IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 27, 2019, accepted March 9, 2019, date of publication March 22, 2019, date of current version April 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2907043

Fast Adaptive K-Means Subspace Clustering for High-Dimensional Data

XIAO-DONG WANG[1], **RUNG-CHING CHEN**[2], **FEI YAN**[1], **ZHI-QIANG ZENG**[1], **AND CHAO-QUN HONG**[1]

[1]College of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China
[2]Department of Information Management, Chaoyang University of Technology, Taichung 413, Taiwan

Corresponding author: Rung-Ching Chen (crching@cyut.edu.tw)

This work was supported in part by the National Natural Science Foundation of China, China, under Grant 61871464, in part by the National Natural Science Foundation of Fujian Province, China under Grant 2017J01511, in part by the Scientific Research Fund of Fujian Provincial Education Department under Grant JAT170417 and Grant JAT160357, and in part by the Ministry of Science and Technology, Taiwan, under Grant MOST-104-2221-E-324-019-MY2, Grant MOST-106-2221-E-324-025, Grant MOST-107-2221-E-324-018-MY2, and in part by the ''Climbing'' Program of Xiamen University of Technology under Grant XPDKQ18012.

ABSTRACT In many real-world applications, data are represented by high-dimensional features. Despite the simplicity, existing K-means subspace clustering algorithms often employ eigenvalue decomposition to generate an approximate solution, which makes the model less efficiency. Besides, their loss functions are either sensitive to outliers or small loss errors. In this paper, we propose a fast adaptive K-means (FAKM) type subspace clustering model, where an adaptive loss function is designed to provide a flexible cluster indicator calculation mechanism, thereby suitable for datasets under different distributions. To find the optimal feature subset, FAKM performs clustering and feature selection simultaneously without the eigenvalue decomposition, therefore efficient for real-world applications. We exploit an efficient alternative optimization algorithm to solve the proposed model, together with theoretical analyses on its convergence and computational complexity. Finally, extensive experiments on several benchmark datasets demonstrate the advantages of FAKM compared to state-of-the-art clustering algorithms.

INDEX TERMS Dimension reduction, feature selection, K-means, discriminative embedded clustering, adaptive learning.

I. INTRODUCTION

Clustering has been widely used as one of the most fundamental techniques in machine learning [1], [2]. Over the past decades, we have witnessed its significant effectiveness in many applications, such as multimedia annotation [3], [4], remote sensing [5], gene expression analysis [6], [7], and so on.

K-means clustering (KM) is a frequently used clustering method [8] and has been extensively applied in many applications for its efficiency and simplicity. Typically, KM iteratively determines the assignment of each data point to the cluster centroids according to certain similarity measurements and updates the cluster centroids subsequently. However, as illustrated in Fig.1, KM is known to be sensitive to the noises and outliers by minimizing the squared l_2-norm based loss function. For instance, in Fig.1(a), the outliers (dotted blue circles) will be assigned larger weight value (thick line) and lead to a strong bias when updating the cluster centroids, resulting in a misclassified point (red circle). To overcome this problem, a robust KM (RKM) with the $l_{2,1}$-norm was proposed in [9] and was proven its effectiveness in multi-view applications. Du *et al.* [10] also extended KM with multiple kernels based on the $l_{2,1}$-norm error measurement. Nevertheless, these extensions of KM tend to overpenalize for small loss error. Take the Fig.1(b) for example, RKM assigns larger weight value to data points in high-density regions and smaller weight value to those in low-density regions. Although such a strategy can properly handle the outliers in Fig.1(a), the weight value of the data points in high-density regions are easily gotten over-overestimated, resulting in a misclassified point (blue square).

Driven by the rapid development of multimedia technologies, we have witnessed a boosting growth of

The associate editor coordinating the review of this manuscript and approving it for publication was Chuan Zhou.

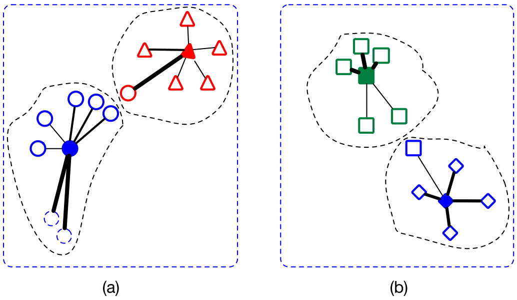


FIGURE 1. Two toy examples of KM and RKM. (a) KM suffers from outliers, (b)RKM suffers from over-penalizing.

high-dimensional data. As demonstrated in previous works [11]–[13], it is impractical to directly cope with these high-dimensional data for unnecessary noises and relevant features they contain. This problem also frequently occurs in KM and its extensions, and thus degrades their clustering performance. One direct way to solve this problem is to map the high-dimensional data into the low-dimensional one by dimension reduction approaches, such as Locally Linear Embedding (LLE) [14] and Principle Component Analysis (PCA) [15]–[17], followed by KM subsequently. For instance, a clustering algorithm based on PCA and KM, named PCAKM, was introduced in [18]. To better utilize the manifold information, Yin *et al.* [19] applied LLE to select the feature subspace and performed KM on the projected data. Nevertheless, a major drawback of these algorithms is that they perform subspace learning and clustering separately, where the obtained subspace may not be optimal for subsequent clustering tasks.

To efficiently improve the accuracy of subspace clustering, researchers found that it is beneficial to perform subspace learning and clustering jointly [20]–[23]. For instance, the Linear Discriminant Analysis (LDA) and KM were adaptively employed into a joint framework (LDAKM) in [20]. In this model, clustering is adaptively cooperated with subspace learning, making it more effective to handle relevant features. However, the LDA component in LDAKM fails when the number of data samples is smaller than that of features. To fix this problem, several extensions of LDA, including Maximum Margin Criterion (MMC) [21], Orthogonal Centroid Method (OCM) [22], and Orthogonal Least Squares Discriminant Analysis (OLSDA) [23] were proposed. Compared with the LDAKM, these extensions of LDA achieve many successes in various applications when cooperating with KM [24]. To better explore discriminative information among high-dimensional data, Hou *et al.* [24] proposed a general framework to perform PCA and KM simultaneously called Discriminative Embedded Clustering (DEC). Xu *et al.* [25] proposed a multi-view Re-weighted Discriminatively Embedded K-means (RDEKM), which can iteratively control the distribution of different views in low-dimensional feature space. Nonetheless, there are several limitations of these KM type discriminative subspace learning methods. On the one hand, all of these methods attempt to find the optimal subspace using the orthogonal linear transformation. In this way, one needs to employ eigenvalue decomposition to compute an approximate solution. Such imposed eigenvalue decomposition will result in a heavy computational burden, making it inapplicable to extremely high-dimensional data. What is worse, it is also difficult to understand the relationship between the transformed low-dimensional feature space and the original one.

Feature selection is another powerful subspace learning method to find the most representative feature subset and preserve the original data structure simultaneously [26], [27]. Boutsidis *et al.* [28] selected features by constructing a probability distribution for feature space, and later described a method where one could select features using deterministic technique [29]. However, they are two-stage approaches, which means they need to find feature subset first and perform clustering concerning the selected features. In addition, they also neglect the discriminative information among data. Wang *et al.* [30] implemented a one-stage KM type subspace clustering with trace ratio LDA and feature learning. Nevertheless, it also depends on the eigenvalue decomposition.

To address the above challenges, in this paper, a novel fast adaptive K-means subspace clustering (named FAKM) is introduced. Inspired by DEC and prevalent success of adaptive learning [31]–[33], we intend to design a robust and elastic discriminative subspace clustering. Compared with most of KM type subspace clustering, FAKM is capable of handling the outliers and the over-penalizing problem by an adaptive loss function. Meanwhile, to efficiently preserve the information in the original feature space, we construct a special feature selection matrix. With such a matrix, FAKM can efficiently select the most representative subspace without the help of eigenvalue decomposition and thus is suitable for real-world applications. To efficiently optimize the proposed objective function, we also develop an iterative algorithm and prove its convergence.

The main contributions of this paper are listed as follows.

1.FAKM combines the feature selection and clustering into a coherent framework, which exploits the reduced dimensional subspace by a special selection matrix instead of eigenvalue decomposition, making it applicable for very high-dimensional data.

2.For real-world applications, we introduce an adaptive loss function to measure the fitness over dimensionally reduced data and the cluster centroids, which enhances the robustness of FAKM to noises and redundant features.

3.We proposed an efficient iterative approach to optimize FAKM with theoretical analysis on its convergence and the computational complexity.

The rest of this paper is organized as follows. The related works are first reviewed in Section II. Then, we elaborate the framework of our FAKM algorithm and provide an iterative optimization approach in Section III. Section IV reports and analyzes comparison results on various kinds of datasets. Finally, we show the conclusion and future works in Section V.

TABLE 1. Important notations.

Notation	Description
D	Dimensionality of samples;
d	Dimensionality of embedded subspace;
n	Number of samples;
c	Number of classes;
λ	Balance parameter;
σ	Adaptive parameter;
$x_i \in \mathbb{R}^D$	The i-th sample;
$y_i \in \mathbb{R}^d$	The i-th sample in low dimensional space;
$f_i \in \mathbb{R}^c$	Class indicator vector of the i-th sample;
$X = [x_1, x_2, \cdots, x_n]$	Data matrix of the whole samples;
$Y = [y_1, y_2, \cdots, y_n]^T$	Data matrix of X in low-dimensional space;
$Q \in \mathbb{R}^{D\times d}$	Linear transformation matrix;
$W \in \mathbb{R}^{D\times d}$	Selection matrix;
$G \in \mathbb{R}^{d\times c}$	Cluster centroid matrix;
$F \in \{0, 1\}^{n\times c}$	Cluster indicator matrix;

II. RELATED WORKS

In this section, we mainly focus on robust KM and KM type discriminative subspace learning, which are closely related to FAKM. Before going into the details, let us introduce some important notations.

A. NOTATIONS

Given a dataset $X = \{x_i \in \mathbb{R}^D | i = 1, 2, \cdots, n\}$ with D features, the main task of subspace clustering is to partition X into c clusters in the embedded feature space $\{y_i \in \mathbb{R}^d | i = 1, 2, \cdots, n\}$ by a linear transformation matrix $Q \in \mathbb{R}^{D\times d}$ $(d \ll D)$. For simplicity, we assume X is centered, i.e., $\sum_i x_i/n = \mathbf{0}$, and denote $Y = [y_1, y_2, \cdots, y_n]^T \in \mathbb{R}^{n\times d}$, $Tr(.)$ is the trace operator. $X_{i:}$ stands for the i-th row of matrix X. Table 1 summarizes the other notations, of which the meanings will be explained when they are first used.

B. ROBUST K-MEANS CLUSTERING

KM aims at grouping given dataset X into c disjoint clusters $\{\mathcal{C}_1, \mathcal{C}_1, \cdots, \mathcal{C}_c\}$. It tries to find the optimal solution according to the clustering fitness using squared Euclidean distance measurement. Let ***Ind*** denote the cluster indicator matrix set, that is, $\boldsymbol{Ind} = \{\widetilde{F} | \widetilde{F} \in \{0, 1\}^{n\times c}, \sum_{j=1}^{c} \widetilde{F}_{ij} = 1, \forall i = 1, 2, \cdots, n\}$. The objective function of KM can be formulated as

$$\min_{F\in \boldsymbol{Ind}} \sum_{i=1}^{n}\sum_{k=1}^{c} F_{ik}\|x_i - \bar{x}_k\|_2^2, \tag{1}$$

where $F = [f_1, f_2, \cdots, f_n]^T \in \boldsymbol{Ind}$ is the cluster indicator matrix. $F_{ik} = 1$ if x_i belongs to the k-th cluster, and $F_{ik} = 0$ otherwise; $\bar{x}_k$ is the centroid of k-th cluster.

Although KM is simple and can be solved efficiently, the squared l_2-norm loss imposed in Eq.(1) is very sensitive to outliers. With the development of $l_{2,1}$-norm technologies [26], [34], [35], amount of robust loss functions are designed and shown their empirical successes in various applications. For instance, Cai *et al.* [9] and Du *et al.* [10] designed an $l_{2,1}$-norm fitness function for robust clustering as follows:

$$\min_{F\in \boldsymbol{Ind}} \sum_{i=1}^{n} \sqrt{\sum_{k=1}^{c} F_{ik}\|x_i - \bar{x}_k\|_2^2}. \tag{2}$$

By introducing such an $l_{2,1}$-norm, the impact of outliers in KM can be reduced. However, due to the non-smoothness of the $l_{2,1}$-norm, Eq.(2) is hard to be directly optimized. Although the re-weighted optimization approach is efficient to solve this problem [36], it is sensitive to small loss [37].

C. K-MEANS WITH DISCRIMINATIVE SUBSPACE LEARNING

The primary principle of LDAKM is to generate labels for LDA by KM and to perform KM in the low-dimensional feature space obtained by LDA in a joint manner. Particularly, it tries to minimize the following objective function:

$$\min_{F\in \boldsymbol{Ind}} \sum_{i=1}^{n}\sum_{k=1}^{c} F_{ik}\|y_i - g_k\|_2^2, \tag{3}$$

where g_k is the cluster centroid of k-th class in the embedded feature space.

To make use of discriminative information among data, LDAKM also computes the total scatter matrix S_t, within class scatter matrix S_w and between class scatter matrix S_b as

$$\begin{aligned} S_t &= \sum_{i=1}^{n} x_i x_i^T \\ S_b &= \sum_{k=1}^{c} n_k \bar{x}_k \bar{x}_k^T \\ S_w &= \sum_{k=1}^{c}\sum_{i\in\mathcal{C}_k} (x_i - \bar{x}_k)(x_i - \bar{x}_k)^T. \end{aligned} \tag{4}$$

Assuming that the transformation is linear, i.e., $Y = X^T Q$, LDAKM is able to find the optimal transformation matrix Q by maximizing the following objective function:

$$\max_{Q^T Q = I} Tr((Q^T S_b Q)(Q^T S_w Q)^{-1}), \tag{5}$$

where the orthogonal constraint $Q^T Q = I$ is imposed to avoid a trivial solution.

As discussed in previous section, LDAKM is difficult to handle the "small-sample-size" problem, where the number of features is smaller than that of instances. In such a scenario, the objective function of LDA in Eq.(5) fails as S_w tends to be singular. To cope with this problem, Orthogonal Centroid Method K-means (OCMKM) subspace learning was proposed [22], [24], which is to maximize the following objective function:

$$\max_{Q^T Q = I} Tr(Q^T S_b Q). \tag{6}$$

Maximum Margin Criterion K-means (MMCKM) is another extension of LDAKM. It tries to maximize the average margin among classes [21], [24], which has the following objective function:

$$\max_{Q^T Q=I} Tr(Q^T (S_b - S_w)Q). \tag{7}$$

Another way of solving the trivial solution problem of LDA is to reform its fitness function [23], [24]. For instance, Orthogonal Least Squares Discriminant Analysis based K-means clustering (OLSDAKM) transforms LDAKM to the least squares problem under the orthogonal constraint [23]. OLSDAKM tries to optimize the objective function as follows:

$$\min_{Q^T Q=I} Tr(Q^T S_w Q). \tag{8}$$

DEC is another school of discriminative subspace learning approaches, which performs PCA and KM simultaneously. Its objective function can be defined as

$$\max_{Q^T Q=I} Tr(Q^T (S_t - \lambda S_w)Q), \tag{9}$$

where λ is a balance parameter.

Different from LDAKM, the problems in Eq.(6), Eq.(7), Eq.(8), and Eq.(9) can be solved by general eigenvalue decomposition of S_b, $S_b - S_w$, S_w, and $S_t - \lambda S_w$ respectively, without the inverse operation of S_w. Therefore, these methods are more suitable for the real-word applications.

Though these discriminative subspace clustering algorithms have been demonstrated their effectiveness in many applications, they can be hardly applied to extremely high-dimensional data due to their dependence on eigenvalue decomposition. What is more, it is hard to directly explore the relationship between the newly transformed low-dimensional data and the original one.

To preserve the original features in the low-dimensional feature space, Wang *et al.* [30] proposed a discriminative learning method with Trace Ratio Formulation and K-means Clustering (TRACK), which jointly unifies trace ratio LDA, K-means, and regularization feature learning into a single objective function. However, similar to LDAKM, TRACK still suffers from ‘‘small-sample-size’’ problem and high computational complexity problem.

III. THE PROPOSED METHOD

In this section, we first conduct the formulation of FAKM and design an efficient iterative algorithm to optimize it. After that, the computational complexity of FAKM is also discussed.

A. SELECTION MATRIX FOR CLUSTERING

To effectively select features, we first introduce a column vector w_i that has the following form:

$$w_i = [\underbrace{0, \cdots, 0}_{i-1}, 1, \underbrace{0, \cdots, 0}_{D-i}]^T.$$

Then we define a special selection matrix W as

$$W = [w_{I(1)}, w_{I(2)}, \cdots, w_{I(d)}], \tag{10}$$

where the vector I is a permutation of $\{1, 2, \cdots, D\}$.

From Eq.(10), one can observe that W is extremely sparse and is indeed a column-full-rank transformation matrix. Then, with this selection matrix, one can transform the D-dimensional data x_i into d-dimensional data y_i ($d \ll D$) as follows:

$$y_i = W^T x_i. \tag{11}$$

Considering the difficulty of dealing with high-dimensional data, we aim to partition them in the low-dimensional feature space. Thereafter, we integrate Eq.(11) into KM and arrive at

$$\begin{aligned} \min_{F\in\boldsymbol{Ind},W,G} \sum_{i=1}^{n}\sum_{k=1}^{c} F_{ik}\|W^T x_i - g_k\|_2^2 &= \min_{F\in\boldsymbol{Ind},W,G} \sum_{i=1}^{n} \|W^T x_i - Gf_i\|_2^2 \\ &= \min_{F\in\boldsymbol{Ind},W,G} \|X^T W - FG^T\|_F^2, \end{aligned} \tag{12}$$

where $G = [g_1, g_2, \cdots, g_c] \in \mathbb{R}^{d\times c}$ is the cluster centroid matrix and $g_k (1 \le k \le c)$ is the centroid of k-th cluster.

Notably, thanks to the special structure of the feature selection W in Eq.(10), compared with the KM type subspace clustering, the framework in Eq.(12) has the following three advantages:

(1) The embedding clustering algorithm is computationally efficient due to the extreme sparsity of W (detailed analysis can be found in Section III-C);

(2) The selection matrix W leads to an easy original feature preservation when performing dimension reduction;

(3) The particular structure of W makes the eigenvalue decomposition operation used in the KM type subspace clustering unnecessarily. Thus, it is easy to tackle extremely high-dimensional data.

B. ADAPTIVE DISCRIMINATIVE CLUSTERING WITH FEATURE SELECTION

To explore discriminative information among clusters, we want to preserve the representative features on which the data points within the same group are close to each other while the data points of different groups are far from each other. Inspired by the recent developments on discriminative subspace clustering [24], we propose the following objective function:

$$\max_{F\in\boldsymbol{Ind},W,G} Tr(W^T S_t W) - \lambda\|X^T W - FG^T\|_F^2. \tag{13}$$

Similar to Eq.(1), the second term in Eq.(13) involves the squared Frobenius norm, which is easily affected by the data noises or outliers. To improve the robustness of clustering, following the previous works [31], [32], we impose an adaptive loss function into our objective function. For arbitrary matrix $A = [a_1, a_2, \cdots, a_n] \in \mathbb{R}^{m\times n}$, the adaptive loss function of matrix A is defined as

$$\|A\|_\sigma = \sum_i \frac{(1+\sigma)\|a_i\|_2^2}{\|a_i\|_2 + \sigma}, \tag{14}$$

where $\sigma > 0$ is an adaptive parameter.

One can easily verify that $\|A\|_\sigma$ is nonnegative, convex, and twice differentiable, thus is desirable for loss function and optimization. Additionally, when $\sigma \to \infty$, $\|A\|_\sigma$ is actually the Frobenius norm; When $\sigma \to 0$, $\|A\|_\sigma$ tends to become the group sparsity $l_{2,1}$-norm $\|A\|_{2,1}$, which has been proven its robustness to outliers in previous works [26], [38], [39]. Thus, the loss function in Eq.(14) is definitely an elastic function with an adaptive parameter σ. After imposing the adaptive loss function Eq.(14) into Eq.(13), our objective function becomes

$$\max_{F\in\boldsymbol{Ind},W,G} Tr(W^TS_tW) - \lambda\|X^TW - FG^T\|_\sigma. \tag{15}$$

C. OPTIMIZATION

In this section, a simple and efficient iterative algorithm is presented to solve problem (15). More specifically, we alternatively update one while keeping the others fixed. According to the theory in [31], the problem in Eq.(15) can be transformed into the following problem:

$$\max_{F\in\boldsymbol{Ind},W,G,\tau_i} Tr(W^TS_tW) - \lambda\sum_{i=1}^{n}\tau_i\|W^Tx_i - Gf_i\|_2^2, \tag{16}$$

where $\tau_i = (1+\sigma)\frac{\|W^Tx_i - Gf_i\|_2 + 2\sigma}{2(\|W^Tx_i - Gf_i\|_2 + \sigma)^2}$.

Denote Δ as a diagonal matrix with its i-th diagonal element as τ_i and $U = [u_1, u_2, \cdots, u_n]^T = X^TW - FG^T$, where $u_i \in \mathbb{R}^{d\times 1}(1 \leq i \leq n)$ is the i-th column vector of U. We have

$$\max_{F\in\boldsymbol{Ind},W,G,\Delta} Tr(W^TS_tW) - \lambda Tr(U^T\Delta U). \tag{17}$$

Note that the objective function in Eq.(17) is not jointly convex with the variables F and G. Besides, Δ is dependent on W, G, and F, and each entry of F is a discrete integer value. We propose the following optimization steps to solve it.

Step 1: Solving F while fixing W, Δ and G

When W, Δ, and G are fixed, the optimization problem in Eq.(17) becomes

$$\begin{aligned}\min_{F\in\boldsymbol{Ind}} &\sum_{i=1}^{n}\tau_i\|W^Tx_i - Gf_i\|_2^2\\ &= \min_{F\in\boldsymbol{Ind}}\sum_{i=1}^{n}\tau_i\sum_{k=1}^{c}\|W^Tx_i - g_k\|_2^2F_{ik}.\end{aligned} \tag{18}$$

The minimization of the objective function in Eq.(18) with respect to F can be decomposed into solving n independent sub-problems. Considering the discrete structure of F, we can find its optimal cluster indicator for each data points as follows:

$$F_{ij} = \begin{cases} 1, & j = \arg\min_k \|W^Tx_i - g_k\|_2^2 \\ 0, & \text{Otherwise.}\end{cases} \tag{19}$$

Step 2: Solving G, W while fixing Δ and F

To optimize W, one needs to find the optimal permutation of $\{1, 2, \cdots, D\}$ in Eq.(10). It is unrealistic to estimate W using the method of exhaustion for the high-dimensional data. Conversely, we determine W iteratively by the cluster structure fitness and discriminative power of the embedded data.

When the variables Δ and F are given, we can derive the optimal solution of G in a closed form. By setting the derivative of Eq. (17) with respect to G to zero, we obtain

$$G = W^TX\Delta F(F^T\Delta F)^{-1}. \tag{20}$$

Substituting G into (17), we arrive at

$$\begin{aligned}\max_W Tr(W^T(S_t - \lambda\widetilde{S}_w)W) &= \max_W Tr(W^TMW)\\ &= \max_W \sum_{i=1}^{d} Tr(w_i^TMw_i),\end{aligned} \tag{21}$$

where $M = S_t - \lambda\widetilde{S}_w$, $\widetilde{S}_w = XNX^T$ and $N = \Delta - \Delta F(F^T\Delta F)^{-1}F^T\Delta$[1].

Recalling the particular structure of W in Eq.(10), we can effectively obtain the optimal solution of the problem in Eq.(21) by locating the first d largest diagonal elements of matrix M. Unfortunately, in practice, storing matrix M requires $O(D^2)$ memory cost and is extremely expensive for high-dimensional data. Noticing the sparsity of matrix N, we can calculate the diagonal elements of matrix M efficiently as follows:

$$M_{ii} = \|X_{i:}\|_2^2 - \lambda\|(XN^{\frac{1}{2}})_{i:}\|_2^2. \tag{22}$$

Step 3: Updating Δ by calculating its i-th element as: $\tau_i = (1+\sigma)\frac{\|u_i\|_2 + 2\sigma}{2(\|u_i\|_2+\sigma)^2}$.

Following the proof given in IV-E, we can easily verify that the above solving strategy will converge. Nevertheless, as the solution of F in Eq.(19) is sensitive to the initialization, the final solution of FAKM is not adequate. Specifically, we first get the optimal solution of F and use it to solve W and G accordingly. But when F needs to be updated next, its initial cluster centroids are derived from the previous F, resulting in an unstable solution. To address this problem, following [24], when updating F, we randomly initialize it several times and select the one with the smallest objective function value in Eq.(19). Mathematically, in the k-th iteration, we derive the optimal F_k^*, G_k^* and W_k^*. In the $(k+1)$-th iteration, the random initializations are $\{F_{k+1}^1, F_{k+1}^2, \cdots, F_{k+1}^l, \}$, where l is the number of random initializations and is empirically set to 20 in our experiment. We update F by the following rule:

$$F_{k+1}^* = \begin{cases} F_{k+1}^j, & \|X^T(W_k^*)^T - F_{k+1}^j(G_k^*)^T\|_\sigma < \\ & \|X^T(W_k^*)^T - F_k^*(G_k^*)^T\|_\sigma \\ F_k^*, & \text{Otherwise,}\end{cases} \tag{23}$$

where F^* is calculated by Eq.(19) with W_i^* and G_i^*.

We summarize the iterative optimization process in Algorithm 1.

[1] In practice, to ensure that $F^T\Delta F$ is invertible, we will add it with a small constant ϵ, that is $F^T\Delta F + \epsilon$.

```
Algorithm 1 The algorithm to Solve the Problem (17).
Input:
    The input data $X \in \mathbb{R}^{D\times n}$
    The reduced dimension number $d$, the number of clusters $c$, regularization parameter $\lambda$, and adaptive parameter $\sigma$
Output:
    Selection matrix $W$
    Cluster indicator matrix $F$
    Cluster centroid matrix $G$
Initialize $\Delta$ as an identity matrix, randomly initialize $W$ and $G$
repeat
    Update the $F$ using Eq.(19) by Eq.(23)
    Update $G$ by Eq.(20)
    Update $W$ by finding the $d$ largest diagonal elements of $M$ in Eq.(22)
    Update $\Delta$ by calculating its $i$-th element as $\tau_i = (1+\sigma)\frac{\|u_i\|_2+2\sigma}{2(\|u_i\|_2+\sigma)^2}$
until Convergence
Return $W$, $F$, and $G$
```

D. COMPLEXITY ANALYSIS

The most computational cost of Algorithm 1 involves three components. The first component is traditional KM in embedded feature space and has a computational complexity $O(dcn)$. The second component is the calculation of G, which has computational complexity $O(dcn + c^2 n)$. The third one is optimization for W in the problem (21), we consider the sparsity of N and calculate the diagonal elements of M with complexity $O(Dn)$. Moreover, to find the optimal W, we need to seek the d largest diagonal elements with complexity $O(D+dlogd)$. Thus, suppose the number of iterations for embedded K-means clustering in Eq.(19) is T_k, and the number of iterations for the whole algorithm is T_t. For high-dimensional data $c \ll D$ and $d \ll D$, the computational complexity of FAKM is $O(T_t(T_k(dcn) + dcn + c^2n + Dn + dlogd)) \sim O(Dn)$. Besides, FAKM involves the storage of several matrices, i.e., X, F and G, requiring $O(Dn+cn+dc)$ memory cost. Therefore, the computational cost of the proposed algorithm is linear with respect to the dimensionality of data points. According to the analyses above, our algorithm is capable of handling high-dimensional data. The complexity of FAKM and the other related algorithms are listed in Table 2, where SRDEKM stands for the single-view version of RDEKM.

E. CONNECTION WITH PREVIOUS METHODS

Proposition 1: DEC is a special case of the proposed FAKM clustering method, when $\sigma \to \infty$.

Proof: As seeing from Eq.(21), when $\sigma \to \infty$, we have $\Delta \to I$ and $\widetilde{S}_w \to S_w$. Considering the objective function of DEC in Eq.(9), we can confirm that DEC is a special case of FAKM with a different transformation matrix when $\sigma \to \infty$.

TABLE 2. Complexity of the compared algorithms.

Method	Computation	Memory
KM	$O(Dcn)$	$O(Dn+cn+dc)$
LDAKM	$O(D^2n+dcn)$	$O(D^2+Dn+cn+dc)$
MMCKM	$O(D^2n+dcn)$	$O(D^2+Dn+cn+dc)$
OLSDAKM	$O(D^2n+dcn)$	$O(D^2+Dn+cn+dc)$
OCMKM	$O(D^2n+dcn)$	$O(D^2+Dn+cn+dc)$
SRDEKM	$O(D^2n+dcn)$	$O(D^2+Dn+cn+dc)$
DEC	$O(D^2n+dcn)$	$O(D^2+Dn+cn+dc)$
FAKM	$O(Dn+dlogd+dcn)$	$O(Dn+cn+dc)$

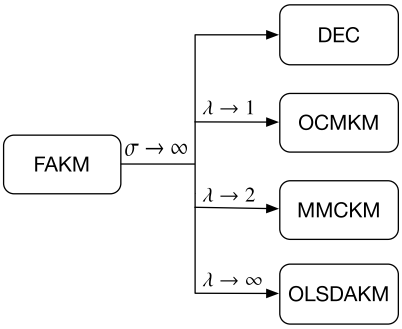


FIGURE 2. Connections of FAKM and other related methods.

Proposition 2: OCMKM, MMCKM, and OLSDAKM are special cases of FAKM, when $\sigma \to \infty$ and $\lambda \to 1$, $\lambda \to 2$, $\lambda \to \infty$ respectively.

Proof: Similar to the proof of *proposition 1*, when $\sigma \to \infty$, we have $\Delta \to I$ and $\widetilde{S}_w \to S_w$. Since $S_t = S_w+S_b$, the objective function of FAKM in Eq.(21) is reduced to the objective function of OCMKM, MMCKM, and OLSDAKM, when $\lambda \to 1$, $\lambda \to 2$, $\lambda \to \infty$ respectively. The relations of FAKM and other related methods are illustrated in Fig.2.

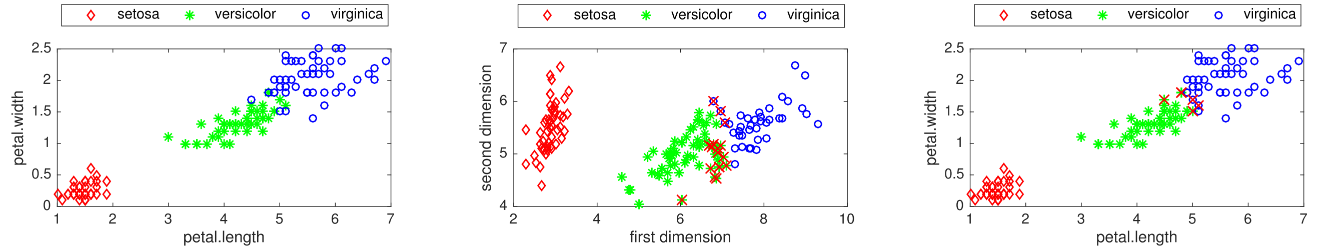
IV. EXPERIMENTS

A. EXPERIMENT SETUP

We have conducted analytical experiments on a diversity of seven public datasets to evaluate the performance. For each dataset, we normalize all the values in the range of $[-1, 1]$. These datasets include: two face image datasets (Yale[2], and Umist[3]), three UCI datasets[4] (Glass, Breast and Vehicle), one text dataset (WebKB [40]), one document dataset (TDT2[5]). For TDT2, it consists of 11201 on-topic documents collected from 96 semantic categories. In this dataset, following [41], we removed those documents belong to more than two categories and used the top 10 categories. Detailed information of the these datasets is summarized in Table 3.

To test the efficiency of the compared algorithms for high-dimensional data, we also generate two groups of synthetic datasets with increasing feature size based on TDT2 and WebKB. Concretely, we randomly select p features from the original datasets, where p is a scalar in

[2] http://www.cvc.yale.edu/projects/yalefaces/yalefaces.html
[3] http://images.ee.umist.ac.uk/danny/database.html
[4] http://archive.ics.uci.edu/ml/
[5] http://www.nist.gov/speech/tests/tdt/tdt98/index.html



(a) (b) (c)

FIGURE 3. Clustering results of DEC and FAKM on iris dataset with reduced dimensionality as 2. (a) Original data with ground truth labels. (b) Clustering result of DEC. (c) Clustering result of FAKM.

TABLE 3. A brief description of the selected datasets.

Datasets	Classes	# of instances	Dimensions	# of reduced dimensions
Glass	6	214	9	{3,4,5,6,7,8}
Breast	2	699	10	{2,3,· · · ,8,9}
Vehicle	4	846	18	{3,4,· · · ,9,10}
Umist	20	575	644	{100,200,300,400,500}
Yale	15	165	1024	{100,200,· · · ,900,1000}
WebKB	7	814	4029	{50,100,150,200,250,300}
TDT2	10	653	36771	{10,50,90,· · · ,370,410,450}

the range of $\{400, 800, \cdots, 3800, 4000\}$ for WebKB and $\{2000, 4000, \cdots, 34000, 36000\}$ for TDT2. Note that the generated synthetic datasets may be less practical for real-world applications due to the randomly selective strategy. However, by using these synthetic datasets, we can better understand algorithm behaviors with different scales of dimensionality.

We compare FAKM with other state-of-the-art KM type subspace clustering algorithms, of which the brief descriptions are listed as follows:

- **KM** is the traditional K-means algorithm and is used as the baseline in this experiment.
- **LDAKM** combines LDA and KM in a joint framework, which could utilize the discriminative power among embedded data [20].
- **OCMKM** is an OCM-based subspace clustering algorithm [22]. It achieves the optimal dimensionality reduction transformation matrix by preserving the clustering information among data.
- **MMCKM** is a MMC type subspace clustering algorithm [21], which imposes a maximum margin criterion to address the "small-sample-size" problem.
- **OLSDAKM** reformulates the objective function of traditional LDA into an orthogonal least squares problem [23], thereby easier to be applied to real-world applications.
- **TRACK** accommodates trace ratio LDA and K-means clustering procedure, and selects the discriminative features using structured sparsity-inducing norms regularization technique.
- **SRDEKM** is a single-view version of re-weighted discriminatively embedded K-means (RDEKM) in [25], which is designed to reduce the influence of noises by imposing a non-squared least-absolute criteria.
- **DEC** is a general discriminative subspace learning framework, where it simultaneously optimizes PCA and KM. Besides, it imposes a balance parameter to control the contributions of within scatter matrix and between scatter matrix [24].

Following the previous works [24], [30], to achieve a fair comparison, all the parameters (if any) of the compared algorithms are tuned by a "grid-search" strategy from $\{10^{-6}, 10^{-4}, 10^{-2}, \cdots, 10^{2}, 10^{4}, 10^{6}\}$, and the best clustering results are recorded with the optimal parameters. We repeat the experiment 50 times independently and report the average results together with the variance.

Two evaluation metrics, i.e., Accuracy (ACC) and Normalized Mutual Information (NMI), are used to measure the clustering performance. Detailed explanations of ACC and NMI can be found in [42] and [43] respectively. For these metrics, higher value indicates better performance.

B. TOY EXAMPLE

We first evaluate FAKM on a 4-dimensional iris dataset. This dataset consists of three groups (i.e., setosa, versicolor, and virginica). Original data with ground truth labels are illustrated in Fig.3(a), where two kinds of features (i.e., the width of petal and the length of petal) are selected for visualization.

We choose DEC and FAKM for comparison. The reduced dimension is set to 2. We first use DEC and FAKM to cluster the original 4-dimensional data separately, and then project the original data to two 2-dimensional data using the obtained optimal transformation matrices. The clustering results of DEC and FAKM are illustrated in Fig.3(b) and Fig.3(c) respectively. The miss-classified data points are marked with the red cross. The results show that FAKM outperforms DEC by generating less miss-classified points. Besides, in Fig.3(b), it is clear that the projected data of DEC show a different cluster structure from the original data in Fig.3(a). In contrast, Fig.3(c) shows that FAKM is more capable of preserving the original data structure by selecting the most representative features, that is, the width of petal and the length of petal.

TABLE 4. Clustering results of compared algorithms (ACC%).

	Glass	Breast	Vehicle	Umist	Yale	WebKB	TDT2
KM	46.79 ± 4.07	95.28 ± 0.00	36.95 ± 0.75	42.79 ± 2.17	43.68 ± 4.37	56.64 ± 1.18	20.60 ± 2.36
LDAKM	47.29 ± 4.05	95.42 ± 0.00	38.26 ± 1.87	43.50 ± 2.53	44.92 ± 3.50	55.66 ± 2.20	
MMCKM	46.73 ± 2.47	95.42 ± 0.00	38.06 ± 0.00	43.74 ± 2.01	45.50 ± 2.86	53.28 ± 2.64	-
OLSDAKM	47.50 ± 4.28	82.98 ± 0.66	40.41 ± 1.41	43.56 ± 2.23	47.36 ± 3.32	53.57 ± 3.23	-
OCMKM	45.50 ± 2.15	95.42 ± 0.00	37.12 ± 0.00	43.66 ± 2.06	47.48 ± 2.73	53.62 ± 2.27	-
TRACK	46.62 ± 5.81	95.42 ± 0.00	44.04 ± 2.90	**49.77 ± 4.38**	46.90 ± 4.27	60.77 ± 0.93	-
DEC	47.24 ± 2.82	95.42 ± 0.00	40.40 ± 0.90	43.56 ± 2.45	47.73 ± 3.14	57.13 ± 3.53	-
SRDEKM	47.87 ± 3.73	83.12 ± 0.00	42.07 ± 1.98	43.85 ± 2.38	46.06 ± 2.77	52.93 ± 2.99	-
FAKM($\sigma = \infty$)	49.13 ± 4.98	95.42 ± 0.00	43.99 ± 2.10	44.25 ± 3.48	47.37 ± 4.07	58.49 ± 6.54	35.72 ± 3.45
FAKM	**49.53 ± 5.96**	**95.57 ± 0.00**	**44.13 ± 2.38**	44.35 ± 3.27	**48.56 ± 4.82**	**67.09 ± 2.42**	**36.22 ± 3.25**

TABLE 5. Clustering results of compared algorithms (NMI%).

	Glass	Breast	Vehicle	Umist	Yale	WebKB	TDT2
KM	32.77 ± 2.49	70.49 ± 0.00	11.32 ± 2.28	64.57 ± 1.70	51.44 ± 3.88	15.12 ± 1.37	9.79 ± 2.29
LDAKM	33.61 ± 2.82	71.17 ± 0.00	10.87 ± 1.28	64.81 ± 1.84	52.09 ± 2.44	17.22 ± 1.55	-
MMCKM	**34.25 ± 1.75**	71.17 ± 0.00	10.41 ± 0.03	65.23 ± 1.59	51.86 ± 2.70	17.45 ± 0.94	-
OLSDAKM	32.60 ± 2.51	29.71 ± 2.07	9.63 ± 1.27	64.92 ± 1.87	53.76 ± 2.27	17.41 ± 0.91	-
OCMKM	33.83 ± 1.24	71.17 ± 0.00	10.02 ± 0.02	**65.53 ± 1.24**	53.78 ± 2.03	17.60 ± 0.66	-
TRACK	32.65 ± 2.68	71.21 ± 0.10	16.81 ± 0.64	65.13 ± 3.91	53.00 ± 2.86	9.00 ± 2.00	-
DEC	33.62 ± 1.37	71.17 ± 0.00	10.20 ± 0.00	**65.53 ± 1.41**	54.19 ± 2.20	**18.53 ± 0.92**	-
SRDEKM	33.92 ± 4.10	29.72 ± 0.00	11.85 ± 3.68	65.02 ± 1.79	52.44 ± 3.01	17.34 ± 0.91	-
FAKM($\sigma = \infty$)	33.62 ± 4.68	71.17 ± 0.00	17.50 ± 0.99	63.79 ± 2.27	53.28 ± 3.64	16.48 ± 1.58	31.09 ± 3.86
FAKM	33.81 ± 3.83	**71.92 ± 0.00**	**17.87 ± 2.41**	63.89 ± 2.39	**54.63 ± 3.55**	16.72 ± 1.42	**31.13 ± 3.07**

C. CLUSTERING PERFORMANCE COMPARISON

To demonstrate the effectiveness of adaptive learning, we set $\sigma = \infty$ of FAKM, which is then reduced to the optimization problem in Eq.(13).

Table 4-5 show the results of clustering, where "-" means "out of memory error" while running the experiment. We have the following observations:

- Most of the KM type subspace clustering algorithms usually outperform KM on each dataset, except that MMCKM, OLSDAKM, and OCMKM fail KM in terms of ACC on some datasets, e.g., WebKB and Glass. However, as shown in Fig.4, the dimensionality can be selectively reduced, making the computation cost of subsequent learning tasks much lower.
- Compared with LDAKM, most of the other subspace clustering algorithms obtain better results on the relatively high-dimensional dataset, that is Umist and Yale. The reason may be that these subspace clustering algorithms improve LDAKM by implementing several flexible fitness functions, which are appropriate for high-dimensional data.
- In most cases, DEC outperforms MMCKM, OCMKM, and OLSDAKM by constructing a more generic discriminative clustering framework.
- Compared with the clustering method with feature learning, i.e., TRACK, FAKM obtains better results on all the benchmark datasets except Umist. The key reason lies in that FAKM is capable of balancing the power of different kinds of scatter matrices, which is more flexible than TRACK.
- FAKM consistently outperforms FAKM ($\sigma = \infty$) and DEC. This observation shows that it is beneficial to incorporate the adaptive learning with clustering.

D. INFLUENCE OF DIMENSION REDUCTION

The following two experiments aim to study the influence of dimension reduction imposed in FAKM.

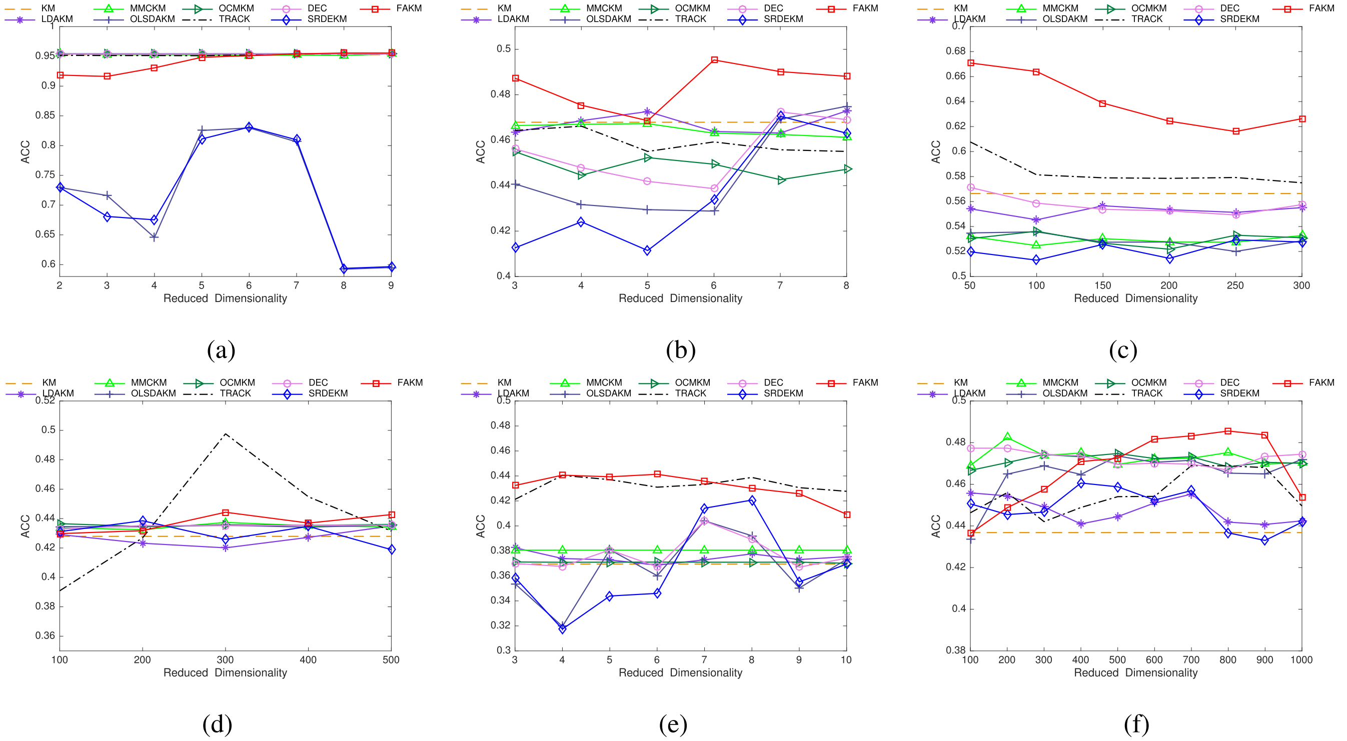
The first experiment is to verify the effectiveness of feature learning in FAKM. To achieve this goal, we simply set the selection matrix in Eq.(15) as a D-dimensional identity matrix, and generate the following equation:

$$\min_{F \in \boldsymbol{Ind}, G} \|X^T - FG^T\|_\sigma . \tag{24}$$

Obviously, without feature learning, FAKM reduced to KM with an adaptive loss function, we name it FAKM-ND. The performance comparison results in terms of ACC are shown in Fig.5. The results show that by imposing the adaptive loss function, FAKM-ND gets better results than KM. FAKM achieves the best results on all the benchmark datasets. Therefore, we argue that the adaptive clustering and feature learning are beneficial for clustering.

Our second experiment is to test the impact of reduced dimensionality variety. The parameter setting is following the strategy in Section IV-A. Similarly, for each reduced dimensionality, we repeat the experiment 50 times and report the average results. Fig. 4 shows the clustering results on several datasets with respect to the reduced dimensionality. We can conclude that:

- Not all of the methods achieve a constant higher ACC consistently when the reduced dimensionality



(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 4. (a) Breast. (b) Glass. (c) WebKB. (d) Umist. (e) Vehicle. (f) Yale. Clustering performance on various datasets w.r.t. different number of reduced dimensionality.

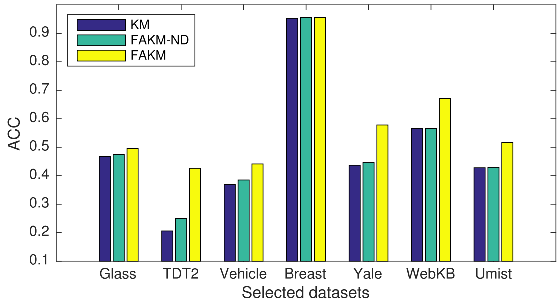


FIGURE 5. Influence of dimension reduction in FAKM.

increases. Therefore, we can assert that dimension reduction benefits clustering.

- When only few dimensionalities preserved, the performance of the compared subspace clustering degrades due to too much information loss. For example, when $d \leq 6$ for DEC on Glass dataset and $d \leq 5$ for FAKM on Breast dataset, the performance of these algorithms is poorer than that of the benchmark algorithm KM.
- Compared with other clustering algorithms, FAKM is more capable of locating the optimal reduced dimensionality. Take the Yale dataset for example, ACC curve of FAKM reaches its peak when the reduced dimensionality is set to 800.
- Of all the different algorithms and datasets, FAKM achieves the best clustering accuracy in most cases. For example, the performance of FAKM improves the second best performance of TRACK by 6.32% on WebKB dataset, which shows a better property of FAKM.

E. CONVERGENCE ANALYSIS

To proof the convergence of the Algorithm 1, we need the following lemma:

Lemma 1 For arbitrary vectors p, q with the same size, the following inequality holds:

$$\frac{\|q\|_2^2}{\|q\|_2+\sigma} - \frac{\|q\|_2+2\sigma}{2(\|q\|_2+\sigma)^2}\|q\|_2^2 \geq \frac{\|p\|_2^2}{\|p\|_2+\sigma} - \frac{\|q\|_2+2\sigma}{2(\|q\|_2+\sigma)^2}\|p\|_2^2. \tag{25}$$

The proof of **Lemma 1** can be found in [31].

*Theorem 1 The iterative approach in **Algorithm 1** monotonically increases the objective function value of optimization problem (15) in each iteration until convergence.*

Proof: Suppose that we have updated W_t, G_t in the t-th iteration. In the $(t+1)$-th iteration, we fix W_t, Δ_t and G_t to optimize F_{t+1} using Eq.(19). Recalling the updating rule in Eq.(23), we have the following inequality:

$$\begin{aligned} &Tr(W_t^T S_t W_t) - \lambda\|X^T W_t - F_t G_t^T\|_\sigma \\ &\quad \leq Tr(W_t^T S_t W_t) - \lambda\|X^T W_t - F_{t+1} G_t^T\|_\sigma . \end{aligned} \tag{26}$$

TABLE 6. Computation time comparison on different datasets with fixed reduced dimensionality.

	KM	LDAKM	MMCKM	OLSDAKM	OCMKM	TRACK	DEC	SRDEKM	FAKM
Glass	0.01± 0.01	0.03± 0.01	0.41± 0.12	0.42± 0.08	0.44± 0.11	0.73± 0.21	0.37± 0.05	0.10± 0.05	0.14± 0.04
Breast	0.01± 0.00	0.05± 0.03	0.26± 0.04	0.32± 0.05	0.34± 0.04	2.87± 0.13	0.28± 0.04	0.13± 0.02	0.12± 0.02
Vehicle	0.02± 0.01	0.19± 0.08	0.78± 0.11	0.54± 0.11	0.86± 0.15	5.61± 0.43	0.96± 0.15	0.42± 0.10	0.21± 0.05
Umist	0.41± 0.10	4.77± 0.36	8.55± 0.88	5.64± 0.50	7.85± 1.36	4.99± 0.43	9.12±1.72	1.32± 0.13	2.41± 0.69
Yale	0.17± 0.04	9.76± 0.89	3.50± 0.60	10.07± 1.45	7.63± 1.03	3.42± 0.34	2.01± 0.29	3.90± 0.17	0.48± 0.09
WebKB	11.26± 2.85	71.55± 3.69	144.61± 10.39	154.26± 14.46	145.22± 11.62	43.42± 4.01	140.11± 13.88	187.11± 44.06	4.25± 0.50
TDT2	64.73± 19.53	-	-	-	-	-	-	-	29.48± 1.45

Next, we utilize Δ_t and F_{t+1} to update G and W using Eq.(21). Let $f(W) = Tr(W^T S_t W)$, $u_i^t = W_t^T x_i - G_t(f_i)_{t+1}$, and $u_i^{t+1} = W_{t+1}^T x_i - G_{t+1}(f_i)_{t+1}$, we get

$$f(W_t) - \lambda \sum_i d_i^t \|u_i^t\|_2^2 \leq f(W_{t+1}) - \lambda \sum_i d_i^t \|u_i^{t+1}\|_2^2. \qquad (27)$$

Note that $d_i^t = (1+\sigma)\frac{\|u_i^t\|_2 + 2\sigma}{2(\|u_i^t\|_2+\sigma)^2}$, so we arrive at

$$f(W_t) - \lambda(1+\sigma)\sum_i \frac{\|u_i^t\|_2 + 2\sigma}{2(\|u_i^t\|_2+\sigma)^2}\|u_i^t\|_2^2 \leq f(W_{t+1}) - \lambda(1+\sigma)\sum_i \frac{\|u_i^t\|_2 + 2\sigma}{2(\|u_i^t\|_2+\sigma)^2}\|u_i^{t+1}\|_2^2. \qquad (28)$$

According to *Lemma 1*, we have

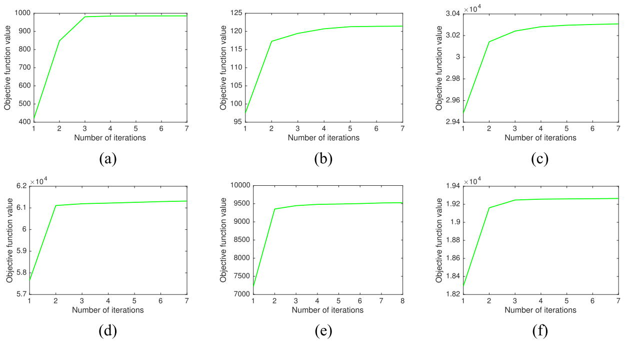
$$-\lambda(1+\sigma)\sum_i\left(\frac{\|u_i^t\|_2^2}{\|u_i^t\|_2+\sigma} - \frac{\|u_i^t\|_2 + 2\sigma}{2(\|u_i^t\|_2+\sigma)^2}\|u_i^t\|_2^2\right) \leq -\lambda(1+\sigma)\sum_i\left(\frac{\|u_i^{t+1}\|_2^2}{\|u_i^{t+1}\|_2+\sigma} - \frac{\|u_i^t\|_2 + 2\sigma}{2(\|u_i^t\|_2+\sigma)^2}\|u_i^{t+1}\|_2^2\right). \qquad (29)$$

Summing Eq.(28) and Eq.(29) in two sides, and combining the results with Eq.(26), we obtain

$$Tr(W_t^T S_t W_t) - \lambda\|X^T W_t - F_t G_t^T\|_\sigma \leq Tr(W_{t+1}^T S_t W_{t+1}) - \lambda\|X^T W_{t+1} - F_{t+1} G_{t+1}^T\|_\sigma. \qquad (30)$$

Since Eq.(15) has an obvious upper bound $Tr(XX^T)$, therefore, the iterative steps in Algorithm 1 will monotonically increase the objective value in Eq.(15) until it converges.

To evaluate the convergence of the objective function of FAKM, we plot the objective function value on each iteration on six datasets. The parameters λ and σ are set to 1, which are the median values of the tuned range of parameters. The results are illustrated in Fig.6. We can observe that FAKM is efficient and converges quickly within 10 iterations on these datasets.



(a) (b) (c)
(d) (e) (f)

FIGURE 6. Convergence analysis of FAKM on 6 datasets. (a) Breast. (b) Glass. (c) Umist. (d) WebKB. (e) TDT2. (f) Yale.

F. PARAMETER SENSITIVITY

FAKM contains two parameters denoted as λ and σ in Eq.(15). To learn how they affect the clustering performance, we constructed an experiment on the parameter sensitivity.

We show the results on six datasets in Fig.7. In this experiment, we set the reduced dimensionality as the median value of its tuned range for each dataset. From the results, we notice that the performance is sensitive to λ on the selected datasets. If λ is properly selected, the performance of FAKM will be significantly improved. Besides, if we select the parameter λ near its optimal value, the performance of FAKM varies within a small range in most cases. This observation is very helpful for us to determine λ. As to the impact of σ, the performance of FAKM varies with different σ on each dataset. For example, the best performance is achieved when σ is smaller than 10^{-4} on Breast and is larger than 10^4 on TDT2 dataset.

G. COMPUTATION TIME COMPARISON

For the computation time investigation, we recorded the clustering time of all the algorithms on each dataset. All algorithms were implemented in Matlab 2015b on an Intel(R) Xeon(R) CPU E5-2630 v4 2.2GHz PC with 16G memory and Ubuntu 16.04 operating system. For the selected datasets, we fixed their reduced dimensionality as c. The computation time (in seconds) comparison results on seven datasets are illustrated in Table 6, where "-" means "out of memory error" while running the experiment. We have the following observations:

- When clustering data with few features, such as Breast, Vehicle and Glass, KM spends the least time. Besides,

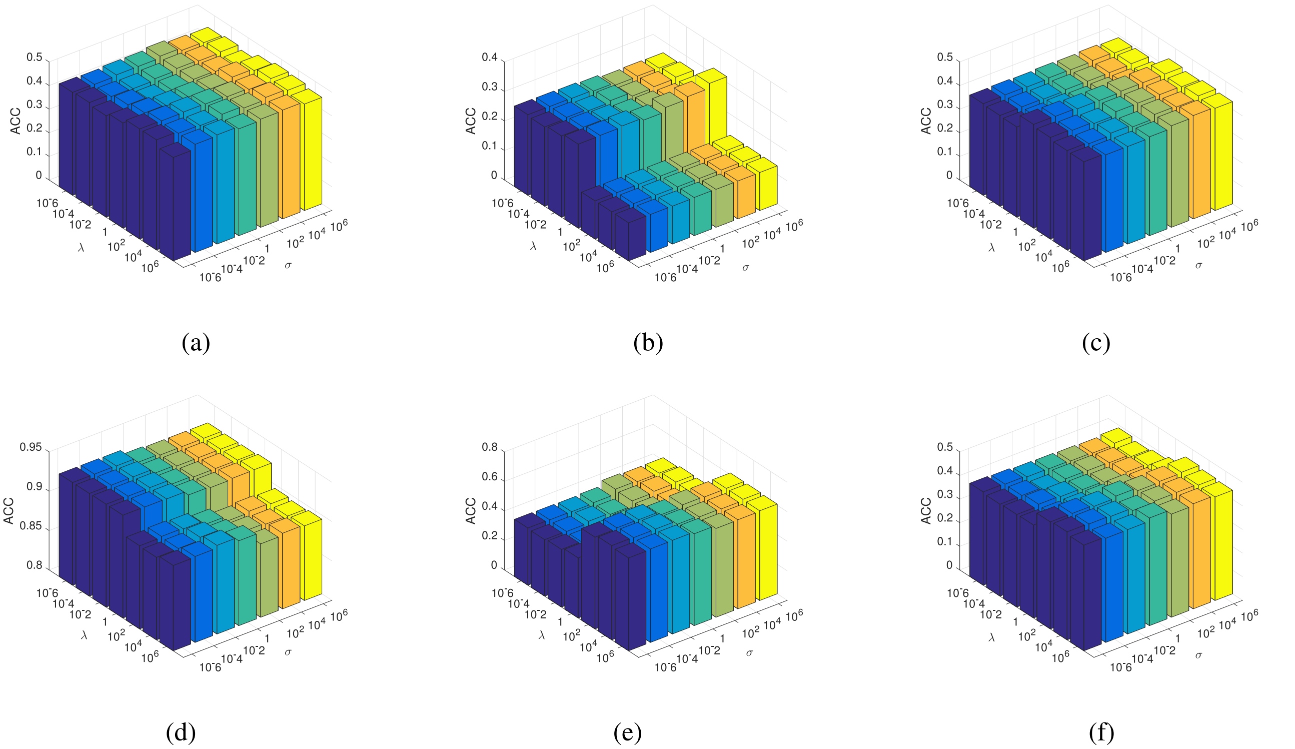


FIGURE 7. Parameter sensitivity analysis of FAKM on 6 datasets. (a) Glass. (b) TDT2. (c) Vehicle. (d) Breast. (e) WebKB. (f) Yale.

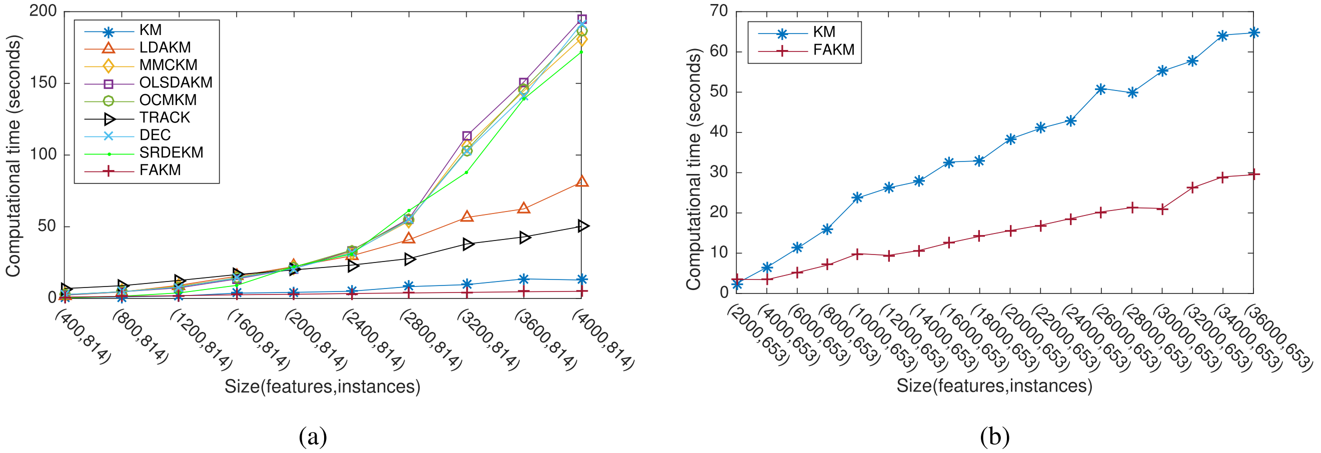


FIGURE 8. Computational time analysis on the synthetic datasets for the algorithms compared. (a) Synthetic dataset on WebKB. (b) Synthetic dataset on TDT2.

the other subspace learning algorithms achieve the comparable computational time on these datasets. Nonetheless, when handling data with a relative larger dimensionality, for example, Umist and Yale, FAKM and KM is able to avoid the intensive computation of eigenvalue decomposition and thus perform more efficiently than the other compared algorithms.

- Although FAKM has the linear complexity with KM with respect to D and n, the time cost is practically different, the elementary reason lies in that their convergence iterations vary on distinct datasets.
- When handling data with extremely high dimensionality, for example, WebKB and TDT2, FAKM is faster than KM. The primary reason is that KM iteratively optimizes cluster indicator labels in the original feature space, while FAKM performs in the reduced feature space, which is quite suitable for extremely high-dimensional data.

The results of computational time with increasing feature sizes on the synthetic datasets are shown in Fig.8. From Fig.8(a), we can see that both FAKM and KM have the linear computation complexity with respect to dimensionality, and are much faster than the other subspace clustering algorithms. For the extremely high-dimensional data, that is, TDT2, we only report the results of KM and FAKM on this dataset, because of the "out of memory error" problem of the other subspace clustering algorithms. From Fig.8(b), we can observe that FAKM has an obvious advancement of saving the computation over KM when the number of features increasing.

V. CONCLUSION

In this paper, we proposed a novel fast adaptive K-means clustering model, namely FAKM, to cope with the challenges of existing KM type subspace clustering algorithms. We jointly integrated KM and feature selection into a single framework according to the basic assumption that actual cluster structure can be well extracted in the embedded feature space. Additionally, to ease the impact of redundant features and outliers, we also imposed an adaptive loss function to elastically calculate the cluster indicator matrix. Furthermore, an efficient alternative optimization algorithm is designed to solve the proposed method, together with theoretical analysis on its convergence and computational complexity. Extensive experiments on several benchmark datasets demonstrated the advantage of FAKM. In the future work, we will extend it to multi-view clustering.

REFERENCES

[1] X. Chang, F. Nie, Z. Ma, Y. Yang, and X. Zhou, "A convex formulation for spectral shrunk clustering," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2532–2538.

[2] X. Song, W. Li, D. Ma, Y. Wu, and D. Ji, "An enhanced clustering-based method for determining time-of-day breakpoints through process optimization," *IEEE Access*, vol. 6, pp. 29241–29253, 2018.

[3] M. Luo, X. Chang, L. Nie, Y. Yang, A. G. Hauptmann, and Q. Zheng, "An adaptive semisupervised feature analysis for video semantic recognition," *IEEE Trans. Cybern.*, vol. 48, no. 2, pp. 648–660, Feb. 2018.

[4] Y. Yang, D. Xu, F. Nie, S. Yan, and Y. Zhuang, "Image clustering using local discriminant models and global integration," *IEEE Trans. Image Process.*, vol. 19, no. 10, pp. 2761–2773, Oct. 2010.

[5] Z. Lv, T. Liu, J. A. Benediktsson, and H. Du, "A novel land cover change detection method based on K-means clustering and adaptive majority voting using bitemporal remote sensing images," *IEEE Access*, to be published.

[6] Y. Yang, H. T. Shen, F. Nie, R. Ji, and X. Zhou, "Nonnegative spectral clustering with discriminative regularization," in *Proc. 25th AAAI Conf. Artif. Intell.*, 2011, pp. 555–560.

[7] C.-G. Li, C. You, and R. Vidal, "Structured sparse subspace clustering: A joint affinity learning and subspace clustering framework," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2988–3001, Jun. 2017.

[8] K. Peng, V. C. M. Leung, and Q. Huang, "Clustering approach based on mini batch kmeans for intrusion detection system over big data," *IEEE Access*, vol. 6, pp. 11897–11906, 2018.

[9] X. Cai, F. Nie, and H. Huang, "Multi-view k-means clustering on big data," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 2598–2604.

[10] L. Du *et al.*, "Robust multiple kernel K-means using L21-norm," in *Proc. 24th Int. Conf. Artif. Intell.*, 2015, pp. 3476–3482.

[11] C. Hou, C. Zhang, Y. Wu, and Y. Jiao, "Stable local dimensionality reduction approaches," *Pattern Recognit.*, vol. 42, no. 9, pp. 2054–2066, 2009.

[12] F. Nie, D. Xu, I. W. Tsang, and C. Zhang, "Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction," *IEEE Trans. Image Process.*, vol. 19, no. 7, pp. 1921–1932, Jul. 2010.

[13] Y. Yang, F. Nie, D. Xu, J. Luo, Y. Zhuang, and Y. Pan, "A multimedia retrieval framework based on semi-supervised ranking and relevance feedback," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 723–742, Apr. 2012.

[14] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

[15] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Hoboken, NJ, USA: Wiley, 2000.

[16] X. Chang, F. Nie, Y. Yang, C. Zhang, and H. Huang, "Convex sparse PCA for unsupervised feature learning," *ACM Trans. Knowl. Discovery Data*, vol. 11, no. 1, Jul. 2016, Art. no. 3.

[17] R. Wang, F. Nie, X. Yang, F. Gao, and M. Yao, "Robust 2DPCA with non-greedy l_1-norm maximization for image analysis," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 1108–1112, May 2015.

[18] C. Hou, F. Nie, Y. Jiao, C. Zhang, and Y. Wu, "Learning a subspace for clustering via pattern shrinking," *Inf. Process. Manage.*, vol. 49, no. 4, pp. 871–883, Jul. 2013.

[19] X. Yin, S. Chen, and E. Hu, "Regularized soft K-means for discriminant analysis," *Neurocomputing*, vol. 103, pp. 29–42, Mar. 2013.

[20] C. Ding and T. Li, "Adaptive dimension reduction using discriminant analysis and k-means clustering," in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, New York, NY, USA, 2007, pp. 521–528.

[21] H. Li, T. Jiang, and K. Zhang, "Efficient and robust feature extraction by maximum margin criterion," *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 157–165, Jan. 2006.

[22] H. Park, M. Jeon, and J. B. Rosen, "Lower dimensional representation of text data based on centroids and least squares," *BIT Numer. Math.*, vol. 43, no. 2, pp. 427–448, 2003.

[23] F. Nie, S. Xiang, Y. Liu, C. Hou, and C. Zhang, "Orthogonal vs. uncorrelated least squares discriminant analysis for feature extraction," *Pattern Recognit. Lett.*, vol. 33, no. 5, pp. 485–491, Apr. 2012.

[24] C. Hou, F. Nie, D. Yi, and D. Tao, "Discriminative embedded clustering: A framework for grouping high-dimensional data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 6, pp. 1287–1299, Jun. 2015.

[25] J. Xu, J. Han, F. Nie, and X. Li, "Re-weighted discriminatively embedded K-means for multi-view clustering," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 3016–3027, Jun. 2017.

[26] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint l_2,1-norms minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1813–1821.

[27] R. Chen, N. Sun, X. Chen, M. Yang, and Q. Wu, "Supervised feature selection with a stratified feature weighting method," *IEEE Access*, vol. 6, pp. 15087–15098, 2018.

[28] C. Boutsidis, P. Drineas, and M. W. Mahoney, "Unsupervised feature selection for the k-means clustering problem," in *Proc. 22nd Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 153–161. [Online]. Available: http://dl.acm.org/citation.cfm?id=2984093.2984111

[29] C. Boutsidis and M. Magdon-Ismail, "Deterministic feature selection for K-means clustering," *IEEE Trans. Inf. Theory*, vol. 59, no. 9, pp. 6099–6110, Sep. 2013.

[30] D. Wang, F. Nie, and H. Huang, *Unsupervised Feature Selection via Unified Trace Ratio Formulation and K-means Clustering (TRACK)*. Berlin, Germany: Springer, 2014, pp. 306–321.

[31] F. Nie, H. Wang, H. Huang, and C. Ding, "Adaptive loss minimization for semi-supervised elastic embedding," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2013, pp. 1565–1571.

[32] Y. Liu, Y. Guo, H. Wang, F. Nie, and H. Huang, "Semi-supervised classifications via elastic and robust embedding," in *Proc. 31st AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, , Feb. 2017, pp. 2294–2300.

[33] M. Luo, F. Nie, X. Chang, Y. Yang, A. G. Hauptmann, and Q. Zheng, "Adaptive unsupervised feature selection with structure regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 944–956, Apr. 2018.

[34] F. Nie, W. Zhu, and X. Li, "Unsupervised feature selection with structured graph optimization," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 1302–1308. [Online]. Available: http://dl.acm.org/citation.cfm?id=3015812.3016004

[35] 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.

[36] X. Chang, F. Nie, Y. Yang, and H. Huang, "A convex formulation for semi-supervised multi-label feature selection," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1171–1177.

[37] M. Qian and C. Zhai, "Joint adaptive loss and l_2/l_0-norm minimization for unsupervised feature selection," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–8.

[38] Y. Yang, Z. Ma, A. G. Hauptmann, and N. Sebe, "Feature selection for multimedia analysis by sharing information among multiple tasks," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 661–669, Apr. 2013.

[39] X.-D. Wang, R.-C. Chen, F. Yan, and Z.-Q. Zeng, "Semi-supervised feature selection with exploiting shared information among multiple tasks," *J. Vis. Commun. Image Represent.*, vol. 41, pp. 272–280, Nov. 2016.

[40] Z. Li, Y. Yang, J. Liu, X. Zhou, and H. Lu, "Unsupervised feature selection using nonnegative spectral analysis," in *Proc. 26th AAAI Conf. Artif. Intell.*, 2012, pp. 1026–1032.

[41] X. He and P. Niyogi, "Locality preserving projections," in *Proc. Adv. Neural Inf. Process. Syst.*, S. Thrun, L. K. Saul, and P. B. Schölkopf, Eds. Cambridge, MA, USA: MIT Press, 2004, pp. 153–160.

[42] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 12, pp. 1624–1637, Dec. 2005.

[43] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Mar. 2003.

XIAO-DONG WANG received the B.E. and M.E. degrees from the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, in 2007 and 2010, respectively, and the Ph.D. degree from the Department of Information Management, Chaoyang University of Technology, Taichung, Taiwan, in 2019. He is currently an Associate Professor with the College of Computer and Information Engineering, Xiamen University of Technology. His current research interests include pattern recognition, image processing, and embedded system structure.

RUNG-CHING CHEN received the B.S. degree from the Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, in 1987, the M.S. degree from the Institute of Computer Engineering, National Taiwan University of Science and Technology, in 1990, and the Ph.D. degree from the Department of Applied Mathematics in Computer Science Sessions, National Chung Tsing University, in 1998. He is currently a Professor with the Department of Information Management, Taichung, Taiwan. His research interests include networks technology, domain ontology, pattern recognition and knowledge engineering, the IoT and data analysis, and the applications of artificial intelligent.

FEI YAN received the B.E. degree from the College of Technology, Hunan Normal University, Changsha, China, in 2007, and the M.E. degree from the College of Computer Science and Electronic Engineering, Hunan University, China, in 2010. She is currently a Lab Master with the College of Computer and Information Engineering, Xiamen University of Technology. Her research interests include patter recognition and data hiding.

ZHI-QIANG ZENG received the bachelor's degree in automation from Sichuan University, Chengdu, China, in 1994, the M.Sc. degree in computer science from Xi'an Jiaotong University, China, in 2004, and the Ph.D. degree in computer science from Zhejiang University, China, in 2007. In 2008, he joined the Computer Science Department, Xiamen University of Technology, as a Research Associate. His interests include pattern recognition and machine learning, in particular, support vector machines and general kernel methods.

CHAO-QUN HONG received the Ph.D. degree from Zhejiang University, Zhejiang, China, in 2011. He is currently an Associate Professor with the Department of Computer and Information Engineering, Xiamen University of Technology, Xiamen, China. He has authored or co-authored more than 30 scientific articles. His research interests include high-performance computing, image processing, computer vision, and pattern recognition.