

FAST SPECTRAL CLUSTERING WITH SELF-WEIGHTED FEATURES

XIANG ZHU¹, ZHILING CAI², ZINIU YU², JUNLIANG WU², WILLIAM ZHU^{2,*}

¹*School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu, China*

²*Institute of Fundamental and Frontier Sciences,
University of Electronic Science and Technology of China, Chengdu, China*

Abstract. As one of the mainstream clustering methods, the spectral clustering has aroused more and more attention recently because of its good performance, especially in nonlinear data sets. However, traditional spectral clustering models have high computational complexity. Meanwhile, most of these models fail in distinguishing the noisy and useful features in practice, which leads to the limitation of clustering performance. In this paper, we propose a new fast spectral clustering with self-weighted features (FSCSWF) to achieve good clustering performance through learning and assigning optimal weights for features in a low computational complexity. Specifically, the FSCSWF selects anchors from original samples, then learns the weights of features and the similarity between anchors and samples interactively in a local structure learning framework. This interactive learning makes the learnt similarity can better measure the relationship between anchors, and samples due to the optimal weights make the data points become more discriminative. Moreover, the connectivity constraint are embedded to make sure that the connected components of bipartite graph constructed by the learnt similarity can indicate clusters directly. In this way, the FSCSWF can achieve good clustering performance and has a low computational complexity, which is linear to the number of samples. Extensive experiments on synthetic and practical data sets illustrate the effectiveness and efficiency of the FSCSWF with respect to state-of-the-art methods.

Keywords. Anchor points; Bipartite graph; Local structure learning; Spectral clustering; Self-weighted features.

1. INTRODUCTION

Clustering, as a widely used unsupervised learning technique, aims to divide samples into a specific number of groups based on the similarity among the samples. It is widely used in pattern recognition [1], machine learning [2, 3], image analysis [4], data mining [5], to name a few. After decades of development, many clustering algorithms have been proposed, such as K-means [6], Ratio-cut (R-cut) [7], CURE [8], DBSCAN [9]. From the viewpoint of involved methods, existing clustering methods can be roughly divided into: spectral clustering [10, 11, 12, 13], hierarchical clustering [14], density-based clustering [15, 16, 17], subspace clustering [18, 19], deep clustering [20] and so on.

*Corresponding author.

E-mail address: wfzhu@uestc.edu.cn (W. Zhu).

Received July 20, 2021; Accepted August 23, 2021.

In recent years, the spectral clustering has received more and more attention because it has a wide range of applications. Spectral clustering algorithm, which can be used to detect non-convex patterns as well as nonlinear data sets [10, 21], is unsuitable for other clustering methods based on Euclidean geometry. The basic framework of the spectral clustering algorithm can be divided into three steps. First, build a similarity graph (SG) according to the data points; then perform eigenvalue decomposition to obtain the low-dimensional representation of data points based on SG [22]; finally, get the clustering results through partitioning on the low-dimensional representation of data points. In practice, based on different methods to construct similarity graph, spectral clustering can be divided into different models: local structure learning [23, 24], and global structure learning [25, 26], etc. However, the SG establishment and the eigenvalue decomposition have high computational complexity, which are $O(n^2)$ and $O(n^3)$, respectively, where n is the number of samples. The high computational complexity makes the traditional spectral clustering methods unsuitable for large-scale data sets.

In order to reduce the computational complexity, many fast spectral clustering (FSC) methods were proposed. Choromanska *et al.* [27] employed the Nyström approximation to the Laplacian matrix to perform clustering; Chen *et al.* [28] scaled down the computational complexity by sequentially reducing the dimension of the Laplacian matrix; Wang *et al.* [29] constrained the co-clustering factor matrices to be cluster indicator matrices to decouple the optimization problem into smaller size subproblems. Moreover, anchor-based clustering can also effectively reduce complexity through selecting anchor points from the original samples to learn the optimal bipartite graph; see, e.g., [13, 30, 31, 32, 33]. Due to the number of anchor points is much less than that of samples, the computational complexity corresponding to the process of constructing the bipartite graph can be reduced greatly. However, existing FSC models follow the assumption that there is no rank relationship for features including noisy and useful features, which is not adaptive in practical and limit the clustering performance.

In this paper, we propose a new fast spectral clustering method, called FSCSWF, which can learn the optimal weights of all features inspired by the self-weighted technique mentioned in SWCAN [34] for the anchor-based FSC model in a low computational complexity. First, our method selects anchor points from original samples by the K-means method which makes the anchor points more representative than random ways. Then, aiming to make the similarity between anchor points and samples can better measure the relationship between them, our method learns the weights of all features and the similarity interactively in a local structure learning framework. In the learning process, the learnt similarity constructs the bipartite graph and the connectivity constraint are embedded to make sure that the connected components of bipartite graph can indicate clusters directly. The FSCSWF takes advantage of the learnt optimal weights of features to make the data points more discriminative so as to learn the better similarity which can help improve the clustering performance. Moreover, the use of bipartite graph makes the computational complexity is linear to the number of samples. Therefore, the FSCSWF can achieve good clustering performance with a low computational complexity. Extensive experiments illustrate that the FSCSWF has better clustering performance compared to other state-of-the-art techniques on synthetic and practical data sets.

To sum up, the main contributions of this paper are as follows.

(1) We propose a new fast spectral clustering with self-weighted features (FSCSWF) framework, to learn the weights of features and the similarity between anchors and samples interactively to make the similarity can better measure the relationship between anchors and samples.

(2) We embed the connectivity constraint to the framework of the FSCSWF to ensure the connected components of bipartite graph, which is constructed by the learnt similarity, can indicate clusters directly.

(3) We propose an effective algorithm, whose overall computational complexity is linear to the number of samples, to realize the interactive learning of weights and similarity as well as clustering on samples.

The rest of this paper is organized as follows. In Section 2, we list the notations used in this article and revisit the related works for spectral clustering and anchor-based clustering. In Section 3, we propose the FSCSWF model to learn optimal weights of different features as well as the similarity between samples and anchors simultaneously. In Section 4, we develop an efficient algorithm to optimize the FSCSWF and compare time complexity. In Section 5, we perform extensive experiments to verify the effectiveness of the FSCSWF. At last, we draw the conclusion in Section 6.

2. RELATED WORK

2.1. Notations. In this subsection, we give some general notations, which will be used later. First, all the matrices are written as uppercase throughout this paper, such as M . For a matrix M , the transpose of M is M^T , and the trace of M is $\text{Tr}(M)$. m_i means the i th row of M , and m_{ij} means the i, j -th element of M . Besides, $\|v\|_2$ is the l_2 -norm of vector v , and $\|M\|_F$ is the F -norm of matrix M . The specific definitions of notations are listed in TABLE 1.

TABLE 1. Notations and abbreviations

Notation	Definition
n	the number of samples
m	the number of anchors
c	the number of clusters
d	the dimension of features
x_i	i th sample
u_j	j th anchor
$X \in \mathbb{R}^{n \times d}$	the data matrix
$U \in \mathbb{R}^{m \times d}$	the anchor matrix
$Z \in \mathbb{R}^{n \times m}$	z_{ij} denotes the similarity between x_i and u_j
$S \in \mathbb{R}^{(n+m) \times (n+m)}$	the bipartite graph constructed by Z
$D_s \in \mathbb{R}^{(n+m) \times (n+m)}$	the degree matrix of S
$L_s \in \mathbb{R}^{(n+m) \times (n+m)}$	the normalized Laplacian matrix of S
$\Theta \in \mathbb{R}^{d \times d}$	the wight matrix

2.2. Spectral clustering. Denote that the data set is $X = [x_1; x_2; \dots; x_n] \in \mathbb{R}^{n \times d}$, and spectral clustering constructs an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, W\}$, where \mathcal{V} is the set of verticals in this graph, which are the data points x_i , $i = 1, 2, \dots, n$; \mathcal{E} denotes the edge set and each edge on

behalf of the similarity relationship of one pair of vertexes; $W \in \mathbb{R}^{n \times n}$, called similarity matrix, denotes weights of the edges [35]. Let $L = D - W$, which is called the Laplacian matrix [36], and the matrix D is named as the degree matrix which is a diagonal matrix with the i th diagonal element $D_{ii} = \sum_{j=1}^n W_{ji}$.

Spectral clustering models make the use of the c eigenvectors of L corresponding to c smallest eigenvalues, where c is the number of clusters, to obtain the spectral embedded representation $F \in \mathbb{R}^{n \times c}$. Based on this, Ratio-cut (R-cut) [7] and Normalized-cut (N-cut) [12] have been proposed. In practice, the objective function of R-cut can be written as follows:

$$\begin{aligned} \min_F \text{Tr}(F^T L F), \\ \text{s.t. } F^T F = I_n, \end{aligned} \quad (2.1)$$

where F is the spectral embedded representation, and L is the Laplacian matrix defined before. As for N-cut, the objective function is

$$\begin{aligned} \min_F \text{Tr}(F^T \tilde{L} F), \\ \text{s.t. } F^T F = I_n, \end{aligned} \quad (2.2)$$

where F is the spectral embedded representation, and \tilde{L} is the normalized Laplacian matrix which is defined as $\tilde{L} = I - D^{-1/2} W D^{-1/2}$. Because the graph construction and eigen-decomposition both have high computational complexity, traditional spectral clustering models are inapplicable to large-scale data.

2.3. Anchor-based clustering. To alleviate the problem of high computational complexity, anchors or landmarks were widely in used in scalable spectral clustering; see, e.g., [13, 37] for more details. The core idea in anchor-based spectral clustering models is the construction of the small graph $Z \in \mathbb{R}^{n \times m}$ to represent the similarity between anchors and original samples, where n is the number of samples and m is the number of anchors and use Z to generate the bipartite graph. Generally, the small graph Z can be obtained by the kernel function, global structure learning, local structure learning and so on. As for the kernel method, taking the Gaussian kernel function for example, Z can be computed as [13]:

$$z_{ij} = \frac{K_\sigma(x_i, u_j)}{\sum_{j' \in U_{< i >}} K_\sigma(x_i, u_{j'})}, j \in U_{< i >}, \quad (2.3)$$

where $K_\sigma = \exp(-\frac{\|x_i - u_j\|_2^2}{2\sigma^2})$, and $U_{< i >} \in \mathbb{R}^{r \times d}$ is composed of the r nearest anchors of the sample x_i .

As for the global structure learning, Z can be learnt by solving the following problem [33]:

$$\min_{Z: \mathbf{1} = \mathbf{1}, Z \geq 0} \|X - ZU\|_F^2 + \alpha \|Z\|_F^2, \quad (2.4)$$

where α is the regularization parameter. Different from the kernel method and global structure learning method, local structure learning can capture the local structure information of samples and anchors well and learn the matrix Z by solving the following problem [35, 38]:

$$\min_{Z: \mathbf{1} = \mathbf{1}, Z \geq 0} \sum_{i=1}^n \sum_{j=1}^m \|x_i - u_j\|_2^2 z_{ij} + \gamma \|Z\|_F^2, \quad (2.5)$$

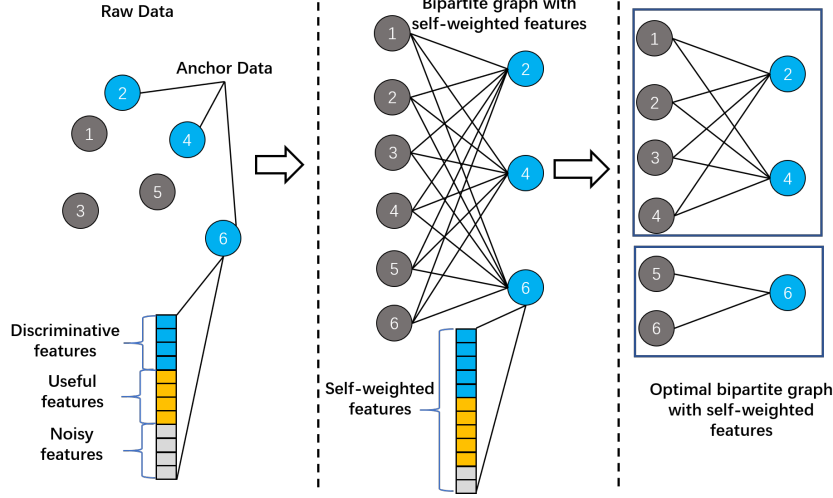


FIGURE 1. Schematic diagram of the framework of the FSCSWF. There are three main steps in the framework of the FSCSWF: select anchors from samples and the data points contain original features; construct bipartite graph of anchors and samples and learn optimal weights of features; clustering data points into specific groups based on the optimal bipartite graph.

where γ is the regularization parameter.

3. METHODOLOGY

In this section, we propose a new fast spectral clustering with self-weighted features, called FSCSWF, to obtain good clustering performance through assigning optimal weights for all features with a low computational complexity.

Denote that $X = [x_1; x_2; \dots; x_n] \in \mathbb{R}^{n \times d}$ as the original samples and $U = [u_1; u_2; \dots; u_m] \in \mathbb{R}^{m \times d}$ as the generated anchor points from X through the K-means method instead of the random sampling in order to make the anchors more representative. The similarity between original samples and the anchor points can be learnt through a local structure learning method as following:

$$\begin{aligned} \min_Z \sum_{i=1}^n \sum_{j=1}^m \|x_i - u_j\|_2^2 z_{ij} + \gamma \|Z\|_F^2, \\ \text{s.t. } Z \cdot \mathbf{1} = \mathbf{1}, Z \geq 0. \end{aligned} \quad (3.1)$$

However, the similarity matrix Z can be learnt through solving (3.1), the difference of features dose not get enough attention in (3.1). In order to make the samples more discriminative so as to improve the clustering performance, we need to assign optimal weights for features based on the importance of features [34]. Instead of assigning the weights based on the prior knowledge of samples, we propose a method to learn the weights of all features as well as the similarity between samples and anchor points simultaneously through embedding the weight matrix Θ into (3.1):

$$\begin{aligned}
& \min_{\Theta, Z} \sum_{i=1}^n \sum_{j=1}^m \|x_i \Theta - u_j \Theta\|_2^2 z_{ij} + \gamma \|Z\|_F^2, \\
& \text{s.t. } 0 < \theta_k < 1, \sum_{k=1}^d \theta_k = 1, \\
& \Theta = \text{diag}(\theta_k), Z \cdot \mathbf{1} = \mathbf{1}, Z \geq 0,
\end{aligned} \tag{3.2}$$

where $\Theta \in \mathbb{R}^{d \times d}$ is a diagonal matrix and its diagonal elements are the weights of features. In this way, the weights as well as the similarity can be learnt simultaneously and interactively to both achieve optimal values. Then the bipartite S graph can be constructed based on Z as [39]:

$$S = \begin{bmatrix} 0 & Z \\ Z^T & 0 \end{bmatrix}. \tag{3.3}$$

And we can obtain the the degree matrix of S as:

$$D_S = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}, \tag{3.4}$$

where D_1 and D_2 are both diagonal matrices and the i th diagonal element of D_1 is defined as $\sum_{j=1}^m z_{ij}$ while the j th diagonal element of D_2 is defined as $\sum_{i=1}^n z_{ij}$. Correspondingly, we calculate the normalized Laplacian matrix as [40]:

$$L_S = I - D_S^{-1/2} S D_S^{-1/2}. \tag{3.5}$$

The normalized Laplacian matrix defined by (3.5) has one important property [36, 41]:

Theorem 3.1. *The multiplicity k of the eigenvalue 0 of the normalized matrix L_S is equal to the number of connected components in the graph associated with S*

According to Theorem 3.1, if $\text{rank}(L_S) = (n + m) - c$, then the n data points as well as m anchor points can be divided into c clusters, which is an ideal clustering result. Therefore, we add the connectivity constraint to the framework of the FSCSWF to ensure the connected components of bipartite graph S can indicate clusters directly. Correspondingly, the objective function of the FSCSWF is

$$\begin{aligned}
& \min_{\Theta, Z} \sum_{i=1}^n \sum_{j=1}^m \|x_i \Theta - u_j \Theta\|_2^2 z_{ij} + \gamma \|Z\|_F^2, \\
& \text{s.t. } 0 < \theta_k < 1, \sum_{k=1}^d \theta_k = 1, \Theta = \text{diag}(\theta_k), \\
& Z \cdot \mathbf{1} = \mathbf{1}, Z \geq 0, \text{rank}(L_S) = (n + m) - c.
\end{aligned} \tag{3.6}$$

The framework of the FSCSWF is shown in FIGURE 1.

4. OPTIMIZATION ALGORITHM

4.1. Optimization strategy. To avoid solving the difficult rank constrain in problem (3.6), we transfer problem (3.6) to an equivalent problem through relaxing the rank constraint to a matrix form and then solve the problem by iterative method.

We denote $\sigma_h(L_s)$ as the h th smallest eigenvalue of L_s . Then, problem (3.6) can be relaxed as following problem if the value of λ is large enough.

$$\begin{aligned} \min_{\Theta, Z} \quad & \sum_{i=1}^n \sum_{j=1}^m \|x_i \Theta - u_j \Theta\|_2^2 z_{ij} + \gamma \|Z\|_F^2 + \lambda \sum_{k=1}^c \sigma_h(L_s), \\ \text{s.t.} \quad & 0 < \theta_k < 1, \sum_{k=1}^d \theta_k = 1, \Theta = \text{diag}(\theta_k), \\ & Z \cdot \mathbf{1} = \mathbf{1}, Z \geq 0. \end{aligned} \quad (4.1)$$

According to Ky Fan's Theorem [42], we have

$$\begin{aligned} \sum_{h=1}^c (L_s) &= \min_F \text{Tr}(F^T L_s F), \\ \text{s.t.} \quad & F^T F = I, F \in \mathbb{R}^{n \times c}, \end{aligned} \quad (4.2)$$

where c is the number of clusters. Therefore, problem (3.6) can be transformed as following problem, which is easier to tackle.

$$\begin{aligned} \min_{\Theta, Z, F} \quad & \sum_{i=1}^n \sum_{j=1}^m \|x_i \Theta - u_j \Theta\|_2^2 z_{ij} + \gamma \|Z\|_F^2 + \lambda \text{Tr}(F^T L_s F), \\ \text{s.t.} \quad & 0 < \theta_k < 1, \sum_{k=1}^d \theta_k = 1, \Theta = \text{diag}(\theta_k), \\ & Z \cdot \mathbf{1} = \mathbf{1}, Z \geq 0, F^T F = I. \end{aligned} \quad (4.3)$$

To cope with problem (4.3), we need to solve three sub-problems for Θ , F and Z , respectively. In this section, we obtain the optimal solution for the three variables by fixing any two of them and update the rest one.

4.1.1. *Update Θ with F and Z fixed.* Then, problem (4.3) can be rewritten as its equivalent form

$$\begin{aligned} \min_{\Theta} \quad & \sum_{i=1}^n \sum_{j=1}^m \|x_i \Theta - u_j \Theta\|_2^2 z_{ij}, \\ \text{s.t.} \quad & \text{Tr}(\Theta) = 1, \Theta > 0. \end{aligned} \quad (4.4)$$

In order to solve problem (4.4), we need to transform it into matrix form based on the following lemma.

Lemma 4.1.

$$\begin{aligned} \sum_{i=1}^n \sum_{j=1}^m \|x_i \Theta - u_j \Theta\|_2^2 z_{ij} &= \text{Tr}[\Theta^T (X^T D_1 X + U^T D_2 U - 2X^T Z U) \Theta], \\ \text{s.t.} \quad & D_1 = \text{diag}(\sum_{j=1}^m z_{ij}), D_2 = \text{diag}(\sum_{i=1}^n z_{ij}), \\ & \text{Tr}(\Theta) = 1, \Theta > 0. \end{aligned}$$

Proof.

$$\begin{aligned}
& \sum_{i=1}^n \sum_{j=1}^m \|x_i \Theta - u_j \Theta\|_2^2 z_{ij} \\
&= \sum_{i=1}^n \sum_{j=1}^m (x_i \Theta - u_j \Theta)(x_i \Theta - u_j \Theta)^T z_{ij} \\
&= \sum_{i=1}^n x_i \Theta \left(\sum_{j=1}^m z_{ij} \right) \Theta^T x_i^T + \sum_{j=1}^m u_j \Theta \left(\sum_{i=1}^n z_{ij} \right) \Theta^T u_j^T - 2 \sum_{i=1}^n \sum_{j=1}^m x_i \Theta z_{ij} \Theta^T x_i^T \\
&= \text{Tr}(\Theta^T X^T D_1 X \Theta) + \text{Tr}(\Theta^T U^T D_2 U \Theta) - 2 \text{Tr}(\Theta^T X^T Z U \Theta) \\
&= \text{Tr}[\Theta^T (X^T X + U^T D_2 U - 2X^T Z U) \Theta], \quad D_1 = I_n.
\end{aligned}$$

□

Following Lemma 4.1, denote $P \in \mathbb{R}^{d \times d} = X^T X + U^T D_2 U - 2X^T Z U$ and (4.4) can be rewritten as following:

$$\begin{aligned}
& \min_{\Theta} \text{Tr}(\Theta^T P \Theta), \\
& \text{s.t. } \text{Tr}(\Theta) = 1, \quad \Theta > 0.
\end{aligned} \tag{4.5}$$

Denote p_k^* as the k th diagonal element of P , and we can obtain the following relationship:

$$\text{Tr}(\Theta^T P \Theta) = \sum_{k=1}^d \theta_k^2 p_k^*. \tag{4.6}$$

As for the two constraints of Θ , we first ignore the non-negative one $\Theta > 0$ and consider $\text{Tr}(\Theta) = 1$ only. Then Lagrange function of problem (4.5) can be written as

$$L(\theta_k, \eta) = \sum_{k=1}^d \theta_k^2 p_k^* + \eta \left(\sum_{k=1}^d \theta_k - 1 \right), \tag{4.7}$$

where $\eta > 0$ is the the Lagrangian multiplier. We solve θ_k by setting the derivative $\frac{\partial L}{\partial \theta} = 2\theta_k p_k^* + \eta = 0$ and then can obtain the following result.

$$\theta_k = \frac{-\eta}{2p_k^*}. \tag{4.8}$$

Combining the constraint $\text{Tr}(\Theta) = 1$, which means $\sum_{k=1}^d \theta_k = 1$, with (4.8), the closed form solution of θ_k can be written as

$$\theta_k = \frac{1}{p_k^* \sum_{k=1}^d \frac{1}{p_k^*}}. \tag{4.9}$$

As for the non-negative constraint $\Theta > 0$, if we can prove that the value of the k th diagonal element of P is greater than zero ($p_k^* > 0$), then, without loss of generality, this constrain can be satisfied based on (4.9).

Because of $P = X^T X + U^T D_2 U - 2X^T Z U$, its diagonal element p_k^* can be written as

$$\begin{aligned}
p_k^* &= x_{:,k}^T x_{:,k} + u_{:,k}^T D_2 u_{:,k} - 2x_{:,k}^T Z u_{:,k} \\
&\Rightarrow \text{Tr}(x_{:,k}^T x_{:,k}) + \text{Tr}(u_{:,k}^T D_2 u_{:,k}) - 2\text{Tr}(x_{:,k}^T Z u_{:,k}) \\
&\Rightarrow \sum_{i=1}^n \sum_{j=1}^m \|x_{i,k} - u_{j,k}\|_2^2 z_{ij},
\end{aligned}$$

where $x_{:,k} \in \mathbb{R}^{n \times 1}$ and $u_{:,k} \in \mathbb{R}^{m \times 1}$ are the k th column of X and U , respectively. Because of $0 < z_{ij} < 1$ and $\|x_{i,k} - u_{j,k}\|_2^2 \geq 0$, we can prove that $p_k^* \geq 0$, which leads to $\Theta > 0$. Therefore, the optimal solution of Θ can be achieved by solving problem (4.9).

4.1.2. *Update F with Θ and Z fixed.* Problem (4.3) can be written as following:

$$\begin{aligned}
&\min_F \text{Tr}(F^T L_S F) \\
&= \min_F \text{Tr}(F^T (I - D_s^{-1/2} S D_s^{-1/2}) F) \\
&= \max_F \text{Tr}(F^T D_s^{-1/2} S D_s^{-1/2} F), \\
&\text{s.t. } F^T F = I.
\end{aligned} \tag{4.10}$$

Supposing that $F \in \mathbb{R}^{(n+m) \times c} = [M; N]$, $M \in \mathbb{R}^{n \times c}$, $N \in \mathbb{R}^{m \times c}$ as well as combining the structure of S and D_S defined in (3.3) and (3.4) respectively, problem (4.10) can be rewritten as

$$\begin{aligned}
&\max_F \text{Tr}(M^T D_1^{-1/2} Z D_2^{-1/2} N) \\
&\text{s.t. } M^T M + N^T N = I.
\end{aligned} \tag{4.11}$$

According to [40], we can solve problem (4.11) by a SVD decomposition method described in Lemma 4.2.

Lemma 4.2. *The optimal solutions to the problems:*

$$\max_{X^T X + Y^T Y = I} \text{Tr}(X^T Q Y)$$

are $X = \frac{\sqrt{2}}{2} U_1$, $Y = \frac{\sqrt{2}}{2} V_1$ where U_1 and V_1 are corresponding to the top c left and right singular vectors of Q , respectively.

According to Lemma 4.2, we suppose that $D_1^{-1/2} Z D_2^{-1/2}$ as Q . Therefore, the optimal solutions of M and N are

$$M = \frac{\sqrt{2}}{2} M_z, \quad N = \frac{\sqrt{2}}{2} N_z, \tag{4.12}$$

where M_z and N_z are corresponding to the top c left and right singular vectors of $D_1^{-1/2} Z D_2^{-1/2}$, respectively. Therefore, F can be updated as $[\frac{\sqrt{2}}{2} M_z; \frac{\sqrt{2}}{2} N_z]$.

Algorithm 1 Optimization Algorithm for the FSCSWF

Input: Data $X = [x_1; x_2; \dots; x_n] \in \mathbb{R}^{n \times d}$, cluster number c , anchor number m , tradeoff parameter λ and the number of neighbors \tilde{k} .

Output: Results of clustering.

- 1: Run K-means on X to obtain the anchor matrix $U \in \mathbb{R}^{m \times d}$.
 - 2: Initialize Z by solving Eq.(4.17) where $l_{ij} = \|x_i - u_j\|_2^2$ and calculate γ through (4.18).
 - 3: **while** not converge **do**
 - 4: Update F by solving Eq.(4.12).
 - 5: Update Θ by solving Eq.(4.9).
 - 6: Update Z by solving Eq.(4.17) where $l_{ij} = \|x_i\Theta - u_j\Theta\|_2^2 + \lambda \cdot \left\| \frac{f_i}{\sqrt{d_i}} - \frac{f_{n+j}}{\sqrt{d_{n+j}}} \right\|_2^2$.
 - 7: **end while**
 - 8: $F = [M; N] \in \mathbb{R}^{(n+m) \times c}$, run K-means on M to achieve the clustering results.
-

4.1.3. *Update Z with Θ and F fixed.* The constraints of problem (4.3) will be simplified largely and the remaining ones are $Z \cdot \mathbf{1} = \mathbf{1}, Z \geq 0$ with the objective function as

$$\begin{aligned} \min_Z \sum_{i=1}^n \sum_{j=1}^m \|x_i\Theta - u_j\Theta\|_2^2 z_{ij} + \gamma \|Z\|_F^2 + \lambda \text{Tr}(F^T L_s F) \\ \text{s.t. } Z \cdot \mathbf{1} = \mathbf{1}, Z \geq 0. \end{aligned} \quad (4.13)$$

According to [43],

$$\begin{aligned} \text{Tr}(F^T L_s F) &= \frac{1}{2} \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} \left\| \frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right\|_2^2 s_{ij} \\ &= \sum_{i=1}^n \sum_{j=1}^m \left\| \frac{f_i}{\sqrt{d_i}} - \frac{f_{n+j}}{\sqrt{d_{n+j}}} \right\|_2^2 z_{ij}, \end{aligned} \quad (4.14)$$

where f_i is the i th row of F , while d_i is the i th row of D_s . Let $\|x_i\Theta - u_j\Theta\|_2^2 = d_{ij}^{xu}$ and $\left\| \frac{f_i}{\sqrt{d_i}} - \frac{f_{n+j}}{\sqrt{d_{n+j}}} \right\|_2^2 = d_{ij}^f$. Then, problem (4.13) can be transformed into

$$\begin{aligned} \min_Z \sum_{i=1}^n \sum_{j=1}^m (d_{ij}^{xu} z_{ij} + \gamma z_{ij}^2 + \lambda d_{ij}^f z_{ij}) \\ = \min_Z \sum_{i=1}^n \sum_{j=1}^m (\gamma z_{ij}^2 + (d_{ij}^{xu} + \lambda d_{ij}^f) z_{ij}) \\ = \min_Z \sum_{i=1}^n \left\| z_i + \frac{1}{2\gamma} l_i \right\|_2^2 \\ \text{s.t. } l_i = d_i^{xu} + \lambda d_i^f, Z \cdot \mathbf{1} = \mathbf{1}, Z \geq 0. \end{aligned} \quad (4.15)$$

The Lagrange function of problem (4.15) can be constructed as

$$L(z_i, \eta, \beta_i) = \frac{1}{2} \sum_{i=1}^n \left\| z_i + \frac{1}{2\gamma} l_i \right\|_2^2 - \alpha(z_i \cdot \mathbf{1} - 1) - \beta_i z_i^T, \quad (4.16)$$

where $\alpha > 0$ and $\beta_i > 0$ are the Lagrangian multipliers. According to KKT conditions, the closed form solution of z_i and γ can be set as [44]

$$z_i = \left(-\frac{l_i}{2\gamma_i} + \alpha \right)_+, \quad (4.17)$$

$$\gamma = \frac{1}{n} \sum_{i=1}^n \gamma_i = \frac{1}{n} \sum_{i=1}^n \left(\frac{\tilde{k}}{2} l_{i, \tilde{k}+1} - \frac{1}{2} \sum_{j=1}^{\tilde{k}} l_{ij} \right), \quad (4.18)$$

where \tilde{k} is the number of nearest neighbours.

The complete optimization algorithm of the FSCSWF is summarized in Algorithm 1. According to the theory of alternating optimization proposed in [45], the value of our objective function defined in (4.3) will decrease monotonously after each iteration until it converges. Besides, the value of our objective function is non-negative because its three terms are defined by l_2 -norm or F -norm. Therefore, our objective function has a lower bound, such as 0, and the iteration algorithm converges.

4.2. Computational complexity analysis. In this section, we analyse the computational complexity of three main parts of our algorithm: updating F , Θ , and Z .

Suppose that t, n, m, d, \tilde{k} are the number of optimization iterations, data points, anchor points, data features, and nearest neighbors. The computational complexity of our algorithm mainly comes from the following steps:

(1) Updating F by SVD decomposition of $D_1^{-1/2} Z D_2^{-1/2} \in R^{n \times m}$ has the computational complexity of $O((m^3 + mn)t)$;

(2) Updating Z has the the computational complexity of $O((mn\tilde{k})t)$.

(3) The main computational complexity of updating Θ exists in calculating $P \in R^{d \times d} = X^T X + U^T D_2 U - 2X^T Z U$, which takes $O((d^2 n + dm^2 + d^2 m + dmn)t)$.

Therefore, the overall computational complexity of the FSCSWF is linear to the number of data points n . When dealing with large-scale data which has a large value of n , because of $m \ll n$, our algorithm has a lower computational complexity compared with some spectral clustering and graph-based algorithms, such as CAN [23] with $O(n^2 d + (n^3 + n^2 d)t)$, CLR [24] with $O(n^3 t)$ and SWCAN [34] with $O((n^3 + nd \max(n, d))t)$.

5. EXPERIMENTS

5.1. Data sets.

5.1.1. Synthetic data sets. To test the clustering performance of our method, we do visualization experiments on two popular synthetic data sets: Two-moon data set and Three-ring data sets. The Two-moon data set can be divided into two classes, and the numbers as well as features of this data set are 600 and 2 respectively. The Three-ring data set can be divided into three classes, and the numbers as well as features of this data set are 1000 and 3 respectively.

TABLE 2. The description of Practical benchmark data sets

Data sets	Numbers	Features	Classes
Iris	150	4	3
Lung	203	3312	5
Jaffe	213	1024	10
UMist	575	10304	20
TR45	690	8261	10
BA	1404	320	36
Coil20	1440	1024	20
USPS	2007	256	10
MNIST	10000	784	10

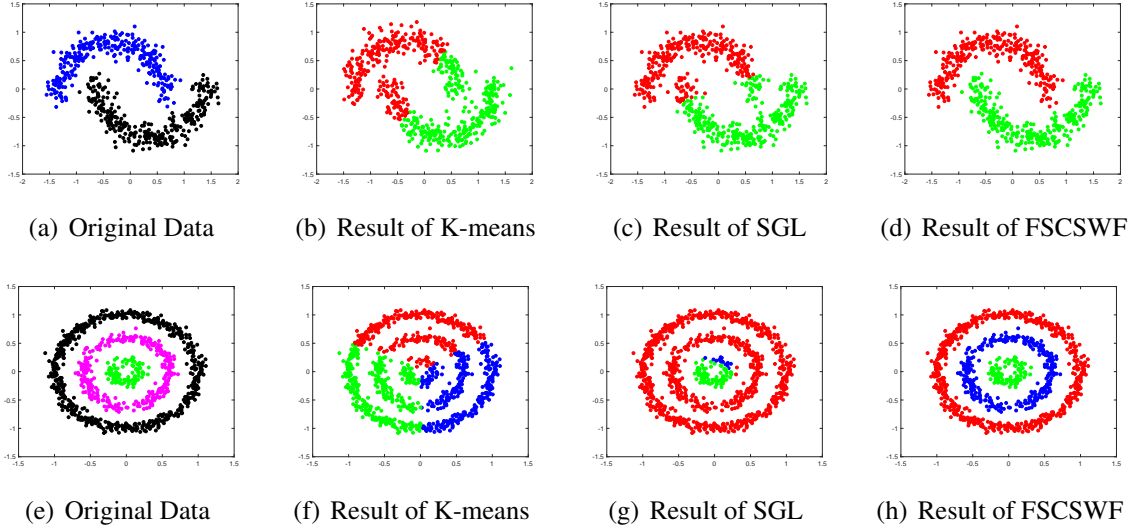


FIGURE 2. Clustering results on synthetic data sets. (a)-(d) are results on Two-moon data set. (e)-(h) are results on Three-ring data set.

5.1.2. *Practical benchmark data sets.* Apart from the synthetic data sets, we also use nine practical benchmark data sets to test the performance of our method: TR45 [33], Iris [46], Lung [47], Jaffe [48], UMist [49], BA [50], Coil20 [51], USPS [52], MNIST [53]. The details of these data sets are summarized in TABLE 2. In addition, we employ the PCA to increase the efficiency of experiments as in [19].

5.2. **Experiments Setup.** In order to demonstrate the performance of the method (FSCSWF), we choose the following comparison methods: K-means [6], N-cut [12], LSC-K [13], LSC-R [13], SGL [33], and ESCG [54]. The summaries of these methods are:

K-means: One of the simplest and the best known unsupervised learning algorithms, which assigns each data point to one of the clusters by minimizing the within-cluster sum of squares.

N-cut: A classic algorithm, which measures both the total dissimilarity between the different clusters as well as the total similarity within the clusters.

TABLE 3. Clustering performance on practical benchmark data sets of various methods (the best performance are highlighted and time is measured in seconds)

Iris					Lung				Jaffe			
	ACC	NMI	Purity	Time	ACC	NMI	Purity	Time	ACC	NMI	Purity	Time
K-means	0.8867	0.7364	0.8867	0.0031	0.7094	0.5279	0.8966	0.1221	0.9155	0.8995	0.9155	0.0054
N-cut	0.9067	0.7787	0.9067	0.0027	0.7438	0.3219	0.7882	0.0053	0.8920	0.8616	0.8920	0.0039
LSC-R	0.9267	0.8253	0.9267	0.0251	0.6207	0.4206	0.8227	0.0348	0.9624	0.9438	0.9624	0.0319
LSC-K	0.8933	0.7782	0.8933	0.0214	0.7586	0.5092	0.8768	0.1193	0.9859	0.9780	0.9859	0.0497
ESCG	0.9600	0.8697	0.9600	0.0260	0.9212	0.6822	0.9261	0.0222	0.9859	0.9783	0.9859	0.0370
SGL	0.9133	0.7694	0.9133	1.2719	0.9163	0.6953	0.9261	2.2152	0.9953	0.9917	0.9953	0.4330
FSCSWF	0.9667	0.8983	0.9667	0.0388	0.9458	0.7933	0.9458	1.8797	0.9953	0.9918	0.9953	0.3538
UMist					TR45				BA			
	ACC	NMI	Purity	Time	ACC	NMI	Purity	Time	ACC	NMI	Purity	Time
K-means	0.3948	0.6032	0.4626	0.0388	0.6319	0.6182	0.7667	0.0287	0.4345	0.5848	0.4594	0.1840
N-cut	0.4765	0.6266	0.5130	0.0406	0.7217	0.6875	0.8406	0.0375	0.3832	0.5461	0.4124	0.5372
LSC-R	0.4661	0.6598	0.5287	0.1709	0.6971	0.6373	0.8058	0.1736	0.4487	0.5817	0.4751	0.7773
LSC-K	0.4922	0.6676	0.5426	0.2690	0.7333	0.6801	0.8464	0.2396	0.4679	0.5946	0.4922	1.1237
ESCG	0.6643	0.7379	0.6870	0.1371	0.7029	0.6679	0.8203	0.3137	0.4274	0.5589	0.4523	1.1542
SGL	0.5739	0.6996	0.6278	7.1644	0.7522	0.7051	0.7754	1.9401	0.4815	0.6021	0.5028	3.9955
FSCSWF	0.6817	0.7499	0.6939	0.8495	0.8261	0.7526	0.8464	0.6175	0.4843	0.6134	0.5121	3.0414
Coil20					USPS				MNIST			
	ACC	NMI	Purity	Time	ACC	NMI	Purity	Time	ACC	NMI	Purity	Time
K-means	0.6583	0.7341	0.6687	0.0848	0.7025	0.6718	0.7683	0.0985	0.5611	0.5385	0.6171	1.3367
N-cut	0.6660	0.7450	0.7201	0.0490	0.6692	0.6294	0.7349	0.2917	0.5504	0.4781	0.6016	8.4015
LSC-R	0.6993	0.7809	0.7285	0.3471	0.6288	0.6414	0.7374	0.5913	0.6742	0.6600	0.7125	3.6337
LSC-K	0.7153	0.7790	0.7194	0.6041	0.7524	0.7442	0.8251	0.8283	0.7277	0.7214	0.7770	11.7420
ESCG	0.8083	0.8728	0.8313	0.2340	0.7419	0.6701	0.7509	0.2577	0.5533	0.5642	0.6196	7.6364
SGL	0.7646	0.8185	0.7681	6.1577	0.7803	0.7974	0.8515	5.9714	0.6283	0.6044	0.6709	39.5280
FSCSWF	0.8049	0.8737	0.8257	3.4176	0.8490	0.7944	0.8535	3.8042	0.7679	0.7852	0.8067	48.3621

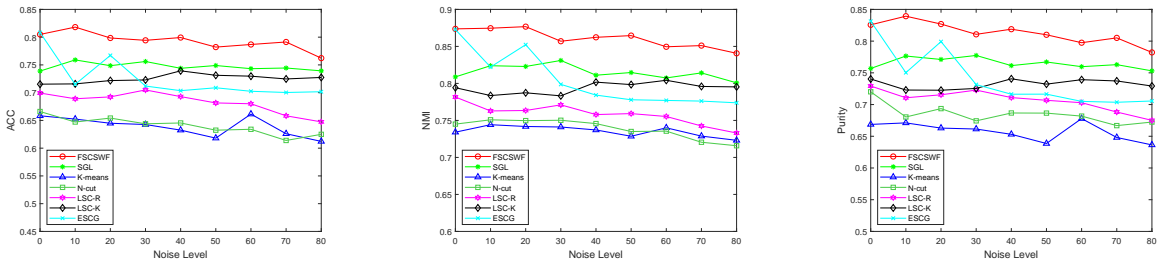


FIGURE 3. Performance of different methods on Coil20 with different standard deviations of Gaussian noise (from 0 to 80).

LSC-K: A landmark-based spectral clustering method using K-means to select landmarks.

LSC-R: A landmark-based spectral clustering method using random sampling to select landmarks.

ESCG: An efficient spectral clustering method uses the shortest path algorithm to generate bipartite graph.

SGL: An effective graph-based subspace clustering method based on anchor points and bipartite graph, which was proposed recently.



FIGURE 4. The images with white Gaussian noise. From left to right, the standard deviation σ_n increases from 0 to 80 (with an interval of 10).

Moreover, some parameters of the above methods need to be determined before we start our experiments. To be fair, for LSC-K, LSC-R, ESCG, SGL and FSCSWF, the number of anchors (or landmarks) m are selected from the same range under each specific data set. We select the optimal values of tradeoff parameters of SGL and FSCSWF from the same range: 10^{-4} to 10^4 . The optimal values of the number of nearest neighbors \tilde{k} of FSCSWF are selected from [2, 3, 4, 6, 8, 10, 15]. For LSC-K and LSC-R, the parameter r is set as 6 empirically according to [13]. For N-cut and ESCG, we use RBF kernel of the form $K = \exp(-(\|x_i - x_j\|_2^2)/(t \cdot d_{max}^2))$ where d_{max} is the maximum distance between samples to construct adjacency matrix and we turn the parameter t in the range [0.01, 0.1, 1, 10, 100].

The clustering performance evaluation metrics we employ are accuracy (ACC) [55], normalized mutual information (NMI) [56] and Purity [57]. The ACC is defined as the maximum overall accuracy rate among all possible permutation mappings between clusters and true class labels. The NMI is used to measure the similarity of clustering results by measuring the normalized statistical information shared between two distributions. Purity calculates the ratio of the number of correctly clustered samples against the total number of samples.

5.3. Comparison results.

5.3.1. Clustering performance on synthetic data sets. We use two synthetic data sets (Two-moon and Three-ring data sets) to test the clustering performance of K-means, SGL, and FSCSWF. The experimental results are visually shown in FIGURE 2 where different colors represent different clusters of data. We can obviously see from the results that our algorithm FSCSWF, which matches the data points into exactly two or three groups, can outperform K-means and SGL on both synthetic data sets. Especially, the FSCSWF achieves perfect clustering performance (ACC = 1, NMI = 1 and Purity = 1) on both synthetic data sets.

5.3.2. Clustering performance on practical benchmark data sets. We tune the parameters of each method under fair range to obtain the best clustering performance. The experimental results of our algorithm and other state-of-art methods on nine practical benchmark data sets are reported in TABLE 3. The results show that FSCSWF outperforms other methods in most cases. To be specific, FSCSWF improves SGL, ESCG, LSC-K by 5.37%, 6.18% and 8.83% on ACC; 6.32%, 7.23% and 8.89% on NMI; 4.61%, 4.59% and 5.41% on Purity in average, respectively. As for other methods, we could see that ESCG performs best on data set Coil20 but poorly on data set BA and MNIST; LSC-K has a relatively higher and more stable performance than LSC-R, which could result from the landmarks selected by K-means are more representative than random ways.

We run seven algorithms under the same environment and record the time cost for each data set in TABLE 3. Compared with K-means, N-cut, LSC-K, LSC-R, ESCG and SGL, FSCSWF

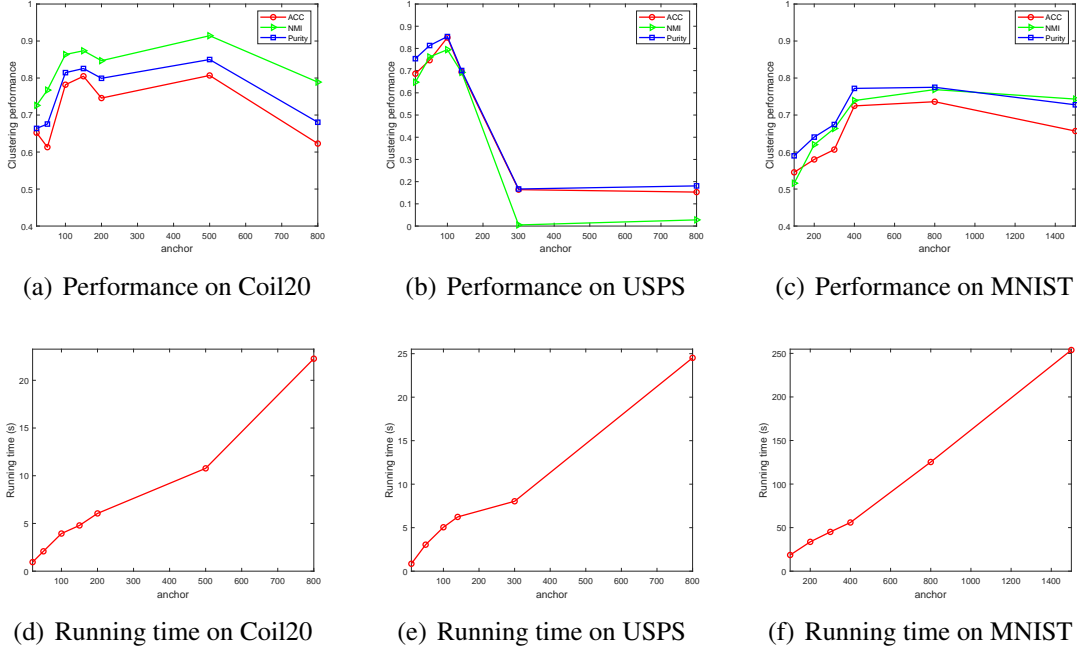


FIGURE 5. Influence of different anchor numbers on clustering performance and running time when \tilde{k} and λ are fixed.

could achieve better results with a reasonable time consumption in most cases, which illustrates the effectiveness and efficiency of our algorithm.

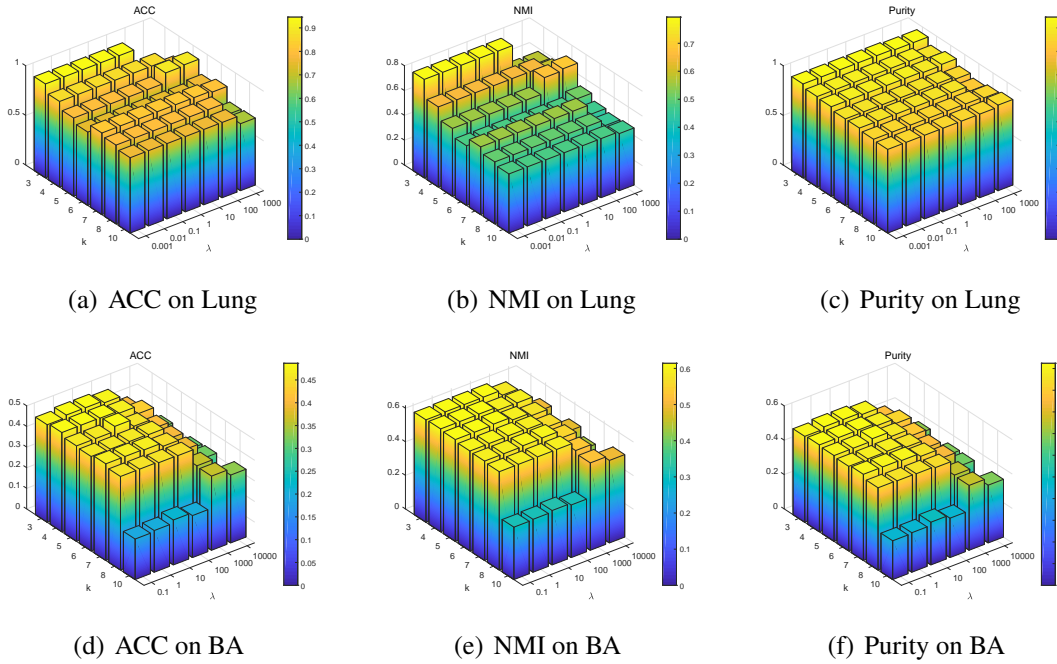


FIGURE 6. Influence of different values of tradeoff parameters on clustering performance when number of anchors are fixed.

5.3.3. Robustness against noise. In order to demonstrate the robustness of the FSCSWF against noise, we choose Coil20 data set as the experimental benchmark and select half of the image number of each class, 36 images, to add Gaussian noise via $\tilde{x} = x + N(0, \sigma_n^2)$, where σ_n means the standard deviation. The standard deviation varies from 0 to 80 and the examples are shown in FIGURE 4 visually.

To avoid randomness, we perform five experiments on the corrupted Coil20 data set and then take the average of the results. From the results shown in FIGURE 3 we could see that our method performs better than other methods in all levels of added noise and shows ideal robustness. Specifically, the average clustering performance (ACC, NMI, Purity) of FSCSWF in all levels of noise are 0.7916, 0.8596 and 0.8113 respectively, which improves SGL by 4.36%, 4.38% and 4.52% respectively and LSC-K by 6.49%, 6.59% and 8.02% respectively. Because the optimal weights of features are learnt, the samples become more discriminative which really contributes to the ability to resist noise of FSCSWF.

5.4. Parameters sensitivity and convergence analysis.

5.4.1. Number of anchors m . We choose three medium-size data sets: Coil20, USPS and MNIST to analyse the influence of the number of anchors m . FIGURE 5 shows the clustering performance and running time of FSCSWF on three data sets under various m . With the variety of m , the clustering performance of FSCSWF is relatively stable in Coil20 and MNIST while is quite sensitive in USPS data set. FSCSWF achieves optimal performance with a smaller amount of anchors compared to the number of data points, specifically on Coil20 ($m/n = 10.4\%$), USPS ($m/n = 4.98\%$) and MNIST ($m/n = 8\%$). This means that selecting enough anchors to construct bipartite graph is essential to final results while overmuch anchors will not be beneficial to the clustering performance. This phenomenon may results from the large amount of useless information brought by too much anchors. From FIGURE 5 we could see that as the number of anchors increases, running time ascends significantly, which illustrates that overmuch anchors also influence the efficiency of algorithm.

In a word, achieving the optimal results using a reasonable number of anchors reflects both effectiveness and efficiency of FSCSWF.

5.4.2. Tradeoff parameter \tilde{k} and λ . In this section, we analyse the sensitivity of FSCSWF with respect to two tradeoff parameters: the number of nearest neighbors \tilde{k} and λ . Taking Lung and BA data set as the research objects, the number of anchors are fixed and parameters \tilde{k} varies over the range of $[3, 4, 5, 6, 7, 8, 9, 10]$. Parameter λ varies over $[10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3]$ and $[10^{-1}, 1, 10, 10^2, 10^3, 10^4]$ on Lung and BA respectively. The results are exhibited in FIGURE 6. It can be seen that our method's performance is quite robust to the change of \tilde{k} and λ . Except for the NMI on Lung data set, which has a decline from 70% to 50% with \tilde{k} increasing, others are almost level off with the variety of two parameters, which illustrates the stability of FSCSWF in practice.

5.4.3. Convergence analysis. Apart from the above analysis, we provide convergent curves of FSCSWF on different data sets in FIGURE 7 to verify the convergence of our algorithm. The values of objective functions are normalized and show that the number of iteration on all practical data sets are less than 15, which demonstrates that our algorithm converges rapidly in practice.

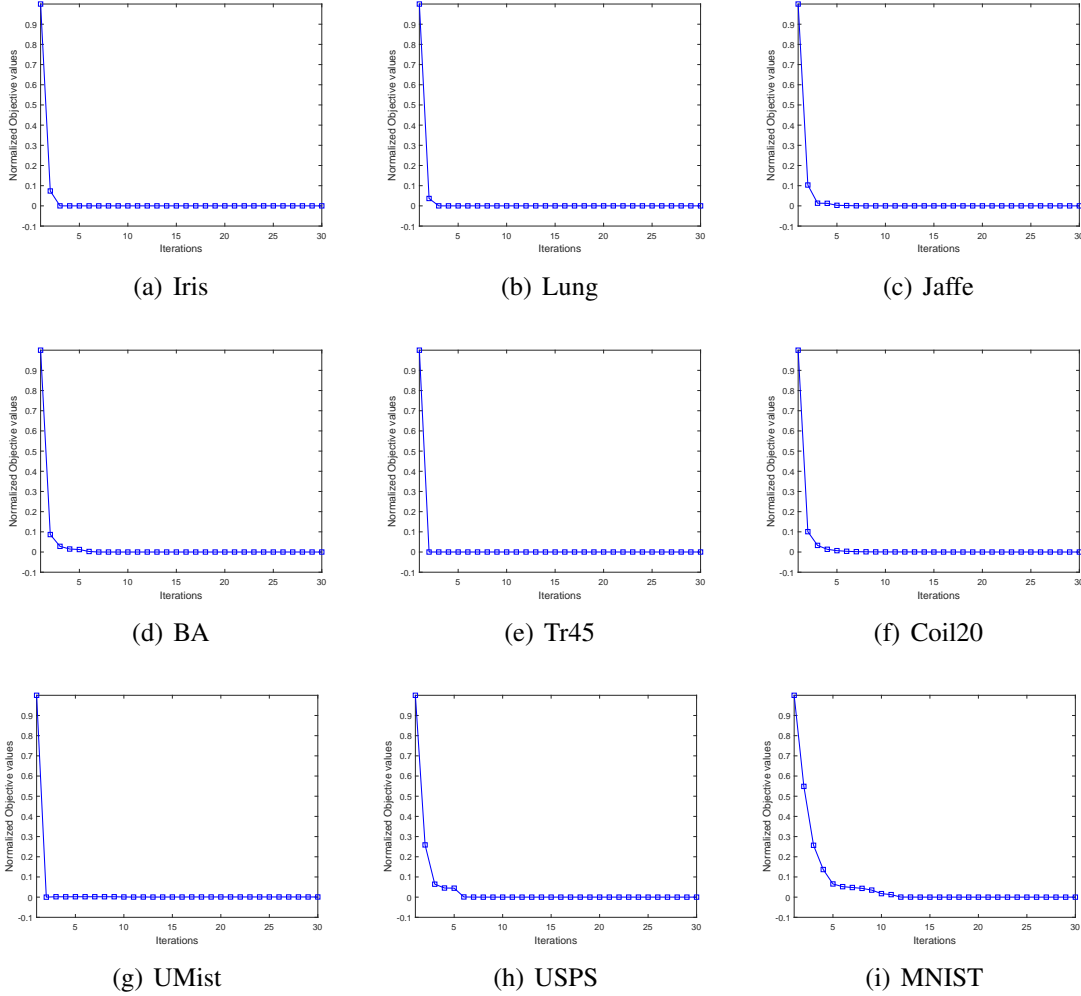


FIGURE 7. Convergence curves of FSCSWF on nine practical benchmark data sets.

6. CONCLUSION

In this paper, we propose a new fast spectral clustering with self-weighted features, FSCSWF, to achieve good clustering performance in a low computational complexity. In the framework of the FSCSWF, the weights of features and the similarity between anchors and samples are learnt in a local structure learning framework interactively, which makes the learnt similarity can measure the relationship between anchors and samples better. Meanwhile, the connectivity constraint is embedded to the framework of the FSCSWF to ensure the connected components of bipartite graph constructed by the learnt similarity can indicate clusters directly. The FSCSWF improves the clustering performance by assigning optimal weights for features so as to make the samples more discriminative in clustering. Besides, the FSCSWF takes advantage of anchors to construct bipartite graph, which makes the computational complexity of the FSCSWF is linear to the number of samples. Extensive experiments on synthetic and practical data sets illustrate the effectiveness and efficiency of the FSCSWF with respect to state-of-the-art methods.

Acknowledgements

The authors are grateful to the reviewers for useful suggestions which improved the contents of this paper. The work was supported by the National Natural Science Foundation of China (61772120).

REFERENCES

- [1] D. Horn , A. Gottlieb, Algorithm for data clustering in pattern recognition problems based on quantum mechanics, *Phys. Rev. Lett.* 88 (2001), 018702.
- [2] F. Nie, X. Dong, X. Li, Unsupervised and semisupervised projection with graph optimization, *IEEE Trans. Neural Netw. Learn. Syst.* 32 (2021), 1547-1559.
- [3] G. Gan, Application of data clustering and machine learning in variable annuity valuation, *Insurance Math. Econom.* 53 (2013), 795-801.
- [4] J. Yu, R. Hong, M. Wang, J. You, Image clustering based on sparse patch alignment framework, *Pattern Recogn.* 47 (2014), 3512-3519.
- [5] Z. Cai, X. Yang, T. Huang, W. Zhu, A new similarity combining reconstruction coefficient with pairwise distance for agglomerative clustering, *Info. Sci.* 508 (2020), 173-182.
- [6] J. MacQueen, Some methods for classification and analysis of multi-variate observations, In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281-297, 1967.
- [7] L. Hagen, A.B. Kahng, New spectral methods for ratio cut partitioning and clustering, *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 11 (1992), 1074-1085.
- [8] S. Guha, R. Rastogi, K. Shim, Cure: An efficient clustering algorithm for large databases, *ACM Sigmod Record*, 27 (1998), 73-84.
- [9] M. Ester, H.P. Kriegl, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, *KDD*, 96 (1996), 226-231.
- [10] A. Ng, M. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, *Advances in Neural Information Processing Systems*, 14 (2011), 849-856.
- [11] Q. Wang, Z. Qin, F. Nie, X. Li, Spectral embedded adaptive neighbors clustering, *IEEE Trans. Neural Netw. Learn. Syst.* 30 (2019), 1265-1271.
- [12] J. Shi, J. Malik, Normalized cuts and image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22 (2000), 888-905.
- [13] X. Chen, D. Cai, Large scale spectral clustering with landmark-based representation, In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 25, 2011.
- [14] S.C. Johnson, Hierarchical clustering schemes, *Psychometrika*, 32 (1967), 241-254.
- [15] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, *Science*, 344 (2014), 1492-1496.
- [16] Z. Guo, T. Huang, Z. Cai, W. Zhu, A new local density for density peak clustering, In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 426-438, 2018.
- [17] R. Li, X. Yang, X. Qin, W. Zhu, Local gap density for clustering high-dimensional data with varying densities, *Knowledge-Based Systems*, 184 (2019), 104905.
- [18] X. Peng, H. Tang, L. Zhang, Z. Yi, S. Xiao, A unified framework for representation-based subspace clustering of out-of-sample and large-scale data, *IEEE Transactions on Neural Networks and Learning Systems*, 27 (2015), 2499-2512.
- [19] X. Peng, Z. Yu, Z. Yi, H. Tang, Constructing the l2-graph for robust subspace learning and subspace clustering, *IEEE Transactions on Cybernetics*, 47 (2017), 1053-1066.
- [20] X. Peng, H. Zhu, J. Feng, C. Shen, H. Zhang, J.T. Zhou, Deep clustering with sample-assignment invariance prior, *IEEE Transactions on Neural Networks and Learning Systems*, 31 (2020), 4857-4868.
- [21] M. Filippone, F. Camastra, F. Masulli, S. Rovetta, A survey of kernel and spectral methods for clustering, *Pattern Recognition* 41 (2008), 176-190.
- [22] U.V. Luxburg, A tutorial on spectral clustering, *Stat. Comput.* 17 (2007), 395-416.
- [23] F. Nie, X. Wang, H. Huang, Clustering and projected clustering with adaptive neighbors, *Association for Computing Machinery*, New York, 2014.

- [24] F. Nie, X. Wang, M. Jordan, H. Huang, The constrained laplacian rank algorithm for graph-based clustering, In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 2016.
- [25] Z. Kang, X. Zhao, C. Peng, H. Zhu, J.T. Zhou, X. Peng, W. Chen, Z. Xu, Partition level multiview subspace clustering, *Neural Netw.* 122 (2020), 279-288.
- [26] C. Zhang, H. Fu, Q. Hu, X. Cao, Y. Xie, D. Tao, D. Xu, Generalized latent multi-view subspace clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42 (2018), 86-99.
- [27] A. Choromanska, T. Jebara, H. Kim, M. Mohan, C. Monteleoni, Fast spectral clustering via the nyström method, In: *International Conference on Algorithmic Learning Theory*, pp. 367-381, 2013.
- [28] B. Chen, B. Gao, T.Y. Liu, Y.F. Chen, W.Y. Ma, Fast spectral clustering of data using sequential matrix compression, In: *European Conference on Machine Learning*, pp. 590-597, 2006.
- [29] H. Wang, F. Nie, H. Huang, F. Makedon, Fast nonnegative matrix tri-factorization for large-scale data co-clustering, In: *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.
- [30] X. Yang, W. Yu, R. Wang, G. Zhang, F. Nie, Fast spectral clustering learning with hierarchical bipartite graph for large-scale data, *Pattern Recognition Lett.* 130 (2020), 345-352.
- [31] Y. Li, F. Nie, H. Huang, J. Huang, Large-scale multi-view spectral clustering via bipartite graph, In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, 2015.
- [32] R. Wang, F. Nie, W. Yu, Fast spectral clustering with anchor graph for large hyperspectral images, *IEEE Geoscience and Remote Sensing Letters* 14 (2017), 2003-2007.
- [33] Z. Kang, Z. Lin, X. Zhu, W. Xu, Structured graph learning for scalable subspace clustering: From single view to multiview, *IEEE Transactions on Cybernetics*, 2021. DOI: 10.1109/TCYB.2021.3061660.
- [34] F. Nie, D. Wu, R. Wang, X. Li, Self-weighted clustering with adaptive neighbors, *IEEE Transactions on Neural Networks and Learning Systems*, 31 (2020), 3428-3441.
- [35] R. Wang, F. Nie, Z. Wang, F. He, X. Li, Scalable graph-based clustering with nonnegative relaxation for large hyperspectral image, *IEEE Transactions on Geoscience and Remote Sensing*, 57 (2019), 7352-7364.
- [36] F.R.K. Chung, F.C. Graham, *Spectral Graph Theory*, CBMS Number 92, AMS Publications, 1997.
- [37] W. Liu, J. He, S.F. Chang, Large graph construction for scalable semi-supervised learning, In: *International Conference on Machine Learning*, 2010.
- [38] X. Yang, G. Lin, Y. Liu, F. Nie, L. Lin, Fast spectral embedded clustering based on structured graph learning for large-scale hyperspectral image, *IEEE Geoscience and Remote Sensing Letters*, 2020. doi: 10.1109/LGRS.2020.3035677.
- [39] S. Huang, Z. Xu, I.W. Tsang, Z. Kang, Auto-weighted multi-view co-clustering with bipartite graphs, *Info. Sci.* 512 (2020), 18-30.
- [40] F. Nie, X. Wang, C. Deng, H. Huang, Learning a structured optimal bipartite graph for co-clustering, In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 4132-4141, 2017.
- [41] B. Mohar, *The laplacian spectrum of graphs. graph theory, combinatorics and applications*, vol. 2 (kalamazoo, mi, 1988), pp. 871-898, Wiley, New York, 1991.
- [42] K. Fan, On a theorem of weyl concerning eigenvalues of linear transformations i, *Proceedings of the National Academy of Sciences of the United States of America*, 35 (1949), 652.
- [43] Y. Liu, Y. Cai, X. Yang, F. Nie, W. Ye, Fast adaptive neighbors clustering via embedded clustering, *Neurocomputing* 399 (2020), 331-341.
- [44] F. Nie, G. Cai, X. Li, Multi-view clustering and semi-supervised classification with adaptive neighbours, In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 2017.
- [45] J.C. Bezdek, R.J. Hathaway, Convergence of alternating optimization, *Neural, Parallel & Scientific Computations*, 11 (2003), 351-368.
- [46] A. Asuncion, D.J. Newman, *Uci Machine Learning Repository*, 2007.
- [47] D. Singh, et al., Gene expression correlates of clinical prostate cancer behavior, *Cancer Cell*, 1 (2002), 203-209.
- [48] M. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, Coding facial expressions with gabor wavelets, In: *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 200-205, 1998.

- [49] D.B. Graham, N.M. Allinson, Characterising virtual eigensignatures for general purpose face recognition, In: Wechsler H., Phillips P.J., Bruce V., Soulié F.F., Huang T.S. (eds) Face Recognition. NATO ASI Series (Series F: Computer and Systems Sciences), vol 163. Springer, Berlin, 1998.
- [50] H. Li, X. Ye, A. Imakura, T. Sakurai, Ensemble learning for spectral clustering, In: 2020 IEEE International Conference on Data Mining (ICDM), pp. 1094-1099, 2020.
- [51] S.A. Nene, S.K. Nayar, H. Murase, Columbia object image library (coil-20), Technical Report, 1996.
- [52] J.J. Hull, A database for handwritten text recognition research, IEEE Transactions on Pattern Analysis and Machine Intelligence, 16 (1994), 550-554.
- [53] L. Deng, The mnist database of handwritten digit images for machine learning research [best of the web, IEEE Signal Processing Magazine 29 (2012), 141-142.
- [54] J. Liu, C. Wang, M. Danilevsky, J. Han, Large-scale spectral clustering on graphs, In: Twenty-Third International Joint Conference on Artificial Intelligence, Citeseer, 2013.
- [55] M. Wu, B. Schölkopf, A local learning approach for clustering, Adv. Neural Info. Processing Sys. 19 (2006), 1529-1536.
- [56] A. Strehl, J. Ghosh, Cluster ensembles—a knowledge reuse framework for combining multiple partitions, J. Machine Learn. Res. 3 (2002), 583-617.
- [57] H. Schütze, C.D. Manning, P. Raghavan, Introduction to Information Retrieval, vol. 39, Cambridge University Press Cambridge, 2008.