Fast Self-Supervised Clustering With Anchor Graph

Jingyu Wang, *Member, IEEE*, Zhenyu Ma, Feiping Nie, *Member, IEEE*, and Xuelong Li, *Fellow, IEEE*

***Abstract*— Benefit from avoiding the utilization of labeled samples, which are usually insufficient in the real world, unsupervised learning has been regarded as a speedy and powerful strategy on clustering tasks. However, clustering directly from primal data sets leads to high computational cost, which limits its application on large-scale and high-dimensional problems. Recently, anchor-based theories are proposed to partly mitigate this problem and field naturally sparse affinity matrix, while it is still a challenge to get excellent performance along with high efficiency. To dispose of this issue, we first presented a fast semisupervised framework (FSSF) combined with a balanced K-means-based hierarchical K-means (BKHK) method and the bipartite graph theory. Thereafter, we proposed a fast self-supervised clustering method involved in this crucial semisupervised framework, in which all labels are inferred from a constructed bipartite graph with exactly k connected components. The proposed method remarkably accelerates the general semisupervised learning through the anchor and consists of four significant parts: 1) obtaining the anchor set as interim through BKHK algorithm; 2) constructing the bipartite graph; 3) solving the self-supervised problem to construct a typical probability model with FSSF; and 4) selecting the most representative points regarding anchors from BKHK as an interim and conducting label propagation. The experimental results on toy examples and benchmark data sets have demonstrated that the proposed method outperforms other approaches.**

***Index Terms*— Bipartite graph, label propagation, self-supervised learning, semisupervised framework, special selection.**

I. Introduction

LEARNING-BASED methods have been regarded as one of the most prominent researching strategies in machine learning [1]–[3], pattern recognition [4]–[6], and data mining [7], [8] scenarios. Generally, the learning techniques are divided into three categories: supervised learning [9], semisupervised learning (SSL) [10], and unsupervised learning [11]. The first group fully utilizes labeled samples, such as support vector machine [12], [13] and linear discriminant analysis [14], [15]. The second group merely exploits a few labeled samples, where the rest are all substituted with unlabeled sample points. The third group completely uses unlabeled samples, in which principle component analysis [16], [17] is a fundamentally typical dimensionality reduction (DR) [18] approach.

Since the labeled data in the real world are extremely deficient, learning label information is treated as an indispensable task in classification [19] and regression missions [20]. However, as the acquisition of labeled samples is tedious and laborious, sometimes even impossible, prior knowledge is usually not available when dealing with practical problems [21]. Furthermore, label annotations of low quality also have a serious impact on the performance of algorithms. Consequently, semisupervised and unsupervised learning are more generally favored than supervised learning.

SSL leverages both limited labeled data and abundant unlabeled data to learn a more efficient model. For instance, the general semisupervised model combined with novel class discovery [22] is proposed to predict outliers in unknown data. To tackle large graph problems, the neighbor graph construction with SSL is introduced [23] and large-scale SSL [24] in classification is also proposed to enhance robustness with $l_{2,p}$-norm. However, SSL needs to utilize labeled data after all. Hence, compared with SSL, unsupervised learning possesses more efficiency in most applications, which predicts labels of unknown data without auxiliary information.

Clustering is playing an important role in unsupervised learning, which aims to classify samples into different groups on the basis of certain criteria. Many classical clustering methods have been introduced, including K-means [25], fuzzy K-means [26], spectral clustering (SC) [27], hierarchical clustering [28], and nonnegative matrix factorization [29]. The target of K-means is to update centers of all clusters until satisfying the convergence condition. Although K-means is commonly used to conduct fast clustering and evaluate the performance of DR, it is applicable only in convex distribution. Comparatively, SC is a graph-based clustering technique seeking the optimal graph-based decomposition, which is capable of tackling more complex data structures compared with K-means. Recently, numerous researchers have proposed many innovative and comprehensive analyses in terms of optimizing SC, such as spectral embedding clustering combined with spectral embedding [30] and nonnegative SC [31]. However, the construction of similarity matrix on the spectral graph in these approaches needs to select appropriate criteria

Manuscript received April 20, 2020; revised September 22, 2020 and January 10, 2021; accepted January 27, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61976179 and Grant 61502391; in part by the Nation Key Research and Development Program under Grant 2018XXX08241041 and Grant 2018YFB1305702; in part by the Fundamental Research Funds for the Central Universities under Grant 3102019HTXM005, Grant 3102017HQZZ003, and Grant 3102019HTXS001; and in part by the Key Industrial Innovation Chain Project in Industrial Domain of Key Research and Development Program of Shaanxi Province under Grant 2018ZDCXLGY030203. *(Corresponding author: Feiping Nie.)*

Jingyu Wang is with the School of Astronautics, School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, China (e-mail: jywang@nwpu.edu.cn).

Zhenyu Ma is with the School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: zhenyu.ma@mail.nwpu.edu.cn).

Feiping Nie and Xuelong Li are with the School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, China (e-mail: feipingnie@gmail.com; xuelong_li@nwpu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3056080.

Digital Object Identifier 10.1109/TNNLS.2021.3056080

strictly and also decide how to build the affinity matrix. The introduction of the Gaussian kernel function makes SC not parameter-free, which is time-consuming when solving large-scale problems on account of the overall computational complexity of general SC is $O(n^2d)$, where n and d is the number of samples and features, respectively. Thus, landmark-based sparse representation strategy [32] is proposed to apply to SC and construct a more sparse similarity graph.

As a special kind of unsupervised learning pattern, self-supervision-based learning [33], [34] has attracted much attention from researchers in recent years to further improve clustering performance. Generally, the self-supervised clustering process mainly adopts auxiliary tasks to explore and acquire its own supervision information from existed unsupervised data. Thereafter, the obtained supervised signals will be utilized for subsequent training or supervised learning. In the past two years, a self-supervised clustering module has been widely used in various deep learning networks. For instance, Zhang *et al.* [35] proposed a self-supervised convolutional subspace clustering network to simultaneously achieve feature learning and subspace clustering. In order to cope with the outliers problem in multiview clustering, Sun *et al.* [36] proposed a self-supervised deep multiview subspace clustering algorithm, in which the clustering results and affinity matrix are mutually trained by an integrated deep learning framework to obtain superior clustering performance under multiviews.

Comparatively, in recent years, there are a few researches on self-supervision based clustering in the field of machine learning. Ye *et al.* [37] proposed an affinity learning-based self-supervised diffusion (SSD) for SC to deal with the sensitivity of SC to a fixed affinity matrix, where the clustering results in each iteration provides supervisory signals for the diffusion process. Furthermore, a novel self-supervised clustering method is also proposed to effectively discover new user intentions [38]. However, the above-mentioned few self-supervised clustering studies in the field of machine learning still have potential limitations. Concretely, attributed to the property of traditional SC, the iterative process in SSD might bring serious computational burden when tackling the large-scale problem. The study for new user intentions still leverages a small number of labeled samples in this clustering framework, which is essentially not in the realm of self-supervision.

Thence, in order to acquire clustering results with higher quality and faster processing speed, we focus on performing self-supervised clustering tasks from the perspective of semisupervised label propagation. More specifically, it is pondered whether it is possible to use sparse theory to integrate semisupervised ideas with unsupervised methods in clustering tasks. Thus, we propose a fast clustering technique referred to as fast self-supervised clustering (FSSC) with anchor graph in this article, which integrates a fast semisupervised framework (FSSF) with unsupervised methods in clustering tasks. The procedures of our algorithm are listed as follows.

1) We utilize the efficient BKHK algorithm to find anchor set from the original data set, which has lower computational complexity and better effectiveness in comparison with K-means. Therefore, this anchor generated technique is relatively practical especially dealing with large-scale problems possessing complex data structures.
2) Using the bipartite graph learned from samples and anchors is significant to build sparse similarity matrix W, where there is only one parameter k meaning k nearest anchors for each sample.
3) Inspired by the SSL framework, we adopt virtual labels of representative anchors from BKHK to get soft label matrix by fulfilling semisupervised clustering problem utilizing the proposed FSSF. The soft label matrix expresses the meaning of probabilities of the data belonging to classes in virtual labels.
4) The novel special selection strategy is proposed to find out the most representative points, the number of which is equivalent to the number of real classes c. This procedure is relatively significant to select ultimate real labels. The SSL is conducted to operate label propagation at the end of our method.

It is essential to emphasize some aspects of our method.

1) The major measures to accelerate the entire algorithm are threefold: the BKHK algorithm, the construction of a naturally sparse bipartite graph, and inversion lemma.
2) We take the advantage of fake labels of each anchor point to operate self-supervised learning, which is exactly an unsupervised learning strategy with the proposed FSSF.

The rest of this article is scheduled as follows. Related works and notations are discussed in Section II. The details of FSSF are significantly demonstrated in Section III. The proposed method FSSC with FSSF is expounded in Section IV. Corresponding experiments on toy and benchmark data sets are elaborated in Section V. Finally, we give the conclusion and prospective works in Section VI.

II. Notations and Related Works

In this section, some related works, including spectral graph theory and some pivotal graph-based clustering techniques, will be briefly reviewed first. Meanwhile, some relative notations will be described for a vivid description of these algorithms and theories. The BKHK algorithm will be roughly described at last for the preparation of our main method.

A. Graph-Based Learning

In recent studies, graph-based clustering [39] techniques have been widely and mainly applied in an unsupervised manner. First, we denote the data matrix $X = [x_1; x_2; \ldots; x_n] \in \mathbb{R}^{n\times d}$, where n and d are the number of samples and dimensionality, respectively. $x_i \in \mathbb{R}^d$ expresses the ith data point, which is a row vector. Supposed that e_{ij} means the similarity weight to connect x_i and x_j, Euclidean distance is usually adopted to be simple. Second, these approaches employ a graph $G = (V, E)$ to model the original data set, where $V = X$ is the graph vertex set and E is the edge set. Associated with each edge $e_{ij} \in E$, W_{ij} is a nonnegative weight indicating the similarity between x_i and x_j. Generally, the graph can be divided into directed graph ($W_{ij} \neq W_{ji}$) [40] and undirected graph ($W_{ij} = W_{ji}$) [41]. In this article, we focus on the

undirected graph with local neighborhood information and next a brief review on commonly used similarity graphs is provided.

1) The ε-Neighborhood Graph: For two random data points, if distances between them are smaller than ε, we connect them with the same scale, which is regarded as an unweighted graph. However, the scale of ε is not easy to adjust based on the density distribution of primal data sets.

2) k-Nearest Neighbor Graph: We connect vertex x_i with data point x_j if x_j belongs to the K-nearest neighbors (KNN) of x_i, which leading to a problem that the similarity graph will be not symmetric unless we operate $W' = (W^{\mathrm{T}} + W)/2$ [42]. While we impose a stronger condition which means two sample points are nearest neighbors to each other, the corresponding graph is referred to as the *mutual KNN graph*.

3) The Fully Connected Graph: Full points are simply connected with nonnegative similarity values with each other. For instance, the Gaussian kernel function [43]

$$W_{ij} = W_{ji} = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right) \tag{1}$$

can be utilized, where the parameter σ is the width of Gaussian function. This parameter plays a similar role as the parameter ε in occasion of *the ε-neighborhood graph*. We are capable of utilizing various criteria, including Euclidean distance, Mahalanobis distance, Minkowski distance, and cosine similarity for improvement of graph.

4) Adaptive Gaussian method graph: In order to pursue the parameter-free measurement, the self-tuning graph with Gaussian function [44] is proposed. The details of the similarity between two vertex x_i and x_j can be written as

$$W_{ij} = \exp\left(\frac{-d^2(x_i, x_j)}{\sigma_i \sigma_j}\right) \tag{2}$$

where σ_i represents the local scaling for x_i, which is the distances between vertex x_i and its farthest neighborhood. Therefore, this parameter can be formulated as

$$\sigma_i = d(x_i, x_k) \tag{3}$$

where x_k is the kth neighbor for x_i. This strategy will be also used in following bipartite graph construction.

To consider the connection between each point and avoid high computational complexity caused by too much iterations using K-means in large-scale data sets, classical SC methods are divided into two steps: 1) solving a relaxed continuous optimization problem [we denote $tr(\cdot)$ is the trace operator]

$$H' = \arg\min_{H^{\mathrm{T}}H=I} \mathrm{tr}(\mathrm{HLH}^{\mathrm{T}}) \tag{4}$$

where L is the graph Laplacian matrix followed by affinity matrix W, to obtain a constrained matrix H and 2) applying K-means or spectral rotations to build indicator matrix of clusters.

However, on the one hand, the affinity matrix directly obtained from original data incorporates noisy and redundant information, which extremely affects the clustering performance. In order to dispose of this issue, Zhu *et al.* [45] adopted low-dimensional subspace and low-rank constraint dynamically to learn similarity matrix effectively and efficiently, the idea of which has been utilized in a novel unsupervised feature selection method [46] to preserve local and global structure simultaneously. In multiview clustering field, since the flaws of fixed common affinity matrix, one-step multiview SC (OMSC) method [47] was proposed with discriminative weights in diverse views, where mapping matrix and affinity matrix for different views are optimized dynamically to acquire better clustering performance. Besides, a fuzzy and robust multiview clustering model [48] was proposed to strengthen the stability of each view, where a more sparse affinity matrix with more abundant and helpful information was obtained in this framework. Furthermore, in multitask SC [49], affinity matrix and mapping function are simultaneously learned with mutual improvement to effectively explore intertask correlation.

On the other hand, since the extra heat kernel parameter and singular value decomposition in affinity matrix from primal data, classical SC is generally not efficient to tackle a large-scale problem, the computational complexity of which is $O(n^2d + n^3)$. Hence, anchor-based theory [50], [51] has been introduced to generate bipartite graph connecting anchor layer and sample layer in recent years, where anchors are a series of representative points that roughly cover entire sample points. Many scholars have utilized bipartite graphs to build a naturally sparse similarity matrix by adjusting the number of anchors and their nearest neighbors. Zhu *et al.* [52] proposed a fast SC to construct a parameter-free large graph with effective neighbor assignment. Subsequently, double anchor layers and even hierarchical bipartite graph were proposed and utilized to explore more explicit connection relationship among all samples, e.g., Representative Point-based Spectral Clustering (RPSC) [53] and SC based on Hierarchical Bipartite Graph (SCHBG) [54]. Random walk-based Laplacian matrix [55] was also proposed to balance the anchors and samples and improve clustering performance. Furthermore, recently, an anchor-based graph has been combined with SSL to efficiently deal with large-scale problem [56]–[58], which are able to effectively dispose of the out-of-sample problem as well.

In this article, inspired by the integration between anchor-based theory and SSL, the proposed FSSC method with FSSF first and quickly obtain the crucial probability model between anchors and samples, which represents the belonging relationship about fake labels for each sample. The novel special selection strategy is then conducted to choose the most representative points with the best quality by extracting the maximum score for each sample, which depicts the probability that the sample point belongs to the outlier. Terminally, clustering results can be gained by the label propagation process. As an essential part to generate anchor points, the BKHK algorithm will be roughly described later.

B. Balanced K-Means Based Hierarchical K-Means

In the past few years, the anchor-based theory has been introduced to accelerate the construction of a connected graph. For a specific quantity of anchors, generally, the number of anchors is far less than that of samples, while too sparse

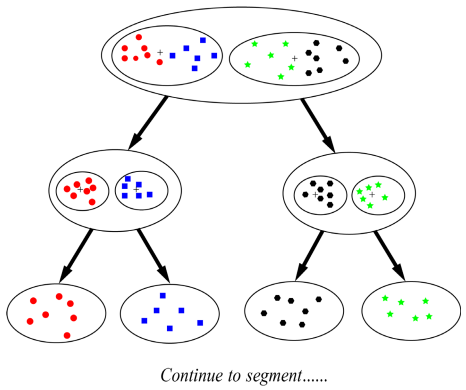

Fig. 1. Similar to cell division, BKHK performs dichotomy at each data group.

representative anchors are not capable of meeting the satisfying performance. On the contrary, dense enough anchors might bring extremely high computational cost when we deal with the large-scale problem. In addition, for concrete selection methods of anchors, on the one hand, though we are able to randomly select representative points to fast view the local structure, it is difficult to reach satisfying performance by the constructed graph from original samples and these randomly selected anchors. On the other hand, utilizing K-means in BKHK can obtain desirable representative anchors with ideal performance; however, the computational complexity is extremely high when solving large-scale problems. Thus, our overall method starts with a speedy and steady BKHK algorithm [52] to find representative anchors.

Fig. 1 vividly shows the entire course of the BKHK algorithm. Similar to the cell division process, this algorithm adopts a balanced binary tree structure, balances to segment almost equal number samples into two clusters, and hierarchically processes each obtained new clusters (if p hierarchies are generated, 2^p representative anchors will be obtained to constitute anchor set U). Since BKHK is really efficient to apply to large data sets that have high dimensions or large sample points, it is adopted to accelerate the graph learning to get the anchor set U as an interim. To simplify the construction of label matrix Y, the index of all anchors learned from primal data sets are recorded.

III. Fast Semisupervised Framework

In this section, the details of the FSSF will be described. First, we will introduce a general SSL framework for clustering tasks. Second, the acceleration strategy for this framework on similarity matrix then will be demonstrated. Since the labels for labeled points and the rest points should be treated differently, the parameter α_l and α_u related to regularization parameter μ will be introduced, the meaning of which will be explained later.

A. General Semisupervised Framework

Consider a graph $G = (V, E)$ with V nodes corresponding to n sample points, in which the first l nodes denote l labeled data points and the rest u nodes denote abundant u unlabeled data points.

Assuming that the number of true class is c, we denote the initial label matrix $Y = [Y_1^{\mathrm{T}}; Y_2^{\mathrm{T}}; \ldots; Y_n^{\mathrm{T}}] \in \mathbb{R}^{n\times(c+1)}$, where $Y_i \in \mathbb{R}^{c+1}(1 \le i \le n)$ are column vectors and the motivation of the $(c+1)$th column is to evaluate the probability of belonging to outlier for each sample point. For l labeled data points, $Y_{ij} = 1$ if the ith sample belongs to jth class and $Y_{ij} = 0$ otherwise. Comparatively, for u unlabeled data points, we initially set $Y_{ij} = 1$ if $j = c + 1$ and $Y_{ij} = 0$ otherwise. The soft label matrix $F = [F_1^{\mathrm{T}}; F_2^{\mathrm{T}}; \ldots; F_n^{\mathrm{T}}] \in \mathbb{R}^{n\times(c+1)}$ is a true probability model and optimized objective matrix, where $F_i \in \mathbb{R}^{c+1}(1 \le i \le n)$ are column vectors and F_{ij} represents the probability of the ith sample belonging to jth class. Generally, the meaning of soft label matrix F is very useful for postprocessing.

Assuming W_{ij} is the similarity matrix among n data points with the Gaussian kernel function and its corresponding degree matrix D is a diagonal matrix, the ith entry of which is $d_i = \sum_j W_{ij}$. Let us denote $\|\cdot\|_F^2$ represents the Frobenius norm of matrix, i.e., $\|M\|_F^2 = \mathrm{tr}(M^{\mathrm{T}}M)$. According to the constructed graph, the general semisupervised framework can be formulated as the following cost function:

$$\mathcal{Q}(F) = \sum_{i,j=1}^{n} W_{ij}\|F_i - F_j\|_F^2 + \sum_{i=1}^{n} \mu_i \|F_i - Y_i\|_F^2. \tag{5}$$

The first term in this cost function is a clustering term, which can reach the target that similar samples have similar labels. For details, if the similarity between ith data point and jth data point is extremely high, in order to solve the minimum of the cost function, F_i can be set almost equal to F_j. If the similarity between them is almost zero or equal to zero, the constraint to these two samples will be relatively weak. We denote that $\mu_i > 0$ is a regularization parameter for each data point. Accordingly, the second term is a regularization term for labels to measure the discrepancy between obtained soft labels F_i and primal labels Y_i. On the one hand, if μ_i is deployed to be zero, the label constraint will be invalid and this problem will be purely regarded as a label propagation process. On the other hand, if μ_i is employed as infinitely large, the initial label Y_i will be maintained. The concrete employment for different samples will be elaborated when two novel parameters are introduced subsequently.

To cope with this problem, we set corresponding SSL model as

$$\min_F \sum_{i,j=1}^{n} W_{ij}\|F_i - F_j\|_F^2 + \sum_{i=1}^{n} \mu_i \|F_i - Y_i\|_F^2. \tag{6}$$

Then, we try to deform the Frobenius norm and gain matrix form of the problem as

$$\min_F \sum_{i,j=1}^{n} W_{ij}[(F_i - F_j)^{\mathrm{T}}(F_i - F_j)] + \sum_{i=1}^{n} \mathrm{tr}[(F_i - Y_i)^{\mathrm{T}}U(F_i - Y_i)] \tag{7}$$

where U is a $n\times n$ diagonal matrix, the ith entry of which is μ_i. The problem can be transformed through identity deformation as

$$\min_F \mathrm{tr}[F^{\mathrm{T}}LF] + \mathrm{tr}[(F - Y)^{\mathrm{T}}U(F - Y)] \tag{8}$$

where $L = D - W$ is referred as an unweighted Laplacian matrix.

Let us denote the optimized function in (8) is $\mathcal{P}(F)$. Since the optimal solution of problem (8) satisfies that the derivative of $\mathcal{P}(F)$ for F is equal to zero, we could gain

$$\left(\frac{\partial \mathcal{P}(F)}{\partial F}\right)|_{F=F^*} = 2LF^* + 2U(F^* - Y) = 0. \tag{9}$$

Through simplification, the optimal solution F^* can be obtained as

$$F^* = (L + U)^{-1}UY \tag{10}$$

where the sum of each row of F^* is equal to 1, which represents a classical probability model for easier follow-up data processing. [We will see the strict proof of probability model following (12) when two novel parameters have been introduced.]

As for the evaluation of probability belonging to outliers for each sample point, which will be demonstrated in the following parts, two parameters α_l and α_u should be introduced to simplify the parameter-setting process.

Since the introduction of evaluation strategy, we need to set different constraints for labeled samples and unlabeled samples. Generally, for data point x_i, whether x_i is labeled or unlabeled, the value of corresponding α is calculated by

$$\alpha_i = d_i/(d_i + \mu_i) \tag{11}$$

where d_i is the ith element of degree matrix D obtained from W and α_i is determined by d_i and regularization parameter μ_i. On this condition, the value range of α_i is limited into $[0, 1]$ which is really efficient for parameter selection. And the corresponding model in framework of certain value on α_i will be specifically illustrated.

Meanwhile, $\beta_i = \mu_i/(d_i + \mu_i)$ is introduced to satisfy $\beta_i = 1 - \alpha_i$. Assuming matrix I_α and I_β, we could easily know that $I_\alpha = I - I_\beta$, where I is a $n \times n$ identity matrix and I_α is a $n \times n$ diagonal matrix with the ith entry being α_i.

Defining $P = D^{-1}W$, (10) becomes

$$\begin{aligned} F^* &= (D - W + U)^{-1}UY \\ &= (I - D^{-1}W + D^{-1}U)^{-1}(D^{-1}U)Y \\ &= (I_\alpha - I_\alpha D^{-1}W + I_\beta)^{-1}I_\beta Y \\ &= (I - I_\alpha P)^{-1}I_\beta Y \end{aligned} \tag{12}$$

which is the ultimate solution of soft label matrix F in the general semisupervised framework.

Based on the concrete constructed details of P and Y, it can be easily found that $P\mathbf{1}_n = \mathbf{1}_n$ and $Y\mathbf{1}_{c+1} = \mathbf{1}_n$, where $\mathbf{1}_n \in \mathbb{R}^{n\times 1}$ and $\mathbf{1}_{c+1} \in \mathbb{R}^{(c+1)\times 1}$ indicate a column vector with elements are all one. Combined with the trait of I_α and I_β, we will have

$$\begin{aligned} I_\alpha P\mathbf{1}_n + I_\beta Y\mathbf{1}_{c+1} = \mathbf{1}_n &\Rightarrow I_\beta Y\mathbf{1}_{c+1} = (I - I_\alpha P)\mathbf{1}_n \\ &\Rightarrow (I - I_\alpha P)^{-1}I_\beta Y\mathbf{1}_{c+1} = \mathbf{1}_n \end{aligned} \tag{13}$$

which embodies that the obtained solution F_* in this semisupervised framework possesses that characteristic of probability model.

Furthermore, the difference in concrete value and corresponding meaning between α_l and α_u will be elaborated and discussed as follows.

1) Assuming a labeled data point x_i at first, when we absolutely guarantee the validity of the initial label, this label should remain unchanged, where α_i will be set to be zero. In this condition, μ_i should be deployed to be large enough in (11), which will promote the effect of fitting term on the problem (6) and contributing to complete fixation of the label of x_i. Otherwise, we should make α_i to be positive value to make the initial label adjustable, which can effectively solve the noises in initial labels.
2) Second, for one unlabeled data point x_j, if we ensure that it is impossible to exist novel class in data set, which means every sample all belongs to assumed c classes, α_j will be set to be 1 and the value μ_j is low enough compared with d_j, where the optimization only conduct clustering term according to the primitive graph. Generally, we set α_i to be a relatively large value but lower than 1, since the number of outliers is usually low on earth.

B. Acceleration on Similarity Matrix W

However, the similarity matrix W in (12) purely constructed by the Gaussian kernel function is directly gained from primal data set X, which contributes that the solution process of general semisupervised clustering framework is not fast enough. Therefore, in order to accelerate general framework, we consider to utilize anchor set U from BKHK to construct naturally sparse bipartite graph [59] to build connection between anchor set U and original data set X. We assume that the number of anchors is m and the objective bipartite graph matrix is $B = [b_1^{\mathrm{T}}; b_2^{\mathrm{T}}; \ldots; b_n^{\mathrm{T}}] \in \mathbb{R}^{n\times m}$, which means the similarity between samples and representative anchors. The original problem can be formulated as

$$\min_{b_i\mathbf{1}=1, b_i\geq 0} \sum_{j=1}^{n} h_{ij}b_{ij} + \gamma \sum_{j=1}^{n} b_{ij}^2 \tag{14}$$

where $h_{ij} = \|x_i - y_j\|_2^2$, which represents the square of distance between ith sample and jth anchor using Euclidean distance to be simple. The first term is the regularization term, which means the smoothness between anchors and original samples. The second term is the sparse term, where B could become as sparse as possible if the value of γ is set to be large enough. Thus, the problem (14) becomes

$$\max_{\gamma, \|b_i\|_0 = k} = \gamma \tag{15}$$

where k embodies the connected components for each original sample in bipartite graph matrix B, which is a nonzero integer and similar to the parameter k in the KNN algorithm. The value of k varies on different data sets owing to distinct data distribution itself. In general, the parameter k is set from 3 to 30.

There exists two constraints of this problem: The first is the equality constraint $b_i\mathbf{1} = 1$, which avoids the appearance

of all zero vector; the second is inequality constraint $b_i \geq 0$, since the B_{ij} symbolizes the similarity meaning. Thus, through Lagrange solution we have the bipartite graph

$$\hat{b}_{ij} = \begin{cases} \dfrac{h_{i,k+1} - h_{ij}}{k h_{i,k+1} - \sum_{j'=1}^{k} h_{ij'}}, & \text{if } j \leq k \\ 0, & \text{if } j > k. \end{cases} \tag{16}$$

Moreover, we optimize the naturally sparse bipartite graph matrix B obtained by the (16): assuming x_i belongs to anchor set U, the nearest anchors for x_i will be itself. Therefore, the selection of neighbors is optimized by disposing of zero value to improve a little performance.

When accomplishing the construction of bipartite graph matrix B, the symmetric similarity matrix $W \in \mathbb{R}^{n\times n}$ then can be constructed as

$$W = B\Lambda^{-1}B^{\mathrm{T}} \tag{17}$$

where the matrix $\Lambda \in \mathbb{R}^{m\times m}$ is a diagonal matrix and the ith element θ_i is the sum of ith column of B, which means $\theta_i = \sum_i b_{ij}$. We can easily get corresponding matrix $D = I$ from the bipartite graph. The Laplacian matrix L will be equal to $I - W$. Thus, we will have

$$L = I - B\Lambda^{-1}B^{\mathrm{T}}. \tag{18}$$

In this condition, the number of labeled data points is m, which is equivalent to that of representative anchors. Initial label matrix $Y \in \mathbb{R}^{n\times(m+1)}$ and soft label matrix $F \in \mathbb{R}^{n\times(m+1)}$ are also correspondingly changed in dimension. Due to $D = I$ from bipartite graph theory earlier, α_i will be set to be $1/(1 + \mu_i)$ and β_i is equal to $\mu_i/(1 + \mu_i)$, comparatively.

According to the solution process of general semisupervised framework, the optimal soft label matrix F^* can be obtained as

$$F^* = (I - I_\alpha W)^{-1} I_\beta Y. \tag{19}$$

Therefore, the FSSF has been proposed with optimized naturally sparse similarity matrix W, where the specific selection of all parameters will be elaborated in Section IV to tackle clustering problems.

IV. Fast Self-Supervised Clustering

In this section, the details of the proposed FSSC method are described. The setting of parameters based on the proposed FSSF in self-supervised clustering will be demonstrated concretely followed by the special selection strategy to discover the most representative c points with the best quality for label propagation from m anchors.

We exploit FSSF to conduct the clustering process in a self-supervised manner, where m representative anchors are considered as the foundation of a bipartite graph and m labeled points simultaneously. The labels are completely artificial, which are marked by the selected order of each representative point in the last hierarchy of the BKHK structure. According to the assessment strategy on outliers, we are able to find proper features for each sample to extract corresponding representative points, where anchor set U from BKHK is regarded as an interim to get final c points belonging to c classes in overall clustering. The label propagation will be conducted at the end of our method.

The computational complexity analysis for each step and the overall algorithm will be stated at last of this section. Thus, our approach can be divided into the following parts.

A. Self-Supervised Clustering Algorithm With Fake Label

The motivation of the self-supervised clustering based on FSSF is to construct a probability model for easier post-processing. The initial and artificial labels for m representative points are usually virtual, which play an important transitive effect in FSSF to obtain matrix F. As for I_α and I_β, on the one hand, we generally set positive α_i for labeled data x_i to tackle the appearance of noises. However, different from the known labels in traditional SSL, m representative points are selected to roughly and evenly cover primal data points, the acquired fake labels of which are merely related to the original structures of n samples. In order to maintain the original distribution of entire representative points, the m different labels of m representative points in matrix Y will be considered as no noises theoretically. Furthermore, this assumption is validated in experimental results in benchmark data sets. On the other hand, for unlabeled data x_j, $\alpha_j = 1$ means that we will lose the capability of filtering outliers. Accordingly, $\alpha_j = 1$ will be deployed to be relatively large to alter their labels as much as possible and lower than 1 to retain the function to detect and filter outliers in the semisupervised label propagation process.

Moreover, we utilize matrix inversion lemma [60] to further accelerate the computation of (19) when solving the large-scale problem (in this condition, the computation of the inversion of a general $n \times n$ matrix will be usually time-consuming). This inversion will be transformed into the computation of $m \times m$ matrix. Assuming $Q_1 = (I - I_\alpha B\Lambda^{-1}B^{\mathrm{T}})^{-1} - I$, through $(I - I_\alpha B\Lambda^{-1}B^{\mathrm{T}})(I + Q_1) = I$, Q_1 can be transformed into

$$Q_1 = (I - I_\alpha B\Lambda^{-1}B^{\mathrm{T}})^{-1} I_\alpha B\Lambda^{-1}B^{\mathrm{T}}. \tag{20}$$

We assume $Q_2 = I_\alpha B\Lambda^{-1}B^{\mathrm{T}}$. Since $F^* = (Q_1 + I)I_\beta Y$, F^* will be deformed as follows:

$$\begin{aligned} F^* &= \big((I - I_\alpha B\Lambda^{-1}B^{\mathrm{T}})^{-1}Q_2 + I\big)I_\beta Y \\ &= \Big(\Big(-I_\alpha B\big((-I_\alpha B)^{-1} + \Lambda^{-1}B^{\mathrm{T}}\big)\Big)^{-1}Q_2 + I\Big)I_\beta Y \\ &= -\big((-I_\alpha B)^{-1} + \Lambda^{-1}B^{\mathrm{T}}\big)^{-1}\Lambda^{-1}B^{\mathrm{T}}I_\beta Y + I_\beta Y \\ &= -\Big(\Lambda^{-1}\big(\Lambda(-I_\alpha B)^{-1} + B^{\mathrm{T}}\big)\Big)^{-1}\Lambda^{-1}B^{\mathrm{T}}I_\beta Y + I_\beta Y \\ &= -\big(\Lambda(-I_\alpha B)^{-1} + B^{\mathrm{T}}\big)^{-1}B^{\mathrm{T}}I_\beta Y + I_\beta Y \\ &= -\Big(\big(\Lambda + B^{\mathrm{T}}(-I_\alpha B)\big)(-I_\alpha B)^{-1}\Big)^{-1}B^{\mathrm{T}}I_\beta Y + I_\beta Y \\ &= I_\alpha B\big(\Lambda + B^{\mathrm{T}}(-I_\alpha B)\big)^{-1}B^{\mathrm{T}}I_\beta Y + I_\beta Y. \end{aligned} \tag{21}$$

The contributions of obtained soft label matrix F are as follows.

1) It is relatively convenient for follow-up processing attributed to the probability model of compact F and we

can judge the rough class through each F_{ij} for m classes from selected anchors. Though the class belonging to m anchors does not represent the ultimate clustering results, we are able to extract significant information to continue following the clustering process.

2) Benefit from the matrix inversion lemma in (20), the inversion process for $n \times n$ matrix has been changed into the inversion of $m \times m$, which dramatically mitigates the computational cost due to $m \ll n$ in most cases. Concretely, in (20), the computational complexity of $(\Lambda + B^{\mathrm{T}}(-I_\alpha B))^{-1}$ is $O(m^2 n) + O(m^3) + O(mn)$ and computational complexity of the rest matrix calculation is $O(m^2 n) + O(mn)$. Therefore, we need $O(m^2 n) + O(m^3)$ to calculate soft label matrix F according to (20), which is linearly related to the number of samples n.

B. Selection of c Representative Points

F_{ij} only represents the probability of ith sample belonging to jth representative anchor or outliers instead of the belonging relationship between samples and true classes. For details, since it is sure that the number of true classes in the data set is c, there is still some doubt that how to express the probability of each sample belonging to c classes. The focus on this problem is how to use m obtained anchors to get c representative points from n primal samples. Thus, we consider extracting a unique score for each sample from matrix F to represent significance not being outliers.

Attributed to the $(m+1)$th column of $F \in \mathbb{R}^{n\times(m+1)}$ represents the underlying probability of becoming outliers, we delete this column maintaining first m columns to get $\hat{F} \in \mathbb{R}^{n\times m}$ for preprocessing. For sample x_i, we can easily find that the greater the sum of the ith row of $\hat{F}$, the less likely it that the ith sample will be identified as an outlier in the data set when we are sure that the true label contains c classes. We set $\alpha \neq 1$ to avoid the sum of the row for each unlabeled data is all 1, which has a negative impact to regard the sum of the row as a proper choosing score.

Therefore, a special selection strategy to extract c representative points with the best quality from n samples will be introduced, which aims to find out one corresponding representative point belonging to one class in each of the c classes and avoid the absence of representative points in a class. It should be significantly emphasized that there is no correlation between c representative points and c cluster centers actually. On the one hand, the larger extracted score for the ith sample merely embodies that the ith sample is more likely to be included among known c true classes. On the other hand, the selection of subsequent representative points is subject to previous representative points, which only aims to choose representative points from different classes, respectively. It is only an ideal situation that the selected c representative points can refer to the cluster center of each category, the probability of which is extremely small.

Since the number of selected representative points is exactly c, whenever we choose a representative point, we should remove the sample points with high similarity. We adapt to the correction of the sum of all other rows for samples, where more similar to the representative point, the smaller corrected sum of rows on similarity will be. Thus, the probability of the other high similarity points being selected will be very small.

Algorithm 1 FSSC

Input: The representative anchor set U and the order of all anchors $Rank$.
Self-supervised clustering with FSSF:
Set Y of all samples according to U and $Rank$.
Build B utilizing Eq.(16).
while not converge **do**
  Calculate F by Eq.(19).
  Accelerate computing of F by Eq.(21).
end while
The selection of c representative points:
$i = 1$.
while $i < c$ **do**
  Extract first m columns of F.
  Calculate the sum of each row $score_i$ for x_i.
  Select maximum of $score_i$ and record the order.
  Amend other scores.
  $i = i + 1$.
end while
Label Propagation:
while not converge *do*
  Calculate the T via computing Eq.(26) quickly.
end while
Output: The result of anticipated clustering label Z with n data points from converged T_{final}.

Inspired by feature selection, we denote the sum of ith row of $\hat{F}$ to be the score for ith sample point as

$$\mathrm{score}(x_i) = \sum_{j}^{m} F_{ij} \tag{22}$$

where we need to choose the maximum of scores of n samples, which means that larger scores, more possible not to be outlier and representative one of c classes with best quality.

When the first representative point et.x_i is selected, the scores of other samples will be amended by the similarity between x_i and other points in W. Assuming the scores of x_i is z_1, the explanation can be formulated as

$$z_1 = \arg\max \mathrm{score}(x_i). \tag{23}$$

We assume one point x_j different from x_i, the feature score of which is $\mathrm{score}(x_j)$, the modification process should be

$$\mathrm{score}(x_j)_{\mathrm{new}} = (1 - W_{ij})\mathrm{score}(x_j) \tag{24}$$

where the scores of other samples will be amended by (24) one by one. Then, we will choose the second largest score x_h when accomplishing all the amending. The algorithm ends until c representative points are selected completely. These c representative points will be treated as the initial points with true labels to operate label propagation processing.

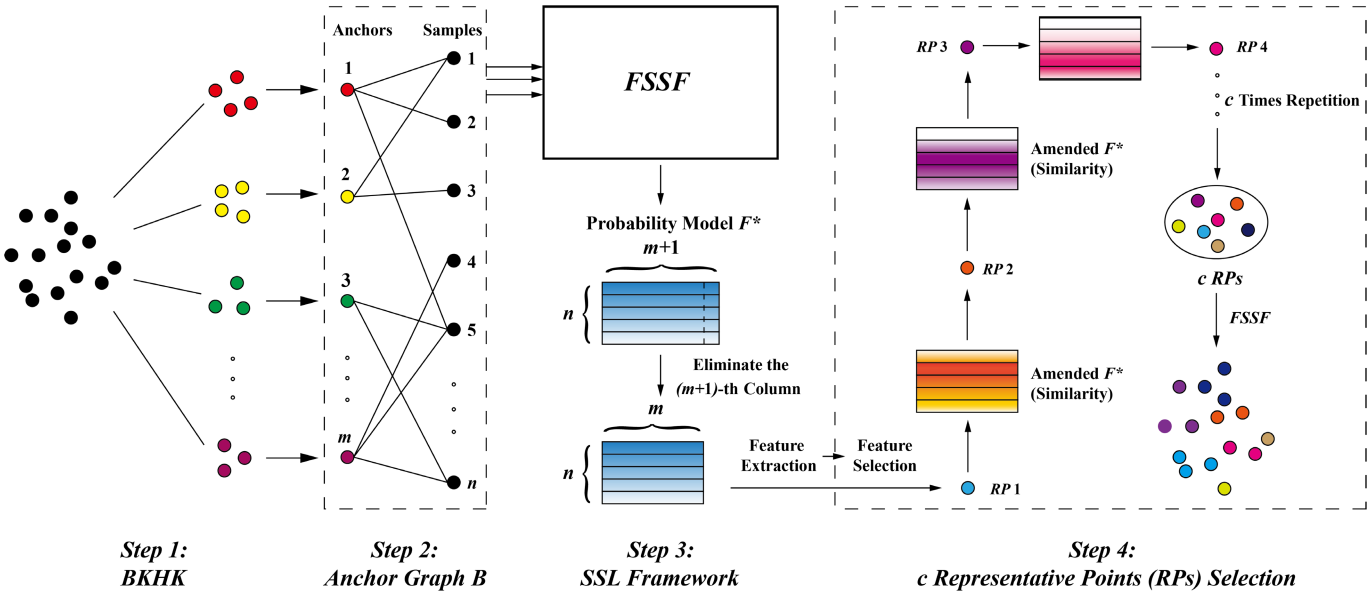

Fig. 2. Illustration of FSSC model from BKHK to final label propagation.

C. Label Propagation

The general SSL framework in Section III is also adapted to conduct label propagation, which can be formulated as

$$\min_{T} \sum_{i,j=1}^{n} W_{ij}\|T_i - T_j\|_F^2 + \sum_{i=1}^{n} \hat{\mu}_i \|T_i - \hat{Y}_i\|_F^2 \tag{25}$$

where $T \in \mathbb{R}^{n\times(c+1)}$ is the terminal propagating result and $\hat{Y} \in \mathbb{R}^{n\times(c+1)}$ is the initial label matrix based on the most representative points. We still utilize the affinity matrix W obtained from Section III to improve the performance. And the problem (25) can be processed by Lagrange solution. Thus, the label propagation procedure becomes

$$T^* = (I - W + \hat{U})^{-1}\hat{U}\hat{Y} \tag{26}$$

where $\hat{U}$ is a diagonal matrix with the ith entry being $\hat{\mu}_i$. Then, $\hat{\alpha}_i = d_i/(d_i + \hat{\mu}_i)$ and $\hat{\beta}_i = \hat{\mu}_i/(d_i + \hat{\mu}_i)$ are defined as the same as general semisupervised framework. In this condition, for labeled data point x_i, $\hat{\alpha}_i$ will be set to be zero, since c representative points do not exist noises exactly. Comparatively, for unlabeled data point x_j, $\hat{\alpha}_j$ will be set to be one due to the absence about outliers. Meanwhile, we denote diagonal matrix $\hat{I}_\alpha$ and $\hat{I}_\beta$ with the ith entry $\hat{\alpha}_i$ and $\hat{\beta}_i$, respectively. Thus, the iteration format becomes

$$T^* = (I - \hat{I}_\alpha P)^{-1}\hat{I}_\beta \hat{Y}. \tag{27}$$

Furthermore, we are capable of accelerating the computation of (27), which is the same as the meaning of (19).

Through certain iterations of label propagation, we could find out the largest T_{ij} from T_i, which means ith data point belongs to jth class on ground truth (GT). Therefore, the anticipated label $Z \in \mathbb{R}^n$ vector can be obtained. To make the general model more impressive and intuitive, we utilize a portrayal to illustrate the FSSC algorithm, as shown in Fig. 2. Subsequently, the overall algorithm is summarized in Algorithm I.

D. Computational Complexity Analysis

The computational complexity of SC is $O(n^2d + n^3)$, which is really time-consuming to solve the large-scale problem, while the proposed method possesses more efficient computational performance. Concretely, the calculated complexity of FSSC can be separated fourfold.

1) BKHK is conducted based on the dichotomous K-means algorithm. The computational complexity of the first dichotomous K-means on the primal data set is $O(nd)$. In subsequent hierarchies of a binary tree, each time the number of samples of binary K-means will be halved, while the number of executions will be doubled. On this condition, the computational complexity is still $O(nd)$. In addition, since the number of decomposed layers is $\log(m)$, the calculated cost of BKHK is $O(nd\log(m))$ to obtain m anchors from n samples.
2) It takes us $O(nmd)$ to build the major matrix $H \in \mathbb{R}^{n\times m}$ in (14) between m anchors and n samples. Besides, the occupied time of (16) is so little that its time complexity can be ignored to construct bipartite graph matrix B. Therefore, we spend $O(nmd)$ in building anchor-based graph B and speeding up the semisupervised framework.
3) The majority of the computational cost when acquiring soft label matrix F derives from the calculation of (21). Benefit from the acceleration of bipartite graph and matrix inversion lemma, the corresponding computational complexity is optimized from $O(n^3)$ to $O(m^2n + m^3)$. Generally, due to $m \ll n$, it is determined as $O(m^2n)$ in FSSF.
4) It takes us $O(nmc)$ to find out the most representative c sample points by updated feature-like selection. Subsequently, propagating labels through the most representative c points with the best quality to all sample points is spending $O(m^2n + nmd)$ by FSSF.

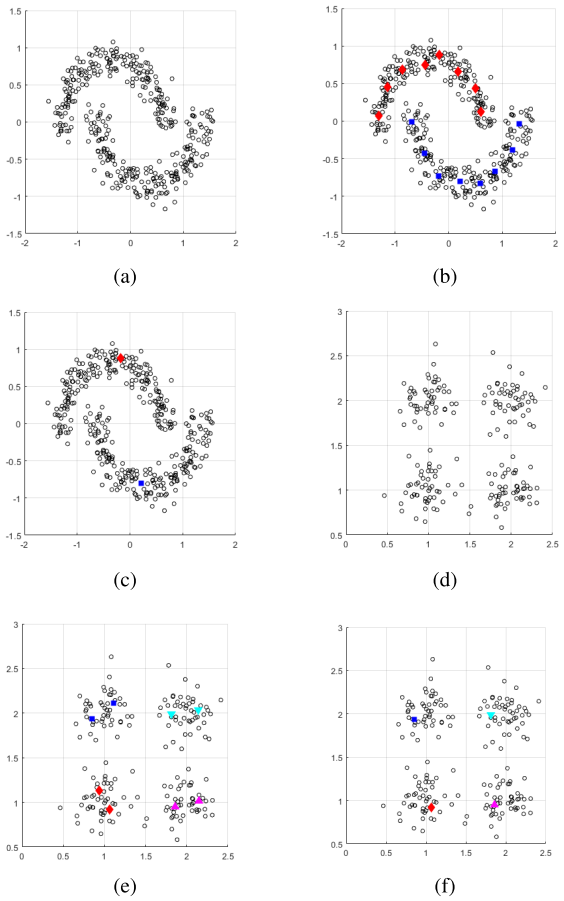

Fig. 3. Experiments on two toy examples. (a) Original data of **Two Moon**. (b) Anchors of **Two Moon**. (c) Final points of **Two Moon**. (d) Original data of **Spheres**. (e) Anchors of **Spheres**. (f) Final points of **Spheres**.

Considering that $m \ll n$ and $c \ll n$, the overall computational complexity of the FSSC method is $O(nmd)$.

V. Experimental Results

In this section, we first validate our approach and exhibit the main stages of the proposed method with two toy examples graphically. Second, the parameter sensitivity of our method based on the number of anchors m, α_l and α_u will be concretely analyzed on five benchmark data sets with clustering metrics ACCuracy (ACC), normalized mutual information (NMI), and clustering time. Finally, compared results with K-means, SC [61], LSC-R [62], LSC-K [62], FSC [52], and FRWL-B [55] are assessed, where we run every method ten times in these five data sets, calculate the mean of all results, and report the metrics ACC, NMI, and running time. These experimental results can demonstrate the effectiveness of our algorithm and can also validate the computational complexity analysis in Section IV-D.

A. Validation on Toy Example

We give two toy examples to analyze and validate our method. Fig. 3(a) shows the primal two moon toy data containing two classes which have the same number of samples with 0.12 noise and Fig. 3(d) shows initial spheres toy data which consists of four classes.

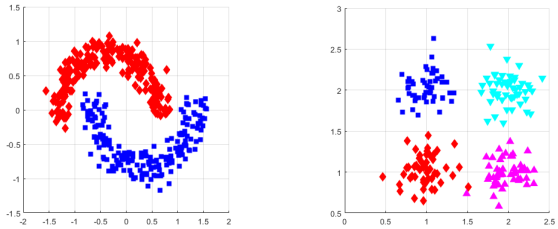

Fig. 4. Clustering results on two toy examples. (a) Two moon data. (b) Spheres data.

TABLE I
Description of Data Sets

Datasets	Num of Instances	Dimensions	Classes
PalmData25	2000	256	100
Abalone	4177	7	29
USPS	9298	256	10
Letter	20000	16	26
MNIST	70000	784	10

From Fig. 3(b), it can be seen that the number of labeled representative anchors is set to be 16 and it is reasonable for 400 samples and two classes. We can easily find that the selection of anchor points is roughly uniform within and between classes, where the red diamonds represent the anchor points of the first class and the blue squares represent the anchor points of the second class. Combined with the theory of the construction of bipartite graph, for sample x_i, it will be most similar to the nearest anchor point of the same category. The anchor results on the spheres data set have similar characteristics to the two moon data set, which is shown in Fig. 3(e).

Fig. 3(c) and (f) show the results of selection about c representative points in these two data sets, which express that there is only one representative point of each category.

The clustering results in these two data sets are shown in Fig. 4, where the clustering accuracy of two moon data can achieve 100% and that of spheres data can reach 99.5%. Though the m anchors and c representative points sometimes lightly fluctuate in different experiments, the desirable results can be always maintained in these two toy examples.

B. Parameter Sensitivity

We begin via describing our experimental benchmark data sets. The specific characteristics of these data sets about the number of instances, dimensions, and classes of GT are all listed on Table I. Two of them, Abalone and Letter, are from UCI machine learning repository [63]. While PalmData25 with 16×16 image scale, US Postal handwritten digit (USPS) with 16×16 image scale, and Mixed National Institute of Standards and Technology handwritten digit (MNIST) with 24×24 image scale belong to image data sets.

The three essential parameters in our method are the number of the representative anchors m and the regularization parameter α_l and α_u. The influence of these parameters on running time and performance will be validated as follows.

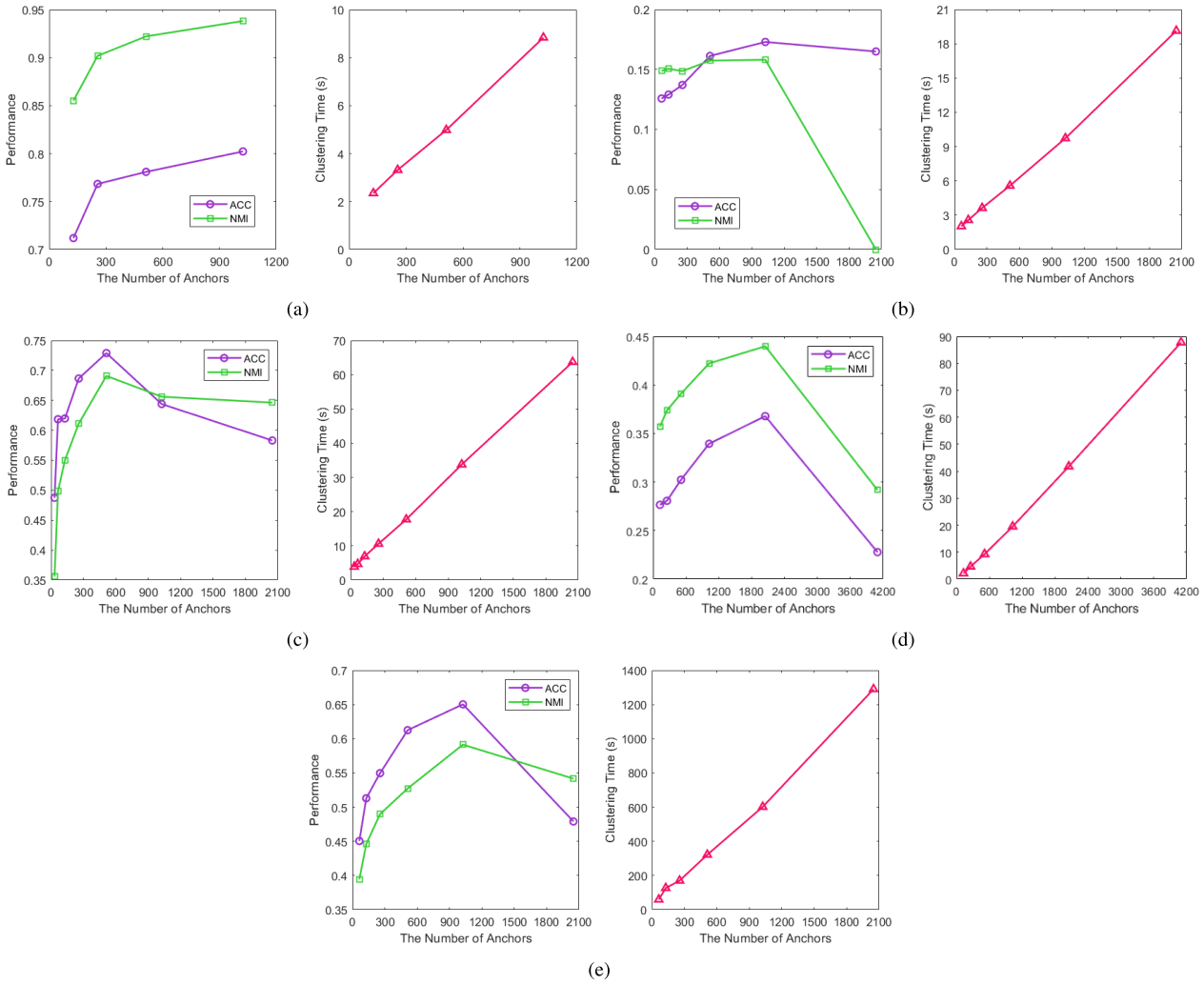

Fig. 5. Trend of ACC, NMI, and clustering time via adjusting the number of anchors on five benchmark data sets. (a) PalmData25. (b) Abalone. (c) USPS. (d) Letter. (e) MNIST.

1) Number of Anchors m: The experimental results on these five benchmark data sets are exhibited in Fig. 5. The first picture of Fig. 5(b)–(e) shows that dense enough representative anchors are indispensable to build a reasonable connecting relationship between anchors and original samples, while overly large amounts of anchors which usually contain redundant information will be useless for clustering performance. Therefore, an appropriate number of anchors plays an essential role to improve final performance. In addition, the second picture of Fig. 5(a)–(e) demonstrates that running time linearly ascend as the number of representative anchors increases, which matches the comprehensive complexity analysis in Section IV-D.

2) Framework Parameter α_l and α_u: The specific parameter sensitivity on α_l and α_u with clustering metric ACC are portrayed in Figs. 6–10, which correspond PalmData25, Abalone, USPS, Letter, and MNIST, respectively. As for the different combined value of α_l and α_u, we record corresponding mean results of ten times experiments merely on ACC, which is sufficient to illustrate the limitation of these two parameters. According to the description of these two parameters in Section III, α_l controls the fake labels of m representative anchors, while α_u controls the detection capability for outliers.

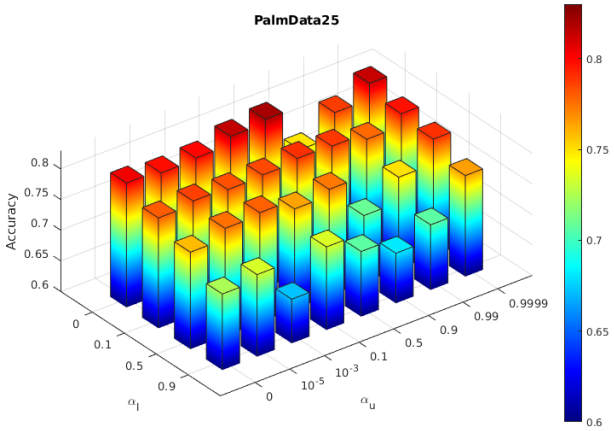

Fig. 6. Clustering accuracy of PalmData25 via adjusting parameter α_l and α_u when fixing k and m.

On the one hand, Figs. 6–10 show that the employment of $\alpha_l = 0$ effectively improve clustering performance, which

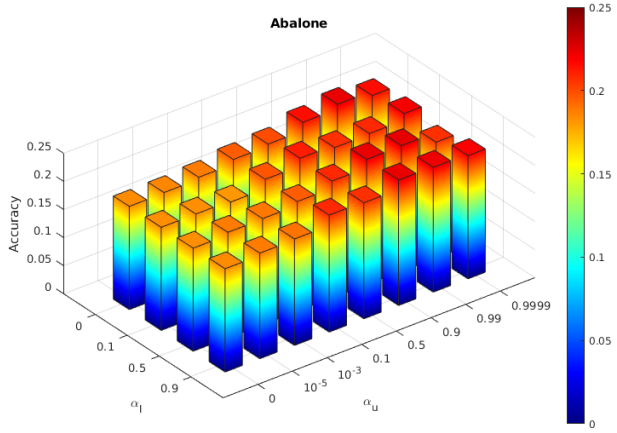

Fig. 7. Clustering accuracy of Abalone via adjusting parameter α_l and α_u when fixing k and m.

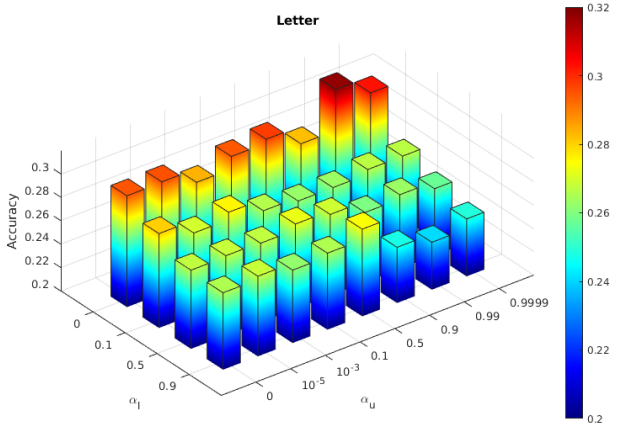

Fig. 9. Clustering accuracy of Letter via adjusting parameter α_l and α_u when fixing k and m.

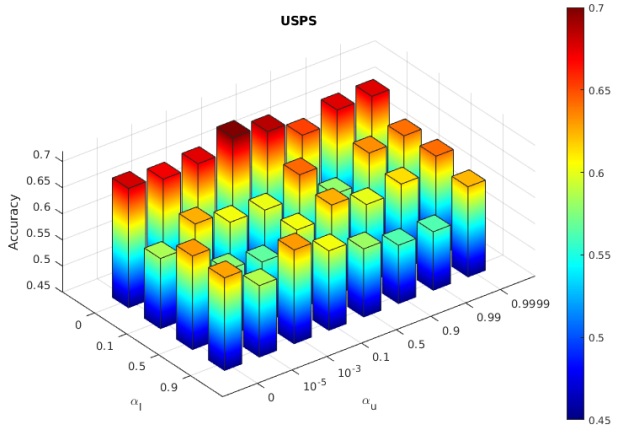

Fig. 8. Clustering accuracy of USPS via adjusting parameter α_l and α_u when fixing k and m.

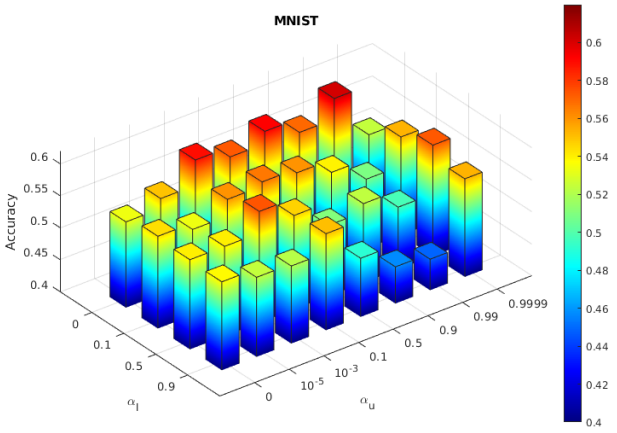

Fig. 10. Clustering accuracy of MNIST via adjusting parameter α_l and α_u when fixing k and m.

validates that artificial labels of representative anchors are fairly reliable (virtual labels theoretically do not exist noises mentioned in Section IV-A) to conduct label propagation in our FSSF. Thus, the α_l in final label propagation from c representative points to n samples will be accurately assigned to be zero. In addition, Fig. 7 shows that the influence of α_l on ACC are able to be ignored on Abalone, which attributes to the characteristic of data itself.

On the other hand, for unlabeled samples, Figs. 6–10 also show that it is critical to select suitable value of α_u to detect outliers and avoid them to become one of c representative points in final label propagation stage. For instance, Fig. 10 shows that overly large and small values of α_u, which means overly weak and strong capability to avert the occurrence of outliers in the final representative point set, will result in drawbacks for final clustering performance in MNIST data set. While for Abalone, the clustering ACC will constantly and slowly augment as the α_u increases. Thus, the optimal value of α_u should be dynamically tuned based on the characteristic of various data sets.

As a whole, we set the value of α_l to be zero as default in subsequently compared experiments. Furthermore, combined with experimental results by adjusting α_u on all data sets, it will be assigned to be 0.99 for convenience.

C. Comparison Results

To illustrate the effectiveness and efficiency of the proposed method, we compare it with K-means, SC, landmark-based SC using random sampling to select landmarks (LSC-R) [62], landmark-based SC using K-means to select landmarks (LSC-K) [62], fast SC (FSC) [52], and FSC based on random walk Laplacian with BKHK (FRWL-B) [55]. Except for two traditional clustering approaches K-means and SC, the rest compared methods all belong to anchor graph-based SC techniques. More concretely, LSC-R randomly selects anchors from original samples, while LSC-K utilizes K-means for anchor-selection. FSC and FRWL-B adopt BKHK to find anchors and conduct subsequent spectral analysis. In order to acquire a better comparison effect, for these five anchor graph-based methods, we choose 1024 anchors in PalmData25, 1024 anchors in Abalone, 512 anchors in USPS, 512 anchors in Letter, and 512 anchors in MNIST to construct a homogeneous anchor graph with the same scale. We still perform 10 times duplicated experiments for each method and record mean results.

To demonstrate the effectiveness of our method, the experimental results in different methods on the five benchmark data sets are reported in Tables II and III with ACC and

TABLE II
COMPARISON IN TERMS OF ACC (%)

Datasets	PalmData25		Abalone		USPS		Letter		MNIST	
K-means	69.46	(±0.29)	14.82	(±0.24)	65.46	(±0.77)	25.52	(±0.27)	56.14	(±**0.97**)
SC	77.88	(±**0.24**)	13.26	(±0.14)	65.38	(±**0.69**)	26.75	(±**0.16**)	–	
LSC-R[62]	74.83	(±2.49)	12.77	(±0.73)	59.23	(±5.69)	25.40	(±1.22)	49.21	(±3.12)
LSC-K[62]	77.45	(±1.42)	13.79	(±0.48)	66.81	(±0.79)	26.47	(±1.03)	58.27	(±2.26)
FSC[52]	80.04	(±0.40)	12.50	(±**0.12**)	67.45	(±2.36)	31.32	(±0.48)	58.42	(±1.47)
FRWL-B[55]	81.02	(±0.38)	13.19	(±0.21)	67.76	(±1.72)	31.97	(±0.39)	59.76	(±1.13)
FSSC(ours)	**82.25**	(±0.47)	**21.21**	(±0.55)	**72.90**	(±1.14)	**33.94**	(±0.80)	**61.26**	(±1.41)

TABLE III
COMPARISON IN TERMS OF NMI (%)

Datasets	PalmData25		Abalone		USPS		Letter		MNIST	
K-means	89.30	(±0.39)	15.94	(±0.77)	60.69	(±**0.21**)	34.44	(±0.28)	49.38	(±**0.27**)
SC	93.73	(±0.24)	14.87	(±0.15)	67.54	(±0.24)	36.75	(±**0.19**)	–	
LSC-R	89.40	(±1.70)	12.13	(±0.78)	60.27	(±2.26)	31.82	(±1.57)	43.21	(±2.52)
LSC-K	90.81	(±1.36)	13.74	(±1.33)	65.28	(±0.64)	33.83	(±1.35)	53.94	(±1.00)
FSC	93.80	(±0.16)	15.35	(±**0.09**)	68.93	(±0.87)	42.19	(±0.40)	55.43	(±0.75)
FRWL-B	93.00	(±**0.11**)	14.77	(±**0.09**)	67.84	(±0.63)	41.27	(±0.33)	54.81	(±0.83)
FSSC(ours)	**94.88**	(±0.16)	**16.02**	(±0.24)	**69.09**	(±0.78)	**42.21**	(±0.76)	**55.89**	(±1.89)

TABLE IV
COMPARISON IN TERMS OF RUNNING TIME (S)

Datasets	PalmData25	Abalone	USPS	Letter	MNIST
SC	9.9	44.9	457.7	1365.4	–
LSC-R	**2.5**	**6.2**	**7.0**	16.5	**102.9**
LSC-K	**3.7**	6.5	13.1	17.9	447.6
FSC	8.1	7.5	23.0	14.8	303.4
FRWL-B	8.4	7.5	23.5	**14.1**	300.3
FSSC(ours)	10.1	**6.0**	**11.9**	**13.9**	**269.2**

NMI, respectively. To indicate the efficiency of our method, we also compare clustering time between the proposed method and other five graph-based SC techniques. More specifically, Table II shows that our method reaches the highest performance in all data sets especially in Abalone and USPS. However, the standard deviation of our method is fairly large, which embodies the instability of our algorithm resulting from the random initialization of K-means and label propagation in our FSSF. Table III shows that our method also obtains the best performance with relatively unstable results upon NMI metric.

In Table IV, we bold the two results with the least computational cost among the six algorithms for each benchmark data set. From Tables II–IV, it can be first seen that the clustering performance of traditional K-means is poor. Second, conventional clustering method SC is not capable of running on MNIST, and its computational burden is extremely high with the increasing number of samples. Third, though the computational cost of LSC-R is generally advantageous in all data sets, the experimental results upon ACC and NMI are overall inferior to other graph-based approaches, sometimes even lower than K-means. Furthermore, attributed to the randomly anchor-selection strategy, the standard deviation of LSC-R on ACC and NMI is extremely large. Compared with LSC-R, LSC-K has a significant improvement in accuracy and stability, while it is still difficult to achieve satisfactory clustering results. Finally, the stability of FSC and FRWL-B is greater than our method except for the USPS data set; however, their computational cost is distinctly larger than our method in Abalone, USPS, and MNIST.

As a whole, our method has obvious superiority on metrics ACC and NMI, and it performs better on complexity cost than other related approaches especially with a large number of samples. Therefore, we can believe that our method could outperform other methods to reach effectiveness and efficiency in real applications.

VI. CONCLUSION

This study designed a novel anchor-based clustering method in a self-supervised manner, which is referred to as FSSC. This approach operates unsupervised clustering with semisupervised thoughts through the construction of fake labels, which utilizes BKHK to generate targeted representative anchors with fast access at first. Second, a parameter-free bipartite graph connecting anchors and original samples is constructed. Third, we present a novel FSSF to perform FSSC and obtain a crucial probability model. Next, we introduce a feature-like selection strategy to find out the most representative c points, where each point indicates one class. Terminally, label propagation is conducted together with FSSF to accomplish label prediction of all samples based on c selected points. Several experimental results have been provided to show the effectiveness and efficiency of our method. In the future, we should improve the initialization upon generation of anchors to dispose of undesirable stability in our algorithm. Meanwhile, we will also pay more attention to adaptively adjust the capability of feature amend to find optimal representative points for subsequent label propagation.

REFERENCES

[1] M. Ozay, I. Esnaola, F. T. Yarman Vural, S. R. Kulkarni, and H. V. Poor, "Machine learning methods for attack detection in the smart grid," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 8, pp. 1773–1786, Aug. 2016.

[2] M. A. Muñoz, L. Villanova, D. Baatar, and K. Smith-Miles, "Instance spaces for machine learning classification," *Mach. Learn.*, vol. 107, no. 1, pp. 109–147, Jan. 2018.

[3] X. Li, M. Chen, F. Nie, and Q. Wang, "Locality adaptive discriminant analysis," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2201–2207.

[4] M. Hausknecht, W.-K. Li, M. Mauk, and P. Stone, "Machine learning capabilities of a simulated cerebellum," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 510–522, Mar. 2017.

[5] W. Kim, M. S. Stankovic, K. H. Johansson, and H. J. Kim, "A distributed support vector machine learning over wireless sensor networks," *IEEE Trans. Cybern.*, vol. 45, no. 11, pp. 2599–2611, Nov. 2015.

[6] X. Li, M. Chen, F. Nie, and Q. Wang, "A multiview-based parameter free framework for group detection," in *Proc. AAAI*, 2017, pp. 4147–4153.

[7] X.-D. Wang, R.-C. Chen, Z.-Q. Zeng, C.-Q. Hong, and F. Yan, "Robust dimension reduction for clustering with local adaptive learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 657–669, Mar. 2019.

[8] Z. Feng and Y. Zhu, "A survey on trajectory data mining: Techniques and applications," *IEEE Access*, vol. 4, pp. 2056–2067, 2016.

[9] Z. Li, Z. Zhang, J. Qin, Z. Zhang, and L. Shao, "Discriminative Fisher embedding dictionary learning algorithm for object recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 3, pp. 786–800, Mar. 2020.

[10] X. Fang *et al.*, "Flexible affinity matrix learning for unsupervised and semisupervised classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1133–1149, Apr. 2019.

[11] H. Jia, Y.-M. Cheung, and J. Liu, "A new distance metric for unsupervised learning of categorical data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 5, pp. 1065–1079, May 2016.

[12] F. Cai and V. Cherkassky, "Generalized SMO algorithm for SVM-based multitask learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 6, pp. 997–1003, Jun. 2012.

[13] A. Dong, F.-L. Chung, Z. Deng, and S. Wang, "Semi-supervised SVM with extended hidden features," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2924–2937, Dec. 2016.

[14] B. Leng, J. Zeng, M. Yao, and Z. Xiong, "3D object retrieval with multitopic model combining relevance feedback and LDA model," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 94–105, Jan. 2015.

[15] Y. Aliyari Ghassabeh, F. Rudzicz, and H. A. Moghaddam, "Fast incremental LDA feature extraction," *Pattern Recognit.*, vol. 48, no. 6, pp. 1999–2012, Jun. 2015.

[16] C. Alzate and J. A. K. Suykens, "Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 2, pp. 335–347, Feb. 2010.

[17] Z. Khan, F. Shafait, and A. Mian, "Joint group sparse PCA for compressed hyperspectral imaging," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4934–4942, Dec. 2015.

[18] Z. Lai, Y. Xu, J. Yang, L. Shen, and D. Zhang, "Rotational invariant dimensionality reduction algorithms," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3733–3746, Nov. 2017.

[19] L. Gan, J. Xia, P. Du, and Z. Xu, "Dissimilarity-weighted sparse representation for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 11, pp. 1968–1972, Nov. 2017.

[20] J. Gan, G. Wen, H. Yu, W. Zheng, and C. Lei, "Supervised feature selection by self-paced learning regression," *Pattern Recognit. Lett.*, vol. 132, pp. 30–37, Apr. 2020.

[21] J. Wang, X. Wang, K. Zhang, K. Madani, and C. Sabourin, "Morphological band selection for hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 8, pp. 1259–1263, Aug. 2018.

[22] F. Nie, S. Xiang, Y. Liu, and C. Zhang, "A general graph-based semi-supervised learning with novel class discovery," *Neural Comput. Appl.*, vol. 19, no. 4, pp. 549–555, Jun. 2010.

[23] L. Berton, "Graph construction based on neighborhood for semisupervised," Ph.D. dissertation, Univ. São Paulo, São Paulo, Brazil, 2016.

[24] L. Zhang *et al.*, "Large-scale robust semisupervised classification," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 907–917, Mar. 2019.

[25] W.-L. Zhao, C.-H. Deng, and C.-W. Ngo, "K-means: A revisit," *Neurocomputing*, vol. 291, pp. 195–206, May 2018.

[26] A. Karlekar, A. Seal, O. Krejcar, and C. Gonzalo-Martín, "Fuzzy k-means using non-linear s-distance," *IEEE Access*, vol. 7, pp. 55121–55131, 2019.

[27] R. Langone and J. A. K. Suykens, "Fast kernel spectral clustering," *Neurocomputing*, vol. 268, pp. 27–33, Dec. 2017.

[28] H. Zheng and J. Wu, "Which, when, and how: Hierarchical clustering with human–machine cooperation," *Algorithms*, vol. 9, no. 4, p. 88, Dec. 2016.

[29] C. Fevotte and N. Dobigeon, "Nonlinear hyperspectral unmixing with robust nonnegative matrix factorization," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4810–4819, Dec. 2015.

[30] Y. Pang, J. Xie, F. Nie, and X. Li, "Spectral clustering by joint spectral embedding and spectral rotation," *IEEE Trans. Cybern.*, vol. 50, no. 1, pp. 247–258, Jan. 2020.

[31] R. Zhang, F. Nie, M. Guo, X. Wei, and X. Li, "Joint learning of fuzzy K-means and nonnegative spectral clustering with side information," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2152–2162, May 2019.

[32] D. Cai and X. Chen, "Large scale spectral clustering via landmark-based sparse representation," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1669–1680, Aug. 2015.

[33] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, "Using self-supervised learning can improve model robustness and uncertainty," in *Proc. NeurIPS*, 2019, pp. 15637–15648.

[34] Q. Ma, S. Li, W. Zhuang, S. Li, J. Wang, and D. Zeng, "Self-supervised time series clustering with model-based dynamics," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Aug. 31, 2020, doi: 10.1109/TNNLS.2020.3016291.

[35] J. Zhang *et al.*, "Self-supervised convolutional subspace clustering network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5473–5482.

[36] X. Sun, M. Cheng, C. Min, and L. Jing, "Self-supervised deep multi-view subspace clustering," in *Proc. ACML*, vol. 101, 2019, pp. 1001–1016.

[37] J. Ye, Q. Li, J. Yu, X. Wang, and H. Wang, "Affinity learning via self-supervised diffusion for spectral clustering," *IEEE Access*, vol. 9, pp. 7170–7182, 2021.

[38] T. Lin, H. Xu, and H. Zhang, "Constrained self-supervised clustering for discovering new intents (student abstract)," in *Proc. AAAI*, 2020, pp. 13863–13864.

[39] M. A. Lozano and F. Escolano, "Graph matching and clustering using kernel attributes," *Neurocomputing*, vol. 113, pp. 177–194, Aug. 2013.

[40] N. Paudel, L. Georgiadis, and G. F. Italiano, "Computing critical nodes in directed graphs," *ACM J. Experim. Algorithmics*, vol. 23, pp. 1–24, Nov. 2018.

[41] L. Gellert and R. Sanyal, "On degree sequences of undirected, directed, and bidirected graphs," *Eur. J. Combinatorics*, vol. 64, pp. 113–124, Aug. 2017.

[42] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The constrained Laplacian rank algorithm for graph-based clustering," in *Proc. AAAI*, 2016, pp. 1969–1976.

[43] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering," *Pattern Recognit.*, vol. 41, no. 1, pp. 176–190, Jan. 2008.

[44] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proc. NIPS*, 2004, pp. 1601–1608.

[45] X. Zhu, S. Zhang, Y. Li, J. Zhang, L. Yang, and Y. Fang, "Low-rank sparse subspace for spectral clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 8, pp. 1532–1543, Aug. 2019.

[46] X. Zhu, S. Zhang, R. Hu, Y. Zhu, and J. Song, "Local and global structure preservation for robust unsupervised spectral feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 3, pp. 517–529, Mar. 2018.

[47] X. Zhu, S. Zhang, W. He, R. Hu, C. Lei, and P. Zhu, "One-step multi-view spectral clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 10, pp. 2022–2034, Oct. 2019.

[48] X. Zhu, S. Zhang, Y. Zhu, W. Zheng, and Y. Yang, "Self-weighted multi-view fuzzy clustering," *ACM Trans. Knowl. Discovery Data*, vol. 14, no. 4, p. 48, 2020,

[49] Y. Yang, Z. Ma, Y. Yang, F. Nie, and H. T. Shen, "Multitask spectral clustering by exploring intertask correlation," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 1069–1080, May 2015.

[50] X. Chen, R. Chen, Q. Wu, Y. Fang, F. Nie, and J. Z. Huang, "LABIN: Balanced min cut for large-scale data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 3, pp. 725–736, Mar. 2020.

[51] X. Chen, W. Hong, F. Nie, D. He, M. Yang, and J. Z. Huang, "Spectral clustering of large-scale data by directly solving normalized cut," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1206–1215.

[52] W. Zhu, F. Nie, and X. Li, "Fast spectral clustering with efficient large graph construction," in *Proc. ICASSP*, 2017, pp. 2492–2496.

[53] L. Yang, X. Liu, F. Nie, and M. Liu, "Large-scale spectral clustering based on representative points," *Math. Problems Eng.*, vol. 2019, Dec. 2019, Art. no. 5864020.

[54] X. Yang, W. Yu, R. Wang, G. Zhang, and F. Nie, "Fast spectral clustering learning with hierarchical bipartite graph for large-scale data," *Pattern Recognit. Lett.*, vol. 130, pp. 345–352, Feb. 2020.

[55] C. Wang, F. Nie, R. Wang, and X. Li, "Revisiting fast spectral clustering with anchor graph," in *Proc. ICASSP*, 2020, pp. 3902–3906.

[56] F. He, F. Nie, R. Wang, X. Li, and W. Jia, "Fast semisupervised learning with bipartite graph for large-scale data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 2, pp. 626–638, Feb. 2020.

[57] F. He, F. Nie, R. Wang, H. Hu, W. Jia, and X. Li, "Fast semi-supervised learning with optimal bipartite graph," *IEEE Trans. Knowl. Data Eng.*, early access, Jan. 21, 2020, doi: 10.1109/TKDE.2020.2968523.
[58] W. Liu, J. He, and S. Chang, "Large graph construction for scalable semi-supervised learning," in *Proc. ICML*, 2010, pp. 679–686.
[59] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in *Proc. KDD*, 2014, pp. 977–986.
[60] M. Šorel and F. Šroubek, "Fast convolutional sparse coding using matrix inversion lemma," *Digit. Signal Process.*, vol. 55, pp. 44–51, Aug. 2016.
[61] S. Mehrkanoon, C. Alzate, R. Mall, R. Langone, and J. A. K. Suykens, "Multiclass semisupervised learning based upon kernel spectral clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 4, pp. 720–733, Apr. 2015.
[62] X. Chen and D. Cai, "Large scale spectral clustering with landmark-based representation," in *Proc. AAAI*, 2011, pp. 313–318.
[63] M. M. R. Khan, R. B. Arif, M. A. B. Siddique, and M. R. Oishe, "Study and observation of the variation of accuracies of KNN, SVM, LMNN, ENN algorithms on eleven different datasets from UCI machine learning repository," *CoRR*, vol. abs/1809.06186, pp. 124–129, Sep. 2018.

Jingyu Wang (Member, IEEE) received the Ph.D. degree in signal, image, and automation from the Université Paris-Est, Paris, France, in 2015.

He is currently an Associate Professor with the School of Astronautics, School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an, China. His research interests include information processing, computer vision, and intelligent perception.

Zhenyu Ma is currently pursuing the M.E. degree with the School of Astronautics, Northwestern Polytechnical University, Xi'an, China.

His research interests include machine learning and computer vision.

Feiping Nie (Member, IEEE) received the Ph.D. degree in computer science from Tsinghua University, Beijing, China, in 2009.

He has authored over 100 articles in top journals and conferences, including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the *International Journal of Computer Vision*, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON NEURAL NETWORKS, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the *ACM Transactions on Knowledge Discovery from Data*, the *Bioinformatics*, the International Conference on Machine Learning, the Conference on Neural Information Processing Systems, the Knowledge Discovery and Data Mining Conference, the International Joint Conference on Artificial Intelligence, the Association for the Advancement of Artificial Intelligence, the International Conference on Computer Vision, the Conference on Computer Vision and Pattern Recognition, and the *ACM Multimedia*. His current research interests include machine learning and its applications, such as pattern recognition, data mining, computer vision, image processing, and information retrieval.

Dr. Nie is currently serving as an associate editor or a PC member for several prestigious journals and conferences in the related fields. His articles have been cited over 5000 times (Google scholar).

Xuelong Li (Fellow, IEEE) is currently a Full Professor with the School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an, China.