Clustering Ensemble Meets Low-rank Tensor Approximation

Yuheng Jia[1,2], **Hui Liu**[*2], **Junhui Hou**[2], **Qingfu Zhang**[2]
[1]School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
[2]Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong SAR
yhjia@seu.edu.cn, hliu99-c@my.cityu.edu.hk, {jh.hou, qingfu.zhang}@cityu.edu.hk

Abstract

This paper explores the problem of clustering ensemble, which aims to combine multiple base clusterings to produce better performance than that of the individual one. The existing clustering ensemble methods generally construct a co-association matrix, which indicates the pairwise similarity between samples, as the weighted linear combination of the connective matrices from different base clusterings, and the resulting co-association matrix is then adopted as the input of an off-the-shelf clustering algorithm, e.g., spectral clustering. However, the co-association matrix may be dominated by poor base clusterings, resulting in inferior performance. In this paper, we propose a novel low-rank tensor approximation based method to solve the problem from a global perspective. Specifically, by inspecting whether two samples are clustered to an identical cluster under different base clusterings, we derive a coherent-link matrix, which contains limited but highly reliable relationships between samples. We then stack the coherent-link matrix and the co-association matrix to form a three-dimensional tensor, the low-rankness property of which is further explored to propagate the information of the coherent-link matrix to the co-association matrix, producing a refined co-association matrix. We formulate the proposed method as a convex constrained optimization problem and solve it efficiently. Experimental results over 7 benchmark data sets show that the proposed model achieves a breakthrough in clustering performance, compared with 11 state-of-the-art methods. *To the best of our knowledge, this is the first work to explore the potential of low-rank tensor on clustering ensemble, which is fundamentally different from previous approaches.* Last but not least, our method only contains one parameter, which can be easily tuned.

Introduction

Clustering is an important but very challenging unsupervised task, the goal of which is to partition a set of samples into homogeneous groups (Jia et al. 2018). Numerous applications can be formulated as a clustering problem, such as recommender systems (Song, Tekin, and Van Der Schaar 2014), community detection (Wu et al. 2018), and image segmentation (Li et al. 2019). Over the past decades, a large number of clustering techniques were proposed, e.g., K-means (Jain 2010), spectral clustering (Ulrike 2007), matrix factorization (Jia et al. 2020a; Wu et al. 2018; Jia et al. 2020c), hierarchical clustering (Johnson 1967), Gaussian mixture models (Moore 1999), and so on. As each method has its own advantages as well as drawbacks, no method could always outperform others (Vega-Pons and Ruiz-Shulcloper 2011). Additionally, a clustering method usually contains a few hyper-parameters, on which its performance heavily depends (Jia, Hou, and Kwong 2020). Moreover, the hyper-parameters are difficult to tune, and some methods are quite sensitive to initialization, like K-means. Those dilemmas increase the difficulty in choosing an appropriate clustering method for a typical clustering task. To this end, clustering ensemble was introduced, i.e., given a set of base clusterings produced by different methods or the same method with different hyper-parameters/initializations, it aims to generate a consensus clustering with better clustering performance than the base clusterings (Sagi and Rokach 2018; Boongoen and Iam-On 2018). Unlike supervised ensemble learning, clustering ensemble is more difficult (Tao et al. 2017, 2016), as the commonly used strategies in supervised ensemble learning, such as voting, cannot be directly applied to clustering ensemble, when labels of samples are unavailable.

To realize clustering ensemble, the existing methods generally first learn a pairwise relationship matrix from the base clusterings, and then apply off-the-shelf clustering methods like spectral clustering to the resulting matrix to produce the final clustering result (Tao, Liu, and Fu 2017). Based on how to generate the pairwise relationship matrix, we roughly divide the existing methods into two categories. (1) The first kind of methods treat the base clusterings as new feature representations (as shown in Fig. 1-A), to learn a pairwise relationship matrix. For example, (Gao et al. 2016) formulated clustering ensemble as a convex low-rank matrix representation problem. (Zhou, Zheng, and Pan 2019) used a Frobenius norm regularized self-representation model to seek a dense affinity matrix for clustering ensemble. (2) The second kind of methods rely on the co-association matrix (as shown in Fig. 1-C), which summarizes the co-occurrence of samples in the same cluster of the base clusterings. The concept of co-association matrix was first proposed by (Fred and Jain 2005), and since then it became popular as an important fundamental method in clustering ensemble. (Liu et al. 2017) theoretically bridged the co-association based method

*Corresponding author.

to weighted K-means clustering, which largely reduces the computational complexity. Recently, many advanced co-association matrix construction methods were proposed. For example, (Huang, Wang, and Lai 2018) considered the uncertainty of each base clustering and proposed a locally weighted co-association matrix. (Huang et al. 2018) used the cluster-wise similarities to enhance the traditional co-association matrix. (Zhou et al. 2020) proposed a self-paced strategy to learn the co-association matrix. See the detailed discussion about the related works in the next section. We observe that, the constructed co-association matrices of the prior works are variants of a weighted linear combination of the connective matrices (as shown in Fig. 1-B) from different base clusterings. When the performance of some base clusterings are poor, they will dominate the co-association matrix and degrade the clustering performance severely.

In this paper, we propose a novel constrained low-rank tensor approximation (LTA) model to refine the co-association matrix from a global perspective. Specifically, as shown in Fig. 1-D, we first construct a coherent-link matrix, whose element examines whether two samples are from the same cluster in all the base clusterings or not. We then stack the coherent-link matrix and the conventional co-association matrix to form a 3-dimensional (3-D) tensor shown in Fig. 1-E, which is further low-rank approximated. By exploring the low-rankness, the proposed model can propagate the highly reliable information of the coherent-link matrix to the co-association matrix, producing a refined co-association matrix, which is adopted as the input of an off-the-shelf clustering method to produce the final clustering result. Technically, the proposed model is formulated as a convex optimization problem and solved by an alternative iterative method. We evaluate the proposed model on 7 benchmark data sets, and compare it with 11 state-of-the-art clustering ensemble methods. The experimental comparisons substantiate that the proposed model significantly outperforms state-of-the-art methods. *To the best of our knowledge, this is the first work to explore the potential of low-rank tensor on clustering ensemble.*

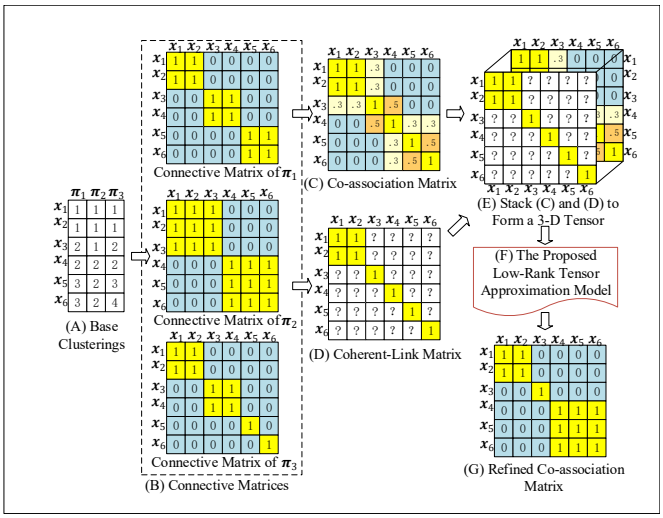

Figure 1: Illustration of the proposed method by taking 3 base clusterings denoted by π_1, π_2 and π_3, and 6 input samples denoted by $\mathbf{x}_1, \cdots, \mathbf{x}_6$ as an example. By exploring the low-rankness of the formed 3-D tensor, the limited but highly reliable information contained in the coherent-link matrix can be leveraged to enhance the quality of the co-association matrix.

Related Work

Notation. We denote tensors by boldface swash letters, e.g., $\boldsymbol{\mathcal{A}}$, matrices by boldface capital letters, e.g., $\mathbf{A}$, vectors by boldface lowercase letters, e.g., $\mathbf{a}$, and scalars by lowercase letters, e.g., a. Let $\boldsymbol{\mathcal{A}}(i,j,k)$ denote the (i,j,k)-th element of 3-D tensor $\boldsymbol{\mathcal{A}}$, $\mathbf{A}(i,j)$ denote the (i,j)-th element of matrix $\mathbf{A}$, and $\mathbf{a}(i)$ denote the i-th entry of vector $\mathbf{a}$. The i-th frontal slice of tensor $\boldsymbol{\mathcal{A}}$ is denoted as $\boldsymbol{\mathcal{A}}(:,:,i)$.

Rank of tensors. In this paper, we use the tensor nuclear norm induced by tensor singular value decomposition (t-SVD) (Kilmer et al. 2013) to measure the rank of a tensor. Specifically, the t-SVD of a 3-D tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ can be represented as

$$\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{U}} * \boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}}^{\mathsf{T}}, \tag{1}$$

where $\boldsymbol{\mathcal{U}} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\boldsymbol{\mathcal{V}} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are two orthogonal tensors, $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ is an f-diagonal tensor,

Algorithm 1 t-SVD of a 3-D tensor (Zhang et al. 2014)

Input: 3-D tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$.

1: Perform FFT on $\boldsymbol{\mathcal{A}}$, i.e., $\boldsymbol{\mathcal{A}}_f = \text{fft}(\boldsymbol{\mathcal{A}}, [\,], 3)$;
2: **for** $k = 1 : n_3$; **do**
3: Perform SVD on each frontal slice of $\boldsymbol{\mathcal{A}}_f$, i.e., [U,S,V]=SVD($\boldsymbol{\mathcal{A}}_f(:,:,k)$) ;
4: $\boldsymbol{\mathcal{U}}_f(:,:,k) = \mathbf{U}, \boldsymbol{\mathcal{S}}_f(:,:,k) = \mathbf{S}, \boldsymbol{\mathcal{V}}_f(:,:,k) = \mathbf{V}$;
5: **end**
6: Perform inverse FFT on $\boldsymbol{\mathcal{U}}_f$, $\boldsymbol{\mathcal{S}}_f$ and $\boldsymbol{\mathcal{V}}_f$, i.e., $\boldsymbol{\mathcal{U}} = \text{ifft}(\boldsymbol{\mathcal{U}}_\text{f}, [], 3), \boldsymbol{\mathcal{S}} = \text{ifft}(\boldsymbol{\mathcal{S}}_\text{f}, [], 3)$ and $\boldsymbol{\mathcal{V}} = \text{ifft}(\boldsymbol{\mathcal{V}}_\text{f}, [], 3)$;

Output: $\boldsymbol{\mathcal{U}}$, $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{V}}$.

$*$ and $\cdot^{\mathsf{T}}$ denote tensor product and tensor transpose, respectively. The detailed definitions of the above-mentioned tensor related operators can be find in (Zhang et al. 2014). Since the tensor product can be efficiently computed in the Fourier domain (Kilmer et al. 2013), the t-SVD form of a tensor can be obtained with fast Fourier transform (FFT) efficiently as shown Algorithm 1. Given t-SVD, the tensor nuclear norm (Zhang et al. 2014) is defined as the sum of the absolute values of the diagonal entries of $\boldsymbol{\mathcal{S}}$, i.e.,

$$\|\boldsymbol{\mathcal{A}}\|_{\circledast} = \sum_{i=1}^{\min(n_1,n_2)} \sum_{k=1}^{n_3} |\boldsymbol{\mathcal{S}}(i,i,k)|. \tag{2}$$

Formulation of Clustering Ensemble. Given a data set $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ of n samples with each sample $\mathbf{x}_i \in \mathbb{R}^{d \times 1}$, and m base clusterings $\boldsymbol{\Pi} = [\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \cdots, \boldsymbol{\pi}_m] \in \mathbb{R}^{n \times m}$, where each base clustering $\boldsymbol{\pi}_i \in \mathbb{R}^{n \times 1}$ is an n-dimensional vector with the j-th element $\boldsymbol{\pi}_i(j)$ indicating the clustering membership of the j-th sample $\mathbf{x}_j$ in $\boldsymbol{\pi}_i$. For clustering ensemble, the cluster indicators

in different base clusterings are generally different. Fig. 1-A shows a toy example of 6 samples and 3 base clusterings. The objective of clustering ensemble is to combine multiple base clusterings to produce better performance than that of the individual one.

Prior Art. Based on how to use $\mathbf{\Pi}$, we roughly divide previous clustering ensemble methods into two categories. The methods in the first category treat $\mathbf{\Pi}$ as a representation of samples and then construct a pairwise affinity matrix $\mathbf{P} \in \mathbb{R}^{n\times n}$ accordingly, which can be generally expressed as

$$\min_{\mathbf{P}} f(\mathbf{\Pi}, \mathbf{P}) + \lambda\phi(\mathbf{P}), \tag{3}$$

where $f(\mathbf{\Pi}, \mathbf{P})$ is the fidelity term and $\phi(\mathbf{P})$ imposes specific regularization on $\mathbf{P}$. For example, in (Gao et al. 2016), $f(\cdot,\cdot)$ and $\phi(\cdot)$ denote the Frobenius norm and the nuclear norm, respectively, while those in (Zhou, Zheng, and Pan 2019) are both the Frobenius norm. The second kind of methods first transform each base clustering as a connective matrix (as shown in Fig. 1-B), i.e.,

$$\mathbf{A}_k(i,j) = \delta(\pi_k(i), \pi_k(j)), \tag{4}$$

where $\mathbf{A}_k \in \mathbb{R}^{n\times n}$ is the k-th connective matrix constructed from $\boldsymbol{\pi}_k$, and

$$\delta(\pi_k(i), \pi_k(j)) = \begin{cases} 1 & \text{if } \pi_k(i) = \pi_k(j) \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

And then, the methods in the second category build a co-association matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ (Fred and Jain 2005) according to the connective matrices, i.e.,

$$\mathbf{A}(i,j) = \frac{1}{m}\sum_{k=1}^{m} \mathbf{A}_k(i,j). \tag{6}$$

As the co-association matrix naturally converts the base clusterings to a pairwise similarity measure, it becomes the cornerstone of clustering ensemble. Recently, many advanced co-association matrix construction methods were proposed to enhance the clustering performance, which can be generally unified in the following formula:

$$\mathbf{A}(i,j) = \sum_{k=1}^{m} \boldsymbol{\omega}(k) \times \mathbf{A}_k(i,j), \tag{7}$$

where $\boldsymbol{\omega} \in \mathbb{R}^{m\times 1}$ is the weight vector constructed with different strategies. For example, (Zhou et al. 2020) used a self-paced learning strategy to construct $\boldsymbol{\omega}$. (Huang, Wang, and Lai 2018) considered the uncertainties of the base clustering, and proposed a locally-weighted weight vector. (Huang et al. 2018) used the cluster-wise similarities to construct the weight vector.

Proposed Method

As shown in Eq. (7), the previous methods construct a co-association matrix as the linear combination of connective matrices, and thus are vulnerable to some poor base clusterings. To this end, we propose a novel low-rank tensor approximation based method to refine the initial co-association matrix from a global perspective.

Problem Formulation

To refine the co-association matrix, we first construct a coherent-link matrix (as shown in Fig. 1-D), which inspects whether two samples are clustered to the same category under all the base clusterings. It is worth pointing out that the elements of the coherent-link matrix are highly reliable information we could infer from the base clusteirngs. Specifically, we could directly get the coherent-link matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ from the co-association matrix in Eq. (6), i.e,

$$\mathbf{M}(i,j) = \begin{cases} 1 & \text{if } \mathbf{A}(i,j) = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

We then stack the coherent-link matrix and the co-association matrix to form a 3-D tensor $\boldsymbol{\mathcal{P}} \in \mathbb{R}^{n\times n\times 2}$, with $\boldsymbol{\mathcal{P}}(:,:,1) = \mathbf{M}$, and $\boldsymbol{\mathcal{P}}(:,:2) = \mathbf{A}$. As the elements of both the coherent-link matrix and the co-association matrix express the pairwise similarity between samples, ideally, the formed tensor should be low-rank. Moreover, the non-one elements of $\mathbf{M}$ are limited but express the highly reliable similarity between samples, and we thus try to complement the zero elements with reference to the non-zero ones and the co-association matrix. On the contrary, the elements of the co-association matrix is dense but with many error connections, and we try to refine it by removing the incorrect connections which is depicted by $\mathbf{E} \in \mathbb{R}^{n\times n}$, by leveraging the information from the coherent-link matrix. In addition, the elements of $\boldsymbol{\mathcal{P}}$ should be bounded in $[0,1]$, and each frontal slice of $\boldsymbol{\mathcal{P}}$ should be symmetric. Taking all the above analyses into account, the proposed method is mathematically formulated as a constrained optimization problem, written as

$$\begin{aligned} &\min_{\boldsymbol{\mathcal{P}},\mathbf{E}} \|\boldsymbol{\mathcal{P}}\|_{\circledast} + \lambda\|\mathbf{E}\|_F^2 \\ &\text{s.t. } \boldsymbol{\mathcal{P}}(i,j,1) = \mathbf{M}(i,j), \text{ if } \mathbf{M}(i,j) = 1, \\ &\quad \boldsymbol{\mathcal{P}}(:,:,1) = \boldsymbol{\mathcal{P}}(:,:,1)^{\mathsf{T}}, 0 \le \boldsymbol{\mathcal{P}}(i,j,1) \le 1, \forall i,j, \\ &\quad \boldsymbol{\mathcal{P}}(:,:,2) + \mathbf{E} = \mathbf{A}, \\ &\quad \boldsymbol{\mathcal{P}}(:,:,2) = \boldsymbol{\mathcal{P}}(:,:,2)^{\mathsf{T}}, 0 \le \boldsymbol{\mathcal{P}}(i,j,2) \le 1, \forall i,j, \end{aligned} \tag{9}$$

where $\lambda > 0$ is the coefficient to balance the error matrix, and a Frobenius norm is imposed on $\mathbf{E}$ to avoid trivial solution, i.e., $\boldsymbol{\mathcal{P}}(:,:,2) = 0$. By optimizing Eq. (9), it is expected that the limited but highly reliable information in $\mathbf{M}$ could be propagated to the co-association matrix, while the coherent-link matrix is complemented according to the information from the co-association matrix at the same time.

After solving the problem in Eq. (9), we can obtain a refined co-association matrix $\boldsymbol{\mathcal{P}}^*(:,:,2)$ with $\mathbf{P}^*$ being the optimized solution. Then, one can apply any clustering methods based on pairwise similarity on $\boldsymbol{\mathcal{P}}^*(:,:,2)$ to generate the final clustering result. In this paper, we investigate two popular clustering methods, i.e., spectral clustering (Ng, Jordan, and Weiss 2002) and agglomerative hierarchical clustering (Fred and Jain 2005).

Numerical Solution

We propose an optimization method to solve Eq. (9), based on the inexact Augmented Lagrangian method (Jia, Kwong,

and Hou 2018). Specifically, we first introduce two auxiliary matrices $\mathbf{B}, \mathbf{C} \in \mathbb{R}^{n\times n}$ to deal with the bounded and symmetric constraints on $\boldsymbol{\mathcal{P}}(:,:,1)$ and $\boldsymbol{\mathcal{P}}(:,:,2)$, respectively, and Eq. (9) can be equivalently rewritten as

$$\begin{aligned}&\underset{\boldsymbol{\mathcal{P}},\mathbf{E},\mathbf{B},\mathbf{C}}{\operatorname{argmin}} \|\boldsymbol{\mathcal{P}}\|_{\circledast} + \lambda\|\mathbf{E}\|_F^2 \\ &\text{s.t. } \mathbf{B}(i,j) = \mathbf{M}(i,j), \text{ if } \mathbf{M}(i,j) = 1,\ \mathbf{B} = \mathbf{B}^\mathsf{T}, \\ &\quad 0 \le \mathbf{B}(i,j) \le 1, \forall i,j,\ \mathbf{B} = \boldsymbol{\mathcal{P}}(:,:,1), \\ &\quad \boldsymbol{\mathcal{P}}(:,:,2) + \mathbf{E} = \mathbf{A},\ \mathbf{C} = \boldsymbol{\mathcal{P}}(:,:,2), \\ &\quad \mathbf{C} = \mathbf{C}^\mathsf{T},\ 0 \le \mathbf{C}(i,j) \le 1, \forall i,j.\end{aligned} \tag{10}$$

To handle the equality constraints, we introduce three Lagrange multipliers $\boldsymbol{\Lambda}_1, \boldsymbol{\Lambda}_2$ and $\boldsymbol{\Lambda}_3 \in \mathbb{R}^{n\times n}$, and the augmented Lagrangian form of Eq. (10) becomes

$$\begin{aligned}&\underset{\boldsymbol{\mathcal{P}},\mathbf{E},\mathbf{B},\mathbf{C}}{\operatorname{argmin}} \|\boldsymbol{\mathcal{P}}\|_{\circledast} + \lambda\|\mathbf{E}\|_F^2 + \frac{\mu}{2}\left\|\boldsymbol{\mathcal{P}}(:,:,2) + \mathbf{E} - \mathbf{A} + \frac{\boldsymbol{\Lambda}_2}{\mu}\right\|_F^2 \\ &+ \frac{\mu}{2}\left\|\boldsymbol{\mathcal{P}}(:,:,1) - \mathbf{B} + \frac{\boldsymbol{\Lambda}_1}{\mu}\right\|_F^2 + \frac{\mu}{2}\left\|\boldsymbol{\mathcal{P}}(:,:,2) - \mathbf{C} + \frac{\boldsymbol{\Lambda}_3}{\mu}\right\|_F^2 \\ &\text{s.t. } \mathbf{B}(i,j) = \mathbf{M}(i,j), \text{ if } \mathbf{M}(i,j) = 1, 0 \le \mathbf{B}(i,j) \le 1, \forall i,j, \\ &\quad \mathbf{B} = \mathbf{B}^\mathsf{T}, \mathbf{C} = \mathbf{C}^\mathsf{T},\ 0 \le \mathbf{C}(i,j) \le 1, \forall i,j,\end{aligned} \tag{11}$$

where $\mu > 0$ is the penalty coefficient. Then Eq. (11) can be optimized by solving the following four subproblems iteratively and alternately, i.e., only one variable is updated with the remaining ones fixed at each time.

The $\boldsymbol{\mathcal{P}}$ subproblem. Removing the irrelevant terms, Eq. (11) with respect to $\boldsymbol{\mathcal{P}}$ is written as

$$\underset{\boldsymbol{\mathcal{P}}}{\operatorname{argmin}} \frac{1}{\mu}\|\boldsymbol{\mathcal{P}}\|_{\circledast} + \frac{1}{2}\|\boldsymbol{\mathcal{P}} - \boldsymbol{\mathcal{T}}\|_F^2, \tag{12}$$

where

$$\begin{cases}\boldsymbol{\mathcal{T}}(:,:,1) = \mathbf{B} - \frac{\boldsymbol{\Lambda}_1}{\mu} \\ \boldsymbol{\mathcal{T}}(:,:,2) = \frac{1}{2}\left(\mathbf{A} + \mathbf{C} - \mathbf{E} - \frac{\boldsymbol{\Lambda}_2 + \boldsymbol{\Lambda}_3}{\mu}\right).\end{cases} \tag{13}$$

According to (Zhang et al. 2014), Eq. (12) has a closed-form solution with the soft-thresholding operator of the tensor singular values. Moreover, according to Algorithm 1, t-SVD computes FFT and SVD on the frontal slices of the input 3-D tensor $\boldsymbol{\mathcal{T}}(:,:,i)$ and its FFT version $\boldsymbol{\mathcal{T}}_f(:,:\ .i)$, respectively, which mainly emphasizes the low-rankness of the frontal slices. Differently, we aim to take advantage of the correction between the original co-association matrix and the coherent-link matrix. Therefore, we perform FFT and SVD on the lateral slices of the tensors $\boldsymbol{\mathcal{T}}(:,i,:)$, and $\boldsymbol{\mathcal{T}}_f(:,i,:)$, respectively, to get the t-SVD representation.

The E subproblem. Without the irrelevant terms, the $\mathbf{E}$ subproblem becomes:

$$\min_{\mathbf{E}} \lambda\|\mathbf{E}\|_F^2 + \frac{\mu}{2}\left\|\boldsymbol{\mathcal{P}}(:,:,2) + \mathbf{E} - \mathbf{A} + \frac{\boldsymbol{\Lambda}_2}{\mu}\right\|_F^2. \tag{14}$$

Since Eq. (14) is quadratic function of $\mathbf{E}$, we can get its global minimum by setting the derivative of it to 0, i.e.,

$$\mathbf{E} = \frac{\mu\mathbf{A} - \boldsymbol{\Lambda}_2 - \mu\boldsymbol{\mathcal{P}}(:,:,2)}{2\lambda + \mu}. \tag{15}$$

The B subproblem. The $\mathbf{B}$ subproblem is written as

$$\begin{aligned}&\min_{\mathbf{B}} \frac{\mu}{2}\left\|\mathbf{B} - \left(\boldsymbol{\mathcal{P}}(:,:,1) + \frac{\boldsymbol{\Lambda}_1}{\mu}\right)\right\|_F^2 \\ &\text{s.t. } \mathbf{B}(i,j) = \mathbf{M}(i,j), \text{ if } \mathbf{M}(i,j) = 1, \\ &\quad \mathbf{B} = \mathbf{B}^\mathsf{T},\ 0 \le \mathbf{B}(i,j) \le 1, \forall i,j,\end{aligned} \tag{16}$$

which is a symmetric and bounded constrained least squares problem, and has an optimal solution in element-wise (Jia et al. 2020d), i.e.,

$$\mathbf{B}(i,j) = \begin{cases}\mathbf{M}(i,j) & \text{if } \mathbf{M}(i,j) = 1, \\ 0 & \text{if } \mathbf{T}_1(i,j) \le 0 \ \&\ \mathbf{M}(i,j) \ne 1, \\ 1 & \text{if } \mathbf{T}_1(i,j) \ge 1 \ \&\ \mathbf{M}(i,j) \ne 1, \\ \mathbf{T}_1(i,j) & \text{if } 0 \le \mathbf{T}_1(i,j) \le 1 \ \&\ \mathbf{M}(i,j) \ne 1,\end{cases} \tag{17}$$

where

$$\mathbf{T}_1 = \frac{1}{2}\left(\boldsymbol{\mathcal{P}}(:,:,1) + \boldsymbol{\mathcal{P}}(:,:,1)^\mathsf{T} + \frac{\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_1^\mathsf{T}}{\mu}\right). \tag{18}$$

The C subproblem. The $\mathbf{C}$ subproblem is identical to the $\mathbf{B}$ subproblem without a set of element-wise equality constraints, which is written as

$$\begin{aligned}&\min_{\mathbf{C}} \frac{\mu}{2}\left\|\mathbf{C} - \left(\boldsymbol{\mathcal{P}}(:,:,2) + \frac{\boldsymbol{\Lambda}_3}{\mu}\right)\right\|_F^2 \\ &\text{s.t. } \mathbf{C} = \mathbf{C}^\mathsf{T},\ 0 \le \mathbf{C}(i,j) \le 1, \forall i,j,\end{aligned} \tag{19}$$

and the optimal solution of it is

$$\mathbf{C}(i,j) = \begin{cases}\mathbf{T}_2(i,j) & \text{if } 0 \le \mathbf{T}_2(i,j) \le 1, \\ 0 & \text{if } \mathbf{T}_2(i,j) \le 0, \\ 1 & \text{if } \mathbf{T}_2(i,j) \ge 1,\end{cases} \tag{20}$$

where

$$\mathbf{T}_2 = \frac{1}{2}\left(\boldsymbol{\mathcal{P}}(:,:,2) + \boldsymbol{\mathcal{P}}(:,:,2)^\mathsf{T} + \frac{\boldsymbol{\Lambda}_3 + \boldsymbol{\Lambda}_3^\mathsf{T}}{\mu}\right). \tag{21}$$

Update $\boldsymbol{\Lambda}_1$, $\boldsymbol{\Lambda}_2$, $\boldsymbol{\Lambda}_3$ and μ The Lagrange multipliers and μ are updated by

$$\begin{cases}\boldsymbol{\Lambda}_1 = \boldsymbol{\Lambda}_1 + \mu(\boldsymbol{\mathcal{P}}(:,:,1) - \mathbf{B}) \\ \boldsymbol{\Lambda}_2 = \boldsymbol{\Lambda}_2 + \mu(\boldsymbol{\mathcal{P}}(:,:,2) + \mathbf{E} - \mathbf{A}) \\ \boldsymbol{\Lambda}_3 = \boldsymbol{\Lambda}_3 + \mu(\boldsymbol{\mathcal{P}}(:,:,2) - \mathbf{C}) \\ \mu = \min(1.1\mu, \mu_{\max}),\end{cases} \tag{22}$$

where μ is initialized to 0.0001 (Liu et al. 2019), and $\mu_{\max}$ is the upper-bound for μ. The overall numerical solution is summarized in Algorithm 2, where the stopping conditions is $\max(\|\mathbf{B} - \boldsymbol{\mathcal{P}}(:,:,1)\|_\infty, \|\mathbf{C} - \boldsymbol{\mathcal{P}}(:,:,2)\|_\infty, \|\mathbf{A} - \mathbf{E} - \boldsymbol{\mathcal{P}}(:,:,2)\|_\infty) < 10^{-8}$ with $\|\cdot\|_\infty$ being the maximum of the absolute values of a matrix.

Experiment

We conducted extensive experiments to evaluate the proposed model. To reproduce the results, we made the code publicly available at https://github.com/jyh-learning/TensorClusteringEnsemble.

Algorithm 2 Numerical solution to Eq. (9)

Input: Base clusterings matrix $\mathbf{\Pi}$;
Initialize: $\mathcal{P} = 0, \mathbf{E} = 0, \mathbf{B} = 0, \mathbf{C} = 0$, and $\mu_{\max} = 10^8$;
Construct the co-association matrix $\mathbf{A}$ by Eq. (6);
Construct the coherent-link matrix $\mathbf{M}$ by Eq. (8);
while not converged **do**
  Update $\mathcal{P}$ by solving Eq. (12);
  Update $\mathbf{E}$ by Eq. (15);
  Update $\mathbf{B}$ by Eq. (17);
  Update $\mathbf{C}$ by Eq. (20);
  Update $\mathbf{\Lambda}_1$, $\mathbf{\Lambda}_2$, $\mathbf{\Lambda}_3$ and μ by Eq. (22);
  Check the convergence conditions;
end while
Output: $\mathcal{P}(:, :, 2)$ as the refined co-association matrix.

Data Sets. Following recent clustering ensemble papers (Huang, Wang, and Lai 2018; Huang, Lai, and Wang 2016; Zhou, Zheng, and Pan 2019), we adopted 7 commonly used data sets, i.e., BinAlpha, Multiple features (MF), MNIST, Semeion, CalTech, Texture and ISOLET. Following (Huang, Wang, and Lai 2018), we randomly selected 5000 samples from MNIST and used the subset in the experiments, and for CalTech, we used 20 representative categories out of 101 categories and denoted it as CalTech20.

Generation of Base Clusterings. Following (Huang, Wang, and Lai 2018), we first generated a pool of 100 candidate base clusterings for all the data sets by applying the the K-means algorithm with the value of K randomly varying in the range of $[2, \sqrt{n}]$, where n is the number of input data samples.

Methods under Comparison. We compared the proposed model with 11 state-of-the-art clustering ensemble methods, including TA-CL, TA-SL and PTGP (Huang, Lai, and Wang 2016), LWSC, LWEA and LWGP (Huang, Wang, and Lai 2018), E-HC and E-MC (Huang et al. 2018), DREC (Zhou, Zheng, and Pan 2019), SPCE (Zhou et al. 2020), and SEC (Liu et al. 2017). The codes of all the compared methods are provided by the authors. Ours-EA and Ours-SC denote the proposed model equipped with agglomerative hierarchical clustering and spectral clustering, respectively, to generate the final clustering result.

Evaluation Metrics. We adopted 7 commonly used metrics to evaluate clustering performance, i.e., clustering accuracy (ACC), normalized mutual information (NMI), purity, adjust rand index (ARI), F1-score, precision, and recall. For all the metrics, a larger value indicates better clustering performance, and the values of all the metrics are up-bounded by 1. The detailed definitions of those metrics can be found in (Zhang et al. 2020; Jia et al. 2020b).

Experiment Settings. For each data set, we randomly selected 10 base clusterings from the candidate base clustering pool, and performed different clustering ensemble methods on the selected base clusteirngs. To reduce the influence of the selected base clusterings, we repeated the random selection 20 times, and reported the average performance

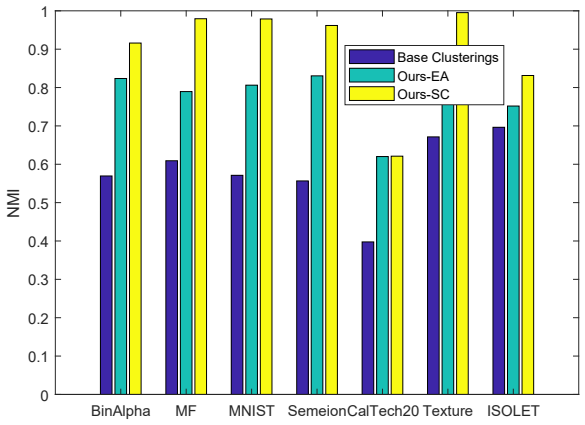

Figure 2: The NMI of our methods against the average NMI of the base clusteings in the candidate base clustering pool.

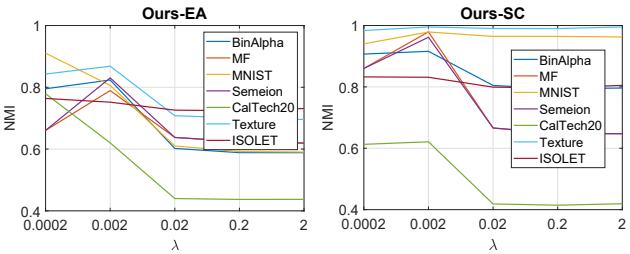

Figure 3: The NMI of our methods against different λ.

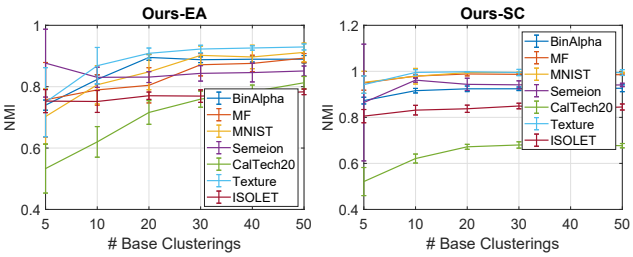

Figure 4: The NMI of our methods with different numbers of base clusterings, where the vertical error bar indicates the standard deviation over 20 repetitions.

over the 20 repetitions. For the compared methods, we set the hyper-parameters according to their original papers. If there are no suggested values, we exhaustively searched the hyper-parameters, and used the ones producing the best performance. The proposed model only contains one hyper-parameter λ, which was set to 0.002 for all the data sets.

Analysis of the Clustering Performance

Tables 1-7 show the clustering performance of all the methods over 7 data sets, where we have the following observations. First, the proposed methods including both Ours-EA and Ours-SC almost always outperform all the compared methods under various metrics, which proves the universality of the refined co-association matrix of the proposed

BinAlpha	TA-CL	TA-SL	PTGP	LWSC	LWEA	LWGP	E-HC	E-MC	DREC	SPCE	SEC	Ours-EA	Ours-SC
ACC	0.429	0.186	0.429	0.424	0.403	0.431	0.375	0.454	0.375	0.298	0.443	0.712	**0.858**
NMI	0.577	0.300	0.574	0.570	0.553	0.575	0.537	0.592	0.518	0.541	0.585	0.824	**0.916**
Purity	0.451	0.197	0.446	0.444	0.413	0.457	0.383	0.478	0.396	0.285	0.470	0.718	**0.876**
ARI	0.291	0.081	0.291	0.284	0.289	0.287	0.269	0.300	0.248	0.227	0.291	0.643	**0.817**
F1-score	0.312	0.126	0.313	0.306	0.313	0.308	0.295	0.320	0.271	0.302	0.311	0.654	**0.822**
Precision	0.276	0.071	0.277	0.272	0.248	0.277	0.220	0.305	0.238	0.294	0.296	0.559	**0.801**
Recall	0.361	0.635	0.361	0.349	0.426	0.348	0.451	0.337	0.323	0.314	0.327	0.791	**0.845**

Table 1: Clustering Performance on BinAlpha (# samples: 1404, dimension: 320, # clusters: 36)

MF	TA-CL	TA-SL	PTGP	LWSC	LWEA	LWGP	E-HC	E-MC	DREC	SPCE	SEC	Ours-EA	Ours-SC
ACC	0.606	0.507	0.648	0.671	0.609	0.649	0.589	0.652	0.362	0.581	0.592	0.718	**0.990**
NMI	0.638	0.536	0.654	0.655	0.650	0.655	0.618	0.652	0.347	0.621	0.602	0.790	**0.979**
Purity	0.644	0.533	0.677	0.690	0.650	0.673	0.616	0.676	0.387	0.615	0.623	0.719	**0.990**
ARI	0.500	0.371	0.523	0.533	0.514	0.530	0.481	0.526	0.257	0.459	0.472	0.685	**0.979**
F1-score	0.554	0.457	0.575	0.583	0.567	0.582	0.541	0.576	0.370	0.527	0.528	0.724	**0.981**
Precision	0.511	0.344	0.534	0.551	0.517	0.530	0.472	0.541	0.311	0.424	0.496	0.586	**0.981**
Recall	0.608	0.712	0.627	0.619	0.628	0.647	0.637	0.618	0.739	0.713	0.566	0.960	**0.981**

Table 2: Clustering Performance on MF (# samples: 2000, dimension: 649, # clusters: 10)

MNIST	TA-CL	TA-SL	PTGP	LWSC	LWEA	LWGP	E-HC	E-MC	DREC	SPCE	SEC	Ours-EA	Ours-SC
ACC	0.654	0.207	0.665	0.613	0.658	0.573	0.609	0.656	0.480	0.543	0.539	0.797	**0.977**
NMI	0.610	0.133	0.622	0.612	0.635	0.594	0.608	0.635	0.434	0.482	0.521	0.806	**0.979**
Purity	0.668	0.209	0.685	0.663	0.676	0.626	0.624	0.691	0.498	0.557	0.585	0.798	**0.980**
ARI	0.504	0.051	0.522	0.483	0.531	0.460	0.495	0.524	0.342	0.429	0.384	0.735	**0.969**
F1-score	0.557	0.219	0.572	0.540	0.582	0.522	0.558	0.574	0.427	0.445	0.450	0.767	**0.972**
Precision	0.523	0.124	0.541	0.490	0.536	0.459	0.448	0.543	0.373	0.316	0.420	0.666	**0.968**
Recall	0.596	0.952	0.607	0.603	0.641	0.609	0.745	0.610	0.576	0.831	0.485	0.918	**0.977**

Table 3: Clustering Performance on MNIST (# samples: 5000, dimension: 784, # clusters: 10)

Semeion	TA-CL	TA-SL	PTGP	LWSC	LWEA	LWGP	E-HC	E-MC	DREC	SPCE	SEC	Ours-EA	Ours-SC
ACC	0.700	0.425	0.692	0.682	0.739	0.620	0.638	0.679	0.450	0.571	0.594	0.846	**0.983**
NMI	0.634	0.418	0.631	0.630	0.656	0.598	0.601	0.635	0.386	0.571	0.569	0.831	**0.962**
Purity	0.707	0.449	0.703	0.702	0.739	0.651	0.645	0.705	0.460	0.607	0.634	0.847	**0.983**
ARI	0.510	0.248	0.507	0.507	0.540	0.465	0.480	0.508	0.290	0.401	0.418	0.790	**0.962**
F1-score	0.563	0.360	0.560	0.559	0.588	0.525	0.540	0.560	0.391	0.477	0.481	0.813	**0.966**
Precision	0.522	0.246	0.522	0.523	0.552	0.466	0.468	0.527	0.329	0.381	0.448	0.748	**0.966**
Recall	0.611	0.712	0.606	0.601	0.631	0.603	0.644	0.599	0.664	0.660	0.524	0.893	**0.966**

Table 4: Clustering Performance on Semeion (# samples: 1593, dimension: 256, # clusters: 10)

CalTech20	TA-CL	TA-SL	PTGP	LWSC	LWEA	LWGP	E-HC	E-MC	DREC	SPCE	SEC	Ours-EA	Ours-SC
ACC	0.343	0.421	0.345	0.324	0.423	0.336	0.450	0.363	0.340	0.495	0.297	**0.726**	0.418
NMI	0.402	0.269	0.401	0.396	0.454	0.406	0.455	0.428	0.350	0.452	0.381	0.620	**0.621**
Purity	0.639	0.520	0.637	0.642	0.665	0.646	0.645	0.660	0.590	0.664	0.633	0.730	**0.788**
ARI	0.265	0.184	0.267	0.222	0.359	0.224	0.351	0.258	0.225	0.395	0.202	**0.785**	0.328
F1-score	0.337	0.363	0.338	0.291	0.432	0.298	0.437	0.332	0.316	0.457	0.269	**0.823**	0.384
Precision	0.561	0.284	0.563	0.529	0.612	0.510	0.538	0.543	0.479	0.503	0.525	**0.764**	0.743
Recall	0.241	0.562	0.243	0.201	0.335	0.211	0.373	0.239	0.253	0.449	0.181	**0.898**	0.259

Table 5: Clustering Performance on CalTech20 (# samples: 2386, dimension: 30,000, # clusters: 20)

Texture	TA-CL	TA-SL	PTGP	LWSC	LWEA	LWGP	E-HC	E-MC	DREC	SPCE	SEC	Ours-EA	Ours-SC
ACC	0.714	0.410	0.732	0.719	0.793	0.686	0.675	0.675	0.416	0.634	0.614	0.863	**0.993**
NMI	0.721	0.438	0.731	0.742	0.782	0.739	0.703	0.718	0.419	0.693	0.638	0.868	**0.995**
Purity	0.729	0.427	0.746	0.744	0.798	0.728	0.685	0.698	0.441	0.658	0.647	0.864	**0.995**
ARI	0.600	0.237	0.619	0.628	0.696	0.609	0.569	0.585	0.298	0.534	0.486	0.816	**0.993**
F1-score	0.639	0.350	0.656	0.663	0.724	0.648	0.614	0.626	0.397	0.590	0.537	0.834	**0.993**
Precision	0.598	0.228	0.627	0.631	0.700	0.592	0.543	0.582	0.337	0.465	0.500	0.780	**0.991**
Recall	0.690	0.897	0.689	0.700	0.752	0.719	0.710	0.677	0.752	0.827	0.586	0.895	**0.996**

Table 6: Clustering Performance on Texture (# samples: 5500, dimension: 20, # clusters: 11)

ISOLET	TA-CL	TA-SL	PTGP	LWSC	LWEA	LWGP	E-HC	E-MC	DREC	SPCE	SEC	Ours-EA	Ours-SC
ACC	0.540	0.394	0.539	0.556	0.578	0.527	0.451	0.581	0.324	0.574	0.554	0.575	**0.675**
NMI	0.718	0.587	0.715	0.721	0.743	0.710	0.667	0.743	0.413	0.818	0.719	0.752	**0.831**
Purity	0.572	0.407	0.567	0.594	0.605	0.564	0.467	0.619	0.350	0.301	0.590	0.583	**0.707**
ARI	0.498	0.316	0.495	0.485	0.552	0.472	0.449	0.516	0.251	0.367	0.483	0.563	**0.639**
F1-score	0.520	0.356	0.517	0.506	0.571	0.495	0.477	0.536	0.303	0.384	0.505	0.584	**0.654**
Precision	0.467	0.237	0.462	0.458	0.511	0.437	0.352	0.496	0.253	0.584	0.463	0.481	**0.625**
Recall	0.591	**0.787**	0.590	0.568	0.648	0.573	0.748	0.584	0.690	0.327	0.555	0.752	0.685

Table 7: Clustering Performance on ISOLET (# samples: 7791, dimension: 617, # clusters: 26)

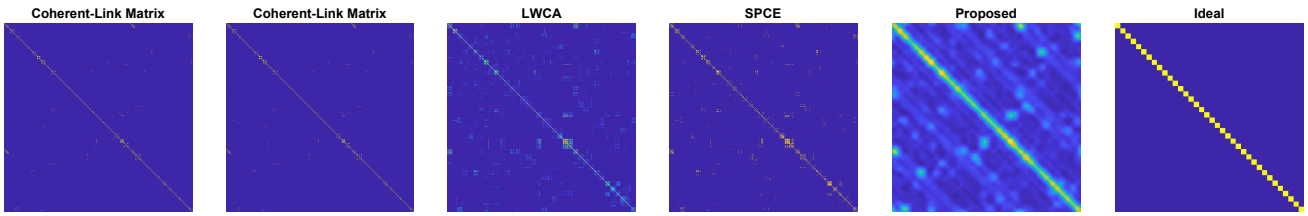

Figure 5: Visual comparison of the learned pairwise similarity matrices for different methods. All the matrices share the same color bar, and the brighter color indicates a larger value.

model to different clustering methods. Moreover, Ours-SC usually performs better than Ours-EA, which means the refined co-association is more suitable for spectral clustering. Second, the improvements of the proposed methods are significant. For example, on BinAlpha, compared with the best method under comparison, Ours-SC increases the ACC from 0.454 to 0.858. On CalTech20, the highest ACC of the compared methods is 0.495, while the ACC of Ours-EA is 0.726. The improvements of the proposed methods in terms of other metrics are also significant. Moreover, the performance of Ours-SC on MF, MNIST, Semeion, Texture are extremely good, i.e., all the metrics are quite close to 1. Those phenomena suggest that the proposed model brings a breakthrough in clustering ensemble. Third, the highly competitive performance of the proposed model is achieved with a fixed hyper-parameter, proving the practicability of the proposed model. Besides, the proposed model is also robust to different data sets, as both Ours-EA and Ours-SC consistently produce superior clustering performance on all the data sets.

Comparison Against Base Clusterings. We compared the average NMI of the our methods with that of all the base clusterings from the candidate clustering pool in Fig. 2. It is clear that, on all the data sets, both Ours-SC and Ours-EA can significantly improve the NMI of the base clusterings, and Ours-SC outperforms Ours-EA in the majority cases.

Sensitivity to Hyper-parameter. Fig. 3 shows the NMI of the proposed methods with different λ on all the data sets, where we can conclude that: first, a smaller λ usually leads to better clustering performance for both Ours-EA and Ours-SC, which demonstrates the importance of removing the incorrect connections from the original co-association matrix; and second, for the majority data sets, the highest NMI occurs when $\lambda = 0.002$ for both Ours-EA and Ours-SC, which proves the highly robustness of the proposed model to different data sets.

Performance with Different Number of Base Clusterings. Fig. 4 illustrates the influence of different numbers of the base clusterings to the proposed model, where we have the following observations. First, with the increase of the number of base clusterings, the NMIs of both Ours-EA and Ours-SC generally increase, indicating that more base clustering are beneficial to the clustering performance. Second, with more base clusterings, the standard deviations generally become smaller for all the data sets, which suggests that more base clusterings can enhance the stability our methods. Third, for the majority data sets, 20 base clusterings are sufficient for our methods to generate high value of NMI.

Comparison of the Learned Pairwise Similarity Matrix. Fig. 5 presents the cohere-link matrix, the traditional co-association matrix, the learned co-association matrices by LWCA (Huang, Wang, and Lai 2018), SPCE (Zhou et al. 2020) and the proposed model, and the ideal affinity matrix of BinAlpha, where all the matrices are normalized to $[0, 1]$ and share the same color bar. Form 5, we can observe that the coherent-link matrix is sparse, but its majority connections are correct, while on the contrary, the co-association matrix is dense, but with many incorrect connections in it. By exploiting the the low-rankness of the 3-D tensor stacked by the coherent-link matrix and the association matrix, the refined co-association matrix of the proposed model is quite close to the ideal one. Although there are some error corrections in it, almost all the relationships of two samples belonging to the same cluster have been correctly recovered, leading to high clustering performance. In contrast, there are many incorrect connections, but without enough correct connections in both the affinity matrices of LWCA and SPCE, which explains why they produced inferior clustering performance than the proposed model.

Conclusion

As the first work, we introduced low-rank tensor approximation to clustering ensemble. Different from previous methods, the proposed model solves clustering ensemble from a global perspective, i.e., exploiting the low-rankness of a 3-D tensor formed by the coherent-link matrix and the co-association matrix, such that the valuable information of the coherent-link matrix can be effectively propagated to the co-association matrix. Extensive experiments have shown that i), the proposed model improves current state-of-the-art performance of clustering ensemble to a new level; ii), the recommended value for the hyper-parameter of the proposed model is robust to different data sets; and iii), only a few base clusterings are required to generate high clustering performance.

Acknowledgements

This work was supported by the Hong Kong Research Grants Council under Grants 21211518, 11219019 and 11202320.

References

Boongoen, T.; and Iam-On, N. 2018. Cluster Ensembles: A Survey of Approaches with Recent Extensions and Applications. *Computer Science Review* 28: 1–25.

Fred, A. L.; and Jain, A. K. 2005. Combining Multiple Clusterings Using Evidence Accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(6): 835–850.

Gao, J.; Yamada, M.; Kaski, S.; Mamitsuka, H.; Zhu, S.; and Kambhampati, S. 2016. A Robust Convex Formulations for Ensemble Clustering. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*, 1476–1482. AAAI Press International Joint Conferences on Artificial Intelligence.

Huang, D.; Lai, J.-H.; and Wang, C.-D. 2016. Robust Ensemble Clustering Using Probability Trajectories. *IEEE Transactions on knowledge and Data Engineering* 28(5): 1312–1326.

Huang, D.; Wang, C.; Peng, H.; Lai, J.; and Kwoh, C. 2018. Enhanced Ensemble Clustering via Fast Propagation of Cluster-Wise Similarities. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 1–13.

Huang, D.; Wang, C.-D.; and Lai, J.-H. 2018. Locally Weighted Ensemble Clustering. *IEEE Transactions on Cybernetics* 48(5): 1460–1473.

Jain, A. K. 2010. Data Clustering: 50 Years Beyond K-means. *Pattern Recognition Letters* 31(8): 651–666.

Jia, Y.; Hou, J.; and Kwong, S. 2020. Constrained Clustering With Dissimilarity Propagation-Guided Graph-Laplacian PCA. *IEEE Transactions on Neural Networks and Learning Systems* 1–13. doi:10.1109/TNNLS.2020.3016397.

Jia, Y.; Kwong, S.; and Hou, J. 2018. Semi-Supervised Spectral Clustering With Structured Sparsity Regularization. *IEEE Signal Processing Letters* 25(3): 403–407.

Jia, Y.; Kwong, S.; Hou, J.; and Wu, W. 2018. Convex Constrained Clustering with Graph-Laplacian Pca. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*, 1–6.

Jia, Y.; Kwong, S.; Hou, J.; and Wu, W. 2020a. Semi-Supervised Non-Negative Matrix Factorization With Dissimilarity and Similarity Regularization. *IEEE Transactions on Neural Networks and Learning Systems* 31(7): 2510–2521.

Jia, Y.; Liu, H.; Hou, J.; and Kwong, S. 2020b. Clustering-Aware Graph Construction: A Joint Learning Perspective. *IEEE Transactions on Signal and Information Processing over Networks* 6: 357–370.

Jia, Y.; Liu, H.; Hou, J.; and Kwong, S. 2020c. Semisupervised Adaptive Symmetric Non-Negative Matrix Factorization. *IEEE Transactions on Cybernetics* 1–1. doi: 10.1109/TCYB.2020.2969684.

Jia, Y.; Wu, W.; Wang, R.; Hou, J.; and Kwong, S. 2020d. Joint Optimization for Pairwise Constraint Propagation. *IEEE Transactions on Neural Networks and Learning Systems* 1–13.

Johnson, S. C. 1967. Hierarchical Clustering Schemes. *Psychometrika* 32(3): 241–254.

Kilmer, M. E.; Braman, K.; Hao, N.; and Hoover, R. C. 2013. Third-order Tensors as Operators on Matrices: A Theoretical and Computational Framework with Applications in Imaging. *SIAM Journal on Matrix Analysis and Applications* 34(1): 148–172.

Li, H.; Kwong, S.; Chen, C.; Jia, Y.; and Cong, R. 2019. Superpixel Segmentation based on Square-wise Asymmetric Partition and Structural Approximation. *IEEE Transactions on Multimedia* 21(10): 2625–2637.

Liu, H.; Jia, Y.; Hou, J.; and Zhang, Q. 2019. Imbalance-aware Pairwise Constraint Propagation. In *In Proceedings of the 27th ACM International Conference on Multimedia*, 1605–1613.

Liu, H.; Wu, J.; Liu, T.; Tao, D.; and Fu, Y. 2017. Spectral Ensemble Clustering via Weighted K-Means: Theoretical and Practical Evidence. *IEEE Transactions on Knowledge and Data Engineering* 29(5): 1129–1143.

Moore, A. W. 1999. Very Fast EM-based Mixture Model Clustering Using Multiresolution Dd-trees. In *Advances in Neural information processing systems*, 543–549.

Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2002. On Spectral Clustering: Analysis and an Algorithm. In *Advances in Neural Information Processing Systems*, 849–856.

Sagi, O.; and Rokach, L. 2018. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8(4): e1249.

Song, L.; Tekin, C.; and Van Der Schaar, M. 2014. Online Learning in Large-scale Contextual Recommender Systems. *IEEE Transactions on Services Computing* 9(3): 433–445.

Tao, Z.; Liu, H.; and Fu, Y. 2017. Simultaneous Clustering and Ensemble. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 1546–1552.

Tao, Z.; Liu, H.; Li, S.; Ding, Z.; and Fu, Y. 2017. From ensemble clustering to multi-view clustering. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2843–2849.

Tao, Z.; Liu, H.; Li, S.; and Fu, Y. 2016. Robust Spectral Ensemble Clustering. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 367–376.

Ulrike, L. 2007. A Tutorial on Spectral Clustering. *Stat. Comput.* 17(4): 395–416. ISSN 1573-1375. doi:10.1007/s11222-007-9033-z. URL https://doi.org/10.1007/s11222-007-9033-z.

Vega-Pons, S.; and Ruiz-Shulcloper, J. 2011. A Survey of Clustering Ensemble Algorithms. *International Journal of Pattern Recognition and Artificial Intelligence* 25(03): 337–372.

Wu, W.; Jia, Y.; Kwong, S.; and Hou, J. 2018. Pairwise Constraint Propagation-Induced Symmetric Nonnegative Matrix Factorization. *IEEE Transactions on Neural Networks and Learning Systems* 29(12): 6348–6361.

Wu, W.; Kwong, S.; Zhou, Y.; Jia, Y.; and Gao, W. 2018. Nonnegative Matrix Factorization with Mixed Hypergraph Regularization for Community Detection. *Information Sciences* 435: 263–281.

Zhang, C.; Fu, H.; Hu, Q.; Cao, X.; Xie, Y.; Tao, D.; and Xu, D. 2020. Generalized Latent Multi-View Subspace Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42(1): 86–99.

Zhang, Z.; Ely, G.; Aeron, S.; Hao, N.; and Kilmer, M. 2014. Novel Methods for Multilinear Data Completion and Denoising based on Tensor-SVD. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3842–3849.

Zhou, J.; Zheng, H.; and Pan, L. 2019. Ensemble Clustering based on Dense Representation. *Neurocomputing* 357: 66–76.

Zhou, P.; Du, L.; Liu, X.; Shen, Y.; Fan, M.; and Li, X. 2020. Self-Paced Clustering Ensemble. *IEEE Transactions on Neural Networks and Learning Systems* 1–15.