Discriminatively Fuzzy Multi-View K-means Clustering with Local Structure Preserving

Jun Yin[1], **Shiliang Sun**[2*], **Lai Wei**[1], **Pei Wang**[1]

[1]College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China
[2]Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China
junyin@shmtu.edu.cn, shiliangsun@gmail.com, weilai@shmtu.edu.cn, w823701615@163.com

Abstract

Multi-view K-means clustering successfully generalizes K-means from single-view to multi-view, and obtains excellent clustering performance. In every view, it makes each data point close to the center of the corresponding cluster. However, multi-view K-means only considers the compactness of each cluster, but ignores the separability of different clusters, which is of great importance to producing a good clustering result. In this paper, we propose Discriminatively Fuzzy Multi-view K-means clustering with Local Structure Preserving (DFMKLS). On the basis of minimizing the distance between each data point and the center of the corresponding cluster, DFMKLS separates clusters by maximizing the distance between the centers of pairwise clusters. DFMKLS also relaxes its objective by introducing the idea of fuzzy clustering, which calculates the probability that a data point belongs to each cluster. Considering multi-view K-means mainly focuses on the global information of the data, to efficiently use the local information, we integrate the local structure preserving into the framework of DFMKLS. The effectiveness of DFMKLS is evaluated on benchmark multi-view datasets. It obtains superior performances than state-of-the-art multi-view clustering methods, including multi-view K-means.

Introduction

As an important unsupervised learning method, clustering has been extensively studied and applied in various fields such as data mining, pattern recognition and machine learning (Xu and Wunsch 2005; Wang et al. 2020). K-means is a popular and widely used clustering method. MacQueen (MacQueen 1967) introduced the K-means algorithm, which aims to find a partition that minimizes the squared error between the empirical mean of a cluster and the data points within that cluster. Based on this, Cheung (Cheung 2003) proposed a generalized version of the traditional K-means algorithm, which is not only suitable for partitioning elliptical data, but also partitions correctly without pre-assigning exact cluster numbers.

With the rapid advancement of information technology, data volumes across various fields are experiencing exponential growth. In numerous practical applications, data is often gathered from diverse domains and different sensors, resulting in the emergence of multi-view data (Sun 2013). Multi-view data refers to situations in which a singular object or entity can be represented using various data sources or distinct feature sets. Each data source or feature set can be considered as a distinct view, and capable of independent utilization in clustering analysis. It is important to note that there exist inherent connections as well as variations among these different views, highlighting the diversity in data representation. For example, for medical image data, ultrasound image, CT image and MRI image can be seen as multiple views; for data acquisition using sensors, data acquired by different sensors can be used as different views of the same data. Multi-view data can provide common semantics to improve the learning effectiveness (Asano et al. 2020; Peng et al. 2020).

Multi-view clustering is a clustering method for multi-view data that aims to improve the quality of clustering results by integrating information from different views. In multi-view clustering, each view corresponds to a distinct aspect or feature representation of the data. By combining multiple views, we can obtain different perspectives of the data, allowing us to capture more comprehensive and accurate information. In recent years, researchers have proposed lots of multi-view clustering models and algorithms (Zhang et al. 2018; Cao et al. 2015; Xu et al. 2022; Nie et al. 2017), and have made great progress. Among them, multi-view clustering methods based on co-training aim at maximizing mutual agreement among all views and reaching the broadest consensus. Kumar et al. (Kumar, Rai, and Daume 2011) proposed a co-regularized multi-view clustering method that applies the graph Laplacian operator to all views and regularizes the feature vectors of the Laplacian operator to obtain consistent clustering results. Ye et al. (Ye et al. 2016) proposed a co-regularized kernel K-means algorithm that automatically learns the weights of different views from the data.

In contrast, the graph-learning-based multi-view clustering methods aim to find a fused graph among all views and then use graph-cutting algorithms or other clustering techniques to obtain the clustering results. Wang et al. (Wang, Yang, and Liu 2019) fused the data graph matrices of all views to generate a unified graph matrix for improving the data graph matrix of each view and directly derived the final

*Corresponding author

clustering results. Liang et al. (Liang et al. 2022) proposed an robust and parameter-free parametric graph-based multi-view clustering method. They defined a convex optimization graph-based multi-view clustering formulation, and used a gradient descent-based algorithm to solve the resulting optimization problem. To obtain accurate similarity graph for multi-view clustering, Li et al. (Li et al. 2022) constructed it in the spectral embedding space and developed a graph learning method which learns spectral embedding and tensor representation simultaneously.

In addition, the subspace learning-based multi-view clustering methods aim to find a shared latent subspace from multiple low-dimensional subspace, which assume that the input views can be generated from the latent subspace, and then use existing clustering algorithms to obtain the final clustering results. Cheng et al. (Cheng, Jing, and Ng 2018) proposed a multi-view clustering method based on tensor self-representation learning of shared subspace, which combines self-representation learning and tensor decomposition techniques with subspace learning, and can effectively address the issue of high dimensionality while considering the intrinsic connections between data. Li et al. (Li et al. 2019) proposed an interactive multi-layer subspace learning algorithm for multi-view clustering, using hierarchical self-representative layers to construct mutually inverse multi-layer subspace representations, and using backward encoding networks to explore the complex relationships between different views. Liu et al. (Liu et al. 2013) put forward multi-view clustering based on Nonnegative Matrix Factorization (NMF), which seeks a common subspace representation of multiple views with a joint NMF.

As a classical clustering method, K-means is designed to solve single-view data clustering problem. To deal with multi-view data, multi-view K-means algorithms (Xu et al. 2017; Xu, Han, and Nie 2016; Chen et al. 2020) are proposed. In order to solve the weighting problem between different views, Liu et al. (Liu et al. 2020a) proposed a cluster-weighted kernel K-means method to assign a weight to each inner cluster of each view, which is learned based on the intra-cluster similarity between the clusters and all corresponding clusters under different views, so that the clusters with high intra-cluster similarity have higher weights in the corresponding clusters. Han et al. (Han et al. 2020) focused on constructing common affiliation matrices with appropriate sparsity on different views and learning the center of mass matrix and its corresponding weights for each view. Cai et al. (Cai, Nie, and Huang 2013) proposed a robust large-scale multi-view K-means clustering method to integrate heterogeneous representations of multi-view data and used the structured sparsity norm to make their method robust to outlier. The computational efficiency of the K-means based approaches decreases significantly when the dimension of data is large. For this reason, Yang et al. (Yang et al. 2023) proposed a multi-view K-means clustering method with multiple anchor graphs to construct anchor graph for each view and integrate these anchor graphs to obtain the label of each sample without any additional processing. For multi-view kernel K-means with incomplete views, to reduce the time and space complexities of imputation of kernel matrix, Liu et al. (Liu et al. 2020b) associated each incomplete base matrix generated from incomplete views with the learned consistent clustering matrix instead of complementing the incomplete kernel matrix.

Multi-view K-means makes great achievements in clustering multi-view data. This category of method mainly pays attention to the compactness of cluster, which makes each data point as close as possible to the center of the cluster that it belongs to in each view. However, multi-view K-means ignores the separability of different clusters. For a excellent clustering result, we hope that different clusters are as far as possible from each other. To address this limitation, this paper proposes Discriminatively Fuzzy Multi-view K-means clustering with Local Structure preserving (DFMKLS). DFMKLS introduces discriminative property into multi-view K-means by minimizing the within-cluster scatter and maximizing the between-cluster scatter simultaneously. Besides, with the idea of fuzzy clustering, DFMKLS relaxes the objective of multi-view K-means and calculates the probability that a data point belongs to each cluster. To utilize the local information of the data to improve the clustering performance, the local structure preserving is also integrated into the framework of DFMKLS. The contribution of the paper is summarized as follows:

(1) We bring discriminative property into multi-view K-means. In every view, the compactness within cluster and the separability between clusters are considered simultaneously.

(2) We introduce the idea of fuzzy clustering into multi-view K-means. Each data point does not strictly belong to one cluster, while the probability that it belongs to each cluster is calculated.

(3) We integrate the local structure preserving into the objective of multi-view K-means to make the global information and local information of multi-view data are utilized simultaneously.

(4) We develop an iteration algorithm with multiplicative update rule to solve the objective of the proposed DFMKLS.

The remainder of the paper is organized as: Section 2 reviews two related works, i.e., K-means and Multi-view K-means. Section 3 gives the proposed DFMKLS, including its formulation and optimization. Section 4 evaluates the effectiveness of DFMKLS by experiments. Section 5 provides the conclusion of the paper.

Related Works

K-means Clustering

K-means (MacQueen 1967) is a typical clustering algorithm that considers the existence of C clusters among the samples and the feature distribution of each cluster is represented by its center, then each data point can be assigned to the cluster nearest to it. The objective function of K-means can be defined as

$$\begin{aligned}&\min_{P,Q} \|X - PQ\|_F^2\\ &\text{s.t. } Q_{ci} \in \{0,1\}, \sum_{c=1}^{C} Q_{ci} = 1, \forall i = 1, 2, \cdots, n\end{aligned} \tag{1}$$

where $X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{d\times n}$ is the input data matrix with n instances and d dimensional features, $P \in \mathbb{R}^{d\times C}$ is the cluster centroid matrix, and $Q \in \mathbb{R}^{C\times n}$ is the cluster assignment matrix and each column of the matrix Q is a one-hot vector. If x_i is assigned to the c-th cluster, then $Q_{ci} = 1$, and $Q_{ci} = 0$, otherwise.

Multi-view K-means Clustering

Robust Multi-view K-Means Clustering (RMKMC) generalizes K-means to multi-view data (Cai, Nie, and Huang 2013). Suppose that a multi-view data set is composed of V views for n instances denoted by $\{x_1^v, x_2^v, \cdots, x_n^v\}_{v=1}^V \in \mathbb{R}^{d^v}$. Let $X^v \in \mathbb{R}^{d^v\times n}$ denote the data of the v-th view, $P^v \in \mathbb{R}^{d^v\times C}$ be the centroid matrix of the v-th view, $Q^v \in \mathbb{R}^{n\times C}$ be the clustering indicator matrix of the v-th view. The clustering results of different views should be unique, meaning that the clustering indicator matrices Q^v of different views should be consistent. To reduce the negative impact of outlier on K-means clustering, RMKMC uses the sparsity-inducing norm, i.e., $\ell_{2,1}$-norm, to replace the F-norm in Eq. (1). The objective function of RMKMC is defined as

$$\begin{aligned} &\min_{P^v,Q,\varepsilon^v} \sum_{v=1}^{V} (\varepsilon^v)^\gamma \left\| X^v - P^v Q \right\|_{2,1} \\ &\text{s.t. } Q_{ci} \in \{0,1\}, \sum_{c=1}^{C} Q_{ci} = 1, \sum_{v=1}^{V} \varepsilon^v = 1, \end{aligned} \tag{2}$$

where ε^v is the weight of the v-th view and γ is the parameter controlling the weight distribution. Eq. (2) learns the weights of different views, allowing important views to obtain big weights.

The Proposed Approach

Formulation

It is easy to know that, the cluster centroid matrix P^v in Eq. (2) can be calculated by $P^v = X^v Q^T \Lambda$, where $\Lambda \in \mathbb{R}^{C\times C}$ is a diagonal matrix with the diagonal element $\Lambda_{cc} = 1/\sum_{i=1}^{n} Q_{ci}$. If we relax the optimization by allowing Q to be any positive number, and treat all the views equally, the objective of multi-view K-means clustering can be represented as

$$\begin{aligned} &\min_{Q,\Lambda} \sum_{v=1}^{V} \left\| X^v - X^v Q^T \Lambda Q \right\|_F^2 \\ &\text{s.t. } Q \geq 0. \end{aligned} \tag{3}$$

In Eq. (3), we simply use F-norm minimization to make each data point close to the center of the cluster which it belongs to. Here, Q_{ci} indicates the probability that the i-th data point belongs to the c-th cluster. Since each column of Q is not one-hot vector, each data point does not definitely belongs to one cluster. Thus, Eq. (3) can be seen as the objective of Fuzzy Multi-view K-means clustering (FMK), and each column of $X^v Q^T \Lambda$ is the fuzzy center.

For a good clustering result, not only each data point is close to the center of the corresponding cluster, but also these centers should be far away from each other, which makes different clusters have stronger separability. To achieve this goal, we maximize the distances between the centers of fuzzy clusters, and obtain the Discriminatively Fuzzy Multi-view K-means clustering (DFMK). Its objective can be formulated as

$$\begin{aligned} &\min_{Q,\Lambda} \sum_{v=1}^{V} \frac{\left\| X^v - X^v Q^T \Lambda Q \right\|_F^2}{\sum_{l,m}^{C} \left\| X^v Q_{l,:}^T \Lambda_{ll} - X^v Q_{m,:}^T \Lambda_{mm} \right\|_2^2} \\ &\text{s.t. } Q \geq 0, \end{aligned} \tag{4}$$

where $Q_{l,:}$ and $Q_{m,:}$ are the l-th and m-th rows of Q, respectively, and $X^v Q_{l,:}^T \Lambda_{ll}$ and $X^v Q_{m,:}^T \Lambda_{mm}$ can be seen as the centers of the l-th and m-th fuzzy clusters. With Eq. (4), the fuzzy within-cluster scatter is minimized and the fuzzy between-cluster scatter is maximized simultaneously.

DFMK mainly focuses on the global structure of data, but ignores the local structure, which is also important for clustering. From the local perspective, we think that, two data points with strong connection should also have similar cluster assignment result. The connection between two data points can be represented by the adjacent relation of them, and the adjacent matrix $S^v \in \mathbb{R}^{n\times n}$ of the v-th view is constructed as

$$S_{ij}^v = \begin{cases} 1, & \text{if } x_i^v \text{ belongs to } K\text{-nearest-neighbors of } x_j^v \\ & \text{or } x_j^v \text{ belongs to } K\text{-nearest-neighbors of } x_i^v \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Then, the objective of keeping local clustering structure is formulated as

$$\min_{Q} \frac{1}{2} \sum_{v=1}^{V} \sum_{i,j}^{n} \left\| Q_{:,i} - Q_{:,j} \right\|_2^2 S_{ij}^v, \tag{6}$$

where $Q_{:,i}$ and $Q_{:,j}$ are the i-th and j-th columns of Q, and can be seen as the clustering assignment results of the i-th and j-th data points.

Combining Eq. (4) and Eq. (6), we can obtain the final objective of Discriminatively Fuzzy Multi-view K-means clustering with Local Structure preserving (DFMKLS) as

$$\begin{aligned} &\min_{Q,\Lambda} \sum_{v=1}^{V} \frac{\left\| X^v - X^v Q^T \Lambda Q \right\|_F^2 + \frac{\alpha}{2} \sum_{i,j}^{n} \left\| Q_{:,i} - Q_{:,j} \right\|_2^2 S_{ij}^v}{\sum_{l,m}^{C} \left\| X^v Q_{l,:}^T \Lambda_{ll} - X^v Q_{m,:}^T \Lambda_{mm} \right\|_2^2} \\ &\text{s.t. } Q \geq 0, \end{aligned} \tag{7}$$

where $\alpha \geq 0$ is a trade-off parameter. With Eq. (7), the within-cluster compactness, the between-cluster diversity and the local cluster structure are preserved simultaneously to generate a more reasonable clustering result.

Optimization

The optimization problem in Eq. (7) is non-convex. However, we can solve it by alternating iteration. The following is the iterative update process.
Step 1: Update Q with fixed Λ.
For each term in Eq. (7), it can be easily obtained that

$$\begin{aligned}&\left\|X^v - X^v Q^T \Lambda Q\right\|_F^2 \\ =&\operatorname{tr}\begin{pmatrix} X^{vT}X^v - 2QX^{vT}X^vQ^T\Lambda \\ +Q^T\Lambda QX^{vT}X^vQ^T\Lambda Q\end{pmatrix},\end{aligned} \tag{8}$$

$$\frac{1}{2}\sum_{i,j}^{n}\|Q_{:,i} - Q_{:,j}\|_2^2 S_{ij}^v = \operatorname{tr}\left(QL^vQ^T\right) \tag{9}$$

and

$$\begin{aligned}&\sum_{l,m}^{C}\left\|X^vQ_{l,:}^T\Lambda_{ll} - X^vQ_{m,:}^T\Lambda_{mm}\right\|_2^2 \\ =&2\operatorname{tr}\left(CQ^T\Lambda^2QX^{vT}X^v - Q^T\Lambda\mathbb{1}\Lambda QX^{vT}X^v\right),\end{aligned} \tag{10}$$

where $L^v = D^v - S^v$ is the Laplacian matrix, D^v is a diagonal matrix with the diagonal element $D_{ii}^v = \sum_{j=1}^n S_{ij}^v$, and $\mathbb{1} \in \mathbb{R}^{C\times C}$ is a matrix with all elements equal to 1. Therefore, the optimization problem in Eq. (7) is equivalent to minimizing

$$J(Q) = \sum_{v=1}^{V}\frac{\operatorname{tr}\begin{pmatrix} X^{vT}X^v - 2QX^{vT}X^vQ^T\Lambda \\ +Q^T\Lambda QX^{vT}X^vQ^T\Lambda Q + \alpha QL^vQ^T\end{pmatrix}}{\operatorname{tr}\left(CQ^T\Lambda^2QX^{vT}X^v - Q^T\Lambda\mathbb{1}\Lambda QX^{vT}X^v\right)}. \tag{11}$$

The partial deviation $J(Q)$ with respect to Q is

$$\begin{aligned}&\frac{\partial J(Q)}{\partial Q} \\ =&\sum_{v=1}^{V}\begin{pmatrix} A^v\begin{pmatrix} 2\Lambda QX^{vT}X^vQ^T\Lambda Q + 2\alpha QL^v \\ +2\Lambda QQ^T\Lambda QX^{vT}X^v - 4\Lambda QX^{vT}X^v\end{pmatrix} \\ -B^v\left(2C\Lambda^2QX^{vT}X^v - 2\Lambda\mathbb{1}\Lambda QX^{vT}X^v\right)\end{pmatrix},\end{aligned} \tag{12}$$

where

$$A^v = \frac{1}{\operatorname{tr}\left(CQ^T\Lambda^2QX^{vT}X^v - Q^T\Lambda\mathbb{1}\Lambda QX^{vT}X^v\right)} \tag{13}$$

and

$$B^v = \frac{\operatorname{tr}\begin{pmatrix} X^{vT}X^v - 2QX^{vT}X^vQ^T\Lambda \\ +Q^T\Lambda QX^{vT}X^vQ^T\Lambda Q + \alpha QL^vQ^T\end{pmatrix}}{\left(\operatorname{tr}\left(CQ^T\Lambda^2QX^{vT}X^v - Q^T\Lambda\mathbb{1}\Lambda QX^{vT}X^v\right)\right)^2}. \tag{14}$$

With the multiplicative update rule (Ding, Li, and Jordan 2010), Q is updated as

$$Q_{ij} \to Q_{ij}\sqrt[4]{\frac{\left(\sum_{v=1}^{V}\begin{pmatrix} A^v\begin{pmatrix}\Lambda Q\left(X^{vT}X^v\right)^-Q^T\Lambda Q \\ +\Lambda QQ^T\Lambda Q\left(X^{vT}X^v\right)^- \\ +2\Lambda Q\left(X^{vT}X^v\right)^+ + \alpha QS^v\end{pmatrix} \\ +B^v\begin{pmatrix} C\Lambda^2Q\left(X^{vT}X^v\right)^+ \\ +\Lambda\mathbb{1}\Lambda Q\left(X^{vT}X^v\right)^-\end{pmatrix}\end{pmatrix}\right)_{ij}}{\left(\sum_{v=1}^{V}\begin{pmatrix} A^v\begin{pmatrix}\Lambda Q\left(X^{vT}X^v\right)^+Q^T\Lambda Q \\ +\Lambda QQ^T\Lambda Q\left(X^{vT}X^v\right)^+ \\ +2\Lambda Q\left(X^{vT}X^v\right)^- + \alpha QD^v\end{pmatrix} \\ +B^v\begin{pmatrix} C\Lambda^2Q\left(X^{vT}X^v\right)^- \\ +\Lambda\mathbb{1}\Lambda Q\left(X^{vT}X^v\right)^+\end{pmatrix}\end{pmatrix}\right)_{ij}}} \tag{15}$$

where

$$\left(X^{vT}X^v\right)^+ = \frac{\left|X^{vT}X^v\right| + X^{vT}X^v}{2}, \tag{16}$$

$$\left(X^{vT}X^v\right)^- = \frac{\left|X^{vT}X^v\right| - X^{vT}X^v}{2} \tag{17}$$

and

$$X^{vT}X^v = \left(X^{vT}X^v\right)^+ - \left(X^{vT}X^v\right)^-. \tag{18}$$

Step 2: Update Λ with fixed Q.
Λ is a diagonal matrix and each diagonal element is updated as

$$\Lambda_{cc} = 1/\sum_{i=1}^{n}Q_{ci} \tag{19}$$

We summarize the iterative update of DFMKLS in Algorithm 1. In the beginning, Q can be simply initialized by random matrix. Here, to obtain relatively stable result, we initialize it by the result of spectral clustering. Specifically, the summation of Laplacian matrices from V views, i.e. $\sum_{v=1}^{V} L^v$, is eigen-decomposed and the eigen-vectors are used to perform K-means to obtain the cluster assignment matrix $U \in \mathbb{R}^{C\times n}$, where $U_{ci} = 1$, if x_i belongs to the c-th cluster, and $U_{ci} = 0$, otherwise. Then Q is initialized as $Q = U + E/10$, where $E \in \mathbb{R}^{C\times n}$ is a matrix with all elements equal to 1.

Convergence

On one hand, the value of objective function of DFMKLS in Eq. (7) is non-negative, which means its lower bound is 0.

Algorithm 1: DFMKLS algorithm

Input:
Multi-view data $X^v|_{v=1}^V$ from V views; Trade-off parameter α; Neighbor parameter K.
Output:
Clustering result.
1: Initialize Q.
2: Construct the adjacent matrix $S^v|_{v=1}^V$ using Eq. (5), and compute the diagonal matrix $D^v|_{v=1}^V$ by $D_{ii}^v = \sum_{j=1}^n S_{ij}^v$.
3: **repeat**
4: With fixed Q, update Λ using Eq. (19).
5: With fixed Λ , update Q using Eq. (15).
6: **until** convergence.
7: The label of the ith data point x_i is $\arg\max_c Q_{ci}$.

On the other hand, in the iterative update, with multiplicative update rule, the value of the objective function is monotonically decreasing. Therefore, DFMKLS algorithm must converge to a local minimum.

Computational Complexity

In each iteration, DFMKLS algorithm includes two steps, i.e., Eq. (15) and Eq. (19). It is obviously that the computational complexity of DFMKLS is determined by Eq. (15) and mainly caused by matrix multiplication. There are several matrix multiplication operations in Eq. (15). Because of $C < n$, we can easily obtain that, the computational complexities of two most time consuming operations are $O(Cn^2)$ and $O(\sum_{v=1}^V d^v n^2)$, respectively. Therefore, the computational complexity of DFMKLS is $O(T(Cn^2 + \sum_{v=1}^V d^v n^2))$, where T is the number of iteration.

Experiments

The effectiveness of DFMKLS is evaluated by clustering experiments on four multi-view datasets. Experiments are performed in MATLAB R2014a on a computer with 13th Gen Intel(R) Core(TM) i9-13900K 3.00 GHz CPU, 64.0GB RAM and Windows11 operating system. Accuracy, F-score, Normalized Mutual Information (NMI) and Precision are employed to measure the clustering performance. In the experiments, DFMKLS is compared with the following multi-view clustering methods:
(1) BSV: Best Single View. K-means is performed in each single view and the best clustering result is adopted.
(2) Concat: The features of all the views are concatenated first and K-means is performed on the combined features.
(3) MultiNMF (Liu et al. 2013): Nonnegative Matrix Factorization based Multi-view clustering.
(4) Spec-Pair (Kumar, Rai, and Daume 2011): Pairwise co-regularization multi-view Spectral clustering. The eigenvectors of different views have pairwise similarity in Spec-Pair.
(5) Spec-Cent (Kumar, Rai, and Daume 2011): Centroid based co-regularization multi-view Spectral clustering. The eigenvectors of different views tend towards a common consensus in Spec-Cent.
(6) RMKMC (Cai, Nie, and Huang 2013): Robust Multi-view K-Means Clustering.
(7) MVASM (Han et al. 2020): Multi-View clustering with Adaptive Sparse Memberships and weight allocation.
(8) EMKMC (Yang et al. 2023): Efficient Multi-view K-Means Clustering method with multiple anchor graphs.
(9) GMC (Wang, Yang, and Liu 2019): Graph-based Multi-view Clustering.
(10) CGL (Li et al. 2022): Consensus Graph Learning for multi-view clustering.
(11) FMK: Fuzzy Multi-view K-means. Its objective is presented in Eq. (3)
(12) DFMK: Discriminatively Fuzzy Multi-view K-means. Its objective is presented in Eq. (4).

Dataset	Size	Class	View	Dimension
3Sources	169	6	3	3068/3631/3560
BBC	685	5	4	4659/4633/ 4665/4684
WebKB	1051	2	2	1840/3000
NUS_WIDE	11280	10	3	500/73/128

Table 1: Statistic of Datasets

Datasets

Experiments are conducted on 3sources, BBC, WebKB and NUS_WIDE datasets. Table 1 summarizes four datasets, and their details are presented as follows.
(1) 3Sources (Greene and Cunningham 2009): It consists of 416 news stories of 6 topical labels, which are collected from 3 sources, i.e., BBC, Reuters, and The Guardian. Each source can be seen as one view of a story. In our experiments, we select 169 stories reported in all the 3 sources. The dimensions of three views are 3068, 3631 and 3560, respectively.
(2) BBC (Greene and Cunningham 2006): It consists of 2225 documents belonging to 5 classes, which are collected from BBC news. Four views are obtained by segmentation of the documents. The dimensions of 4 views are 4659, 4633, 4665 and 4684, respectively. Not all the documents have 4 views. In our experiments, we select 685 documents with 4 complete views.
(3) WebKB (Craven et al. 2000): It consists of webpages collected from computer science departments of university, which contains 8,280 documents in 7 categories. In our experiments, we select 1,051 documents in the top two most popular categories. Each document has 2 views, and their dimensions are 1840 and 3000, respectively.
(4) NUS_WIDE (Chua et al. 2009): It consists of 269648 real-world web images. In our experiments, we select 11280 images of mammal, belonging to 10 classes. Each image has 3 views, including 500D bag of words based on SIFT descriptions, 73D edge direction histogram and 128D wavelet texture feature.

Clustering Performance Analysis

In the comparison experiments, for RMKMC, MVASM, EMKMC, GMC, FMK, DFMK and DFMKLS, the cluster-

Method\Index	Accuracy	F-score	NMI	Precision
BSV	0.6183	0.6040	0.6186	0.6608
Concat	0.6251	0.6145	0.6448	0.6748
MultiNMF	0.5148	0.4236	0.4721	0.3846
Spec-Pair	0.5346	0.4370	0.4332	0.4854
Spec-Cent	0.5308	0.4434	0.4395	0.4972
RMKMC	0.4320	0.3466	0.3204	0.3414
MVASM	0.5621	0.4555	0.3121	0.3420
EMKMC	0.5858	0.5324	0.4702	0.6266
GMC	0.6923	0.6047	0.6216	0.4844
CGL	0.6393	0.6166	0.6995	0.7127
FMK	0.7396	0.6866	0.6818	0.6912
DFMK	0.7396	0.7091	0.7066	0.6800
DFMKLS	**0.8107**	**0.7483**	**0.7096**	**0.7598**

Table 2: Cluster Validity Index on 3Sources dataset

Method\Index	Accuracy	F-score	NMI	Precision
BSV	0.5856	0.4948	0.4676	0.4659
Concat	0.6674	0.6057	0.6082	0.5973
MultiNMF	0.4804	0.3817	0.3335	0.2869
Spec-Pair	0.3056	0.3276	0.0379	0.2312
Spec-Cent	0.3092	0.3314	0.0381	0.2316
RMKMC	0.6015	0.4833	0.4357	0.4839
MVASM	0.3314	0.3757	0.0197	0.2339
EMKMC	0.6584	0.5673	0.5020	0.6111
GMC	0.6934	0.6333	0.5628	0.5012
CGL	0.8068	0.7390	0.6972	0.7373
FMK	0.5547	0.5477	0.5148	0.4864
DFMK	0.6496	0.6627	0.5784	0.5708
DFMKLS	**0.8672**	**0.8159**	**0.7341**	**0.8054**

Table 3: Cluster Validity Index on BBC dataset

Method\Index	Accuracy	F-score	NMI	Precision
BSV	0.8754	0.8539	0.3882	0.7666
Concat	0.8844	0.8777	0.5546	0.8432
MultiNMF	0.8647	0.8454	0.3465	0.7515
Spec-Pair	0.9134	0.8891	0.5322	0.8242
Spec-Cent	0.9134	0.8891	0.5322	0.8242
RMKMC	0.9515	0.9319	0.6618	0.9039
MVASM	0.9439	0.9160	0.6785	0.9564
EMKMC	0.9305	0.9061	0.5588	0.8671
GMC	0.7764	0.7867	0.0017	0.6596
CGL	0.5271	0.5688	0.0040	0.6589
FMK	0.8687	0.8434	0.3122	0.7682
DFMK	0.9467	0.9210	0.6591	0.9492
DFMKLS	**0.9610**	**0.9416**	**0.7276**	**0.9638**

Table 4: Cluster Validity Index on WebKB dataset

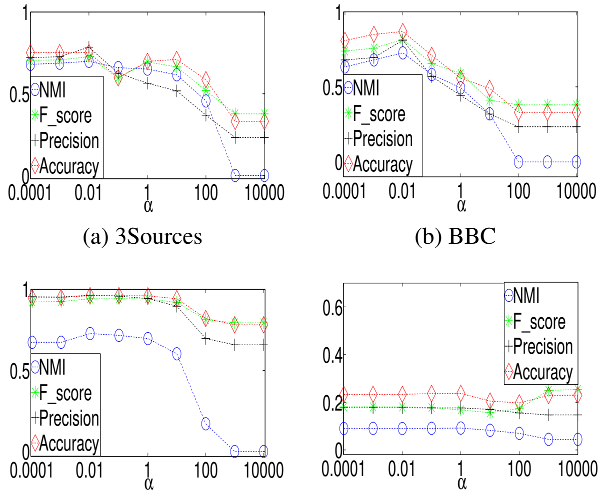
Method\Index	Accuracy	F-score	NMI	Precision
BSV	0.2194	0.1693	0.0791	0.1716
Concat	0.2217	0.1571	0.0812	0.1696
MultiNMF	N/A	N/A	N/A	N/A
Spec-Pair	0.2172	**0.2491**	0.0102	0.1423
Spec-Cent	0.2171	0.2490	0.0099	0.1423
RMKMC	0.2191	0.1548	0.0720	0.1740
MVASM	0.2098	0.1521	0.0699	0.1704
EMKMC	0.1664	0.1342	0.0404	0.1552
GMC	0.2214	0.2444	0.0402	0.1452
CGL	0.2290	0.1590	0.0890	0.1789
FMK	0.2252	0.1899	0.0764	0.1675
DFMK	0.2331	0.1807	0.0905	0.1770
DFMKLS	**0.2453**	0.1886	**0.0966**	**0.1817**

Table 5: Cluster Validity Index on NUS_WIDE dataset

ing results can be obtained directly. For MultiNMF, Spec-Pair, Spec-Cent and CGL, new features are firstly extracted and then K-means is performed on these features. We perform K-means 20 times and report the average value of Accuracy, F-score, NMI and Precision. The parameters of the compared methods are set as the recommendations of the original papers. For DFMKLS, the trade-off parameter α is set as 0.01, and the neighbor parameter K is set as 10. Parameter setting will be analyzed in the following section. From Table 2, Table 3 and Table 4, it can be seen that, on 3Sources, BBC and WebKB datasets, DFMKLS all obtains the best clustering performance, no matter which cluster validity index is adopted. From Table 5, it can be found that, on NUS_WIDE dataset, our DFMKLS has the highest accuracy, F_score and precision. Besides, on all the datasets, DFMK almost performs better than FMK by introducing the discriminant information. By combining the global information and the local information, DFMKLS improves the clustering performance of DFMK further.

Parameter Analysis

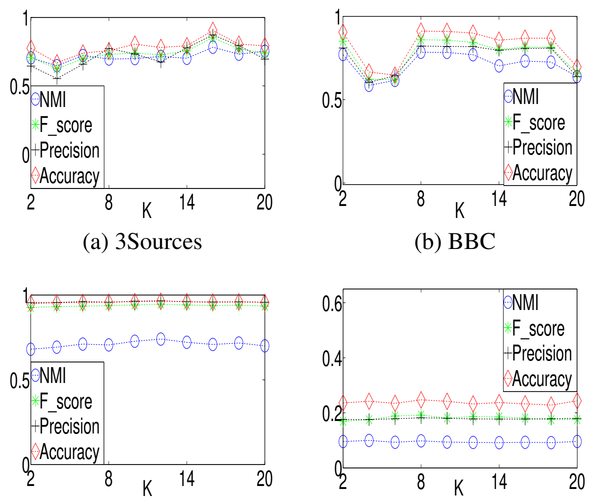
To find the optimal parameters for DFMKLS, we let the parameters α and K vary in a wide range, and evaluate the performances of DFMKLS with different parameters. Experiments are performed on 3Sources, BBC, WebKB and NUS_WIDE datasets. For two parameters, we fix one parameter and change the other one. Specifically, K is fixed as 10 and α is selected from the set $\{10^{-4}, 10^{-3}, \ldots, 10^{3}, 10^{4}\}$. Then, α is fixed as 0.01, and K is selected from the set $\{2, 4, \ldots, 18, 20\}$. Fig.1 and Fig.2 show the values of four cluster validity indexes under different α and K, respectively. From Fig.1, we can see that, too big α leads to the dramatic decrease of clustering validity indexex, especially for NMI. It indicates that, in DFMKLS model, the local structure preserving term cannot occupy too high proportion, the model should focus more on the objective of discriminatively fuzzy multi-view K-means. On all the datasets, DFMKLS almost performs best with $\alpha = 0.01$. From Fig.2, we can see that, on 3sources and BBC datasets, DFMKLS has poor performances with too big or too small K and performs well with $K \in [8, 18]$. K has small effect on the clustering results on WebKB and NUS_WIDE datasets. On all the datasets, DFMKLS obtains relatively good clustering performance with $K = 10$. According to the result of parameter analysis, in our experiments, the parameter α is set as 0.01, and K is set as 10.



(a) 3Sources (b) BBC

(c) WebKB (d) NUS_WIDE

Figure 1: Cluster Validity Index of DFMKLS versus parameter α



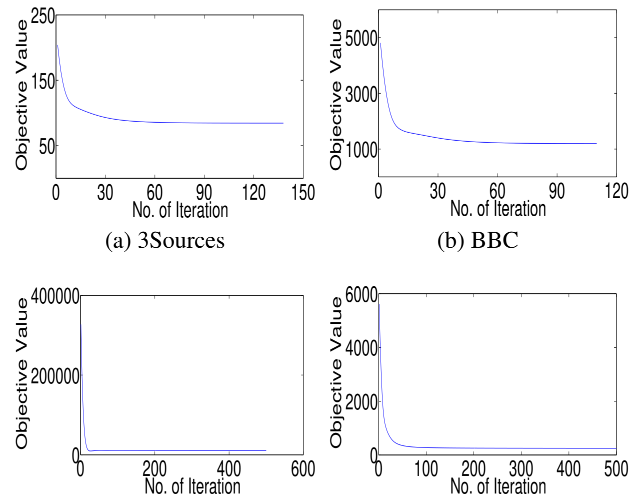
(a) 3Sources (b) BBC

(c) WebKB (d) NUS_WIDE

Figure 2: Cluster Validity Index of DFMKLS versus parameter K

Convergence Analysis

We evaluate the convergence of DFMKLS algorithm on four multi-view datasets. The value of objective function of DFMKLS is recorded in each iteration and displayed in Fig.3. From the figure, we can find that, on all the datasets, the objective value declines rapidly in the beginning of the iteration. After no more than 100 iterations, the curves of the objective value become smooth and steady, and then the algorithm converges. Besides, for larger datasets, i.e., WebKB and NUS_WIDE, DFMKLS needs more iterations to reach convergence.



(a) 3Sources (b) BBC

(c) WebKB (d) NUS_WIDE

Figure 3: The value of objective function of DFMKLS versus iteration

Conclusion

This paper develops a novel multi-view K-means clustering method called DFMKLS. DFMKLS takes into account the compactness of each cluster and the separability of different clusters simultaneously, where the within-cluster scatter is minimized and the between-cluster scatter is maximized. In DFMKLS, a data point does not strictly belong to one cluster. The probability that it belongs to each cluster is calculated and it is assigned to the cluster with the maximum probability. DFMKLS also preserves the local structure of the data to improve the clustering performance. We conduct experiments on four public multi-view datasets and compare DFMKLS with state-of-the-art multi-view clustering methods. Experimental results demonstrate the effectiveness of DFMKLS. In addition, the optimal parameters for DFMKLS are established by parameter analysis experiments, and the convergence of the algorithm is also proved in experiments.

Acknowledgments

This work is supported by National Natural Science Foundation of China (62076096), Shanghai Pujiang Program (22PJD029) and Shanghai Municipal Project (20511100900).

References

Asano, Y.; Patrick, M.; Rupprecht, C.; and Vedaldi, A. 2020. Labelling unlabelled videos from scratch with multi-modal self-supervision. *Proceedings of Advances in Neural Information Processing Systems*, 33: 4660–4671.

Cai, X.; Nie, F.; and Huang, H. 2013. Multi-view k-means clustering on big data. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2598–2604.

Cao, X.; Zhang, C.; Fu, H.; Liu, S.; and Zhang, H. 2015. Diversity-induced multi-view subspace clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 586–594.

Chen, C.; Wang, Y.; Hu, W.; and Zheng, Z. 2020. Robust multi-view k-means clustering with outlier removal. *Knowledge-Based Systems*, 210: 106518.

Cheng, M.; Jing, L.; and Ng, M. K. 2018. Tensor-based low-dimensional representation learning for multi-view clustering. *IEEE Transactions on Image Processing*, 28(5): 2399–2414.

Cheung, Y.-M. 2003. K-means: a new generalized k-means clustering algorithm. *Pattern Recognition Letters*, 24(15): 2883–2893.

Chua, T.-S.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; and Zheng, Y.-T. 2009. NUS-WIDE: a real-world web image database from national university of singapore. In *Proceedings of ACM Conference on Image and Video Retrieval*, 1–9.

Craven, M.; DiPasquo, D.; Freitag, D.; McCallum, A.; Mitchell, T.; Nigam, K.; and Slattery, S. 2000. Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence*, 118(1): 69–113.

Ding, C. H.; Li, T.; and Jordan, M. I. 2010. Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1): 45–55.

Greene, D.; and Cunningham, P. 2006. Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering. In *Proceedings of International Conference on Machine learning*, 377–384.

Greene, D.; and Cunningham, P. 2009. A Matrix Factorization Approach for Integrating Multiple Data Views. In *Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 423–438.

Han, J.; Xu, J.; Nie, F.; and Li, X. 2020. Multi-view k-means clustering with adaptive sparse memberships and weight allocation. *IEEE Transactions on Knowledge and Data Engineering*, 34(2): 816–827.

Kumar, A.; Rai, P.; and Daume, H. 2011. Co-regularized multi-view spectral clustering. *Proceedings of Advances in Neural Information Processing Systems*, 24.

Li, R.; Zhang, C.; Fu, H.; Peng, X.; Zhou, T.; and Hu, Q. 2019. Reciprocal multi-layer subspace learning for multi-view clustering. In *Proceedings of the IEEE International Conference on Computer Vision*, 8172–8180.

Li, Z.; Tang, C.; Liu, X.; Zheng, X.; Zhang, W.; and Zhu, E. 2022. Consensus graph learning for multi-view clustering. *IEEE Transactions on Multimedia*, 24: 2461–2472.

Liang, W.; Liu, X.; Zhou, S.; Liu, J.; Wang, S.; and Zhu, E. 2022. Robust graph-based multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 7462–7469.

Liu, J.; Cao, F.; Gao, X.-Z.; Yu, L.; and Liang, J. 2020a. A cluster-weighted kernel k-means method for multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 4860–4867.

Liu, J.; Wang, C.; Gao, J.; and Han, J. 2013. Multi-view clustering via joint nonnegative matrix factorization. In *Proceedings of the SIAM International Conference on Data Mining*, 252–260.

Liu, X.; Li, M.; Tang, C.; Xia, J.; Xiong, J.; Liu, L.; Kloft, M.; and Zhu, E. 2020b. Efficient and effective regularized incomplete multi-view clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(8): 2634–2646.

MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of Symposium on Mathematical Statistics and Probability*, 281–297.

Nie, F.; Li, J.; Li, X.; et al. 2017. Self-weighted multiview clustering with multiple graphs. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2564–2570.

Peng, L.; Yang, Y.; Wang, Z.; Huang, Z.; and Shen, H. T. 2020. Mra-net: Improving vqa via multi-modal relation attention network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1): 318–329.

Sun, S. 2013. A survey of multi-view machine learning. *Neural Computing and Applications*, 23: 2031–2038.

Wang, H.; Yang, Y.; and Liu, B. 2019. GMC: Graph-based multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 32(6): 1116–1129.

Wang, Z.; Li, Z.; Wang, R.; Nie, F.; and Li, X. 2020. Large graph clustering with simultaneous spectral embedding and discretization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12): 4426–4440.

Xu, J.; Han, J.; and Nie, F. 2016. Discriminatively embedded k-means for multi-view clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5356–5364.

Xu, J.; Han, J.; Nie, F.; and Li, X. 2017. Re-weighted discriminatively embedded k-means for multi-view clustering. *IEEE Transactions on Image Processing*, 26(6): 3016–3027.

Xu, J.; Tang, H.; Ren, Y.; Peng, L.; Zhu, X.; and He, L. 2022. Multi-Level feature learning for contrastive multi-View clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 16051–16060.

Xu, R.; and Wunsch, D. 2005. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3): 645–678.

Yang, B.; Zhang, X.; Li, Z.; Nie, F.; and Wang, F. 2023. Efficient multi-view k-means clustering with multiple anchor graphs. *IEEE Transactions on Knowledge and Data Engineering*, 35(7): 6887–6900.

Ye, Y.; Liu, X.; Yin, J.; and Zhu, E. 2016. Co-regularized kernel k-means for multi-view clustering. In *Proceedings of International Conference on Pattern Recognition*, 1583–1588.

Zhang, Z.; Liu, L.; Shen, F.; Shen, H. T.; and Shao, L. 2018. Binary multi-view clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7): 1774–1782.