Efficient Multi-View K-Means Clustering With Multiple Anchor Graphs

Ben Yang, Xuetao Zhang, Zhongheng Li, Feiping Nie, *Senior Member, IEEE*, and Fei Wang

Abstract—Multi-view clustering has attracted a lot of attention due to its ability to integrate information from distinct views, but how to improve efficiency is still a hot research topic. Anchor graph-based methods and k-means-based methods are two current popular efficient methods, however, both have limitations. Clustering on the derived anchor graph takes a while for anchor graph-based methods, and the efficiency of k-means-based methods drops significantly when the data dimension is large. To emphasize these issues, we developed an efficient multi-view k-means clustering method with multiple anchor graphs (EMKMC). It first constructs anchor graphs for each view and then integrates these anchor graphs using an improved k-means strategy to obtain sample categories without any extra post-processing. Since EMKMC combines the high-efficiency portions of anchor graph-based methods and k-means-based methods, its efficiency is substantially higher than current fast methods, especially when dealing with large-scale high-dimensional multi-view data. Extensive experiments demonstrate that, compared to other state-of-the-art methods, EMKMC can boost clustering efficiency by several to thousands of times while maintaining comparable or even exceeding clustering effectiveness.

Index Terms—Multi-view clustering, k-means, anchor graph, orthogonality

1 Introduction

Multi-view clustering [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14] has grown in popularity due to its exceptional capability to successfully integrate diverse descriptions of an object collected from multiple views or channels. It, as opposed to single-view clustering, can provide more extensive object information and more reasonable clustering performance in practice. However, the introduction of multi-view information brings a large amount of data, which poses a major challenge to the efficiency of conventional multi-view clustering methods, especially when facing large-scale and high-dimensional multi-view data. Currently, one of the hottest topics is how to efficiently cluster these multi-view data.

Many efficient algorithms [15], [16], [17], [18], [19], [20], [21] have been proposed to increase the efficiency of multi-view clustering. The majority of them fall into two categories: anchor graph-based methods [16], [18], [19], [20] and k-means-based methods [21]. Among them, anchor graph-based methods are inspired by spectral clustering (SC). The difference is that SC constructs a full sample graph with a scale of $n \times n$ (where n is the number of samples) between every two samples, whereas anchor graph-based methods produce an anchor graph with a scale of $n \times m$ between samples and their anchors (where m is the number of anchors). In general, m is substantially lower than n, which enables the data scale to be significantly decreased and the clustering efficiency to be greatly increased. For example, to construct an anchor graph between the original data and its anchors, ECCA [22] first generates m anchors by using k-means on the original data and regards the generated centroids as anchors, then constructs a sparse bipartite graph between the original data and the generated anchors. For k-means-based methods, they are equivalent to matrix factorization-based methods [23], [24], [25], inheriting the high clustering efficiency of single-view k-means algorithms, and have been used in numerous efficient multi-view clustering applications.

Though both the anchor graph-based and k-means-based methods have increased the efficiency of multi-view clustering to some degree, their improvements are limited. For instance, anchor graph-based algorithms are still time-consuming in the clustering step when using some standard clustering strategies such as spectral analysis, co-training, subspace learning, etc. K-means-based methods have limited applicability to high-dimensional data since they operate on original multi-view data directly, and so on. To further increase the efficiency of multi-view clustering, in our previous work, we developed a fast multi-view clustering algorithm (FMCNOF) [17] that inherits the benefits of the two methods mentioned above and has been demonstrated to be highly efficient. Nonetheless, the effectiveness of FMCNOF has to be enhanced because it requires all views to provide the same anchors and overlooks the hidden structures of different

- *Ben Yang, Xuetao Zhang, Zhongheng Li, and Fei Wang are with the Institute of Artificial Intelligence and Robotics, College of Artificial Intelligence, Xi'an Jiaotong University, Xi'an 710049, China. E-mail: yangben19940214@stu.xjtu.edu.cn, {xuetaozh, wfx}@xjtu.edu.cn, lizhongheng2010@gmail.com.*
- *Feiping Nie is with the School of Artificial Intelligence, Optics and Electronics (iOPEN), Key Laboratory of Intelligent Interaction and Applications (Ministry of Industry and Information Technology), Northwestern Polytechnical University, Xi'an 710072, China. E-mail: feipingnie@gmail.com.*

Manuscript received 1 June 2021; revised 20 March 2022; accepted 19 June 2022. Date of publication 23 June 2022; date of current version 5 June 2023. The authors would like to thank the National Natural Science Foundation of China under Grant 62088102, in part by the Key Project of Shaanxi Province under Grant 2018ZDCXLGY0607, in part by the Fundamental Research Funds for the Central Universities under Grant xzy022021044, and in part by the China Scholarship Council under Grant 202006280340. (Corresponding author: Feiping Nie.) Recommended for acceptance by Q. He. This article has supplementary downloadable material available at https://doi.org/10.1109/TKDE.2022.3185683, provided by the authors. Digital Object Identifier no. 10.1109/TKDE.2022.3185683

views. As a continuation of FMCNOF, the focus of this study is on how to increase the effectiveness of clustering while guaranteeing high efficiency.

Based on FMCNOF, we present a novel fast multi-view clustering method named efficient multi-view k-means clustering with multiple anchor graphs (EMKMC) in this work to improve the effectiveness of clustering while preserving high efficiency. It first generates different anchors for diverse views to investigate the underlying structure of each view, then constructs anchor graphs for each view separately, and finally incorporates these anchor graphs under the k-means framework to learn the consistent clustering indicator matrix. EMKMC, similar to FMCNOF, can achieve efficiency far exceeding that of the anchor graph-based and k-means-based methods. Compared to FMCNOF, EMKMC can enhance clustering effectiveness while maintaining comparable efficiency because it treats different views individually when generating anchors and introduces a free parameter to control the proportion of each view in the unified representation. To solve the EMKMC model, we designed two optimization algorithms: a Lagrange Multiplier Method (LMM)-based algorithm (EMKMC-L) and a fast optimization-based algorithm (EMKMC-F). Since EMKMC-L requires more steps to reach convergence and more matrix multiplications in each iteration, EMKMC-F is more widely employed in practice due to its higher efficiency. Extensive experiments indicate that by combining the EMKMC model with the EMKMC-F algorithm, compared to other anchor graph-based and k-means-based baselines, our method can achieve several times, even thousands of times, efficiency while assuring comparable or even better clustering effectiveness. Compared to FMCNOF, our method can improve the clustering effectiveness greatly while maintaining comparable clustering efficiency. The following are the primary contributions of this paper:

1) To accelerate multi-view clustering, an efficient multi-view k-means clustering model (EMKMC) is developed, which can efficiently cope with large-scale and high-dimensional multi-view data and achieve reasonable clustering effectiveness by exploring the underlying structure of distinct views.
2) Two optimization algorithms, EMKMC-L and EMKMC-F, are developed to solve the EMKMC model. In practice, EMKMC-F has a higher clustering efficiency than EMKMC-L since it requires fewer iterations and fewer matrix multiplications.
3) Extensive experiments on 9 real-world datasets, including large-scale and high-dimensional datasets, show that EMKMC can maintain comparable or even better effectiveness while increasing efficiency by several times to thousands of times when compared to other anchor graph-based and k-means-based baselines. When compared to FMCNOF, EMKMC can keep comparable clustering efficiency while giving superior clustering effectiveness.

The remaining parts of this paper are organized as follows: Section 2 introduces some related works, while Section 3 introduces some preliminaries. The model of EMKMC is proposed in Section 4. Subsequently, two optimization algorithms, EMKMC-L and EMKMC-F, and their complexity analysis are given in Section 5. Furthermore, to verify the performance of EMKMC and the proposed optimization algorithms, extensive experiments and analysis are given in Section 6. Finally, Section 7 is the conclusion, and the detailed derivation and complexity analysis of EMKMC-L are given in Appendix A and Appendix B, respectively, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2022.3185683. There are many acronyms in this paper, so we summarized them in Table 1.

2 Related Works

Since conventional multi-view clustering has a high computational complexity, numerous accelerated methods [15], [16], [17], [18], [19], [20], [21] has been developed. These accelerated algorithms are classified into two types: anchor graph-based methods [16], [18], [19], [20] and k-means-based methods [21], where k-means-based methods have been demonstrated to be equivalent to matrix factorization-based methods. In this section, we will discuss some of the most popular multi-view clustering methods.

Anchor graph-based methods are born out of SC, which first construct an anchor graph between samples and their anchors rather than the conventional full sample graph, and then apply various clustering strategies to the acquired anchor graph to produce clusters of samples. For instance, co-training is used on obtained anchor graphs to maximize the consistency between different views in GCTMC [26], MVSCB [27] performs general multi-view SC on the Laplacian matrix of anchor graphs directly, LMVSC [28] performs subspace clustering on the obtained anchor graphs to improve the efficiency, and SC and spectral rotation are used in FMVPG [18] to get sample categories, and so on. All of these anchor graph-based methods have been shown to significantly enhance the efficiency of multi-view clustering, but they only reduce the computational complexity in the step of constructing graphs, and the second step is still time-consuming.

The k-means-based methods, which are equal to matrix factorization-based methods, are another type of efficient multi-view clustering algorithm. These algorithms inherit the efficiency of standard k-means and can extract the category structure directly from the original data without any preprocessing. To incorporate the differences between different views, RMKMC [29] relaxes the multi-view k-means problem to the matrix factorization problem, while FMVBKM [21] directly constrains the factor matrix to a discrete matrix indicator to avoid extra k-means operations, and so on. Though these k-means-based algorithms are reasonably efficient in most circumstances since they operate on the original data directly, when the data dimension is large, the complexity of these methods increases rapidly.

The deep learning (DL)-based methods, another state-of-the-art method, improve effectiveness by using neural networks to explore the consensus information between diverse views. For example, [30] proposed a deep adversarial multi-view clustering (DAMC) network to learn the internal structure embedded in multi-view data, [31] proposed a multi-view attribute graph convolutional network

TABLE 1
Acronyms and Their Meanings

Acronyms	Meanings
EMKMC	Efficient multi-view k-means clustering with multiple anchor graphs
SC	Spectral clustering
FMCNOF	Fast multi-view clustering via nonnegative and orthogonal factorization
LMM	Lagrange multiple method
EMKMC-L	LMM-based algorithm for solving EMKMC
EMKMC-F	Fast algorithm for solving EMKMC
GCTMC	Guided co-training for large-scale multi-view spectral clustering
MVSCB	Large-scale multi-view spectral clustering via bipartite graph
LMVSC	Large-scale multi-view subspace clustering
FMVPG	Fast multi-view clustering via prototype graph
RMKMC	Multi-view k-means clustering on Big Data
FMVBKM	Fast multi-view bipartite k-means
DL	Deep learning
DAMC	Deep adversarial multi-view clustering
MAGCN	Multi-view attribute graph convolutional network
ANN	Agglomerative neural network
NOF	Nonnegative and orthogonal factorization
KNN	K-nearest neighbor
NMF	Nonnegative matrix factorization
SVD	Singular value decomposition
MultiNMF	Multi-view clustering via joint nonnegative matrix factorization
RRMA	Multiview clustering based on robust and regularized matrix approximation
AMGL	Parameter-free auto-weighted multiple graph learning
NMFCC	Nonnegative matrix factorization with co-orthogonal constraints
ACC	Clustering accuracy
NMI	Normalized mutual information

(MAGCN) model to map embedded features and learn consensus information by a two-pathway encoding strategy. And [32] employs a constrained Laplacian rank-based agglomerative neural network (ANN) to approximate the optimal consensus view. These algorithms can show satisfactory effectiveness on a variety of datasets, but they all emphasize effectiveness over efficiency. For instance, the computational complexity of ANN is $O(n^3)$. If more focus is given to efficiency, the use of these methods will be limited.

To overcome the drawbacks of the aforementioned methods, we previously developed FMCNOF, a fast multi-view clustering algorithm. FMCNOF combines the high efficiency of anchor graph-based techniques with k-means-based algorithms by applying nonnegative and orthogonal factorization (NOF) on the derived consensus anchor graph. Unfortunately, though FMCNOF can achieve hundreds to thousands of times the efficiency of other efficient methods, its clustering effectiveness is limited since it overlooks distinctions between various views while generating anchors. To address this issue, in this study, we developed EMKMC, an efficient multi-view k-means clustering algorithm that significantly enhances clustering effectiveness while keeping efficiency similar to FMCNOF.

3 Preliminaries

K-means has been widely employed in various clustering tasks as an efficient clustering method, and numerous improved k-means methods have been developed and demonstrated with considerable efficiency. To deal with large-scale multi-view data, [29] proposed a robust k-means-based multi-view clustering method named RMKMC.

RMKMC solves the multi-view k-means problem by converting it to a relaxed matrix factorization problem and directly decomposing the multi-view data into the product of two low-dimensional factor matrices, substantially reducing computational complexity by reducing the data scale. The following is the definition of RMKMC's objective function:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{H}_v,\boldsymbol{\alpha}} \sum_{v=1}^{V} \alpha_v^{\gamma} \left\| \mathbf{X}_v - \mathbf{W}\mathbf{H}_v^{\top} \right\|_{2,1}^2 \\ &s.t.\ \mathbf{W}_{ik} \in \{0,1\}, \sum_{k=1}^{c} \mathbf{W}_{ik} = 1, \sum_{v=1}^{V} \boldsymbol{\alpha}_v = 1, \end{aligned} \tag{1}$$

where $\mathbf{X}_v$, $\mathbf{H}_v$, and $\boldsymbol{\alpha}_v$ are the vth view's data, center matrix, and weight, respectively. $\mathbf{W}$ is the discrete cluster indicator matrix. The high efficiency of k-means is carried over to RMKMC, and the introduction of the $L_{2,1}$-norm improves its robustness to noise. Experiments show that, when compared to other efficient multi-view clustering methods, RMKMC can handle large-scale multi-view data in a short amount of time.

Although RMKMC can enhance the efficiency of multi-view clustering, it still has the following issues: 1) When the data dimension is large, its efficiency decreases dramatically; 2) The optimization step is time-consuming due to the intense matrix multiplications and the introduction of the $L_{2,1}$-norm. As a consequence, it is critical to produce an efficient novel multi-view clustering method that is less affected by data dimensional changes and requires fewer matrix multiplications during optimization.

4 METHODOLOGY

In this section, we elaborate on an efficient multi-view k-means clustering (EMKMC) model inspired by RMKMC. In contrast to RMKMC, EMKMC 1) constrains the cluster indicator matrix to an orthogonal matrix to reduce the freedom of factorization, 2) uses F-norm rather than $L_{2,1}$-norm to simplify the optimization, 3) constructs anchor graphs for all views to improve the efficiency of dealing with high-dimensional data, and it can also explore the hidden structure of different views. EMKMC is divided into two sections: anchor graphs construction and clustering with multiple anchor graphs. Furthermore, the following specifics are provided:

4.1 Anchor Graphs Construction

To reap the benefits of anchor graph-based methods, anchor graphs are adopted into the EMKMC model rather than the original data. It can increase the efficiency of conventional multi-view k-means clustering, particularly in large-scale and high-dimensional clustering problems.

Anchors must be generated first before anchor graphs can be constructed. Because of their low cost and high performance, two methods are commonly used to generate anchors: random sampling and k-means. Following FMCNOF, k-means is utilized in this work to generate all views' anchors since it can provide better clustering effectiveness than random sampling with the same number of anchors. By performing k-means on the original data, we can divide them into some classes, and all centroids of these classes were collected to form anchors. To avoid the information loss in FMCNOF, which requires connecting all views before generating anchors, EMKMC generates anchors for different views independently to maximize the exploitation of each view's hidden structure. It will be able to get higher-quality anchors.

Assume $\mathcal{A} = [\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_V]^\top$ is the set of anchors corresponding to the original data $\mathcal{X} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_V]^\top$ and $\mathcal{Z} = [\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_V]^\top$ is the set of anchor graphs between $\mathcal{X}$ and $\mathcal{A}$. The anchor graph can be constructed in a variety of ways, the majority of which are based on the distance between two samples. For example, when the distance between $\mathbf{x}_i$ and $\mathbf{a}_j$ is less than a certain threshold, $z_{ij} = 1$; otherwise, $z_{ij} = 0$; or the euclidean distance is directly adopted as z_{ij} in the similarity matrix, and so on. In contrast, a normalized k-nearest neighbor (KNN) anchor graph construction method based on k-nearest neighbors is provided in [33], [34], [35]. Following [33], [34], [35], $z_{ij}^{(v)}$ can be obtained using the formula

$$z_{ij}^{(v)} = \frac{\acute{d}\left(\mathbf{x}_i^{(v)}, \mathbf{a}_{k+1}^{(v)}\right) - \acute{d}\left(\mathbf{x}_i^{(v)}, \mathbf{a}_j^{(v)}\right)}{\sum_{\hat{j}=1}^{k}\left[\acute{d}\left(\mathbf{x}_i^{(v)}, \mathbf{a}_{k+1}^{(v)}\right) - \acute{d}\left(\mathbf{x}_i^{(v)}, \mathbf{a}_{\hat{j}}^{(v)}\right)\right]}, \tag{2}$$

where $\acute{d}(\mathbf{x}_i^{(v)}, \mathbf{a}_{k+1}^{(v)}) = \varphi[d(\mathbf{x}_i^{(v)}, \mathbf{a}_{k+1}^{(v)})]$, and $d(x, a) = ||x - a||_2^2$. $\varphi[d(\mathbf{x}_i^{(v)}, \mathbf{a}_{k+1}^{(v)})]$ means the $k+1$th element after sorting $\{d(\mathbf{x}_i^{(v)}, \mathbf{a}_1^{(v)}), d(\mathbf{x}_i^{(v)}, \mathbf{a}_2^{(v)}), \ldots, d(\mathbf{x}_i^{(v)}, \mathbf{a}_{m_V}^{(v)})\}$ in ascending order. The property of $\sum_{j=1}^{m_v} z_{ij}^{(v)} = 1$ can provide a more reasonable performance in subsequent clustering steps.

4.2 Clustering With Multiple Anchor Graphs

Since the orthogonal constraint limits the flexibility of factorization of nonnegative matrix factorization (NMF), it considerably improves clustering effectiveness [25]. Furthermore, when the F-norm and an orthogonal constraint are combined, the model can be solved quickly by employing some advanced optimization strategies. Based on these findings, NOF is a superior clustering approach, and one-sided NOF has been certified to be equal to relaxed k-means clustering. As a consequence, in this work, the k-means clustering problem is transformed into a one-sided NOF problem to produce an efficient clustering algorithm with comparable or even better clustering effectiveness compared to other state-of-the-art algorithms.

Algorithm 1. Algorithm of the EMKMC-L

Input: Multi-view data $\mathcal{X} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_V]^\top$, the set of anchor numbers $M = [m_1, m_2, \ldots, m_V]^\top$, the number of clusters c, and a free parameter $1 < \gamma < 2$.
Output: Cluster indicator matrix $\mathbf{W}$.
1: Generating anchors $\mathcal{A} = [\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_V]^\top$ for all views by k-means, and then constructing anchor graphs $\mathcal{Z} = [\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_V]^\top$ by Eq. (2).
2: **Initialize** α_v is initialized to $1/V$, $\mathbf{W}$ is initialized to a random orthogonal matrix, and $\mathbf{H}_v$ is initialized to a random positive matrix.
3: **while** not converge **do**
4: Update $\mathbf{W}$ by Eq. (28).
5: Update $\mathbf{H}_v$ by Eq. (32).
6: Update α by Eq. (22).
7: **end while**
8: Output $\mathbf{W}$.

Unlike our previous work, FMCNOF, EMKMC produces different anchors for different views individually and employs the NOF technique in each view to fully exploit the hidden structure contained in each. Thus, more effective clustering performance can be achieved while maintaining high efficiency. The EMKMC model can be represented as follows:

$$\begin{aligned} \min_{\mathbf{1}^\top \boldsymbol{\alpha}=1, \boldsymbol{\alpha}\geq 0} & \sum_{v=1}^{V} \alpha_v^\gamma \left\|\mathbf{Z}_v - \mathbf{W}\mathbf{H}_v^\top\right\|_{\mathrm{F}}^2 \\ & s.t.\ \mathbf{W}^\top\mathbf{W} = \mathbf{I}, \mathbf{H}_v \geq 0, \end{aligned} \tag{3}$$

where α_v is the weight of the vth view and $1 < \gamma < 2$ is a free parameter to control the weight vector distribution. We can learn from Eq. (3) that equivalent weight factors can be obtained when $\gamma \to \infty$. When $\gamma \to 1$, 1 will be assigned to the weight factor of the view with the smallest $\mathbf{H}_v$ and 0 to the weights of the other views. Using such a strategy, we can not only avoid the trivial solution to the weight distribution of the different views but also control the weights of important views that are greater and the weights of unimportant views that are smaller to ensure improved clustering effectiveness.

5 OPTIMIZATION

In this section, we develop two optimization algorithms to solve the EMKMC, one LMM-based named EMKMC-L and one fast named EMKMC-F. Furthermore, the computational

complexity of both algorithms is investigated thoroughly. Unlike EMKMC-L, EMKMC-F requires fewer steps to convergence and contains fewer matrix multiplications in each step, which can considerably enhance clustering efficiency.

5.1 An LMM-Based Optimization Algorithm to Solve the EMKMC (EMKMC-L)

In this subsection, an LMM-based iterative algorithm called EMKMC-L is proposed to solve EMKMC, as is the case with most matrix factorization problems. The specific steps of EMKMC-L are given in Algorithm 1, and its comprehensive derivation and complexity analysis are described in Appendix A and Appendix B, respectively, available in the online supplemental material.

Despite its linear computational complexity and ability to reach convergence in a few steps, EMKMC-L has limitations in some high-efficiency required cases. To solve the EMKMC model efficiently, a more efficient optimizing algorithm will be given in the next subsection.

5.2 A Fast Optimization Algorithm to Solve the EMKMC (EMKMC-F)

In this subsection, we develop a fast iterative algorithm called EMKMC-F to solve the EMKMC more quickly. Unlike EMKMC-L, EMKMC-F combines trace properties and orthogonality to simplify the optimization to three decoupling subproblems only with a few matrix multiplications. Furthermore, EMKMC-F can achieve convergence in fewer steps than EMKMC-L. EMKMC-F updates $\mathbf{W}$, $\mathbf{H}_v$, and α independently and keeps other variables fixed when one variable is updated. After the objective function convergence, the cluster indicator $\mathbf{W}$ can be used to identify the category of each sample.

5.2.1 Update $\mathbf{W}$

With $\mathbf{H}_v$ and α fixed, the objective function Eq. (3) can be expressed as follows:

$$\min_{\mathbf{W}^\top\mathbf{W}=\mathbf{I}} \sum_{v=1}^{V} \alpha_v^\gamma \left\|\mathbf{Z}_v - \mathbf{W}\mathbf{H}_v^\top\right\|_{\mathrm{F}}^2. \tag{4}$$

According to the property of F-norm, the above function Eq. (4) can be expanded into the following form:

$$\begin{aligned}\min_{\mathbf{W}^\top\mathbf{W}=\mathbf{I}} \sum_{v=1}^{V} \alpha_v^\gamma \big[&Tr\big(\mathbf{Z}_v^\top\mathbf{Z}_v\big) - 2Tr\big(\mathbf{Z}_v^\top\mathbf{W}\mathbf{H}_v^\top\big)\\ &+Tr\big(\mathbf{H}_v\mathbf{W}^\top\mathbf{W}\mathbf{H}_v^\top\big)\big],\end{aligned} \tag{5}$$

where $Tr(\mathbf{Z}_v^\top\mathbf{Z}_v)$ is a constant that has nothing to do with $\mathbf{W}$ and $Tr(\mathbf{H}_v\mathbf{W}^\top\mathbf{W}\mathbf{H}_v^\top)$ can be converted into a constant that has nothing to do with $\mathbf{W}$ by applying the constraint $\mathbf{W}^\top\mathbf{W}=\mathbf{I}$. Thus, Eq. (5) can be transformed into

$$\begin{aligned}&\min_{\mathbf{W}^\top\mathbf{W}=\mathbf{I}} -\sum_{v=1}^{V} 2\alpha_v^\gamma Tr\Big[\mathbf{W}(\mathbf{Z}_v\mathbf{H}_v)^\top\Big]\\ \Leftrightarrow &\max_{\mathbf{W}^\top\mathbf{W}=\mathbf{I}} \sum_{v=1}^{V} \alpha_v^\gamma Tr\Big[\mathbf{W}(\mathbf{Z}_v\mathbf{H}_v)^\top\Big]\\ \Leftrightarrow &\max_{\mathbf{W}^\top\mathbf{W}=\mathbf{I}} Tr\left[\mathbf{W}\sum_{v=1}^{V} \alpha_v^\gamma(\mathbf{Z}_v\mathbf{H}_v)^\top\right].\end{aligned} \tag{6}$$

Following [17], Eq. (6) can be quickly solved by Theorem 1. First, Singular Value Decomposition (SVD) is applied on $\sum_{v=1}^{V} \alpha_v^\gamma(\mathbf{Z}_v\mathbf{H}_v)^\top$ using the decomposition formulation

$$\left[\mathbf{\Delta}, \mathbf{\Pi}, \mathbf{\Lambda}^\top\right] = svd\left[\sum_{v=1}^{V} \alpha_v^\gamma(\mathbf{Z}_v\mathbf{H}_v)^\top\right]. \tag{7}$$

The sub-problem of solving $\mathbf{W}$ can be further converted into a smaller-scale matrix multiplication problem via SVD, and the value of $\mathbf{W}$ can be determined by a one-step multiplication.

The update rule of $\mathbf{W}$ is as follows:

$$\mathbf{W} = \mathbf{\Lambda}[\mathbf{I}, \mathbf{0}]\mathbf{\Delta}^\top, \tag{8}$$

where $\mathbf{\Lambda}$ and $\mathbf{\Delta}$ are the right and left singular matrices of SVD in Eq. (7).

Theorem 1. *For the following optimization problem:*

$$\max_{\mathbf{P}^\top\mathbf{P}=\mathbf{I}} Tr(\mathbf{P}\mathbf{Q}). \tag{9}$$

Assume that $\mathbf{Q}$'s SVD is $\mathbf{\Phi}$, $\mathbf{\Theta}$, and $\mathbf{\Psi}$. That is $[\mathbf{\Phi}, \mathbf{\Theta}, \mathbf{\Psi}^\top] = svd(\mathbf{Q})$. When the objective function Eq. (9) is maximized, $\mathbf{P}$ can be represented as

$$\mathbf{P} = \mathbf{\Psi}[\mathbf{I}, \mathbf{0}]\mathbf{\Phi}^\top. \tag{10}$$

Proof. Since the whole SVD of $\mathbf{Q}$ is given by $\mathbf{Q} = \Phi\Theta\Psi^\top$, the following equation holds:

$$\begin{aligned}Tr(\mathbf{P}\mathbf{Q}) &= Tr\big(\mathbf{P}\Phi\Theta\Psi^\top\big) = Tr\big(\Theta\Psi^\top\mathbf{P}\Phi\big)\\ &= Tr(\Theta\Omega) = \sum_i \theta_{ii}\delta_{ii},\end{aligned} \tag{11}$$

where $\mathbf{\Omega} = \mathbf{\Psi}^\top\mathbf{P}\Phi$, θ_{ii} and δ_{ii} are the (i,i)th elements of $\mathbf{\Theta}$ and $\mathbf{\Omega}$, respectively.

Thus, it is simply to verify $\Omega\Omega^\top = \mathbf{I}$. And we can derive

$$Tr(\mathbf{P}\mathbf{Q}) = \sum_i \theta_{ii}\delta_{ii} \le \sum_i \theta_{ii}. \tag{12}$$

Hence, $Tr(\mathbf{P}\mathbf{Q})$ reaches its maximum with $\mathbf{\Omega} = [\mathbf{I}, \mathbf{0}]$, and Eq. (10) is the solution of the objective function Eq. (9). □

5.2.2 Update $\mathbf{H}_v$

With $\mathbf{W}$ and α fixed, Eq. (3) can be transformed into the following form:

$$\min_{\mathbf{H}_v\ge 0} \left\|\mathbf{Z}_v - \mathbf{W}\mathbf{H}_v^\top\right\|_{\mathrm{F}}^2. \tag{13}$$

It should be emphasized that $\mathbf{H}_v$ is the vth view's factor matrix, and each $\mathbf{H}_v$ is independent of the others. $\mathbf{H}_v$ can be solved independently without respect for other views, and this technique can make full use of the hidden structures contained in distinct views. Expanding Eq. (13), it can be expressed as follows:

$$\min_{\mathbf{H}_v \geq 0} Tr\Big[\big(\mathbf{Z}_v - \mathbf{W}\mathbf{H}_v^\top\big)^\top \big(\mathbf{Z}_v - \mathbf{W}\mathbf{H}_v^\top\big)\Big]$$
$$= \min_{\mathbf{H}_v \geq 0} Tr\big(\mathbf{Z}_v^\top \mathbf{Z}_v\big) - 2Tr\big(\mathbf{H}_v \mathbf{W}^\top \mathbf{Z}_v\big) + Tr\big(\mathbf{H}_v \mathbf{W}^\top \mathbf{W}\mathbf{H}_v^\top\big). \quad (14)$$

Since $\mathbf{W}^\top\mathbf{W} = \mathbf{I}$, and $\mathbf{Z}_v^\top\mathbf{Z}_v$ is constant with respect to $\mathbf{H}_v$, Eq. (14) can be transferred as follows:

$$\min_{\mathbf{H}_v \geq 0} Tr\big(\mathbf{H}_v^\top \mathbf{H}_v\big) - 2Tr\big(\mathbf{W}^\top \mathbf{Z}_v \mathbf{H}_v\big). \quad (15)$$

By adding a constant $Tr(\mathbf{W}^\top\mathbf{Z}_v\mathbf{Z}_v^\top\mathbf{W})$ into Eq. (15), it can be transferred as follows:

$$\min_{\mathbf{H}_v \geq 0} Tr\big(\mathbf{H}_v^\top \mathbf{H}_v\big) - 2Tr\big(\mathbf{W}^\top \mathbf{Z}_v \mathbf{H}_v\big) + Tr\big(\mathbf{W}^\top \mathbf{Z}_v \mathbf{Z}_v^\top \mathbf{W}\big)$$
$$= \min_{\mathbf{H}_v \geq 0} \big\|\mathbf{H}_v - \mathbf{Z}_v^\top \mathbf{W}\big\|_{\mathrm{F}}^2. \quad (16)$$

Then, $\mathbf{H}_v$ can be updated directly by utilizing the following update rule:

$$\mathbf{H}_v = \big(\mathbf{Z}_v^\top \mathbf{W}\big)_+. \quad (17)$$

5.2.3 Update α

When updating α, the strategy of updating each view independently is used in the same way as it is used when updating $\mathbf{H}_v$. Fixing $\mathbf{W}$ and $\mathbf{H}_v$ and letting $s_v = Tr((\mathbf{Z}_v - \mathbf{W}\mathbf{H}_v^\top)^\top(\mathbf{Z}_v - \mathbf{W}\mathbf{H}_v^\top))$, Eq. (3) can be written as follows:

$$\min_{\mathbf{1}^\top \boldsymbol{\alpha}=1, \boldsymbol{\alpha} \geq 0} \sum_{v=1}^{V} \alpha_v^\gamma s_v. \quad (18)$$

Using the LMM, the augmented Lagrangian function of Eq. (18) can be expressed as

$$L_v = \sum_{v=1}^{V} \alpha_v^\gamma s_v - \lambda\left(\sum_{v=1}^{V} \alpha_v - 1\right). \quad (19)$$

Then, let the partial derivative function of L_v with respect to α_v be equal to 0 as follows:

$$\frac{\partial L_v}{\partial \alpha_v} = \gamma(\alpha_v)^{\gamma-1} s_v - \lambda = 0. \quad (20)$$

Based on this, α_v can be expressed

$$\alpha_v = \left(\frac{\lambda}{\gamma s_v}\right)^{\frac{1}{\gamma-1}}. \quad (21)$$

Substituting the resultant α_v to the constraint $\mathbf{1}\alpha^\top = 1$, the update rule of α_v is

$$\alpha_v = \frac{(\gamma s_v)^{\frac{1}{1-\gamma}}}{\sum_{v=1}^{V}(\gamma s_v)^{\frac{1}{1-\gamma}}}. \quad (22)$$

We updated $\mathbf{W}$, $\mathbf{H}_v$, and α using the three steps indicated above. Repeat this operation until the objective function Eq. (3) converges to directly extract the sample categories. Algorithm 2 summarizes the specific steps of this algorithm.

Algorithm 2. Algorithm of the EMKMC-F

Input: All views data $\mathcal{X} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_V]^\top$, the set of anchor numbers $M = [m_1, m_2, \ldots, m_V]^\top$, the number of clusters c, and the parameter $1 < \gamma < 2$.
Output: Cluster indicator matrix $\mathbf{W}$.
1: Clustering on the input multi-view data $\mathcal{X}$ to get anchors $\mathcal{A} = [\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_V]^\top$ by using k-means separately, and then constructing anchor graphs $\mathcal{Z} = [\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_V]^\top$ for all views between samples and anchors by Eq. (2).
2: **Initialize** α_v is initialized to $1/V$, $\mathbf{W}$ is initialized to a random orthogonal matrix, and $\mathbf{H}_v$ is initialized to a random positive matrix.
3: **while** not converge **do**
4: Update $\mathbf{W}$ by Eq. (8).
5: Update $\mathbf{H}_v$ by Eq. (17).
6: Update α by Eq. (22).
7: **end while**
8: Output the cluster indicator matrix $\mathbf{W}$.

5.3 Complexity Analysis

5.3.1 Computational Complexity of EMKMC-F

EMKMC-F's computational complexity is divided into three parts: 1) The complexity of $O(nt_1\sum_v m_v d_v)$ is necessary to produce anchors, where t_1, m_v, and d_v are the numbers of vth iterations, anchors, and dimensions, respectively; 2) the complexity of $O(n\sum_v m_v d_v)$ is required to construct anchor graphs among all views and their anchors; and 3) in optimization, the complexity of $O(c^2 n)$, $O(n)$, and $O(1)$ is required to update $\mathbf{W}$, $\mathbf{H}_v$, and α, respectively. As a consequence, the overall computational complexity of EMKMC-F is $O(n(t_1 + 1)\sum_v m_v d_v + t_2 c^2 n)$, where t_2 is the number of optimization iterations. In general, $m_v, d_v, t_1, t_2, c \ll n$, hence the computational complexity of EMKMC-F can be approximated to $O(n)$. In particular, when facing large-scale and high-dimensional data, EMKMC-F can still keep high efficiency and comparable clustering effectiveness.

5.3.2 Complexity Analysis of EMKMC-L and EMKMC-F

When compared to other anchor graph-based and k-means-based methods, the computational complexity of EMKMC-L and EMKMC-F is quite low. As a consequence, EMKMC-L and EMKMC-F can both be considered highly efficient methods. Actually, the computational complexity of EMKMC-F is lower than that of EMKMC-L. According to Appendix B, available in the online supplemental material, the computational complexity of EMKMC-L is $O(n(t_1 + 1)\sum_v m_v d_v + nct_3\sum_v m_v)$. To compare the complexity of the optimization part of the two methods more clearly, we made the complexity of their preprocessing equal. The only difference between them is related to the optimization process. EMKMC-L requires $O(nct_3\sum_v m_v)$, but EMKMC-F only requires $O(t_2 c^2 n)$. In reality, t_2 is smaller than t_3, and $\sum_v m_v$, no matter how tiny, is several times of c. As a result, EMKMC-F can outperform EMKMC-L in terms of efficiency, especially as the number of anchors grows.

6 Experiment

This section will evaluate the results of EMKMC-L and EMKMC-F on seven general real-world datasets (WebKB2 [36],

TABLE 2
Specification of the Real-World Multi-View Datasets

Datasets		WebKB	Mnist4	Mnist	Coli20	Yale	Cal7	Reuters	NUS	AWA
# of samples		1051	4000	10000	1440	165	1474	18758	2400	4000
# of features	view1	1840	30	30	30	4096	48	21531	64	2688
	view2	3000	9	9	19	3304	40	24892	144	2000
	view3	/	30	30	30	6750	254	34251	73	252
	view4	/	/	/	/	/	1984	15506	128	2000
	view5	/	/	/	/	/	512	11547	255	2000
	view6	/	/	/	/	/	928	/	500	2000
# of classes		2	4	10	20	15	7	6	12	50

Mnist4 [37], Coil20 [40], Yale [41], Cal7 [38], NUS [39], and AWA [27]) and two large-scale real-world datasets (Mnist [37], Reuters [27]). All datasets are listed in Table 2 in details. As baselines, eight state-of-the-art multi-view clustering algorithms (MultiNMF [42], RMKMC [29], RRMA [43], AMGL [44], LMVSC [28], NMFCC [45], MAGCN [31], and FMCNOF [17]) are introduced. Meanwhile, to verify the performance of all clustering methods, three commonly used evaluation criteria (clustering accuracy (ACC) [46], normalized mutual information (NMI) [47], and clustering purity (Purity) [48]) are used. Experiments show that EMKMC-F can significantly enhance clustering efficiency while maintaining equivalent or even greater clustering effectiveness when compared to other baselines. Finally, the parameter sensitivities of the M and γ are explored to choose optimum parameters to increase clustering performance.

6.1 Datasets

WebKB. WebKB is a webpage dataset of university computer science departments containing 8,280 documents in 7 categories. We selected 1,051 documents in the top two most popular categories to form the new WebKB dataset.

Mnist4/*Mnist*. Mnist is an image dataset with 10,000 samples composed of 10 types of handwritten digits (0-9), and each type of category contains 1,000 images. 4000 images in the first four categories (0-3) were selected to form the Mnist4 dataset. Three popular image features are extracted to form the three-view information of Mnist4 and Mnist.

Coil20. Coil20 is an image dataset containing 1,440 images and 20 types of targets. Each category contains 72 images, and each image is normalized as 32×32 pixels. Three typical image features are extracted to form the three views.

Yale. Yale is a face dataset containing 165 images of 15 people, which contains different expressions of different faces in various states, different lighting performances, and so on. Three high-dimensional features are applied to Yale to obtain a dataset with three high-dimensional feature views.

Cal7. Caltech101 is an image dataset containing 101 categories, of which 1,474 samples in 7 widely used categories are selected to form a subset called Cal7. Five commonly used image features are extracted to form different view features.

Reuters. Reuters is a document dataset composed of five languages and their translations, and there are six types of sample categories. This paper selects English documents and all types of translations to form a new Reuters dataset.

NUS. NUSW is an image dataset containing 30,000 images which are divided into 31 categories. We select the first 12 categories and the first 200 images of each category to get a subset image dataset named NUS, and six popular image features are adapted to NUS to get six view descriptions.

AWA. AWA is an image dataset containing 50 types of animals. 80 images were randomly sampled for each category to form 4,000 samples, and six commonly used image features were collected to form six feature views of the sample.

6.2 Baselines

MultiNMF. This algorithm develops a multi-view NMF-based clustering strategy that combines NMF and PLSA, which can push the solution of different views to a consensus goal instead of directly fixing it. This method has been proven to be able to efficiently complete multi-view clustering because it has linear computational complexity.

RMKMC. This method is a multi-view clustering algorithm based on traditional k-means, which obtains a unified solution by fusing the clustering results of different views. In addition, it directly constrains the decomposition factor as the clustering indicator matrix to obtain the category without additional k-means computational cost.

RRMA. This method is a multi-view matrix factorization clustering algorithm based on graph regular constraints. The $L_{2,1}$-norm is adapted to constrain the degree of approximation before and after decomposition and provides a certain degree of robustness. It inherits the high-efficiency advantage of matrix factorization and has satisfactory performance in some large-scale clustering tasks.

AMGL. By redefining the framework of spectral clustering, AMGL can autonomously learn the weight of each view without introducing attachment parameters. This can greatly improve the multi-view clustering efficiency while improving the effectiveness compared to other traditional methods.

LMVSC. This algorithm is a representative fast multi-view clustering algorithm proposed recently. It learns the anchor graphs instead of the fully connected graphs of samples as the representation matrix of multi-view subspace clustering. It can efficiently cluster multi-view data and has linear computational complexity.

TABLE 3
Running Time Comparison on All Datasets (seconds)

Datasets	MultiNMF	RMKMC	RRMA	NMFCC	MAGCN		AMGL		LMVSC		FMCNOF		EMKMC-L		EMKMC-F	
					TotalT	GraphT	TotalT	GraphT	TotalT	GraphT	TotalT	GraphT	TotalT	GraphT	TotalT	GraphT
WebKB	36.535	3.257	5.220	2.669	281.468	279.740	7.000	0.244	1.842	0.797	0.132	0.041	0.224	0.039	**0.103**	0.039
Mnist4	6.974	10.620	123.485	1.140	95.789	90.269	356.109	8.069	3.684	2.994	**0.073**	0.008	0.596	0.012	0.077	0.012
Mnist	23.418	109.451	1950.640	5.436	816.704	802.789	4305.623	110.690	48.850	12.389	0.232	0.028	3.263	0.026	**0.220**	0.026
Coil20	4.682	2.435	1.917	1.142	11.570	9.526	18.850	0.636	4.851	1.350	0.100	0.005	0.330	0.009	**0.094**	0.009
Yale	214.702	1.099	2.119	2.433	14.324	10.387	0.250	0.034	2.240	0.491	**0.096**	0.028	0.192	0.022	0.097	0.022
Cal7	63.450	6.269	11.419	4.496	458.786	453.157	15.076	1.423	13.324	4.386	0.211	0.053	0.373	0.038	**0.147**	0.038
Reuters	11978.2697	2863.765	—	167.320	—	—	23242.497	1418.251	896.236	104.805	32.045	17.079	33.782	9.258	**24.580**	9.258
NUS	18.433	24.375	25.650	3.747	373.626	367.897	94.496	4.306	11.196	2.275	0.164	0.037	0.490	0.030	**0.157**	0.030
AWA	10153.816	283.807	132.830	74.721	15863.715	15772.452	324.939	18.934	71.856	3.697	2.204	0.846	2.296	0.250	**1.119**	0.250

**The values for the k-means-based methods represent the total running time. The total running time is denoted by "TotalT" (TotalT=GraphT+optimization time) in the two columns for anchor graph-based methods, whereas the time required to construct the anchor graphs is denoted by "GraphT".*

NMFCC. This method develops a novel multi-view NMF model with orthogonal constraints to complete the clustering task, which adds orthogonal constraints to the factor metrics based on graph regular constraints to improve clustering effectiveness. Meanwhile, NMFCC inherits the advantages of NMF's low computational complexity and can efficiently complete large-scale clustering tasks.

MAGCN. This is a novel DL-based multi-view clustering method designed by designing a multi-view attribute graph convolution network model. It is designed with two-pathway encoders to map graph embedding features and learn view-consistency information simultaneously.

FMCNOF. This algorithm is considered as the latest fast multi-view clustering algorithm, which directly obtains the sample categories through orthogonal nonnegative factorization and a fused anchor graph of all views sub-anchor graphs. Compared with other fast multi-view clustering algorithms, FMCNOF can greatly improve the clustering efficiency.

6.3 Parameter Settings

To compare the performance of each method fairly, we set their parameters based on their papers. For simplicity of description, we define the datasets' order as WebKB, Mnist4, Mnist, Coil20, Yale, Cal7, Reuters, NUS, and AWA. Among them, RRMA and AMGL are parameter-free. And a regular coefficient is set to $\lambda_v = 0.01$ for the vth view of MultiNMF. The weight indexes for distinct datasets for RMKMC are chosen as $\log_{10}\gamma = 0.9$, $\log_{10}\gamma = 1.1$, $\log_{10}\gamma = 1.5$, $\log_{10}\gamma = 0.1$, $\log_{10}\gamma = 0.7$, $\log_{10}\gamma = 0.3$, $\log_{10}\gamma = 0.3$, $\log_{10}\gamma = 0.5$, and $\log_{10}\gamma = 1.1$. For NMFCC, the parameters are chosen as $\alpha = 0.01$, $\beta = 0.1$, $\gamma = 0.3$, $\mu = 0.1$, and $q = 5$ for all datasets. LMVSC contains two parameters are the number of anchors and regularization, which are chosen as 2 and 0.01, 4 and 0.01, 50 and 100, 50 and 100, 100 and 100, 100 and 1, 100 and 100, 50 and 100, and 50 and 100, respectively. And for FMCNOF, there is just one parameter, which is the number of anchors, which is set to 9, 9, 15, 25, 22, 16, 15, 19, and 57. To compare the optimization steps of EMKMC-L and EMKMC-F more clearly, they are chosen with the same parameters on the same dataset. The weight index and the number of anchors are 1.6 and $[11, 7]$, 1.5 and $[13, 11, 7]$, 1.3 and $[19, 19, 15]$, 1.4 and $[25, 29, 25]$, 1.1 and $[22, 22, 20]$, 1.7 and $[10, 16, 14, 12, 12, 16]$, 1.6 and $[13, 15, 9, 13, 13]$, 1.6 and $[21, 19, 19, 15, 21, 15]$, and 1.1 and $[57, 59, 57, 57, 53, 59]$, respectively.

6.4 Clustering Results

To evaluate the clustering efficiency and effectiveness of EMKMC-L and EMKMC-F, eight state-of-the-art methods and nine real-world datasets are used. These datasets include seven general-scale datasets and two large-scale datasets. All experiments are carried out on a platform equipped with a 3.50 GHz Intel (R) Xeon (R) CPU E5-1620 v3 with 256 GB and Matlab 2016a (64 b). The average and standard deviation of 10 times the experiment results are recorded. Tables 3 and 4 show the computational time and clustering results of all methods on different datasets. Moreover, Tables 3 and 4 show the top values in bold, while the second values are underlined. Since all methods can obtain the sample category in most cases in a short time, we no longer pay attention to methods with a computation time of more than 10 hours, which are marked by "–" in Tables 3 and 4.

Table 3 compares the running times of all methods on various real-world datasets, including seven general datasets and two large-scale datasets. When compared to anchor graph-based and k-means-based methods, EMKMC-L and EMKMC-F can improve clustering efficiency by several times to thousands of times on all datasets. When confronted with two large-scale datasets, Mnist and Reuters, the efficiency of those baselines drops dramatically, but EMKMC-L and EMKMC-F can still maintain high efficiency. In contrast to FMCNOF, which is solved by an advanced optimization strategy, the efficiency of EMKMC-L is lower than FMCNOF due to the conventional LMM-based optimization method utilized, whereas the efficiency of EMKMC-F is comparable or even higher due to the advanced optimization technique. For example, while clustering on Reuters, because a very high-dimensional connected view is required in FMCNOF, the number of k-means iterations will be substantially more than in EMKMC. Furthermore, the number of anchors in FMCNOF is 15, and the number of anchors in EMKMC is 13, 15, 9, 13, and 13. Consequently, FMCNOF's computing time for anchor graph construction is roughly twice as lengthy as

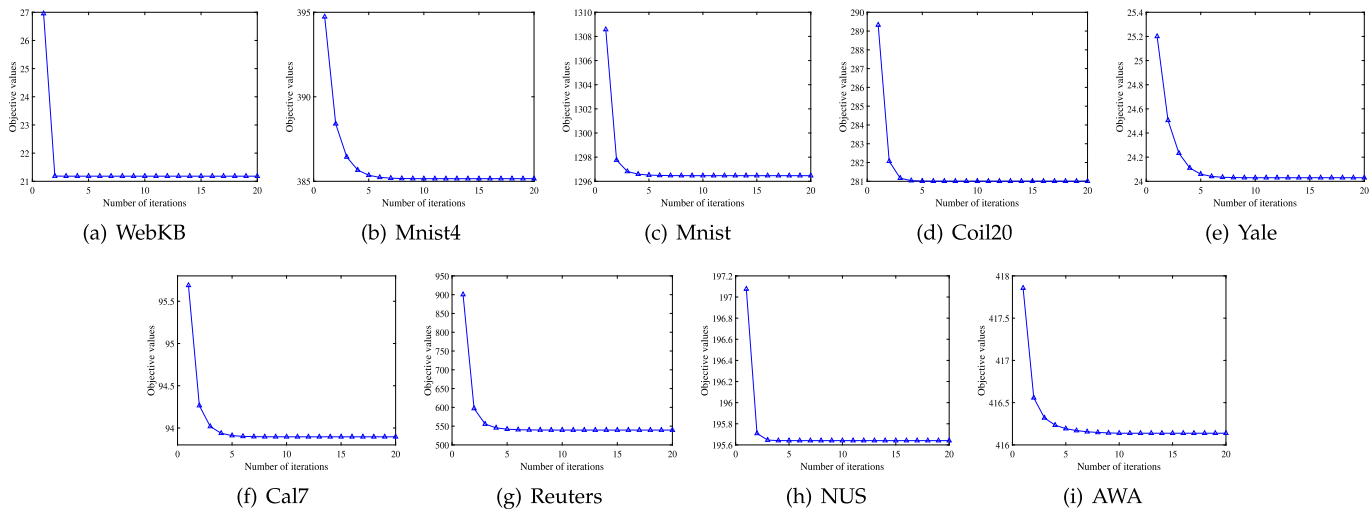
TABLE 4
Clustering Results (ACC, NMI, and Purity) Comparison on all Datasets

Metrics	Datasets	MultiNMF	RMKMC	RRMA	NMFCC	MAGCN	AMGL	LMVSC	FMCNOF	**EMKMC-L**	**EMKMC-F**	PERC
ACC	WebKB	0.639±0.167	0.913±0.006	0.776±0.000	0.812±0.108	0.924±0.007	0.789±0.000	0.882±0.000	0.779±0.086	0.917±0.079	**0.948±0.014**	1.000
	Mnist4	0.786±0.061	0.803±0.120	0.644±0.001	0.564±0.067	0.679±0.017	0.707±0.000	0.727±0.000	0.686±0.060	0.267±0.003	**0.859±0.019**	1.000
	Mnist	0.519±0.017	0.683±0.051	0.560±0.045	0.492±0.042	0.541±0.012	0.631±0.000	0.620±0.000	0.309±0.045	0.116±0.002	**0.711±0.015**	1.000
	Coil20	0.771±0.060	0.338±0.051	0.748±0.201	0.728±0.074	0.889±0.004	0.769±0.000	**0.932±0.000**	0.185±0.082	0.831±0.002	0.832±0.007	0.893
	Yale	0.457±0.028	0.179±0.020	0.398±0.081	0.503±0.026	0.544±0.013	**0.594±0.000**	0.541±0.000	0.189±0.046	0.527±0.038	0.546±0.021	0.919
	Cal7	0.371±0.057	**0.715±0.153**	0.520±0.029	0.287±0.043	0.542±0.002	0.560±0.000	0.370±0.000	0.707±0.012	0.518±0.008	0.516±0.004	0.722
	Reuters	**0.490±0.001**	0.411±0.041	—	0.432±0.019	—	0.272±0.000	0.458±0.000	0.447±0.024	0.173±0.001	0.467±0.040	0.953
	NUS	0.199±0.028	0.220±0.017	0.210±0.007	0.219±0.010	0.211±0.009	0.174±0.000	0.185±0.000	0.200±0.015	0.183±0.008	**0.223±0.007**	1.000
	AWA	0.093±0.004	0.098±0.002	0.096±0.007	0.094±0.004	0.097±0.003	0.055±0.000	0.077±0.000	0.098±0.004	0.092±0.047	**0.100±0.004**	1.000
NMI	WebKB	0.050±0.076	0.563±0.188	0.001±0.000	0.192±0.236	0.558±0.011	0.024±0.000	0.335±0.000	0.284±0.081	0.508±0.095	**0.627±0.062**	1.000
	Mnist4	0.591±0.067	0.666±0.063	0.610±0.000	0.323±0.029	0.528±0.032	0.569±0.000	0.505±0.000	0.513±0.068	0.002±0.001	**0.669±0.036**	1.000
	Mnist	0.433±0.016	0.610±0.015	0.540±0.022	0.389±0.025	0.509±0.005	0.595±0.000	0.550±0.000	0.234±0.034	0.004±0.002	**0.619±0.033**	1.000
	Coil20	0.875±0.035	0.461±0.048	0.891±0.109	0.891±0.035	0.917±0.011	0.795±0.000	**0.977±0.000**	0.228±0.123	0.883±0.091	0.923±0.030	0.945
	Yale	0.483±0.026	0.165±0.043	0.452±0.061	0.544±0.028	**0.596±0.002**	0.594±0.000	0.529±0.000	0.162±0.037	0.591±0.034	0.573±0.018	0.961
	Cal7	0.242±0.014	0.347±0.313	0.371±0.027	0.106±0.037	0.363±0.004	0.305±0.000	0.119±0.000	**0.392±0.022**	0.366±0.007	0.344±0.004	0.878
	Reuters	**0.316±0.001**	0.197±0.067	—	0.255±0.033	—	0.001±0.000	0.277±0.000	0.150±0.018	0.004±0.001	0.280±0.031	0.886
	NUS	0.091±0.012	0.117±0.007	0.102±0.006	0.101±0.003	**0.118±0.003**	0.085±0.000	0.078±0.000	0.080±0.002	0.088±0.035	0.103±0.001	0.873
	AWA	0.145±0.003	0.143±0.002	0.140±0.008	0.156±0.007	0.155±0.004	0.086±0.000	0.131±0.000	0.132±0.017	**0.161±0.015**	0.159±0.018	0.988
Purity	WebKB	0.798±0.029	0.921±0.006	0.781±0.000	0.343±0.124	0.924±0.047	0.789±0.000	0.899±0.000	0.819±0.049	0.917±0.079	**0.948±0.014**	1.000
	Mnist4	0.786±0.061	0.825±0.084	0.732±0.000	0.586±0.030	0.679±0.025	0.707±0.000	0.727±0.000	0.695±0.056	0.267±0.001	**0.859±0.019**	1.000
	Mnist	0.539±0.010	**0.732±0.027**	0.622±0.029	0.530±0.006	0.595±0.007	0.642±0.000	0.676±0.000	0.325±0.041	0.116±0.002	0.711±0.015	0.971
	Coil20	0.793±0.060	0.350±0.050	0.787±0.144	0.800±0.050	0.905±0.008	0.769±0.000	**0.982±0.000**	0.189±0.083	0.831±0.025	0.836±0.073	0.851
	Yale	0.465±0.035	0.182±0.016	0.432±0.068	0.523±0.031	0.570±0.018	0.606±0.000	**0.661±0.000**	0.193±0.043	0.576±0.056	0.552±0.022	0.835
	Cal7	0.722±0.037	0.734±0.167	0.809±0.002	0.632±0.042	0.811±0.002	0.683±0.000	0.501±0.000	0.805±0.017	0.807±0.037	**0.813±0.022**	1.000
	Reuters	**0.551±0.001**	0.474±0.060	—	0.527±0.016	—	0.273±0.000	**0.682±0.000**	0.491±0.013	0.272±0.004	0.534±0.013	0.783
	NUS	0.214±0.002	0.222±0.020	0.225±0.002	0.218±0.009	0.223±0.007	0.195±0.000	**0.349±0.000**	0.212±0.011	0.183±0.081	0.228±0.005	0.653
	AWA	0.102±0.005	0.103±0.002	0.104±0.004	0.101±0.003	0.103±0.003	0.057±0.000	**0.154±0.000**	0.010±0.003	0.101±0.031	0.105±0.006	0.682

*"PERC" means the percentage of EMKMC-F's clustering results to the highest results.

EMKMC's. For the state-of-the-art DL-based method MAGCN, though it only takes a small amount of optimization time to complete the clustering, it takes a long time to construct a full-sample graph because of its a full-sample graph-based method, especially when dealing with large-scale datasets. Thus, EMKMC has a much higher efficiency than MAGCN. Figs. 1 and 2 show the convergence curves of EMKMC-L and EMKMC-F on various datasets, and we can see that the average steps to convergence for EMKMC-L are 6.667 and 4.000 for EMKMC-F. As a consequence, EMKMC-F is the most efficient method when compared to all other methods, including FMCNOF and EMKMC-L, since EMKMC-F is the most efficient in seven datasets while FMCNOF is only the most efficient in two general datasets.

Table 4 shows the clustering results of all methods, and it can be observed that, compared to other baselines, EMKMC-F can achieve the top two high effectiveness in most cases. In particular, the effectiveness of EMKMC-F can outperform the second-place method on some datasets. For example, on WebKB, EMKMC-F's ACC, NMI, and Purity can outperform second place by 2.4%, 6.4%, and 2.4%, respectively. And on Mnist4, it outperforms second place by 5.6%, 0.3%, and 3.4%, respectively. Because of the usage of anchor graphs and the orthogonal constraint, EMKMC-F can obtain equivalent or even better clustering results than RMKMC while improving efficiency several times to hundreds of times. When compared to FMCNOF, the clustering results of EMKMC-F greatly outperform FMCNOF's since it performs k-means for different views separately when generating anchors and allows different anchors generated for different views to explore the underlying structure of distinct views. In the case of EMKMC-L, as it has to relax the



(a) WebKB (b) Mnist4 (c) Mnist (d) Coil20 (e) Yale

(f) Cal7 (g) Reuters (h) NUS (i) AWA

Fig. 1. Convergence curves of EMKMC-L on different real-world datasets and the average iteration step on nine datasets is 6.667.

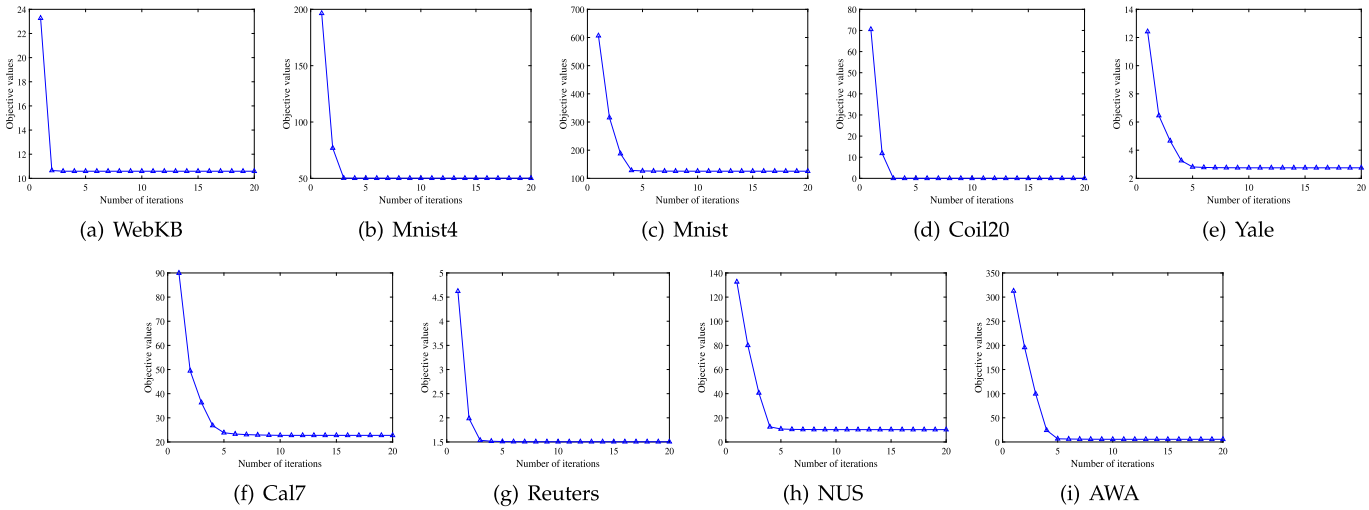


Fig. 2. Convergence curves of EMKMC-F on different real-world datasets and the average iteration step on nine datasets is 4.000.

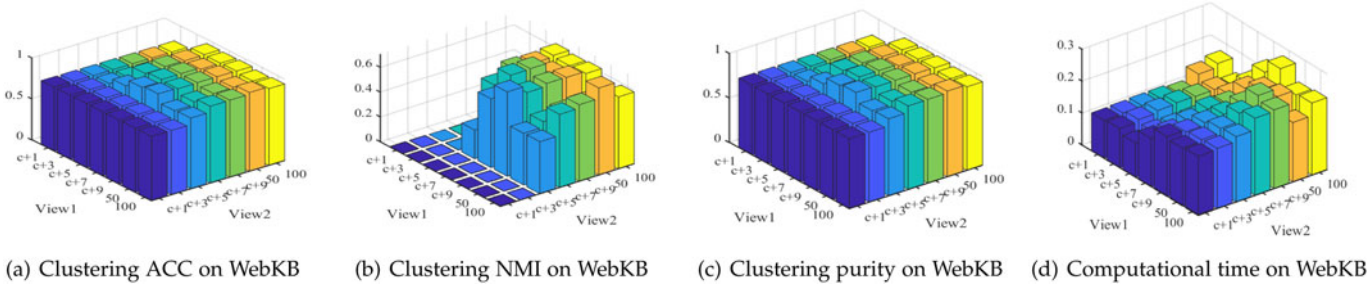


Fig. 3. Clustering ACC, NMI, Purity and computational time of EMKMC-F with different number of anchors on WebKB dataset.

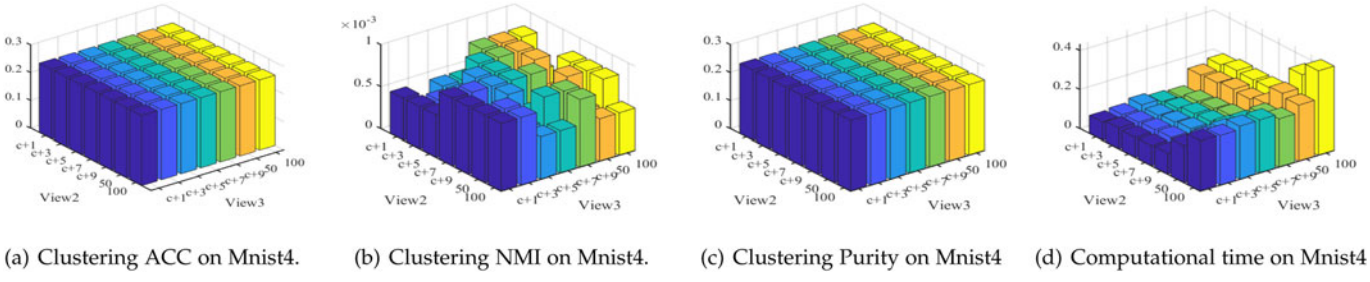
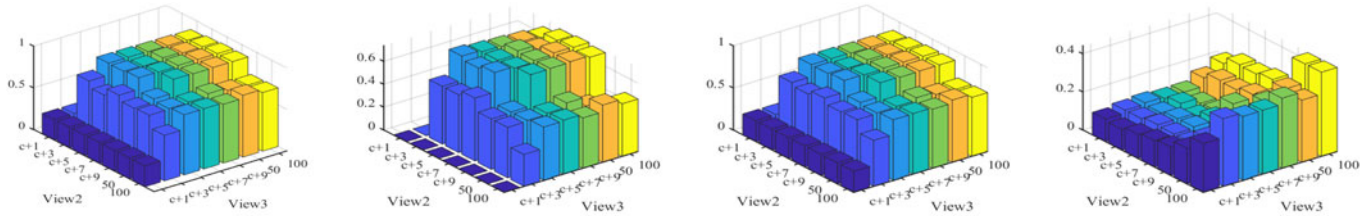


Fig. 4. Clustering ACC, NMI, Purity and computational time of EMKMC-F with different number of anchors of view2 and view3 on WebKB dataset when the number of anchors of view1 is $c+1$.

objective function to a Lagrange function to solve it, some information loss results in some clustering effectiveness loss when compared to EMKMC-F. Compared with the state-of-the-art DL-based method MAGCN, although it can achieve the highest NMI on Yale and NUS datasets, and the second ACC and Purity on some other datasets, its clustering effectiveness is overall lower than EMKMC, especially on some large-scale datasets. Actually, though EMKMC-F can be considered an efficient and effective clustering method when compared to the other methods, it cannot achieve the top two results on all datasets because it evolved from spectral clustering and is still a two-stage clustering method, first constructing anchor graphs and then performing matrix factorization on these anchor graphs, which will result in some information mismatch and a loss of clustering effectiveness. To achieve enhanced clustering effectiveness, we will concentrate our efforts in the future on developing a multi-view clustering method that simultaneously constructs anchor graphs and matrix factorization to reduce the clustering effectiveness loss caused by information mismatch in the current two-step multi-view clustering methods.

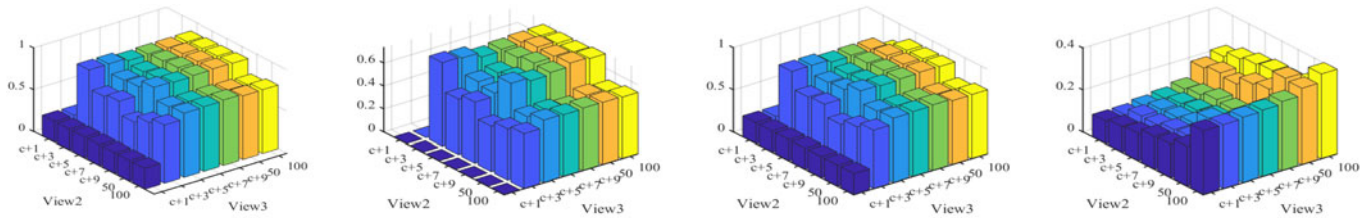
6.5 Parameter Sensitivity

The clustering efficiency and effectiveness of EMKMC-F are affected by two parameters: a set of anchor numbers $M = [m_1, m_2, \ldots, m_V]^\top$ and the weight index γ of different views. WebKB and Mnist4 are used as examples in this section to discuss the effects of different values of these two parameters on EMKMC-F.



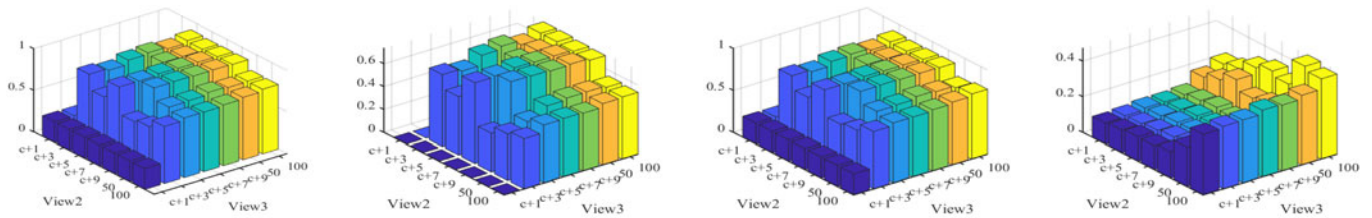
(a) Clustering ACC on Mnist4 (b) Clustering NMI on Mnist4 (c) Clustering purity on Mnist4 (d) Computational time on Mnist4

Fig. 5. Clustering ACC, NMI, Purity and computational time of EMKMC-F with different number of anchors of view2 and view3 on WebKB dataset when the number of anchors of view1 is $c+3$.



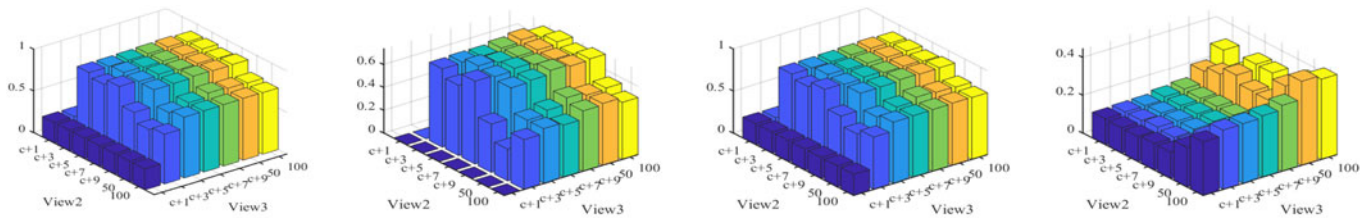
(a) Clustering ACC on Mnist4 (b) Clustering NMI on Mnist4 (c) Clustering Purity on Mnist4 (d) Computational time on Mnist4

Fig. 6. Clustering ACC, NMI, Purity and computational time of EMKMC-F with different number of anchors of view2 and view3 on WebKB dataset when the number of anchors of view1 is $c+5$.



(a) Clustering ACC on Mnist4 (b) Clustering NMI on Mnist4 (c) Clustering purity on Mnist4 (d) Computational time on Mnist4

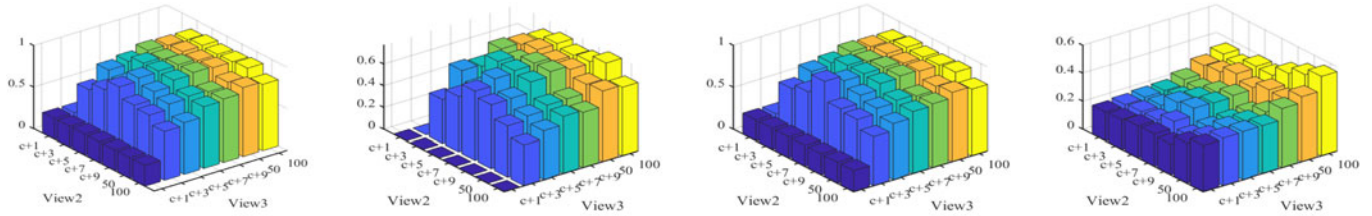
Fig. 7. Clustering ACC, NMI, Purity and computational time of EMKMC-F with different number of anchors of view2 and view3 on WebKB dataset when the number of anchors of view1 is $c+7$.



(a) Clustering ACC on Mnist4 (b) Clustering NMI on Mnist4 (c) Clustering purity on Mnist4 (d) Computational time on Mnist4

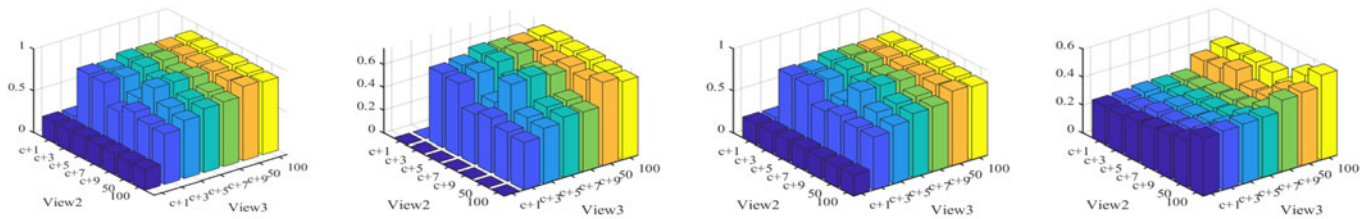
Fig. 8. Clustering ACC, NMI, Purity and computational time of EMKMC-F with different number of anchors of view2 and view3 on WebKB dataset when the number of anchors of view1 is $c+9$.

The selection of anchors $M = [m_1, m_2, \ldots, m_V]^\top$ is critical for achieving higher clustering effectiveness. To fully exploit the underlying structure of distinct views, different from FMCNOF, EMKMC-F generates different anchors for each view. In practice, we choose the number of anchors for WebKB, Mnist4, Mnist, Coil20, Yale, Cal7, Reuters, and NUS from the set $\{c+1, c+3, c+5, c+7, c+9, 50, 100\}$, and the number of anchors for AWA from the set $\{c+1, c+3, c+5, c+7, c+9, 100, 200\}$. Fig. 3 shows the clustering ACC, NMI, Purity, and computational time for the two views of WebKB when a different number of anchors are chosen. Figs. 4, 5, 6, 7, 8, 9, and 10 show the clustering ACC, NMI, Purity, and computational time when views 2 and 3 have different numbers of anchors and view 1 has a fixed number of anchors. Experiments show that while the number of anchors is small, the clustering ACC, NMI, and Purity increase significantly as the number of anchors grows. When the number of anchors reaches a



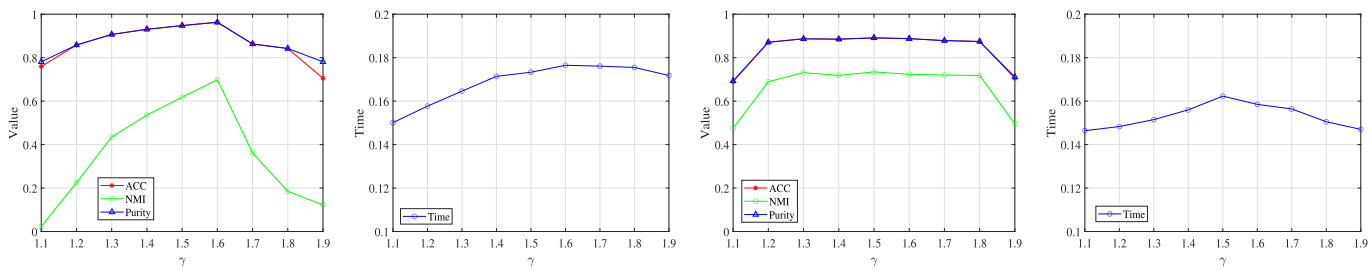
(a) Clustering ACC on Mnist4 (b) Clustering NMI on Mnist4 (c) Clustering purity on Mnist4 (d) Computational time on Mnist4

Fig. 9. Clustering ACC, NMI, Purity and computational time of EMKMC-F with different number of anchors of view2 and view3 on WebKB dataset when the number of anchors of view1 is 50.



(a) Clustering ACC on Mnist4 (b) Clustering NMI on Mnist4 (c) Clustering purity on Mnist4 (d) Computational time on Mnist4

Fig. 10. Clustering ACC, NMI, Purity and computational time of EMKMC-F with different number of anchors of view2 and view3 on WebKB dataset when the number of anchors of view1 is 100.



(a) Clustering results on WebKB (b) Computational time on WebKB (c) Clustering results on Mnist4 (d) Computational time on Mnist4

Fig. 11. Clustering results and computational time of EMKMC-F with different γ on WebKB and Mnist4 datasets.

certain level and then continues to rise, the clustering effectiveness no longer rises appreciably but instead oscillates or even falls. The computing time usually increases as the number of anchors grows. In principle, a greater number of anchors will result in a longer computing time. In practice, however, because of the machine's hardware, the Matlab platform, and the algorithm's random initialization of the variables are not always constant, there is frequently a drop in computing time with a slight increase in the number of anchors. However, this is only applicable if the number of anchors does not grow much. When the number of anchors grows significantly, so does the computing time. Due to the varying importance of each view, the final clustering result commonly shows a stronger dependence on those more important views, and the number of anchors for those important views has a strong influence on the final clustering results. As a consequence, better clustering efficiency and effectiveness can be gained with an appropriate number of anchors for each view.

The weighted index γ $(1 < \gamma < 2)$ is another parameter of EMKMC-F. As we know, the important views in multi-view data provide richer and more meaningful structural information. To fully utilize this attribute, the index parameter γ is added to the different weights, emphasizing those important views and increasing clustering effectiveness. The clustering ACC, NMI, Purity, and computational time produced by different parameters on the WebKB and Mnist4 datasets are shown in Fig. 11. As can be seen in Fig. 11, when γ is close to 1, the clustering effectiveness and efficiency rise dramatically as γ grows. When γ reaches a particular level, clustering effectiveness and computational time rise slowly or even remain steady. When γ continues to grow over a certain critical value, clustering effectiveness and efficiency suffer considerably. Therefore, to obtain the optimal clustering effectiveness and efficiency, a reasonable choice of γ is necessary.

7 Conclusion

In this work, to increase the efficiency of multi-view clustering, we present an efficient multi-view k-means clustering model (EMKMC). It uses anchor graphs as the input to the k-means-based method instead of original data to decrease computational complexity to $O(n)$. Furthermore, the factor matrix is constrained as an orthogonal indicator matrix to decrease decomposition freedom and improve clustering effectiveness while avoiding further post-processing after optimization. Then, two algorithms are designed to solve

the EMKMC model. The first is called EMKMC-L, and the second is EMKMC-F. We can conclude that EMKMC-F is more efficient and effective than EMKMC-L after analyzing their computational complexity and doing numerous experiments. Extensive experiments on nine real-world datasets show that, when compared to other cutting-edge efficient methods, EMKMC-F can significantly improve clustering efficiency while maintaining or even exceeding clustering effectiveness. The code and datasets of this paper can be found in https://github.com/abyeshizhe/EMKMC.

REFERENCES

[1] S. Bickel and T. Scheffer, "Multi-view clustering," in *Proc. 4th IEEE Int. Conf. Data Mining*, 2004, pp. 19–26.

[2] T. Xia, D. Tao, T. Mei, and Y. Zhang, "Multiview spectral embedding," *IEEE Trans. Cybern.*, vol. 40, no. 6, pp. 1438–1446, Dec. 2010.

[3] A. Kumar, P. Rai, and H. Daumé, "Co-regularized multi-view spectral clustering," in *Proc. 24th Int. Conf. Neural Inf. Process. Syst.*, 2011, pp. 1413–1421.

[4] A. Kumar and H. Daumé, "A co-training approach for multi-view spectral clustering," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 393–400.

[5] H. Gao, F. Nie, X. Li, and H. Huang, "Multi-view subspace clustering," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4238–4246.

[6] Y. Wang, X. Lin, L. Wu, W. Zhang, Q. Zhang, and X. Huang, "Robust subspace clustering for multi-view data by exploiting correlation consensus," *IEEE Trans. Process.*, vol. 24, no. 11, pp. 3939–3949, Nov. 2015.

[7] F. Nie, G. Cai, and X. Li, "Multi-view clustering and semi-supervised classification with adaptive neighbors," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 2408–2414.

[8] K. Zhan, C. Zhang, J. Guan, and J. Wang,, "Graph learning for multiview clustering," *IEEE Trans. Cybern.*, vol. 48, no. 10, pp. 2887–2895, Oct. 2018.

[9] K. Zhan, C. Niu, C. Chen, F. Nie, C. Zhang, and Y. Yang, "Graph structure fusion for multiview clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 10, pp. 1984–1993, Oct. 2019.

[10] J. Wu, Z. Lin, and H. Zha, "Essential tensor learning for multi-view spectral clustering," *IEEE Trans. Process.*, vol. 28, no. 12, pp. 5910–5922, Dec. 2019.

[11] H. Wang, Y. Yang, and B. Liu, "GMC: Graph-based multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1116–1129, Jun. 2020.

[12] J. Wen, Y. Xu, and H. Liu, "Incomplete multiview spectral clustering with adaptive graph learning," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1418–1429, Apr. 2020.

[13] J. Liu, X. Liu, Y. Yang, X. Guo, M. Kloft, and L. He, "Multiview subspace clustering via co-training robust data representation," *IEEE Trans. Neural Netw. Learn. Syst.*, early access. Apr. 9, 2021, doi: 10.1109/TNNLS.2021.3069424.

[14] W. Zhao, C. Xu, Z. Guan, and Y. Liu, "Multiview concept learning via deep matrix factorization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 814–825, Feb. 2021.

[15] Z. Zhang, L. Liu, F. Shen, H. T. Shen, and L. Shao, "Binary multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1774–1782, Jul. 2019.

[16] J. Gao and J. Ye, "Anchors bring ease: An embarrassingly simple approach to partial multi-view clustering," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 4412–4419.

[17] B. Yang, X. Zhang, F. Nie, F. Wang, W. Yu, and R. Wang, "Fast multi-view clustering via nonnegative and orthogonal factorization," *IEEE Trans. Image Process.*, vol. 30, pp. 2575–2586, Dec. 2021.

[18] S. Shi, F. Nie, R. Wang, and X. Li, "Fast multi-view clustering via prototype graph," *IEEE Trans. Knowl. Data Eng.*, early access, May 11, 2021, doi: 10.1109/TKDE.2021.3078728.

[19] Q. Qiang, B. Zhang, F. Wang, and F. Nie, "Fast multi-view discrete clustering with anchor graphs," in *Proc. 35th AAAI Conf. Artif. Intell.*, 2021, pp. 9360–9367.

[20] B. Yang, X. Zhang, B. Chen, F. Nie, Z. Lin, and Z. Nan, "Efficient correntropy-based multi-view clustering with anchor graph embedding," *Neural Netw.* vol. 146, pp. 290–302, Feb. 2022.

[21] F. Nie, S. Shi, and X. Li, "Auto-weighted multi-view co-clustering via fast matrix factorization," *Pattern Recognit.*, vol. 102, pp. 107–207, Jun. 2020.

[22] B. Yang *et al.*, "ECCA: Efficient correntropy-based clustering algorithm with orthogonal concept factorization," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 31, 2021, doi: 10.1109/TNNLS.2022.3142806.

[23] C. Ding and X. He, "K-means clustering and principal component analysis," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, pp. 1–9.

[24] C. Ding, X. He, and H. D. Simon, "Nonnegative lagrangian relaxation of k-means and spectral clustering," in *Proc. Eur. Conf. Mach. Learn.*, 2005, pp. 530–538.

[25] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix tri-factorizations for clustering," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2006, pp. 126–135.

[26] T. L. Liu, "Guided co-training for large-scale multi-view spectral clustering," Jul. 2017, *arXiv:1707.09866*.

[27] Y. Li, F. Nie, H. Huang, and J. Huang, "Large-scale multi-view spectral clustering via bipartite graph," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2750–2756.

[28] Z. Kang, W. Zhou, Z. Zhao, J. Shao, M. Han, and Z. Xu, "Large-scale multi-view subspace clustering in linear time," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 118–125.

[29] 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.

[30] Z. Li, Q. Wang, Z. Tao, Q. Gao, and Z. Yang, "Deep adversarial multi-view clustering network," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 2952–2958.

[31] J. Cheng, Q. Wang, Z. Tao, D. Xie, and Q. Gao, "Multi-view attribute graph convolution networks for clustering," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, 2020, pp. 2973–2979.

[32] Z. Liu, Y. Li, L. Yao, X. Wang, and F. Nie, "Agglomerative neural networks for multiview clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 7, pp. 2842–2852, Jul. 2022.

[33] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The constrained Laplacian rank algorithm for graph-based clustering," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 1969–1976.

[34] W. Zhu, F. Nie, and X. Li, "Fast spectral clustering with efficient large graph construction," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 2492–2496.

[35] C. Wang, F. Nie, R. Wang, and X. Li, "Revisiting fast spectral clustering with anchor graph," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 3902–3906.

[36] D. Cai, X. He, X. Wu, and J. Han, "Nonnegative matrix factorization on manifold," in *Proc. IEEE 8th Int. Conf. Data Mining*, 2008, pp. 15–19.

[37] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[38] F. Li, F. Rob, and P. Pietro, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshop*, 2004, pp. 178–186.

[39] T. S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: A real-world web image database from national university of Singapore," in *Proc. ACM Int. Conf. Image Video Retrieval*, 2009, pp. 48–56.

[40] K. Zhan, F. Nie, J. Wang, and Y. Yang, "Multi-view consensus graph clustering," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1261–1270, Mar. 2019.

[41] H. Zhao, Z. Ding, and Y. Fu, "Multi-view clustering via deep matrix factorization," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 2921–2927.

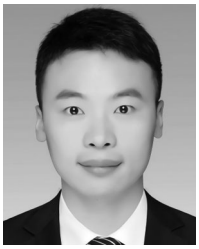
[42] J. Liu, C. Wang, and X. Li, "Multi-view clustering via joint non-negative matrix factorization," in *Proc. SIAM Int. Conf. Data Mining*, 2013, pp. 252–260.

[43] J. Pu, Q. Zhang, L. Zhang, B. Du, and J. You, "Multiview clustering based on robust and regularized matrix approximation," in *Proc. 23rd Int. Conf. Pattern Recognit.*, 2016, pp. 2550–2555.

[44] F. Nie, J. Li, L. Shi, and X. Li, "Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 1881–1887.

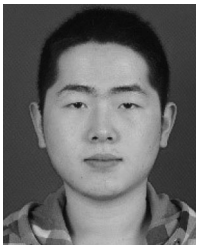
[45] N. Liang, Z. Yang, Z. Li, W. Sun, and S. Xie,, "Multi-view clustering by non-negative matrix factorization with co-orthogonal constraints," *Knowl. Based Syst.*, vol. 194, Apr. 2020, Art. no. 105582.

[46] M. Wu and B. Schölkopf, "A local learning approach for clustering," in *Proc. 19th Int. Conf. Neural Inf. Process. Syst.*, 2006, pp. 1529–1536.
[47] A. Strehi and J. Ghosh, "Cluster ensembles-a knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Mar. 2003.
[48] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge Univ. Press, 2008.



Ben Yang received the BS degree in automation science and technology from Xi'an Jiaotong University, Shaanxi, China, in 2016. Currently, he is working toward the PhD degree with the Institute of Artificial Intelligence and Robotics and College of Artificial Intelligence, Xi'an Jiaotong University. His current research interests include image processing, computer vision, data mining, and machine learning.

Xuetao Zhang received the BS degree in information enginerring, and the MS and PhD degrees in pattern recognition and intelligence system from Xi'an Jiaotong University, China, in 2003, 2006 and 2012, respectively. He visited the Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology from 2009 to 2010. He is currently an associate professor of the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. His research interests include computer vision, human vision, and machine learning.



Zhongheng Li received the BE degree in automation from Xi'an Jiaotong University, Xi'an, China, in 2014, where he is currently working toward the PhD degree in pattern recognition and intelligent system. His main research interests include machine learning, pattern recognition, image processing, and computer vision.

Feiping Nie (Senior Member, IEEE) received the PhD degree in computer science from Tsinghua University, China, in 2009, and currently is full professor in Northwestern Polytechnical University, China. His research interests are machine learning and its applications, such as pattern recognition, data mining, computer vision, image processing and information retrieval. He has published more than 200 papers in the following journals and conferences: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Image Processing*, *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Knowledge and Data Engineering*, ICML, NIPS, KDD, IJCAI, AAAI, ICCV, CVPR, ACM MM. His papers have been cited more than 28000 times and the H-index is 90. He is now serving as associate editor or PC member for several prestigious journals and conferences in the related fields.

Fei Wang received the BS degrees in electronics from Northwest University, the MS degree in communication and information system from the Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, and the PhD degree in pattern recognition and intelligence system from Xi'an Jiaotong University, China, in 1998, 2002 and 2009, respectively. He visited the North Carolina State University from 2012 to 2013. He is currently a professor of Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. His research interests include computer vision and intelligent system.