learning and data mining. He is the author or coauthor of more than 200 scientific papers.

Jiarong Shi is currently pursuing Ph.D. degree in circuits and systems from the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an, China. His current research interests include pattern recognition and machine learning.

Fei Wang received his Ph.D. degree from Tsinghua University, Beijing, China, in 2008. After that, he came to Florida International University as a postdoctorate research associate till August 2009. Currently, he is a postdoc researcher in Healthcare Transformation Group, IBM T.J. Watson Research Center at Hawthorne, NY, USA.

Maoguo Gong is currently a professor in the School of Electronic Engineering, Xidian University. His main research interests include computational intelligence, data mining, and image processing.