# Discriminatively Embedded K-Means for Multi-view Clustering

Jinglin Xu[1], Junwei Han[1], Feiping Nie[2]
[1]School of Automation, [2]School of Computer Science and Center for OPTIMAL,
Northwestern Polytechnical University
Xi'an, 710072, P. R. China
xujignlinlove, junweihan2010, feipingnie@gmail.com

## Abstract

*In real world applications, more and more data, for example, image/video data, are high dimensional and represented by multiple views which describe different perspectives of the data. Efficiently clustering such data is a challenge. To address this problem, this paper proposes a novel multi-view clustering method called Discriminatively Embedded K-Means (DEKM), which embeds the synchronous learning of multiple discriminative subspaces into multi-view K-Means clustering to construct a unified framework, and adaptively control the intercoordinations between these subspaces simultaneously. In this framework, we firstly design a weighted multi-view Linear Discriminant Analysis (LDA), and then develop an unsupervised optimization scheme to alternatively learn the common clustering indicator, multiple discriminative subspaces and weights for heterogeneous features with convergence. Comprehensive evaluations on three benchmark datasets and comparisons with several state-of-the-art multi-view clustering algorithms demonstrate the superiority of the proposed work.*

## 1. Introduction

As a fundamental technique in machine learning, pattern recognition and computer vision fields, clustering is to assign data of similar patterns into the same cluster and reflect the intrinsic structure of the data. In past decades, a variety of classical clustering algorithms such as K-Means Clustering [15] and Spectral Clustering [24, 25] have been invented.

In recent years, due to the rapid development of information technology, we are often confronted with data represented by heterogeneous features. These features are generated by using various feature construction ways. One good example is image/video data. A large number of different visual descriptors, such as SIFT [20], HOG [7], LBP [22], GIST [23], CMT [30] and CENT [29], have been proposed to characterize the rich content of image/video data from different perspectives. Each type of features may capture the specific information about the visual data. To cluster these data, one challenge is how to integrate the strengths of various heterogeneous features by exploring the rich information among them, which certainly can lead to more accurate and robust clustering performance than by using each individual type of features.

Nowadays, the data is often represented by very high dimensional features, which renders another challenge for the clustering. A number of earlier efforts have been devoted to addressing these two challenges. Focusing on one challenge that data is very high dimensional, many dimensionality reduction-based clustering methods [12, 10, 26, 16] have been developed, which mostly concern simultaneous subspace selection by LDA and clustering. These methods generally are more appropriate for single-view data clustering. Although they may be extended to multi-view data clustering task by simply concatenating different views as input or integrating each view of clustering results to the final results, these extended methods still cannot achieve the satisfactory performance due to the lack of intercoordination and complementation between different views during clustering.

Focusing on another challenge that data is represented by multi-view, a school of unsupervised multi-view clustering methods have been presented. Although these methods can achieve interactions among heterogeneous features, there still exist some problems regarding heavy computational complexity or curse of dimensionality. Most of these methods can be roughly classified into two categories: Multi-View K-Means Clustering (MVKM) and Multi-View Spectral Clustering (MVSC). Many MVSC approaches essentially extend the Spectral Clustering from single view to multiple views and are mainly based on similarity graphs or matrices. Although this kind of multi-view clustering algorithms [8, 32, 21, 18, 19, 4, 14, 27, 5] can achieve encouraging performance, they still have two main drawbacks. On the one hand, the construction of the similarity graph for high dimensional data is a heavy work because many

factors must be considered, such as the choice of similarity function and the type of similarity graph. This heavy work may greatly affect the final clustering performance. On the other hand, MVSC algorithms generally need to build proper similarity graph for each view. The more the number of different views, the more complex constructing similarity graphs will be. Thus, MVSC algorithms cannot effectively tackle high-dimensional multi-view data clustering.

Different from MVSC algorithms, MVKM approaches are more superior to deal with high-dimensional data because they do not need to construct a similarity graph for each view. This kind of methods is originally derived from the **G**-orthogonal non-negative matrix factorization (NMF) which is equivalent to relaxed K-Means clustering (RKM) [9]. Recently, *Cai et al*. [3] proposed the robust multi-view K-Means clustering (RMVKM) by using $\ell_{2,1}$-norm [11] to replace the $\ell_2$-norm and learning individual weight for each view. However, RMVKM was performed in the original feature space without any discriminative subspace learning mechanism that may render curse of dimensionality when dealing with multi-view and high dimensional data. In addition, although the work in [31] also extended the model from [10] to the multi-view case, they sum the scatter matrices and produce a separate cluster assignment for each view, which is quite different from the proposed method.

According to above mentioned analysis, both directly extending single-view to multi-view and existing multi-view algorithms are far from thoroughly addressing the multi-view clustering issue. In this paper, we propose a novel unsupervised multi-view scheme aiming to address above two challenges. The proposed method DEKM embeds the synchronous learning of multiple discriminative subspaces into multi-view K-Means clustering to construct a unified framework, and adaptively control the intercoordinations between different views simultaneously.

The highlights of DEKM method are in two aspects. Firstly, learning multiple discriminative subspaces is fulfilled synchronously. Under this unified and embedded framework, DEKM realizes the intercoordination of these subspaces and further makes them complement each other. Secondly, DEKM develops an intertwined and iterative optimization instead of just applying existing methods in an iterative manner, which not only maintains the relative independency on different discriminative subspaces, but also keeps the consistency of clustering results of multiple views. This multi-view extension is the first work among the earliest efforts to sum the clustering objectives via a weighted way. These are quite different from several recent works. Comprehensive evaluations on several benchmark image datasets and comparisons with some state-of-the-art multi-view clustering approaches demonstrate the efficiency and superiority of DEKM.

# 2. The proposed framework

## 2.1. Formulation

According to [17], the trace ratio LDA for single-view was defined as follows:

$$\mathbf{W}=\underset{\mathbf{W}^T\mathbf{W}=\mathbf{I}_m}{\arg\max}\frac{Tr(\mathbf{W}^T\mathbf{S}_B\mathbf{W})}{Tr(\mathbf{W}^T\mathbf{S}_W\mathbf{W})} \tag{1}$$

where $\mathbf{W}\in\mathbb{R}^{d\times m}$ denotes the projection matrix which is a set of orthogonal and normalized vectors. It enables to reduce the dimensionality from $d$ to $m$. $\mathbf{S}_B$ and $\mathbf{S}_W$ denote the between-class scatter matrix and the within-class scatter matrix, respectively.

Suppose that $\mathbf{X}\in\mathbb{R}^{d\times N}$ is the data matrix with $N$ samples and $d$-dimension after centralization and $\mathbf{G}\in\mathbb{R}^{N\times C}$ is the clustering indicator matrix where each row of $\mathbf{G}$ denotes the clustering indicator vector for each sample, and $C$ is the number of clusters. $\mathbf{G}_{ic}=1(i=1,...,N;c=1,...,C)$ if the $i$-th sample belongs to the $c$-th class and $\mathbf{G}_{ic}=0$ otherwise. Using $\mathbf{G}$, $\mathbf{S}_B$ and $\mathbf{S}_W$ can be rewritten as:

$$\begin{aligned}\mathbf{S}_B&=\mathbf{X}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{X}^T\\ \mathbf{S}_W&=\mathbf{X}\mathbf{X}^T-\mathbf{X}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{X}^T\end{aligned} \tag{2}$$

Because of $\mathbf{S}_T=\mathbf{S}_B+\mathbf{S}_W$, (1) is equivalent to the following problem:

$$\mathbf{W}=\underset{\mathbf{W}^T\mathbf{W}=\mathbf{I}_m}{\arg\max}\frac{Tr(\mathbf{W}^T\mathbf{S}_B\mathbf{W})}{Tr(\mathbf{W}^T\mathbf{S}_T\mathbf{W})} \tag{3}$$

We know that (3), as a supervised method, can seek a discriminative subspace to separate different classes maximally. Recently, the combination of dimensionality reduction and clustering has become a hot issue [12, 10, 26, 16]. However, those methods are only designed for single-view issue. In this paper, we firstly design a weighted multi-view LDA and then develop an unsupervised optimization scheme to solve this multi-view framework.

Given $M$ types of heterogeneous features, $k=1,2,...,M$, we suppose $\mathbf{X}_k\in\mathbb{R}^{d_k\times N}$ as the data matrix for the $k$-th view. Referring to the definition of trace ratio LDA, we propose that, for two $d_k\times d_k$ positive semi-definite matrices $\mathbf{S}_B^k$ and $\mathbf{S}_T^k$, the weighted multi-view trace ratio LDA can be defined as finding $M$ differen projection matrices $\mathbf{W}_k|_{k=1}^M$ respectively:

$$\mathbf{W}_k|_{k=1}^M=\underset{\mathbf{W}_k^T\mathbf{W}_k=\mathbf{I}_{m_k}|_{k=1}^M}{\arg\max}\sum_{k=1}^{M}(\alpha_k)^\gamma\frac{Tr(\mathbf{W}_k^T\mathbf{S}_B^k\mathbf{W}_k)}{Tr(\mathbf{W}_k^T\mathbf{S}_T^k\mathbf{W}_k)} \tag{4}$$

where $\mathbf{W}_k$ denotes the projection matrix which reduces the dimensionality from $d_k$ to $m_k$ in the $k$-th view. $\alpha_k$ is the weight for each view and $\gamma$ is the parameter to control the

weights distribution. $\mathbf{S}_B^k$ and $\mathbf{S}_T^k$ denote the $\mathbf{S}_B$ and $\mathbf{S}_T$ in the $k$-th view, respectively:

$$\mathbf{S}_B^k = \mathbf{X}_k\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{X}_k^T, \mathbf{S}_T^k = \mathbf{X}_k\mathbf{X}_k^T \tag{5}$$

It is apparent that the weighted multi-view LDA, *i.e*. (4), is still supervised. However, in the real applications, labeling data is very expensive. Without any label information, we know neither projection matrices $\mathbf{W}_k|_{k=1}^M$ nor clustering indicator matrix $\mathbf{G}$ of (4), which is adverse for doing high-dimensional clustering. Thus, we propose an unsupervised optimization scheme to solve the following weighted multi-view LDA:

$$\begin{aligned}&\max_{\substack{\mathbf{w}_k|_{k=1}^M,\\ \alpha_k|_{k=1}^M,\mathbf{G}}} \sum_{k=1}^M(\alpha_k)^\gamma\left[\frac{Tr(\mathbf{W}_k^T\mathbf{X}_k\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{X}_k^T\mathbf{W}_k)}{Tr(\mathbf{W}_k^T\mathbf{S}_T^k\mathbf{W}_k)}-1\right]\\ &s.t.\mathbf{W}_k^T\mathbf{W}_k=\mathbf{I}_{m_k}|_{k=1}^M, \mathbf{G}\in\mathbf{Ind}, \sum_{k=1}^M\alpha_k=1, \alpha_k\geq 0\end{aligned} \tag{6}$$

where **Ind** is a set of clustering indicator matrices.

## 2.2. Optimization

The key difficulty of solving (6) is that (6) has become an unsupervised complex matter. In other words, the numerator of (6), $\mathbf{X}_k\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{X}_k^T$, actually $\mathbf{S}_B^k$, is closely related to $\mathbf{G}$. However, $\mathbf{W}_k|_{k=1}^M$, $\alpha_k|_{k=1}^M$ and $\mathbf{G}$ are unknown. To simultaneously obtain these variables in a better way, we offer the Theorem 1 to transform (6) into a more tractable framework (7) which is the proposed method DEKM. Actually, $\mathbf{W}_k|_{k=1}^M$ are not decoupled in (7) since $\mathbf{G}$ is also a variable to be optimized.

**Theorem 1.** *Solving (6) is equivalent to solving the following objective function:*

$$\begin{aligned}&\min_{\substack{\mathbf{w}_k|_{k=1}^M,\\ \alpha_k|_{k=1}^M,\mathbf{G}}} \sum_{k=1}^M(\alpha_k)^\gamma\frac{||\mathbf{W}_k^T\mathbf{X}_k-\mathbf{F}_k\mathbf{G}^T||_F^2}{Tr(\mathbf{W}_k^T\mathbf{S}_T^k\mathbf{W}_k)}\\ &s.t.\mathbf{W}_k^T\mathbf{W}_k=\mathbf{I}_{m_k}|_{k=1}^M, \mathbf{G}\in\mathbf{Ind}, \sum_{k=1}^M\alpha_k=1, \alpha_k\geq 0\end{aligned} \tag{7}$$

*Proof.* Obviously, using the properties of matrix trace, (7) can be rewritten as the following formula:

$$\begin{aligned}&\min_{\substack{\mathbf{w}_k|_{k=1}^M,\\ \alpha_k|_{k=1}^M,\mathbf{G}}} \sum_{k=1}^M(\alpha_k)^\gamma\frac{Tr\left[(\mathbf{W}_k^T\mathbf{X}_k-\mathbf{F}_k\mathbf{G}^T)^T(\mathbf{W}_k^T\mathbf{X}_k-\mathbf{F}_k\mathbf{G}^T)\right]}{Tr(\mathbf{W}_k^T\mathbf{S}_T^k\mathbf{W}_k)}\\ &=\min_{\substack{\mathbf{w}_k|_{k=1}^M,\\ \alpha_k|_{k=1}^M,\mathbf{G}}} \sum_{k=1}^M(\alpha_k)^\gamma\frac{\begin{bmatrix}Tr(\mathbf{X}_k^T\mathbf{W}_k\mathbf{W}_k^T\mathbf{X}_k)-2Tr(\mathbf{F}_k^T\mathbf{W}_k^T\mathbf{X}_k\mathbf{G})\\ +Tr(\mathbf{F}_k\mathbf{G}^T\mathbf{G}\mathbf{F}_k^T)\end{bmatrix}}{Tr(\mathbf{W}_k^T\mathbf{S}_T^k\mathbf{W}_k)}\end{aligned} \tag{8}$$

Due to solving the minimum, we get its derivative with respect to $\mathbf{F}_k$. Ignoring irrelevant terms and using the rules of matrix derivative, we can obtain:

$$\mathbf{F}_k = \mathbf{W}_k^T\mathbf{X}_k\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1} \tag{9}$$

Excitingly, $\mathbf{F}_k\in\mathbb{R}^{m_k\times C}$ is the cluster centroid in discriminative subspace for the $k$-th view. Substituting (9) into (8), there is:

$$\begin{aligned}&\min_{\substack{\mathbf{w}_k|_{k=1}^M,\\ \alpha_k|_{k=1}^M,\mathbf{G}}} \sum_{k=1}^M(\alpha_k)^\gamma\left[1-\frac{Tr(\mathbf{W}_k^T\mathbf{X}_k\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{X}_k^T\mathbf{W}_k)}{Tr(\mathbf{W}_k^T\mathbf{S}_T^k\mathbf{W}_k)}\right]\\ \Leftrightarrow &\max_{\substack{\mathbf{w}_k|_{k=1}^M,\\ \alpha_k|_{k=1}^M,\mathbf{G}}} \sum_{k=1}^M(\alpha_k)^\gamma\left[\frac{Tr(\mathbf{W}_k^T\mathbf{X}_k\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{X}_k^T\mathbf{W}_k)}{Tr(\mathbf{W}_k^T\mathbf{S}_T^k\mathbf{W}_k)}-1\right]\end{aligned} \tag{10}$$

Therefore, solving (6) is equivalent to solving (7). □

Further, we decompose (7) into three subproblems and solve them via alternate iteration method.

**Step1: Solving G when $\mathbf{W}_k|_{k=1}^M$, $\mathbf{F}_k|_{k=1}^M$ and $\alpha_k|_{k=1}^M$ are fixed.**

Obtaining $\mathbf{G}$ via a weighted multi-view K-Means clustering is an unsupervised learning stage. The clustering indicator matrix $\mathbf{G}$ is unknown and we search the optimal solution of $\mathbf{G}$ among multiple low-dimensional discriminative subspaces.

We separate $\mathbf{X}_k$ and $\mathbf{G}$ into independent vectors respectively. Then (7) can be replaced by the following problem:

$$\begin{aligned}&\min_{\mathbf{G}}\sum_{k=1}^M(\alpha_k)^\gamma||\mathbf{W}_k^T\mathbf{X}_k-\mathbf{F}_k\mathbf{G}^T||_F^2\\ &=\min_{\mathbf{G}}\sum_{i=1}^N\sum_{k=1}^M(\alpha_k)^\gamma||\mathbf{W}_k^T\mathbf{x}_k^i-\mathbf{F}_k\mathbf{g}_i||_2^2\\ &s.t.\mathbf{G}\in\mathbf{Ind}, \mathbf{g}_i\in\mathbf{G}, g_{ic}\in\{0,1\}, \sum_{c=1}^C g_{ic}=1\end{aligned} \tag{11}$$

where $\mathbf{x}_k^i$ is the $i$-th column of $\mathbf{X}_k$, which corresponds to the $i$-th sample in the $k$-th view and $\mathbf{g}_i$ is the $i$-th row of $\mathbf{G}$, which denotes the clustering indicator vector for the $i$-th sample. Assigning $\mathbf{G}$ into (11) one by one is equivalent to tackling the following problem for the $i$-th sample:

$$c^* = \arg\min_c \sum_{k=1}^M(\alpha_k)^\gamma||\mathbf{W}_k^T\mathbf{x}_k^i-\mathbf{F}_k\mathbf{e}_c||_2^2 \tag{12}$$

where $\mathbf{e}_c$ is the $c$-th row of identity matrix $\mathbf{I}_C$ and $c^*$ means that the $c^*$-th element of $\mathbf{g}_i$ is 1 and others are 0. There are only $C$ kinds of candidate clustering indicator vectors, so we can easily find out the solution of (12).

**Step2: Solving $\mathbf{W}_k|_{k=1}^M$ and $\mathbf{F}_k|_{k=1}^M$ when $\mathbf{G}$ and $\alpha_k|_{k=1}^M$ are fixed.**

Calculating $\mathbf{W}_k|_{k=1}^M$ and $\mathbf{F}_k|_{k=1}^M$ via a weighted multi-view LDA is a supervised learning stage. Moreover, the discriminative subspace $\mathbf{W}_k$ for each view is closely related to the clustering indicator matrix $\mathbf{G}$ and its weight $\alpha_k$.

From (9), we know that $\mathbf{F}_k$ is a function of $\mathbf{W}_k$ and $\mathbf{G}$. When $\mathbf{G}$ and $\alpha_k|_{k=1}^M$ are fixed, substituting (9) into (7) and omitting constant terms, the objective function becomes:

$$\min_{\mathbf{W}_k|_{k=1}^M} \sum_{k=1}^{M} \frac{Tr(\mathbf{W}_k^T \tilde{\mathbf{S}}_W^k \mathbf{W}_k)}{Tr(\mathbf{W}_k^T \mathbf{S}_T^k \mathbf{W}_k)}, s.t. \mathbf{W}_k^T \mathbf{W}_k = \mathbf{I}_{m_k}|_{k=1}^M \quad (13)$$

where $\tilde{\mathbf{S}}_W^k = (\alpha_k)^\gamma [\mathbf{X}_k \mathbf{X}_k^T - \mathbf{X}_k \mathbf{G} (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{X}_k^T]$ denotes the weighted within-class scatter matrix for the $k$-th view. Thus, solving (13) equals to solving the following formula:

$$\max_{\mathbf{W}_k|_{k=1}^M} \sum_{k=1}^{M} \frac{Tr(\mathbf{W}_k^T \tilde{\mathbf{S}}_B^k \mathbf{W}_k)}{Tr(\mathbf{W}_k^T \mathbf{S}_T^k \mathbf{W}_k)}, s.t. \mathbf{W}_k^T \mathbf{W}_k = \mathbf{I}_{m_k}|_{k=1}^M \quad (14)$$

where $\tilde{\mathbf{S}}_B^k = (\alpha_k)^\gamma \mathbf{X}_k \mathbf{G} (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{X}_k^T$ denotes the weighted between-class scatter matrix for the $k$-th view. (14) jointly optimizes $M$ distinct discriminative subspaces in parallel. The solution $\mathbf{W}_k$ for each view is solved by a trace ratio LDA when $\mathbf{G}$ and $\alpha_k|_{k=1}^M$ are fixed.

**Step3: Solving $\alpha_k|_{k=1}^M$ when $\mathbf{W}_k|_{k=1}^M$ and $\mathbf{G}$ are fixed.**

Learning the non-negative normalized weight $\alpha_k$ for each view assigns the more discriminative image feature with higher weight. To derive the solution of $\alpha_k|_{k=1}^M$, we rewrite (7) as:

$$\min_{\alpha_k|_{k=1}^M} \sum_{k=1}^{M} (\alpha_k)^\gamma \mathbf{H}_k, s.t. \sum_{k=1}^{M} \alpha_k = 1, \alpha_k \geq 0 \quad (15)$$

where

$$\mathbf{H}_k = \frac{\|\mathbf{W}_k^T \mathbf{X}_k - \mathbf{F}_k \mathbf{G}^T\|_F^2}{Tr(\mathbf{W}_k^T \mathbf{S}_T^k \mathbf{W}_k)} \quad (16)$$

Thus, the Lagrange function of (15) is:

$$\sum_{k=1}^{M} (\alpha_k)^\gamma \mathbf{H}_k - \lambda (\sum_{k=1}^{M} \alpha_k - 1) \quad (17)$$

where $\lambda$ is the Lagrange multiplier. In order to get the optimal solution, we set the derivative of (17) with respect to $\alpha_k$ to zero and then substitute the result into the constraint $\sum_{k=1}^M \alpha_k = 1$. There is:

$$\alpha_k = \frac{(\gamma \mathbf{H}_k)^{\frac{1}{1-\gamma}}}{\sum_{v=1}^{M} (\gamma \mathbf{H}_v)^{\frac{1}{1-\gamma}}} \quad (18)$$

**Algorithm 1** The algorithm of DEKM method

**Input:**

Data for $M$ views $\{\mathbf{X}_k|k=1,2,...,M\}$, $\mathbf{X}_k \in \mathbb{R}^{d_k \times N}$. The number of clusters $C$. The reduced dimension $m_k$ for each view and the parameter $\gamma$.

**Output:**

The projection matrix $\mathbf{W}_k$, cluster centroid matrix $\mathbf{F}_k$ and weight $\alpha_k$ for the $k$-th view. The common clustering indicator matrix $\mathbf{G}$.

**Initialization:**

Set $t = 0$. Initialize $\mathbf{G} \in$ **Ind**. Initialize $\mathbf{W}_k$ by $\mathbf{W}_k^T \mathbf{W}_k = \mathbf{I}_{m_k}$ and initialize the weight $\alpha_k = 1/M$ for the $k$-th view.

**While** *not converge* **do**

1: Calculate $\mathbf{G}$ by :

$$c^* = \arg\min_c \sum_{k=1}^{M} (\alpha_k)^\gamma ||\mathbf{W}_k^T \mathbf{x}_k^i - \mathbf{F}_k \mathbf{e}_c||_2^2$$

2: Calculate $\mathbf{F}_k$ by $\mathbf{F}_k = \mathbf{W}_k^T \mathbf{X}_k \mathbf{G} (\mathbf{G}^T \mathbf{G})^{-1}$ and update $\mathbf{W}_k|_{k=1}^M$ by

$$\max_{\mathbf{W}_k|_{k=1}^M} \sum_{k=1}^{M} \frac{Tr(\mathbf{W}_k^T \tilde{\mathbf{S}}_B^k \mathbf{W}_k)}{Tr(\mathbf{W}_k^T \mathbf{S}_T^k \mathbf{W}_k)}$$

3: Update $\alpha_k|_{k=1}^M$ by:

$$\alpha_k = \frac{(\gamma \mathbf{H}_k)^{\frac{1}{1-\gamma}}}{\sum_{v=1}^{M} (\gamma \mathbf{H}_v)^{\frac{1}{1-\gamma}}}$$

**End While**, return $\mathbf{W}_k|_{k=1}^M$, $\mathbf{G}$ and $\alpha_k|_{k=1}^M$

To sum up, in Algorithm 1, we can obtain $\mathbf{G}$ via Step1, which is equivalent to the Discriminative K-Means including the interrelations among multi-view features. Updating $\mathbf{W}_k|_{k=1}^M$ via Step2 is the dimensionality reduction for each view. Updating $\alpha_k|_{k=1}^M$ via Step3 fulfills the learning of multiple weights simultaneously. Then we repeat this process iteratively until the objective function value becomes converged.

## 3. Convergence analysis

As mentioned above, DEKM is a unified and embedded multi-view framework solved by an unsupervised optimization scheme. It is obvious that when we transform (6) into (7), it can be divided into three subproblems. Here we show the following proof to verify the convergence of Discriminatively Embedded K-Means (DEKM) algorithm.

**Theorem 2.** *In each iteration, no matter the objective function value of (6) or that of its variant (7), which all decrease until the algorithm converges.*

*Proof.* Supposing after the $t$-th iteration, we have obtained $\mathbf{W}_k^{(t)}|_{k=1}^M$, $\mathbf{G}^{(t)}$ and $\alpha_k^{(t)}|_{k=1}^M$. In the $t+1$-th iteration, we firstly fix $\mathbf{G}$ and $\alpha_k|_{k=1}^M$ as $\mathbf{G}^{(t)}$ and $\alpha_k^{(t)}|_{k=1}^M$ respectively, and then solve $\mathbf{W}_k^{(t+1)}$ for each view. Thus, when $\mathbf{G}^{(t)}$ and $\alpha_k^{(t)}|_{k=1}^M$ are fixed, according to (6), $\mathbf{W}_k^{(t+1)}$ can be solved

by the following equation:

$$\begin{aligned}\mathbf{W}_k^{(t+1)} &= \arg\max_{\mathbf{W}_k} (\alpha_k^{(t)})^\gamma \cdots \\ &\cdots \left\{ \frac{Tr[\mathbf{W}_k^{(t)T}\mathbf{X}_k\mathbf{G}^{(t)}(\mathbf{G}^{(t)T}\mathbf{G}^{(t)})^{-1}\mathbf{G}^{(t)T}\mathbf{X}_k^T\mathbf{W}_k^{(t)}]}{Tr(\mathbf{W}_k^{(t)T}\mathbf{S}_T^k\mathbf{W}_k^{(t)})} - 1 \right\} \\ &= \arg\min_{\mathbf{W}_k} (\alpha_k^{(t)})^\gamma \cdots \\ &\cdots \left\{ \frac{Tr[\mathbf{W}_k^{(t)T}(\mathbf{S}_T^k - \mathbf{X}_k\mathbf{G}^{(t)}(\mathbf{G}^{(t)T}\mathbf{G}^{(t)})^{-1}\mathbf{G}^{(t)T}\mathbf{X}_k^T)\mathbf{W}_k^{(t)}]}{Tr(\mathbf{W}_k^{(t)T}\mathbf{S}_T^k\mathbf{W}_k^{(t)})} \right\}\end{aligned} \tag{19}$$

Referring to the way of argumentation for [6], through rewriting (19) we have:

$$\frac{Tr\big(\mathbf{W}_k^{(t+1)T}\tilde{\mathbf{S}}_W^{k(t)}\mathbf{W}_k^{(t+1)}\big)}{Tr\big(\mathbf{W}_k^{(t+1)T}\mathbf{S}_T^k\mathbf{W}_k^{(t+1)}\big)} \leq \frac{Tr\big(\mathbf{W}_k^{(t)T}\tilde{\mathbf{S}}_W^{k(t)}\mathbf{W}_k^{(t)}\big)}{Tr\big(\mathbf{W}_k^{(t)T}\mathbf{S}_T^k\mathbf{W}_k^{(t)}\big)} \tag{20}$$

where

$$\begin{aligned}\tilde{\mathbf{S}}_W^{k(t)} &= (\alpha_k^{(t)})^\gamma [\mathbf{S}_T^k - \mathbf{X}_k\mathbf{G}^{(t)}(\mathbf{G}^{(t)T}\mathbf{G}^{(t)})^{-1}\mathbf{G}^{(t)T}\mathbf{X}_k^T] \\ &= (\alpha_k^{(t)})^\gamma \mathbf{S}_W^{k(t)}\end{aligned}$$

and it denotes the weighted within-class scatter matrix for the $k$-th view at the $t$-th iteration.

In the same way, we fix $\mathbf{W}_k|_{k=1}^M$ and $\alpha_k|_{k=1}^M$ as $\mathbf{W}_k^{(t)}|_{k=1}^M$ and $\alpha_k^{(t)}|_{k=1}^M$ respectively, and solve for $\mathbf{G}^{(t+1)}$. According to (6), we can obtain:

$$\begin{aligned}\mathbf{G}^{(t+1)} &= \arg\max_{\mathbf{G}} \sum_{k=1}^M (\alpha_k^{(t)})^\gamma \cdots \\ &\cdots \left\{ \frac{Tr[\mathbf{W}_k^{(t)T}\mathbf{X}_k\mathbf{G}^{(t)}(\mathbf{G}^{(t)T}\mathbf{G}^{(t)})^{-1}\mathbf{G}^{(t)T}\mathbf{X}_k^T\mathbf{W}_k^{(t)}]}{Tr(\mathbf{W}_k^{(t)T}\mathbf{S}_T^k\mathbf{W}_k^{(t)})} - 1 \right\} \\ &= \arg\min_{\mathbf{G}} \sum_{k=1}^M (\alpha_k^{(t)})^\gamma \cdots \\ &\cdots \left\{ \frac{Tr[\mathbf{W}_k^{(t)T}(\mathbf{S}_T^k - \mathbf{X}_k\mathbf{G}^{(t)}(\mathbf{G}^{(t)T}\mathbf{G}^{(t)})^{-1}\mathbf{G}^{(t)T}\mathbf{X}_k^T)\mathbf{W}_k^{(t)}]}{Tr(\mathbf{W}_k^{(t)T}\mathbf{S}_T^k\mathbf{W}_k^{(t)})} \right\}\end{aligned} \tag{21}$$

By rewriting (21), there is:

$$\sum_{k=1}^M \frac{Tr\big(\mathbf{W}_k^{(t)T}\tilde{\mathbf{S}}_W^{k(t+1)}\mathbf{W}_k^{(t)}\big)}{Tr\big(\mathbf{W}_k^{(t)T}\mathbf{S}_T^k\mathbf{W}_k^{(t)}\big)} \leq \sum_{k=1}^M \frac{Tr\big(\mathbf{W}_k^{(t)T}\tilde{\mathbf{S}}_W^{k(t)}\mathbf{W}_k^{(t)}\big)}{Tr\big(\mathbf{W}_k^{(t)T}\mathbf{S}_T^k\mathbf{W}_k^{(t)}\big)} \tag{22}$$

where

$$\begin{aligned}\tilde{\mathbf{S}}_W^{k(t+1)} &= (\alpha_k^{(t)})^\gamma \begin{bmatrix} \mathbf{S}_T^k - \mathbf{X}_k\mathbf{G}^{(t+1)}(\mathbf{G}^{(t+1)T}\mathbf{G}^{(t+1)})^{-1} \\ \mathbf{G}^{(t+1)T}\mathbf{X}_k^T \end{bmatrix} \\ &= (\alpha_k^{(t)})^\gamma \mathbf{S}_W^{k(t+1)}\end{aligned}$$

and it is the weighted within-class scatter matrix for the $k$-th view at the $t+1$-th iteration.

Similarly, we fix $\mathbf{W}_k|_{k=1}^M$ and $\mathbf{G}$ as $\mathbf{W}_k^{(t)}|_{k=1}^M$ and $\mathbf{G}^{(t)}$ respectively, and solve for $\alpha_k^{(t+1)}|_{k=1}^M$. According to (6), for each view, $\alpha_k^{(t+1)}$ can be calculated by:

$$\begin{aligned}\alpha_k^{(t+1)} &= \arg\max_{\alpha_k} (\alpha_k^{(t)})^\gamma \cdots \\ &\cdots \left\{ \frac{Tr[\mathbf{W}_k^{(t)T}\mathbf{X}_k\mathbf{G}^{(t)}(\mathbf{G}^{(t)T}\mathbf{G}^{(t)})^{-1}\mathbf{G}^{(t)T}\mathbf{X}_k^T\mathbf{W}_k^{(t)}]}{Tr(\mathbf{W}_k^{(t)T}\mathbf{S}_T^k\mathbf{W}_k^{(t)})} - 1 \right\} \\ &= \arg\min_{\alpha_k} (\alpha_k^{(t)})^\gamma \cdots \\ &\cdots \left\{ \frac{Tr[\mathbf{W}_k^{(t)T}(\mathbf{S}_T^k - \mathbf{X}_k\mathbf{G}^{(t)}(\mathbf{G}^{(t)T}\mathbf{G}^{(t)})^{-1}\mathbf{G}^{(t)T}\mathbf{X}_k^T)\mathbf{W}_k^{(t)}]}{Tr(\mathbf{W}_k^{(t)T}\mathbf{S}_T^k\mathbf{W}_k^{(t)})} \right\}\end{aligned} \tag{23}$$

Thus, (23) can be further rewritten as follows:

$$(\alpha_k^{(t+1)})^\gamma \frac{Tr\big(\mathbf{W}_k^{(t)T}\mathbf{S}_W^{k(t)}\mathbf{W}_k^{(t)}\big)}{Tr\big(\mathbf{W}_k^{(t)T}\mathbf{S}_T^k\mathbf{W}_k^{(t)}\big)} \leq (\alpha_k^{(t)})^\gamma \frac{Tr\big(\mathbf{W}_k^{(t)T}\mathbf{S}_W^{k(t)}\mathbf{W}_k^{(t)}\big)}{Tr\big(\mathbf{W}_k^{(t)T}\mathbf{S}_T^k\mathbf{W}_k^{(t)}\big)} \tag{24}$$

Integrating (20), (22) and (24), we arrive at:

$$\begin{aligned}&\sum_{k=1}^M \frac{Tr\big(\mathbf{W}_k^{(t+1)T}(\alpha_k^{(t+1)})^\gamma\mathbf{S}_W^{k(t+1)}\mathbf{W}_k^{(t+1)}\big)}{Tr\big(\mathbf{W}_k^{(t+1)T}\mathbf{S}_T^k\mathbf{W}_k^{(t+1)}\big)} \\ &\leq \sum_{k=1}^M \frac{Tr\big(\mathbf{W}_k^{(t)T}(\alpha_k^{(t+1)})^\gamma\mathbf{S}_W^{k(t+1)}\mathbf{W}_k^{(t)}\big)}{Tr\big(\mathbf{W}_k^{(t)T}\mathbf{S}_T^k\mathbf{W}_k^{(t)}\big)} \\ &\leq \sum_{k=1}^M \frac{Tr\big(\mathbf{W}_k^{(t)T}(\alpha_k^{(t+1)})^\gamma\mathbf{S}_W^{k(t)}\mathbf{W}_k^{(t)}\big)}{Tr\big(\mathbf{W}_k^{(t)T}\mathbf{S}_T^k\mathbf{W}_k^{(t)}\big)} \\ &\leq \sum_{k=1}^M \frac{Tr\big(\mathbf{W}_k^{(t)T}(\alpha_k^{(t)})^\gamma\mathbf{S}_W^{k(t)}\mathbf{W}_k^{(t)}\big)}{Tr\big(\mathbf{W}_k^{(t)T}\mathbf{S}_T^k\mathbf{W}_k^{(t)}\big)}\end{aligned} \tag{25}$$

Thus, (25) proves that (6) and its variant (7) are lower bounded and their objective function value decreases after each iteration. □

## 4. Experiments

In this section, we evaluate the performance of DEKM on three benchmark datasets in terms of two standard clustering evaluation metrics, namely Accuracy (ACC) [2] and Normalized Mutual Information (NMI) [2]. Before do anything, we need to centralize the data and normalize all values in the range of $[-1, 1]$.

### 4.1. Datasets

In our experiments, by following [3], three benchmark image datasets including Caltech101 [13], MSRC [28] and

Table 1. Descriptions of testing datasets.

| View | MSRCv1 | Caltech101-7 | Handwritten |
|---|---|---|---|
| 1 | CMT (48) | CMT (48) | FAC (216) |
| 2 | HOG (100) | HOG (100) | PIX (240) |
| 3 | LBP (256) | LBP (256) | ZER (47) |
| 4 | SIFT (210) | SIFT (441) | MOR (6) |
| 5 | GIST (512) | GIST (512) | KAR (64) |
| 6 | CENT (1302) | CENT (1302) | FOU (76) |
| Images | 210 | 441 | 2000 |
| Classes | 7 | 7 | 10 |

Handwritten [1] were adopted for evaluations. Figure 1 shows some image examples from above three datasets. Table 1 summarizes the information of each dataset including the number of images and classes, heterogeneous features and the dimensionality of each type of feature.

### 4.2. Toy example

In this section, we conducted a toy experiment to verify the effectiveness of DEKM. For simplicity, we worked on the two-view case given by [19]. We show the projection directions (green solid lines) with different numbers of iterations $Initialization$, $Iteration = 3$, $Iteration = 5$ and $Iteration = 7$ in Figure 2. Performing our method with $\gamma > 1$ on this synthetic data, ACC is $0.9950$ and NMI is $0.9590$. It is observed that DEKM can exactly obtain projection directions which separate different clusters maximally and achieve better and stable accuracy with few iteration steps.

In contrast, if we performed LDA on each individual view, the results are decreased significantly. It further demonstrates that DEKM method has no trivial solution when $\gamma > 1$ and incorporates multiple views effectively.

### 4.3. Performance evaluation

**Comparison methods.** Firstly, we compared the performance of DEKM with Embedded K-Means clustering [26] (EKM) for single-view to simply explain the advantage of multi-view. Secondly, to emphasize the importance of intercoordination among multiple views, we compared the results of DEKM with AEKM which concatenates all views together directly and then performs EKM clustering. Thirdly, we compared DEKM with some baseline methods naive Multi-view K-Means clustering (NMVKM), its robust version LMVKM (NMVKM with $\ell_{2,1}$-norm) and RMVKM [3] to demonstrate the significant advantage of discriminative subspace learning. Finally, when we ignore the weight of each view, DEKM can degenerate to a simple version DEKM (SDEKM), which verifies the necessity of the weight learning.

**Comparison results.** From comparison results shown in Tables 2 and 3, we have the following observations.

In Table 2, DEKM performs significantly better than EKM. It is straightforward to demonstrate the superiority of multi-view. In Table 3, compared with AEKM without any mutual information among multiple views, it is clear that DEKM can boost the clustering performance due to the intercoordinations of different views. In addition, DEKM outperforms other methods (NMVKM, LMVKM, RMVKM and SDEKM). On the one hand, compared with NMVKM, LMVKM and RMVKM, DEKM simultaneously obtains multiple discriminative subspaces which has great effects on the performance of algorithms. On the other hand, unlike SDEKM, DEKM adaptively learns the weight for each view to better integrate heterogeneous image features and then improve the performance of clustering.

Furthermore, we tested the convergence speed of DEKM on three datasets which is shown in Figure 3. It is observed that DEKM algorithm can converge with few iteration steps.

### 4.4. Evaluation of key components of the proposed method

There are three key components in DEKM algorithm: the initialization of $\mathbf{G}$, the dimensionality of embedded subspace $m_k$ for the $k$-th view and the parameter $\gamma$.

**Initialization.** According to [3], it can be seen that NMVKM and RMVKM always simply use general random method to initialize $\mathbf{G}$. However, random initialization greatly affects the results of clustering. Like in [3], the ACC of NMVKM is $0.7002 \pm 0.085$, and the ACC of RMVKM is $0.8142 \pm 0.087$. We can see that the precision level, *i.e.* $8.5\%$ or $8.7\%$, is not high, such that it is hard to always remain high performance. In addition, unstable initialization is difficult to control parameters and obtain ideal results. Thus, we initialize $\mathbf{G}$ in a new way to substantially reduce the dependence of clustering result on initialization and conveniently tune parameters.

We first sort the rows of identity matrix $\mathbf{I}_C$ randomly and get the matrix $\tilde{\mathbf{I}}_C$, and then we use direct product of vector $\mathbf{1}$ and matrix $\tilde{\mathbf{I}}_C$ to produce the initial $\mathbf{G}$:

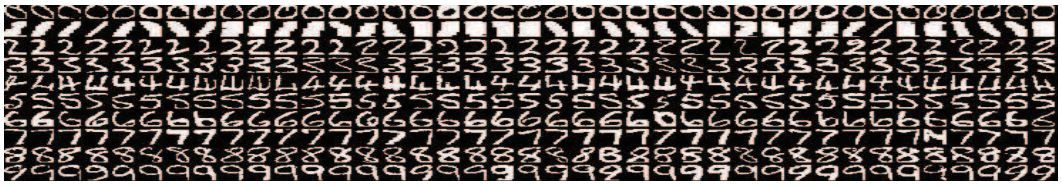
$$\begin{aligned} &\mathbf{G} = \mathbf{1} \otimes \tilde{\mathbf{I}}_C \\ &s.t. \mathbf{I}_C \in \mathbb{R}^{C \times C}, \tilde{\mathbf{I}}_C \in \mathbb{R}^{C \times C}, \mathbf{1} \in \mathbb{R}^{floor(N/C) \times 1} \end{aligned} \tag{26}$$

where $\mathbf{1}$ denotes a column vector with all elements being 1. Sometimes the number of samples $N$ cannot be divisible by the number of clusters $C$, so we need to extra select $r = N - C \times floor(N/C)$ rows from $\tilde{\mathbf{I}}_C$ randomly to fill the indivisible part. In other words, this new initialization makes the number of samples for each label equally as far as possible. Note that we do not care whether these labels are correct or not as long as the numbers of different kinds of labels are equal. As the mapping relationships between the labels of different clusters are nearly invariable, we can obtain more stable initialization.

**Dimension of views.** In above discussions, we assume that the total scatter matrix is always invertible. However, in


(a) Caltech101-7 Dataset images from 7 classes: Face, Motorbike, Dollabill, Garfield, Snoopy, Stop-sign and Windsor-chair


(b) MSRC Dataset images from 7 classes: Cow, Tree, Building, Airplane, Face, Car, and Bicycle



(c) Handwritten numerals Dataset images from 10 digit classes: 0 to 9

Figure 1. Some example images from (a) Caltech101, (b) MSRC and (c) Handwritten numerals data sets.

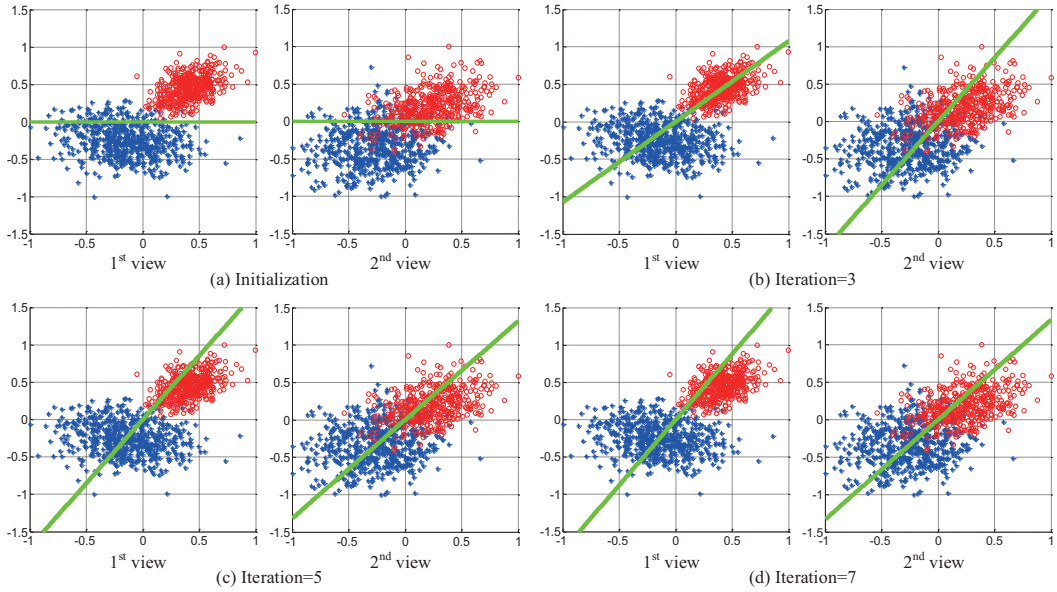



Figure 2. Projection directions of synthetic data with different iterations.

real applications, high-dimensional complex data may lead total scatter matrix to be singular. If so, we can adopt PCA as a preprocessing step to ensure the total scatter matrix invertible.

In this paper, dimensionality $m_k$ is an important parameter because the curse of dimensionality may occur if $m_k$ is large and otherwise there exists overlap of different clusters. We determined $m_k$ heuristically by grid search and choosed the one with the best clustering accuracy. Tables 2 and 3 show that DEKM outperforms other methods greatly, when we find suitable parameter $m_k$. For example, we have performed the test on Caltech101-7 dataset whose total dimensionality of all views reaches up to 2659. Through DEKM algorithm, we not only reduce the total dimensionality of all views from 2659 to 1194, but also learn multiple discriminative subspaces to significantly improve the performance of clustering.

**Parameter $\gamma$.** In DEKM method, we use one parameter $\gamma$ to control the distribution of weights for different views. According to (18) and the characteristic of the function $\frac{1}{1-\gamma}$, two extreme cases are produced. When $\gamma \to \infty$, DEKM can get equal weights $\frac{1}{M}$. When $\gamma \to 1^+$, suppos-

Table 2. Comparison of DEKM and EKM on MSRCv1, Caltech101-7 and Handwritten datasets.

| Method | MSRCv1 | | Caltech101-7 | | Handwritten | |
|---|---|---|---|---|---|---|
| | ACC | NMI | ACC | NMI | ACC | NMI |
| EKM(view1) | 0.5048 ± 0.00 | 0.4365 ± 0.00 | 0.3243 ± 0.00 | 0.1666 ± 0.00 | 0.6340 ± 0.00 | 0.6253 ± 0.00 |
| EKM(view2) | 0.6286 ± 0.00 | 0.5436 ± 0.00 | 0.5578 ± 0.00 | 0.4080 ± 0.00 | 0.7680 ± 0.00 | 0.7313 ± 0.00 |
| EKM(view3) | 0.5048 ± 0.00 | 0.4734 ± 0.00 | 0.3738 ± 0.00 | 0.2948 ± 0.00 | 0.5745 ± 0.00 | 0.5361 ± 0.00 |
| EKM(view4) | 0.4238 ± 0.00 | 0.3277 ± 0.00 | 0.6961 ± 0.00 | 0.6276 ± 0.00 | 0.4280 ± 0.00 | 0.4995 ± 0.00 |
| EKM(view5) | 0.6714 ± 0.00 | 0.6275 ± 0.00 | 0.7007 ± 0.00 | 0.6235 ± 0.00 | 0.6455 ± 0.00 | 0.5462 ± 0.00 |
| EKM(view6) | 0.5476 ± 0.00 | 0.5527 ± 0.00 | 0.6667 ± 0.00 | 0.5635 ± 0.00 | 0.6975 ± 0.00 | 0.6429 ± 0.00 |
| **DEKM** | **0.9238** ± 0.00 | **0.8649** ± 0.00 | **0.8503** ± 0.00 | **0.8231** ± 0.00 | **0.9530** ± 0.00 | **0.9098** ± 0.00 |

Table 3. Clustering Performances of the compared methods on MSRCv1, Caltech101-7 and Handwritten datasets.

| Method | MSRCv1 | | Caltech101-7 | | Handwritten | |
|---|---|---|---|---|---|---|
| | ACC | NMI | ACC | NMI | ACC | NMI |
| NMVKM | 0.7810 ± 0.00 | 0.7122 ± 0.00 | 0.7143 ± 0.00 | 0.7337 ± 0.00 | 0.7810 ± 0.00 | 0.7661 ± 0.00 |
| LMVKM | 0.7762 ± 0.00 | 0.7190 ± 0.00 | 0.7664 ± 0.00 | 0.7208 ± 0.00 | 0.8030 ± 0.00 | 0.7853 ± 0.00 |
| RMVKM | 0.9048 ± 0.00 | 0.8463 ± 0.00 | 0.7846 ± 0.00 | 0.7145 ± 0.00 | 0.9125 ± 0.00 | 0.8539 ± 0.00 |
| AEKM | 0.7810 ± 0.00 | 0.7293 ± 0.00 | 0.7302 ± 0.00 | 0.7299 ± 0.00 | 0.8950 ± 0.00 | 0.8152 ± 0.00 |
| SDEKM | 0.8810 ± 0.00 | 0.8002 ± 0.00 | 0.8254 ± 0.00 | 0.7465 ± 0.00 | 0.9355 ± 0.00 | 0.8753 ± 0.00 |
| **DEKM** | **0.9238** ± 0.00 | **0.8649** ± 0.00 | **0.8503** ± 0.00 | **0.8231** ± 0.00 | **0.9530** ± 0.00 | **0.9098** ± 0.00 |

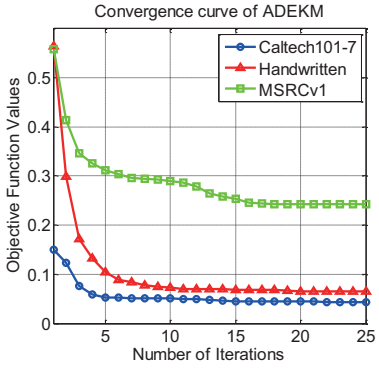



Figure 3. The convergence curve of DEKM on Handwritten, MSRCv1 and Caltech101-7 dataset, respectively.

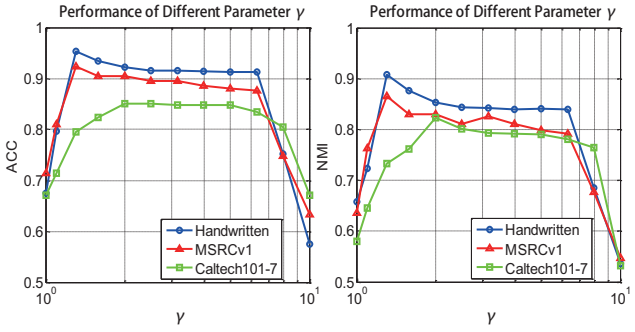



Figure 4. The influence of parameter $\gamma$ on Handwritten, MSRCv1 and Caltech101-7 datasets, respectively.

ing $\mathbf{H}_p = min\{\mathbf{H}_k | k = 1, ..., p, ..., M\}$, we substitute $\mathbf{H}_p$ into (18) and solve its weight:

$$\lim_{\gamma \to 1^+} \alpha_p = \lim_{\gamma \to 1^+} \frac{1}{1 + \sum_{v \neq p} (\mathbf{H}_v / \mathbf{H}_p)^{\frac{1}{1-\gamma}}} = 1 \qquad (27)$$

It can be seen that DEKM assigns 1 to the weight of the view whose $\mathbf{H}_p$ value is the smallest and assign 0 to the weights of other views.

Using such kind of strategy,we not only assure DEKM has no trivial solution when $\gamma > 1$, but also reduce the parameters of the model greatly. In our experiments, we searched $log_{10}\gamma$ in the range from 0 to 1 with incremental step 0.1 to obtain the best parameter $\gamma$. In Figure 4, we show that $\gamma$ dominates the performance of DEKM algorithm on three datasets.

## 5. Conclusion

In this paper, we have proposed an unsupervised clustering framework which embeds multiple discriminative subspaces learning into multi-view K-Means clustering to construct an unified framework, and adaptively control the intercoordinations between multiple views via the weight learning. Besides, our optimization scheme efficiently solved the proposed objective function with global optimality and convergence. Comprehensive evaluations on widely used image benchmarks have demonstrated DEKM is effective for clustering high-dimensional multi-view data.

**Acknowledgements.** This work was supported in part by the National Science Foundation of China under Grants 61522207 and 61473231.

## References


[1] A. Asuncion and D. Newman. Uci machine learning repository, 2007.

[2] D. Cai, X. He, and J. Han. Document clustering using locality preserving indexing. *TKDE*, 17(12):1624–1637, 2005.

[3] X. Cai, F. Nie, and H. Huang. Multi-view k-means clustering on big data. In *AAAI*, 2013.

[4] X. Cai, F. Nie, H. Huang, and F. Kamangar. Heterogeneous image feature integration via multi-modal spectral clustering. In *CVPR*, pages 1977–1984, 2011.

[5] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang. Diversity-induced multi-view subspace clustering. In *CVPR*, pages 586–594, 2015.

[6] X. Chang, F. Nie, Y. Yang, and H. Huang. A convex formulation for semi-supervised multi-label feature selection. In *AAAI*, 2014.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[8] V. R. de Sa. Spectral clustering with two views. In *ICML workshop on learning with multiple views*, pages 20–27, 2005.

[9] C. Ding, X. He, and H. D. Simon. Nonnegative lagrangian relaxation of k-means and spectral clustering. In *ECML*, pages 530–538. 2005.

[10] C. Ding and T. Li. Adaptive dimension reduction using discriminant analysis and k-means clustering. In *ICML*, 2007.

[11] C. Ding, D. Zhou, X. He, and H. Zha. R1-pca: rotational invariant l 1-norm principal component analysis for robust subspace factorization. In *ICML*, 2006.

[12] C. Domeniconi, D. Papadopoulos, D. Gunopulos, and S. Ma. Subspace clustering of high dimensional data. In *SDM*, 2004.

[13] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *CVIU*, 106(1):59–70, 2007.

[14] D. Guo, J. Zhang, X. Liu, Y. Cui, and C. Zhao. Multiple kernel learning based multi-view spectral clustering. In *ICPR*, pages 3774–3779, 2014.

[15] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Applied Statistics*, 28(1):100–108, 1979.

[16] C. Hou, F. Nie, D. Yi, and D. Tao. Discriminative embedded clustering: A framework for grouping high-dimensional data. *TNNLS*, 2014.

[17] Y. Jia, F. Nie, and C. Zhang. Trace ratio problem revisited. *TNN*, 20(4):729–735, 2009.

[18] A. Kumar and H. Daumé. A co-training approach for multi-view spectral clustering. In *ICML*, 2011.

[19] A. Kumar, P. Rai, and H. Daume. Co-regularized multi-view spectral clustering. In *NIPS*, 2011.

[20] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[21] D. Niu, J. G. Dy, and M. I. Jordan. Multiple non-redundant spectral clustering views. In *ICML*, pages 831–838, 2010.

[22] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *TPAMI*, 24(7):971–987, 2002.

[23] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

[24] J. Shi and J. Malik. Normalized cuts and image segmentation. *TPAMI*, 22(8):888–905, 2000.

[25] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[26] D. Wang, F. Nie, and H. Huang. Unsupervised feature selection via unified trace ratio formulation and k-means clustering (track). In *ECMLPKDD*. 2014.

[27] H. Wang, C. Weng, and J. Yuan. Multi-feature spectral clustering with minimax optimization. In *CVPR*, pages 4106–4113, 2014.

[28] J. Winn and N. Jojic. Locus: Learning object classes with unsupervised segmentation. In *ICCV*, 2005.

[29] J. Wu and J. M. Rehg. Where am i: Place instance and category recognition using spatial pact. In *CVPR*, 2008.

[30] H. Yu, M. Li, H.-J. Zhang, and J. Feng. Color texture moments for content-based image retrieval. In *ICIP*, 2002.

[31] X. Zhao, N. Evans, and J.-L. Dugelay. A subspace co-training framework for multi-view clustering. *PR*, 41:73–82, 2014.

[32] D. Zhou and C. J. Burges. Spectral clustering and transductive learning with multiple views. In *ICML*, pages 1159–1166, 2007.