Centerless Multi-View K-means Based on the Adjacency Matrix

Han Lu[1], Quanxue Gao[1*], Qianqian Wang[1], Ming Yang[2], Wei Xia[1]

[1] School of Telecommunications Engineering, Xidian University, Xi'an 710071, P.R.China.
[2] Mathematics department of the University of Evansville, Evansville, IN 47722 USA
luhan0@foxmail.com, {qxgao, qqwang}@xidian.edu.cn, {yangmingmath, xd.weixia}@gmail.com

Abstract

Although K-Means clustering has been widely studied due to its simplicity, these methods still have the following fatal drawbacks. Firstly, they need to initialize the cluster centers, which causes unstable clustering performance. Secondly, they have poor performance on non-Gaussian datasets. Inspired by the affinity matrix, we propose a novel multi-view K-Means based on the adjacency matrix. It maps the affinity matrix to the distance matrix according to the principle that every sample has a small distance from the points in its neighborhood and a large distance from the points outside of the neighborhood. Moreover, this method well exploits the complementary information embedded in different views by minimizing the tensor Schatten p-norm regularize on the third-order tensor which consists of cluster assignment matrices of different views. Additionally, this method avoids initializing cluster centroids to obtain stable performance. And there is no need to compute the means of clusters so that our model is not sensitive to outliers. Experiment on a toy dataset shows the excellent performance on non-Gaussian datasets. And other experiments on several benchmark datasets demonstrate the superiority of our proposed method.

Introduction

Clustering is one of the most representative unsupervised techniques for analyzing data in data mining and artificial intelligence, which has attracted more and more attention in recent years due to a large amount of unlabeled data. Clustering aims to divide the data into C groups, i.e., clusters such that the similarity between data points in the same cluster is high, while the similarity between data points in different clusters is low. With the development of sensor technology, multi-view data are ubiquitous in real applications and help provide some complementary information which is important for clustering. Inspired by this, many multi-view clustering methods have been proposed, and one of the most representative techniques is multi-view K-Means clustering.

The purpose of the traditional K-Means (Hartigan and Wong 1979) is graphing a dataset into some certain clusters by assigning each data sample to the cluster with the nearest centroid. This algorithm suffers from two serious limitations. One is that the initialized cluster centroids affect the results, and the other is that it can only distinguish Gaussian-distributed data. For the first limitation, The general workaround is doing multiple experiments with the randomly initial cluster centroids to obtain a near-local optimal solution, which takes more time. Although some methods can speed up computation such as (Elkan 2003; Arthur and Vassilvitskii 2007), this just alleviates the instability of the results rather than resolves. For the second limitation, kernel K-Means (Kim et al. 2005; Wang et al. 2022; Ren, Sun, and Wei 2021; Liu et al. 2020) are proposed. It is an extension of the standard K-Means algorithm that maps data from the input space to a higher dimensional feature space through a nonlinear transformation and minimizes the clustering error in feature space. Thus non-linearly separated clusters are obtained in the input space. However, the computational and storage costs of the kernel matrix are high due to the increased dimension.

To solve these problems, we propose centerless multi-view K-Means clustering method based on the adjacency matrix. It is well known that K-Means cannot separate nonlinear datasets well. For this, we use the distances calculated by affinity matrices instead of Euclidean distances. Meanwhile, our method avoids to initialize the centers of clusters.

Tensor Schatten p-norm (Gao et al. 2021) exploit the complementary information embedded in different views well. So our method leverages the tensor Schatten p-norm regularizer on the third-order tensor, which consists of discrete clustering assignment matrices of different views, to minimize the divergence between assignment matrices of different views. Finally, we introduce an adaptive weighted strategy to further improve the clustering performance. Code and data are available[1]. The main contributions are as follows:

- Compared with K-Means, our method constructs the distance matrices with the adjacency matrices, so it is suitable for both linearly and non-linearly separate cluster in the input space.
- Our method avoids initializing cluster centroids so the clustering performance is robust. What's more, avoiding compute the means of clusters makes our method insensitive to outliers.

*Corresponding author

[1] https://github.com/luhan0/CMKOA

- We leverage the minimization of tensor Schatten p-norm to fully exploit structural and complementary information among different views. Our proposed adaptive weighted strategy take into account the different contributions of different views.

Notations

For convenience, we introduce the notations used throughout the paper. We use bold calligraphy letters for third-order tensors, $\boldsymbol{\mathcal{M}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, bold upper case letters for matrices, $\mathbf{M}$, bold lower case letters for vectors, $\mathbf{m}$, and lower case letters such as m_{ijk} for the entries of $\boldsymbol{\mathcal{M}}$, m_{ij} is the (i,j)-th entry of $\mathbf{M}$ and $tr(\bullet)$ is the trace of a matrix. Ind is the set of all cluster assignment matrices. Moreover, the i-th frontal slice of $\boldsymbol{\mathcal{M}}$ is $\boldsymbol{\mathcal{M}}^{(i)}$. $\overline{\boldsymbol{\mathcal{M}}}$ is the discrete Fourier transform (DFT) of $\boldsymbol{\mathcal{M}}$ along the third dimension, $\overline{\boldsymbol{\mathcal{M}}} = \text{fft}(\boldsymbol{\mathcal{M}}, [\,], 3)$. Thus, $\boldsymbol{\mathcal{M}} = \text{ifft}(\overline{\boldsymbol{\mathcal{M}}}, [\,], 3)$. The trace of matrix $\mathbf{M}$ is expressed as $tr(\mathbf{M})$. The Frobenius norm of $\boldsymbol{\mathcal{M}}$ is defined as $\|\boldsymbol{\mathcal{M}}\|_F = \sqrt{\sum_{i,j,k} |m_{ijk}|^2}$.

Another Representation of *K*-Means

Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^{\mathrm{T}} \in \mathbb{R}^{N \times d}$ denote the data matrix, where N and d are the number of samples and feature dimensions, respectively. The label matrix is denoted by $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^{\mathrm{T}} \in \{0,1\}^{N \times C}$, where $\mathbf{y}_i$ is indicator vector of the i-th sample; C is the cluster number; $y_{ij} = 1$ if $\mathbf{x}_i$ belongs to the j-th cluster; $y_{ij} = 0$, otherwise.

Given a weighted undirected graph $\mathcal{G}(\mathbf{X}, \mathbf{W})$, where $\mathbf{W}$ is an adjacency matrix which characterizes the relationship between data points $\mathbf{x}_1, \cdots, \mathbf{x}_N$.

K-Means aims to partition data points $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$ into C clusters $\mathbf{X}_1, \ldots, \mathbf{X}_C$ according to the Euclidean distance between the data points and the C centroid points $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_C$ such that the data points in the same clusters are as close as possible and the distances between different clusters are as large as possible. Thus, the objective function of K-Means can be formulated as

$$\min_{\mathbf{X}_1,\ldots,\mathbf{X}_C} \sum_{l=1}^{C} \sum_{\mathbf{x}_i \in \mathbf{X}_l} \|\mathbf{x}_i - \mathbf{u}_l\|_2^2 \tag{1}$$

where $\mathbf{u}_l = \sum_{\mathbf{x}_i \in \mathbf{X}_l} \frac{\mathbf{x}_i}{|\mathbf{X}_l|}$ is the centroid of cluster $\mathbf{X}_l$; $|\mathbf{X}_l|$ is the number of samples in $\mathbf{X}_l$.

By using simple linear algebra, (1) becomes (Pei et al. 2020)

$$\min_{\mathbf{Y} \in Ind} \text{tr}((\mathbf{Y}^{\mathrm{T}}\mathbf{Y})^{-1}\mathbf{Y}^{\mathrm{T}}\mathbf{D}\mathbf{Y}) \tag{2}$$

where $\mathbf{D}$ denotes distance matrix of $\mathbf{X}$, the i-th row j-th column element of $\mathbf{D}$ is $\|\mathbf{x}_i - \mathbf{x}_j\|_2^2$.

Methodology

The Proposed Model

In real applications, different views contain different characteristics of data, thus, they should have different similarity matrices, resulting to different label matrices of different views. Combining the aforementioned insight analysis, we get a new model for discrete multi-view clustering as

$$\begin{aligned} &\min_{\mathbf{Y}^{(v)},\, \alpha_v} \sum_{v=1}^{V} \alpha_v^r \text{tr}((\mathbf{Y}^{(v)^{\mathrm{T}}}\mathbf{Y}^{(v)})^{-1}\mathbf{Y}^{(v)^{\mathrm{T}}}\mathbf{H}^{(v)}\mathbf{Y}^{(v)}) + \lambda \mathcal{R}(\boldsymbol{\mathcal{Y}}) \\ &\quad s.t. \qquad \mathbf{Y}^{(v)} \in Ind,\ \sum_{v=1}^{V} \alpha_v = 1, \alpha_v \geq 0 \end{aligned} \tag{3}$$

where $\mathbf{Y}^{(v)}$ is the label matrix of v-th view; $\mathcal{R}(\bullet)$ represents the regularizer on $\mathbf{Y}^{(v)}$; α_v^r is the adaptive weight for v-th view, which characterizes the importance of the v-th view, V is the number of views, λ is a trade-off parameter. $\mathbf{H}^{(v)}$ is produced by the mapping of Euclidean distances $\mathbf{D}^{(v)}$.

Let $\mathbf{H}^{(v)}$ denotes the distance matrix $\mathbf{D}^{(v)}$ of the v-th view. It is apparent from the above analysis that (3) becomes multi-view K-Means.

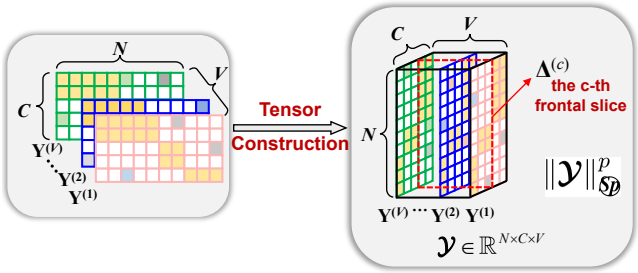


Figure 1: Construction of tensor $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{N \times C \times V}$. $\boldsymbol{\Delta}^{(c)}$ denotes the c-th frontal slice of $\boldsymbol{\mathcal{Y}}$ ($c \in \{1, 2, \cdots, C\}$).

Multi-View Discrete Clustering

In (3), the second term aims to minimize the divergence between $\mathbf{Y}^{(v)}$. To solve this problem, a naive method is to leverage squared F-norm to learn the discrete label matrix. As we all know, F-norm is a one-dimensional and pixel-wise measurement method. Thus, it cannot well exploit the complementary information embedded in $\mathbf{Y}^{(v)}$. It is evident that the low rank approximation using tensor Schatten p-norm performs very well in exploiting the complementary information embedded in views (Gao et al. 2021; Xia et al. 2022c), thus, we minimize the divergence between $\mathbf{Y}^{(v)}$ by using the tensor Schatten p-norm minimization on the third-order tensor $\mathcal{Y}$ which consists of $\mathbf{Y}^{(v)}$. Thus, we have

$$\begin{aligned} &\min_{\mathbf{Y}^{(v)},\, \alpha_v} \sum_{v=1}^{V} \alpha_v^r tr((\mathbf{Y}^{(v)^{\mathrm{T}}}\mathbf{Y}^{(v)})^{-1}\mathbf{Y}^{(v)^{\mathrm{T}}}\mathbf{H}^{(v)}\mathbf{Y}^{(v)}) + \lambda \|\boldsymbol{\mathcal{Y}}\|_{\circledS p}^p \\ &\quad s.t. \qquad \mathbf{Y}^{(v)} \in Ind,\ \sum_{v=1}^{V} \alpha_v = 1, \alpha_v \geq 0 \end{aligned} \tag{4}$$

where the v-th lateral slice of $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{N \times V \times C}$ is $\mathbf{Y}^{(v)}$ (See Fig. 1); $\|\bullet\|_{\circledS p}$ is the tensor Schatten p-norm (see Definition 1).

Definition 1 *(Tensor Schatten p-norm (Gao et al. 2021))* *Given* $\boldsymbol{\mathcal{M}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $h = \min(n_1, n_2)$, *the tensor*

Schatten p-norm of tensor $\mathcal{M}$ is defined as

$$\|\mathcal{M}\|_{\mathcal{S}_p} = \left(\sum_{i=1}^{n_3} \left\|\overline{\mathcal{M}}^{(i)}\right\|_{\mathcal{S}_p}^p\right)^{\frac{1}{p}} = \left(\sum_{i=1}^{n_3}\sum_{j=1}^{h} \sigma_j\left(\overline{\mathcal{M}}^{(i)}\right)^p\right)^{\frac{1}{p}} \tag{5}$$

where $\sigma_j(\overline{\mathcal{M}}^{(i)})$ denotes the j-th singular value of $\overline{\mathcal{M}}^{(i)}$.

Remark 1 *when $p = 1$, tensor Schatten p-norm of $\mathcal{M} \in \mathbb{R}^{n_1\times n_2\times n_3}$ becomes tensor nuclear norm (Semerci et al. 2014; Gao et al. 2021; Xia et al. 2022b,a), i.e., $\|\mathcal{M}\|_* = \sum_{i=1}^{n_3}\sum_{j=1}^{h}\sigma_j\left(\overline{\mathcal{M}}^{(i)}\right)$. Take matrix Schatten p-norm as an example, that is for $\boldsymbol{M} \in \mathbb{R}^{n_1\times n_2}$ and the singular values of $\boldsymbol{M}$ denoted by $\sigma_1, \ldots, \sigma_h$, we have $\|\boldsymbol{M}\|_{\mathcal{S}_p}^p = \sigma_1^p + \cdots + \sigma_h^p$, $p > 0$. Nie et al. (Nie et al. 2012) has shown that $\lim_{p\to 0}\|\boldsymbol{M}\|_{\mathcal{S}_p}^p = \#\{i : \sigma_i \neq 0\} = rank(\boldsymbol{M})$. And for $0 \leq p \leq 1$, i.e. when p is appropriately chosen, the Schatten p-norm can give us quite effective improvements for a tighter approximation of the rank function.*

Remark 2 *The regularizer in the proposed objective (4) is used to explore the complementary information embedded in inter-views cluster assignment matrices $\mathbf{Y}^{(v)}$ ($v = 1, 2, \cdots, V$). Fig. 1 shows the construction of tensor $\mathcal{Y}$, it can be seen that the c-th frontal slice $\boldsymbol{\Delta}^{(c)}$ describes the similarity between N sample points and the c-th cluster in different views. The idea cluster assignment matrix $\boldsymbol{Y}^{(v)}$ should satisfy that the relationship between N data points and the c-th cluster is consistent in different views. Since different views usually show different cluster structures, we impose tensor Schatten p-norm minimization (Gao et al. 2021; Xia et al. 2022b,a) constraint on $\mathcal{Y}$, which can make sure each $\boldsymbol{\Delta}^{(c)}$ has spatial low-rank structure. Thus $\boldsymbol{\Delta}^{(c)}$ can well characterize the complementary information embedded in inter-views.*

Optimization

In (4), it difficult to solve the discrete cluster assignment due to the its non-convex property. To this end, we suppose the number of samples in each cluster are equal. In this case, $\mathbf{Y}^{(v)\mathrm{T}}\mathbf{Y}^{(v)} = \overline{n}\mathbf{I}$, where $\overline{n} = N/C$ is the cluster cardinality. Thus, (4) becomes

$$\begin{aligned}&\min_{\mathbf{Y}^{(v)},\,\alpha_v} \sum_{v=1}^{V}\alpha_v^r \mathrm{tr}(\mathbf{Y}^{(v)\mathrm{T}}\mathbf{H}^{(v)}\mathbf{Y}^{(v)}) + \lambda\|\mathcal{Y}\|_{\mathcal{S}_p}^p \\ &s.t. \qquad \mathbf{Y}^{(v)} \in Ind,\ \sum_{v=1}^{V}\alpha_v = 1, \alpha_v \geq 0\end{aligned} \tag{6}$$

Using augmented Lagrange multiplier (ALM) (Lin, Liu, and Su 2011), we introduce an auxiliary variable $\mathcal{J}$ and rewrite (6) as

$$\begin{aligned}\mathcal{L}(\mathcal{Y}, \mathcal{J}) = &\sum_{v=1}^{V}\alpha_v^r \mathrm{tr}(\mathbf{Y}^{(v)\mathrm{T}}\mathbf{H}^{(v)}\mathbf{Y}^{(v)}) + \lambda\|\mathcal{J}\|_{\mathcal{S}_p}^p \\ &+ \langle\mathcal{Q}, \mathcal{Y} - \mathcal{J}\rangle + \frac{\mu}{2}\|\mathcal{Y} - \mathcal{J}\|_F^2\end{aligned} \tag{7}$$

where $\mathcal{Q}$ is Lagrange multipliers; μ is a penalty parameter. The optimization process could be separated into the following three steps.

• **Solving $\mathcal{Y}$ with fixed α_v and $\mathcal{J}$.** When α_v and $\mathcal{J}$ are fixed, the optimization w.r.t. $\mathcal{Y}$ in (7) becomes

$$\min_{\mathbf{Y}^{(v)}\in Ind} \sum_{v=1}^{V}\alpha_v^r \mathrm{tr}(\mathbf{Y}^{(v)\mathrm{T}}\mathbf{H}^{(v)}\mathbf{Y}^{(v)}) + \frac{\mu}{2}\|\mathcal{Y} - \mathcal{J} + \frac{\mathcal{Q}}{\mu}\|_F^2 \tag{8}$$

Since all $\mathbf{Y}^{(v)}$ ($v = 1, \cdots, V$) are independent, then (8) can be decomposed into V independent sub-optimization problems. So we can obtain $\mathbf{Y}^{(v)}$ ($v = 1, \cdots, V$) of each view by solving

$$\min_{\mathbf{Y}^{(v)}\in Ind} \alpha_v^r \mathrm{tr}(\mathbf{Y}^{(v)\mathrm{T}}\mathbf{H}^{(v)}\mathbf{Y}^{(v)}) + \frac{\mu}{2}\|\mathbf{Y}^{(v)} - \mathbf{S}^{(v)}\|_F^2 \tag{9}$$

where $\mathbf{S}^{(v)} = \mathbf{J}^{(v)} - \frac{1}{\mu}\mathbf{Q}^{(v)}$.

In (9), it is hard to directly get the optimal solution due to the discrete values. Since all rows of $\mathbf{Y}^{(v)}$ are independent, we sequentially solve $\mathbf{Y}^{(v)}$ row by row with fixed the other rows. To update the *k*-th row, we assume the other rows of $\mathbf{Y}^{(v)}$ are known. Then, the optimization with respect to the *k*-th row in (9) becomes

$$\begin{aligned}&\min_{\mathbf{y}_k^{(v)}\in Ind} \alpha_v^r \sum_{i,j=1}^{N} h_{ij}^{(v)}\mathrm{tr}(\mathbf{y}_i^{(v)}\mathbf{y}_j^{(v)\mathrm{T}}) + \frac{\mu}{2}\|\mathbf{y}_k^{(v)} - \mathbf{s}_k^{(v)}\|_F^2 \\ \Leftrightarrow &\min_{\mathbf{y}_k^{(v)}\in Ind} \alpha_v^r \mathbf{y}_k^{(v)\mathrm{T}}(2\sum_{i=1,i\neq k}^{N} h_{ki}^{(v)}\mathbf{y}_i^{(v)}) - \mu\mathbf{y}_k^{(v)\mathrm{T}}\mathbf{s}_k^{(v)}\end{aligned} \tag{10}$$

Since $h_{kk}^{(v)} = 0$, thus, the problem (10) can be rewritten as

$$\begin{aligned}&\min_{\mathbf{y}_k^{(v)}\in Ind} 2\alpha_v^r \mathbf{y}_k^{(v)\mathrm{T}}(\sum_{i=1}^{N} h_{ki}^{(v)}\mathbf{y}_i^{(v)}) - \mu\mathbf{y}_k^{(v)\mathrm{T}}\mathbf{s}_k^{(v)} \\ \Leftrightarrow &\min_{\mathbf{y}_k^{(v)}\in Ind} \mathbf{y}_k^{(v)\mathrm{T}}(2\alpha_v^r \mathbf{Y}_*^{(v)\mathrm{T}}\mathbf{h}_k^{(v)} - \mu\mathbf{s}_k^{(v)})\end{aligned} \tag{11}$$

where $\mathbf{h}_k^{(v)} = [h_{k1}^{(v)}, h_{k2}^{(v)}, \cdots, h_{kn}^{(v)}]^{\mathrm{T}}$, $h_{ii}^{(v)} = 0$; $\mathbf{Y}_*^{(v)}$ is the cluster assignment matrix before $\mathbf{y}_k^{(v)}$ is updated. Then, the optimal solution of (8) can be reformulated as

$$y_{ip}^{(v)} = \begin{cases}1, & p = \arg\min_j (2\alpha_v^r \mathbf{Y}_*^{(v)\mathrm{T}}\mathbf{h}_i^{(v)} - \mu\mathbf{s}_i^{(v)})_j \\ 0, & \text{otherwise.}\end{cases} \tag{12}$$

• **Solving α_v with fixed $\mathcal{Y}$ and $\mathcal{J}$.** When $\mathcal{Y}$ and $\mathcal{J}$ are fixed, the optimization w.r.t. α_v in (7) is equivalent to

$$\min_{\alpha_v} \sum_{v=1}^{V}\alpha_v^r \mathrm{tr}(\mathbf{Y}^{(v)\mathrm{T}}\mathbf{H}^{(v)}\mathbf{Y}^{(v)}) \quad s.t. \quad \sum_{v=1}^{V}\alpha_v = 1, \alpha_v \geq 0 \tag{13}$$

According to the Lagrange multiplier method, we have the following Lagrangian function:

$$\begin{aligned}\mathcal{L}(\alpha_1, \ldots, \alpha_V, \gamma) = &\sum_{v=1}^{V}\alpha_v^r \mathbf{M}_v + \gamma(1 - \sum_{v=1}^{V}\alpha_v) \\ &+ \sum_{v=1}^{V}\mu_v(-\alpha_v)\end{aligned} \tag{14}$$

where γ and μ_v is parameters; $\mathbf{M}_v = \mathrm{tr}(\mathbf{Y}^{(v)^{\mathrm{T}}}\mathbf{H}^{(v)}\mathbf{Y}^{(v)})$. According to the KKT conditions, the optimal solution to (14) satisfies

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \alpha_v} = r\alpha_v^{r-1}\mathbf{M}_v - \gamma - \mu_v = 0, \quad v = 1, \ldots, V \\ \mu_v \alpha_v = 0, \quad v = 1, \ldots, V \\ \mu_v \geq 0, \quad v = 1, \ldots, V \\ 1 - \sum_{v=1}^{V} \alpha_v = 0 \end{cases} \tag{15}$$

then, it is simple to show that

$$\alpha_v = \frac{(\mathbf{M}_v)^{\frac{1}{1-r}}}{\sum_{v=1}^{V}(\mathbf{M}_v)^{\frac{1}{1-r}}} \tag{16}$$

• **Solving $\mathcal{J}$ with fixed $\mathcal{Y}$ and α_v.** In this case, $\mathcal{J}$ can be obtained by solving

$$\mathcal{J}^* = \arg\min_{\mathcal{J}} \frac{\lambda}{\mu}\|\mathcal{J}\|_{\circledS p}^p + \frac{1}{2}\|\mathcal{Y} - \mathcal{J} + \frac{\mathcal{Q}}{\mu}\|_F^2 \tag{17}$$

To solve (17), we first introduce Theorem 1 (Gao et al. 2021).

Theorem 1 *Given third-order tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ whose t-SVD denotes by $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^{\mathrm{T}}$. For the problem*

$$\arg\min_{\mathcal{X}} \mu \|\mathcal{X}\|_{\circledS p}^p + \frac{1}{2}\|\mathcal{X} - \mathcal{A}\|_F^2, \tag{18}$$

the optimal solution is

$$\mathcal{X}^* = \Gamma_\mu[\mathcal{A}] = \mathcal{U} * \mathrm{ifft}(P_\mu(\overline{\mathcal{A}})) * \mathcal{V}^{\mathrm{T}}, \tag{19}$$

where $P_\mu(\bar{\mathcal{A}}) \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is a f-diagonal tensor whose diagonal elements can be obtained by the GST algorithm introduced in Lemma 1 of (Gao et al. 2021).

According to Theorem 1, let $\mathcal{Y} + \frac{\mathcal{Q}}{\mu} = \mathcal{U} * \mathcal{\Sigma} * \mathcal{V}^{\mathrm{T}}$, then the solution of (17) is

$$\mathcal{J}^* = \Gamma_{\frac{\lambda}{\mu}}\left[\mathcal{Y} + \frac{\mathcal{Q}}{\mu}\right] = \mathcal{U} * \mathrm{ifft}(P_{\frac{\lambda}{\mu}}\left(\overline{\mathcal{Y} + \frac{\mathcal{Q}}{\mu}}\right)) * \mathcal{V}^{\mathrm{T}} \tag{20}$$

Finally, Algorithm 1 lists the pseudo code of solving (6).

$\mathbf{H}^{(v)}$ Construction

It is well known that *K*-Means cannot separate clusters well which are non-linearly separable in the input space due to the Euclidean distance between data. To address this problem, inspired by the advantage of the adjacency matrix, which can well characterize intrinsic structure of data with arbitrary shape, and good property of anchor graph, we use the small adjacency matrix $\mathbf{B}^{(v)} \in \mathbb{R}^{N \times \theta}$ $(v = 1, \ldots, V)$ to construct $\mathbf{H}^{(v)}$ $(v = 1, \ldots, V)$ which is called the adjacency matrix, where $\theta \ll N$ is the number of anchors. To be specific:

First, for a given multi-view dataset $\{\mathbf{X}^{(v)}\}_{v=1}^{V}$ with N samples, where $\mathbf{X}^{(v)} \in \mathbb{R}^{N \times d_v}$, we pick out θ samples as anchors. An efficient method named directly alternate sampling (DAS) (Li et al. 2020) are adopted to choose the anchors covering the entire point cloud of data. Let $\{\mathbf{A}^{(v)}\}_{v=1}^{V}$ denotes the anchors, where $\mathbf{A}^{(v)} \in \mathbb{R}^{\theta \times d_v}$.

Second, a parameter-free but effective bipartite graph construct strategy (Li et al. 2020) is applied to construct anchor graphs $\mathbf{B}^{(v)}$ with the gained anchors. The normalized and nonnegative anchor graph $\mathbf{B}^{(v)}$ for v-th view can be obtained by solving

$$\min_{\mathbf{b}^{i(v)}\mathbf{1}=1, \mathbf{b}^{i(v)} \geq \mathbf{0}} \sum_{j=1}^{\theta} b_{ij}^{(v)}\|\mathbf{x}_i^{(v)} - \mathbf{a}_j^{(v)}\|_2^2 + \gamma\|\mathbf{b}^{i(v)}\|_2^2 \tag{21}$$

where $\mathbf{b}^{i(v)}$ represents the i-th row of $\mathbf{B}^{(v)}$, γ is the regularization parameter. Let $p_{ij}^{(v)} = \|\mathbf{x}_i^{(v)} - \mathbf{a}_j^{(v)}\|_2^2$, the following derivation is provided to obtain the closed form solution to (21)

$$\min_{\mathbf{b}^{i(v)}\mathbf{1}=\mathbf{1}, \mathbf{b}^{i(v)} \geq \mathbf{0}} \frac{1}{2}\|\mathbf{b}^{i(v)} + \frac{\mathbf{p}^{i(v)}}{2\gamma}\|_2^2 \tag{22}$$

To solve (22), the Lagrangian function is introduced as follow

$$\begin{aligned} \mathcal{L}(\mathbf{b}^{i(v)}, \eta, \boldsymbol{\mu}) = & \frac{1}{2}\|\mathbf{b}^{i(v)} + \frac{\mathbf{p}^{i(v)}}{2\gamma}\|_2^2 \\ & + \eta(\mathbf{b}^{i(v)}\mathbf{1} - 1) + \mathbf{b}^{i(v)}\boldsymbol{\mu} \end{aligned} \tag{23}$$

where $\eta \in \mathbb{R}$ and $\boldsymbol{\mu} \in \mathbb{R}^{\theta \times 1}$ are the Lagrangian multipliers. By using the KKT conditions, we have:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \mathbf{b}^{i(v)}} = 0 \Rightarrow b_{ij}^{(v)} + \frac{p_{ij}^{(v)}}{2\gamma} - \eta - \mu_j = 0, \quad j = 1, \ldots, \theta \\ b_{ij}^{(v)} \geq 0, \quad j = 1, \ldots, \theta \\ \mu_j \geq 0, \quad j = 1, \ldots, \theta \\ b_{ij}^{(v)}\mu_j = 0 \end{cases} \tag{24}$$

Algorithm 1: Solving the Model (7)

Input: Data matrices $\{\mathbf{X}^{(v)}\}_{v=1}^{V} \in \mathbb{R}^{N \times d_v}$; anchors number θ; cluster number C.
Output: Cluster assignment matrix $\mathbf{K}$

Initialize: λ, Ω, r, p, $\mathcal{J}$, $\mathbf{H}^{(v)}$, $\mathbf{Y}^{(v)}$, $\alpha_v = \frac{1}{V}$, $(v = 1 \ldots V)$, $\rho = 1.1$, $\mu = 10^{-4}$, $\mu_{\max} = 10^{10}$.
Construct $\mathbf{H}^{(v)}$ by (27), $(v = 1 \ldots V)$;
while *not converge* **do**
  Update $\mathcal{Y}$ by solving (12);
  Update $\mathcal{J}$ by solving (17);
  Update $\mathcal{Q}$ by $\mathcal{Q} = \mathcal{Q} + \mu(\mathcal{Y} - \mathcal{J})$;
  Update α_v by (16), $(v = 1 \ldots V)$;
  Update μ by $\mu = \min(\rho\mu, \mu_{\max})$;
end
Calculate the cluster assignment matrix $\mathbf{K}$ by

$$k_{il} = \begin{cases} 1, & l = \arg\max_j \left(\sum_{v=1}^{V} \alpha_v^r \mathbf{Y}^{(v)}\right)_{ij} \\ 0, & \text{otherwise.} \end{cases}$$

return: The cluster assignment matrix $\mathbf{K}$

Let the obtained $\mathbf{b}^{i(v)}$ has exact k nonzero elements which means that i-th sample connects to its k-nearest anchors to satisfy the sparsity. By solving (24), we have $\eta = \frac{1}{k} + \frac{1}{2\gamma k}\sum_{j=1}^{k} p_{ij}^{(v)}$. Meanwhile, following (Li et al. 2020) γ is set as $\gamma = \frac{k}{2}p_{i,k+1}^{(v)} - \frac{1}{2}\sum_{j=1}^{k} p_{i,j}^{(v)}$ and then the solution of $b_{ij}^{(v)}$ in (21) is

$$b_{ij}^{(v)} = \begin{cases} \frac{\|\mathbf{x}_i^{(v)} - \mathbf{a}_{k+1}^{(v)}\|_2^2 - \|\mathbf{x}_i^{(v)} - \mathbf{a}_j^{(v)}\|_2^2}{k\|\mathbf{x}_i^{(v)} - \mathbf{a}_{k+1}^{(v)}\|_2^2 - \sum_{l=1}^{k}\|\mathbf{x}_i^{(v)} - \mathbf{a}_l^{(v)}\|_2^2} & j \leq k \\ 0 & j > k \end{cases} \quad (25)$$

Third, we calculate the symmetric and doubly-stochastic adjacency matrix $\mathbf{W}^{(v)}$ following (Liu, He, and Chang 2010), i.e.,

$$\mathbf{W}^{(v)} = \mathbf{B}^{(v)}{\Delta^{(v)}}^{-1}{\mathbf{B}^{(v)}}^{\mathrm{T}} \quad (26)$$

where $\Delta^{(v)} \in \mathbb{R}^{\theta \times \theta}$ is a diagonal matrix and $\Delta_{jj}^{(v)} = \sum_{i=1}^{N} b_{ij}^{v}$.

Finally, the i-th row j-th column element h_{ij}^{v} of $\mathbf{H}^{(v)}$ can be obtained by

$$h_{ij}^{(v)} = \sqrt{\frac{1}{1 + (\frac{w_{ij}^{(v)}}{\Omega})^4}} \quad (27)$$

which is a novel conversion function inspired by the Butterworth filters, and Ω is a hyperparameter.

Complexity Analysis

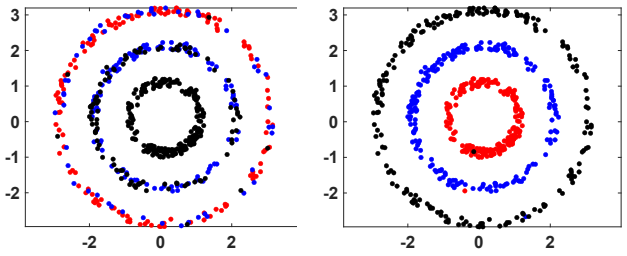
Our method consists of two stages: 1) Construction of $\{\mathbf{H}^{(v)}\}_{v=1}^{V}$, 2) Iterative updating (7). The first stage takes $\mathcal{O}(VN\theta d + VN\theta\log(\theta))$ for the anchor graph construction, where $d = \sum_{v=1}^{V} d_v$; V, θ and N are the number of views, anchors and samples, respectively. The second stage mainly focuses on solving two variables ($\mathbf{Y}^{(v)}$ and $\mathcal{J}$), the complexity in updating these two variables are $\mathcal{O}(NC)$ and $\mathcal{O}(VNC\log(VN) + V^2NC)$, where C is the number of clusters. For $\theta, C \ll N$, the main complexity in this stage is $\mathcal{O}(VNC\log(VN))$. Thus, the main computational complexity of our method is $\mathcal{O}(VN\theta d)$, which is linear to N.

Experiments

We evaluate our model on a toy dataset and six benchmark datasets through some experiments implemented on a Windows 10 desktop computer with a 2.40GHz Intel Xeon Gold 6240R CPU, 64 GB RAM, and MATLAB R2021a (64-bit).

Experiments on Toy Example

To verify the efficiency of our method for non-linearly separate clusters, we construct the three-ring dataset that has 2 views and 3 clusters, each cluster with 200 samples, and the second view is the FFT of the first. Fig. 2 shows the clustering results of our method with different distance matrices. One is from the Euclidean distance matrix (See Fig. 2 (a)), and the other is from the distance transformed by our function (27) (See Fig. 2 (b)). The visual result is in Fig. 2, and it can be seen that, compared with the Euclidean distance



(a) Euclidean distance (b) Our method

Figure 2: Clustering performance of our proposed method with different distance matrices.

matrix, the performance of our method is significantly improved when using (27) to calculate the distance matrix. It indicates that our method can separate clusters well which are non-linearly separable in the input space.

Experimental Settings

Datasets We do experiments on the following six widely used multi-view datasets to investigate the performance of our method.

- ***MSRC*** (Winn and Jojic 2005) includes 7 kinds of objects with 210 images. We choose the 24-dimension CM feature, 576-D HOG feature, 512-D GIST feature, 256-D LBP feature, 254- D CENT feature as 5 views.
- ***ORL***[2] includes 400 pictures of 40 people. Just as (Luo et al. 2018), we extract three types of features from the-dataset: 4096 dimensions intensity feature, 3304 dimensions LBP feature, and 6750 dimensions Gabor feature.
- ***HW*** (Dua and Graff 2019) includes 10 digits with 2,000 images generated from UCI machine learning repository. 76-D FOU feature, 216-D FAC feature, 47-D ZER feature and 6-D MOR feature are employed as 4 views.
- ***Mnist4*** (Deng 2012) includes 4 categories of handwritten digits, i.e., from digit 0 to 3, with 4,000 images. We adopt the 30-D ISO feature, 9-D LDA feature, and 30-D NPE feature as 3 views.
- ***Reuters*** (Apté, Damerau, and Weiss 1994) includes 6 categories with 18758 documents. The 21513-D English, 24892-D France, 34251-D German, 15506-D Italian and 11547-D Spanish are adopted as 5 views.
- ***NUS-WIDE*** (Chua et al. 2009) includes 31 categories with 30000 object images, and the 5 selected views are 64-D CH feature, 225-D CM feature, 144-D CORR feature, 73-D EDH feature and 128-D WT feature.

Methods for Comparison we select the following representative methods: (1) Classical single view spectral clustering **(SC)** (Ng, Jordan, and Weiss 2001) (2) Multi-view K-Means clustering on big data **(RMKMC)** (Cai, Nie, and Huang 2013); (3) Co-regularized multi-view SC **(Co-Reg)** (Kumar, Rai, and III 2011); (4) Consistent and specific

[2]http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

Datasets	MSRC			HW		
Methods	ACC	NMI	Purity	ACC	NMI	Purity
SC (best)	0.663	0.534	0.675	0.639	0.616	0.653
RMKMC	0.700	0.604	0.700	0.804	0.785	0.839
CSMSC	0.758	0.735	0.793	0.806	0.793	0.867
Co-Reg	0.635	0.578	0.659	0.784	0.758	0.795
MVGL	0.690	0.663	0.733	0.811	0.809	0.831
LTCPS	0.981	0.957	0.981	0.920	0.869	0.920
ETLMSC	0.962	0.937	0.962	0.938	0.893	0.938
Ours	**1.000**	**1.000**	**1.000**	**0.998**	**0.995**	**0.998**

Table 1: The results on MSRC and HW datasets.

Datasets	ORL			Mnist4		
Methods	ACC	NMI	Purity	ACC	NMI	Purity
SC (best)	0.727	0.868	0.762	0.713	0.558	0.713
RMKMC	0.543	0.749	0.620	0.895	0.739	0.895
CSMSC	0.857	0.935	0.882	0.643	0.645	0.832
Co-Reg	0.668	0.824	0.713	0.785	0.602	0.786
MVGL	0.765	0.871	0.815	0.912	0.785	0.910
LTCPS	0.981	0.994	0.983	0.929	0.813	0.929
ETLMSC	0.958	0.991	0.970	0.934	0.847	0.934
Ours	**1.000**	**1.000**	**1.000**	**0.988**	**0.955**	**0.988**

Table 2: The results on ORL and Mnist4 datasets.

Datasets	Reuters			NUS-WIDE		
Methods	ACC	NMI	Purity	ACC	NMI	Purity
SC (best)	0.269	0.002	0.272	0.131	0.019	0.140
RMKMC	0.422	0.259	0.531	0.125	0.123	0.221
CSMSC	OM	OM	OM	OM	OM	OM
Co-Reg	0.563	0.326	0.552	0.119	0.114	0.214
MVGL	0.271	0.021	0.281	OM	OM	OM
LTCPS	OM	OM	OM	OM	OM	OM
ETLMSC	OM	OM	OM	OM	OM	OM
Ours	**0.682**	**0.638**	**0.824**	**0.510**	**0.708**	**0.673**

Table 3: The results on Reuters and NUS-WIDE datasets.

multi-view subspace clustering **(CSMSC)** (Luo et al. 2018); (5) Graph learning for multi-view clustering **(MVGL)** (Zhan et al. 2018); (6) Low-rank tensor constrained co-regularized multi-view spectral clustering **(LTCPSC)** (Xu et al. 2020); (7) Essential tensor learning for multi-view spectral clustering **(ETLMSC)** (Wu, Lin, and Zha 2019).

Metrics The widely used 3 metrics are applied to evaluate the performance of our method, (1) Accuracy (ACC) (Cai, He, and Han 2005); (2) Normalized Mutual Information (NMI) (Estévez et al. 2009); (3) Purity (Varshavsky, Linial, and Horn 2005). For all metrics, the higher value means the better clustering performance.

Experimental Results

We do experiments on six datasets and present the metrics comparison of the above methods in Tables. 1, 2, and 3, and specially for SC (best), only the best results among each view are recorded that individually applying single view spectral clustering on. We set the anchor rate $\theta = 0.5$ on MSRC, ORL, Mnist4 and HW, $\theta = 0.1$ on Reuters and $\theta = 0.02$ on NUS-WIDE. The following can observed:

It is observed that multi-view clustering methods achieve better results than single-view methods because they exploit complementary information embedded in different views. Furthermore, our method outperforms not only single-view clustering methods such as SC but also state-of-the-art multi-view clustering algorithms. This is because our method is suitable for both linear and non-linear data. We also consider complementary information among views as well as the difference of each views.

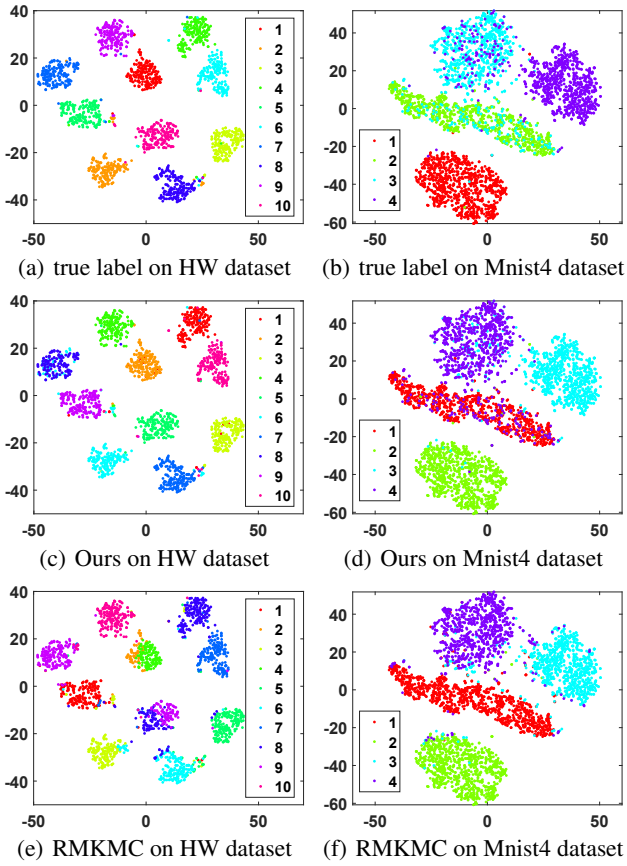
For two large scale datasets Reuters and NUS-WIDE, the out of memory issue occurs in some methods such as ETLMSC. Our method can obtain great results with setting low anchor rate, as shown in Table. 3, especially in NUS-WIDE, our method significantly and consistently outperforms all competitors, which proves the effectiveness on large scale datasets.

T-SNE Visualization T-distributed Stochastic Neighbor Embedding (t-SNE) (Van der Maaten and Hinton 2008) is used to embed the data into a 2-D or 3-D space while respecting relative distances between data samples to visualize high-dimensional data. To further illustrate the clustering performance of our method visually, we apply t-SNE to map the HW and Mnist4 datasets to a 2D plane and label samples of different clusters with different colors (See Fig. 3). On HW, our method divides samples into 10 clusters clearly, and the 4-th, 8-th, and 9-th clusters from RMKMC are dispersed obviously. On Mnist4, although t-SNE does not separate the original data well (shown in Fig. 3 (b)), our results are closer to the original data distribution, and the 4-th cluster from RMKMC is terrible. The reason may be that RMKMC cannot be applied to non-linear data and needs initializing cluster centroids, so the obtained results are unrobust.

Effect of Parameter r In (16), if $r \to \infty$, the weight assigned to each view will be equal, and if $r \to 1$, the weight of the view with the minimum value of $\mathbf{M}_v$ will be 1, and others will be 0. The strategy of using r not only avoids the trivial solution to the weight distribution of the different views but also controls the whole weights by exploiting one parameter.

Fig. 4 reveals the performances of our method on MSRC and ORL datasets by showing three metrics (ACC, NMI, Purity) at different r, and we set r from 3 to 10 with the interval of 1. It can be observed that when r is in the range of 3 to 9, the clustering performance changes less than 0.123, but the results on MSRC datasets decrease 0.272 when r is equal to 10. This confirms that choosing an appropriate r parameter helps to improve the clustering performance and r is insensitive within the range of 3 to 9.

Effect of Parameter p Taking MSRC and ORL datasets as examples, we analyze the effect of p on clustering perfor-

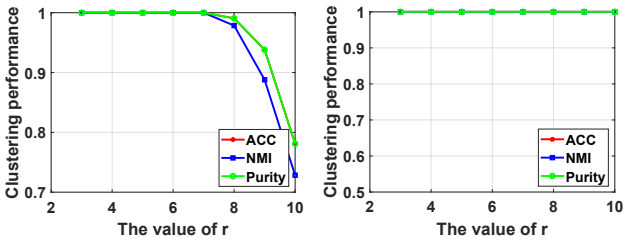


(a) true label on HW dataset (b) true label on Mnist4 dataset

(c) Ours on HW dataset (d) Ours on Mnist4 dataset

(e) RMKMC on HW dataset (f) RMKMC on Mnist4 dataset

Figure 3: t-SNE visualization on HW and Mnist4 datasets compared with RMKMC.

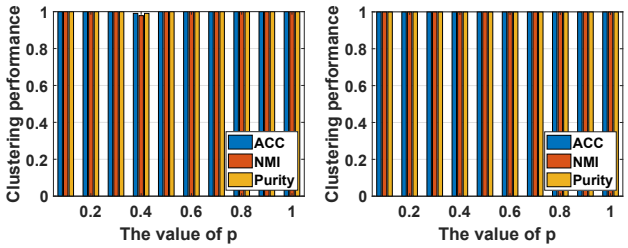


(a) MSRC (b) ORL

Figure 4: Clustering performance vs. r on MSRC and ORL datasets.

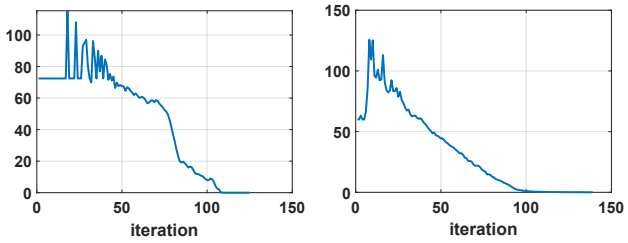
mance. Specifically, we change p from 0.1 to 1.0 with the interval of 0.1, then we report the ACC, NMI, and Purity as shown in Fig. 5. It is observed that the results under different p are close, and when $p = 0.4$, the results on MSRC datasets decrease less than 0.022. This demonstrates that p has some influence on clustering results and it is not sensitive to the results.

Convergence For MSRC and ORL datasets, we calculate the values of the function $\sum_{i=1}^{V} \|\mathbf{J}^{(v)} - \mathbf{Y}^{(v)}\|_F^2$ at each iteration step and record the results in Fig.6. It is observed that



(a) MSRC (b) ORL

Figure 5: Clustering performance vs. p on MSRC and ORL datasets.



(a) MSRC (b) ORL

Figure 6: The values of function with iterations on MSRC and ORL datasets.

the value of the objective function decreases as the number of iterations increases and converges within less than 150 steps eventually. This further indicates that our proposed algorithm can converge in real applications.

Conclusion

In summary, we present a novel centerless multi-view K-Means clustering method, which uses the affinity matrices of different views to construct the corresponding distance matrices. This makes our method suitable for both linearly and non-linearly clusters and eliminates the impact of the initializing cluster centroids. Meanwhile, our method is insensitive to noise points because it does not calculate clustering means. In addition, our method exploits the complementary information embedded in label matrices of different views. Extensive experiments on real-world datasets indicate the efficiency of our method.

Acknowledgments

The work of Han Lu and Quanxue Gao was supported in part by the National Natural Science Foundation of China under Grant 62176203, in part by the Natural Science Basic Research Plan in Shaanxi Province under Grant 2020JZ-19, in part by the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) under Grant 202200035, in part by the Natural Science Foundation of Shandong Province under Grant ZR202102180986, and in part by the Fundamental Research Funds for the Central Universities.

References

Apté, C.; Damerau, F.; and Weiss, S. M. 1994. Automated Learning of Decision Rules for Text Categorization. *ACM Trans. Inf. Syst.*, 12(3): 233–251.

Arthur, D.; and Vassilvitskii, S. 2007. k-means++: the advantages of careful seeding. In *SODA*, 1027–1035.

Cai, D.; He, X.; and Han, J. 2005. Document Clustering Using Locality Preserving Indexing. *IEEE Trans. Knowl. Data Eng.*, 17(12): 1624–1637.

Cai, X.; Nie, F.; and Huang, H. 2013. Multi-View K-Means Clustering on Big Data. In Rossi, F., ed., *IJCAI*, 2598–2604.

Chua, T.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; and Zheng, Y. 2009. NUS-WIDE: a real-world web image database from National University of Singapore. In *CIVR*.

Deng, L. 2012. The MNIST Database of Handwritten Digit Images for Machine Learning Research. *IEEE Signal Process. Mag.*, 29(6): 141–142.

Dua, D.; and Graff, C. 2019. UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA.

Elkan, C. 2003. Using the Triangle Inequality to Accelerate k-Means. In *ICML*, 147–153.

Estévez, P. A.; Tesmer, M.; Perez, C. A.; and Zurada, J. M. 2009. Normalized Mutual Information Feature Selection. *IEEE Trans. Neural Networks*, 20(2): 189–201.

Gao, Q.; Zhang, P.; Xia, W.; Xie, D.; Gao, X.; and Tao, D. 2021. Enhanced Tensor RPCA and its Application. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(6): 2133–2140.

Hartigan, J. A.; and Wong, M. A. 1979. Algorithm AS 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1): 100–108.

Kim, D.; Lee, K. Y.; Lee, D.; and Lee, K. H. 2005. Evaluation of the performance of clustering algorithms in kernel-induced feature space. *Pattern Recognit.*, 38(4): 607–611.

Kumar, A.; Rai, P.; and III, H. D. 2011. Co-regularized Multi-view Spectral Clustering. In *NeurIPS*, 1413–1421.

Li, X.; Zhang, H.; Wang, R.; and Nie, F. 2020. Multi-view Clustering: A Scalable and Parameter-free Bipartite Graph Fusion Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1): 330–344.

Lin, Z.; Liu, R.; and Su, Z. 2011. Linearized Alternating Direction Method with Adaptive Penalty for Low-Rank Representation. In *NeurIPS*, 612–620.

Liu, J.; Cao, F.; Gao, X.; Yu, L.; and Liang, J. 2020. A Cluster-Weighted Kernel K-Means Method for Multi-View Clustering. In *AAAI*, 4860–4867.

Liu, W.; He, J.; and Chang, S. 2010. Large Graph Construction for Scalable Semi-Supervised Learning. In *ICML*, 679–686.

Luo, S.; Zhang, C.; Zhang, W.; and Cao, X. 2018. Consistent and Specific Multi-View Subspace Clustering. In *AAAI*, 3730–3737.

Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2001. On Spectral Clustering: Analysis and an algorithm. In *NIPS*, 849–856.

Nie, F.; Wang, H.; Cai, X.; Huang, H.; and Ding, C. 2012. Robust Matrix Completion via Joint Schatten p-Norm and lp-Norm Minimization. In *ICDM*, 566–574.

Pei, S.; Nie, F.; Wang, R.; and Li, X. 2020. Efficient Clustering Based On A Unified View Of K-means And Ratio-cut. In *NeurIPS*.

Ren, Z.; Sun, Q.; and Wei, D. 2021. Multiple Kernel Clustering with Kernel k-Means Coupled Graph Tensor Learning. In *AAAI*, 9411–9418.

Semerci, O.; Hao, N.; Kilmer, M. E.; and Miller, E. L. 2014. Tensor-Based Formulation and Nuclear Norm Regularization for Multienergy Computed Tomography. *IEEE Trans. Image Process.*, 23(4): 1678–1693.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11): 2579–2605.

Varshavsky, R.; Linial, M.; and Horn, D. 2005. COMPACT: A Comparative Package for Clustering Assessment. In *ISPA Workshops*, volume 3759, 159–167.

Wang, R.; Lu, J.; Lu, Y.; Nie, F.; and Li, X. 2022. Discrete and Parameter-Free Multiple Kernel k-Means. *IEEE Trans. Image Process.*, 31: 2796–2808.

Winn, J. M.; and Jojic, N. 2005. LOCUS: Learning Object Classes with Unsupervised Segmentation. In *ICCV*, 756–763.

Wu, J.; Lin, Z.; and Zha, H. 2019. Essential Tensor Learning for Multi-View Spectral Clustering. *IEEE Trans. Image Process.*, 28(12): 5910–5922.

Xia, W.; Gao, Q.; Wang, Q.; and Gao, X. 2022a. Tensor Completion-Based Incomplete Multiview Clustering. *IEEE Transactions on Cybernetics*, 52(12): 13635–13644.

Xia, W.; Gao, Q.; Wang, Q.; Gao, X.; Ding, C.; and Tao, D. 2022b. Tensorized Bipartite Graph Learning for Multi-View Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–16.

Xia, W.; Zhang, X.; Gao, Q.; Shu, X.; Han, J.; and Gao, X. 2022c. Multiview Subspace Clustering by an Enhanced Tensor Nuclear Norm. *IEEE Transactions on Cybernetics*, 52(9): 8962–8975.

Xu, H.; Zhang, X.; Xia, W.; Gao, Q.; and Gao, X. 2020. Low-rank tensor constrained co-regularized multi-view spectral clustering. *Neural Networks*, 132: 245–252.

Zhan, K.; Zhang, C.; Guan, J.; and Wang, J. 2018. Graph Learning for Multiview Clustering. *IEEE Trans. Cybern.*, 48(10): 2887–2895.