Journal Logo

00 (2023) 1–20

IMC-NLT: Incomplete Multi-view Clustering by NMF and Low-Rank Tensor

Zhenjiao Liu[a], Zhikui Chen[a,*], Yue Li[a], Liang Zhao[a], Tao Yang[b], Reza Farahbakhsh[c], Noel Crespi[c], Xiaodi Huang[d]

[a]*School of Software Technology, Dalian University of Technology, Dalian 116620, China*
[b]*School of Computer Science and Technology, Dalian University of Technology, Dalian 116620, China*
[c]*Samovar, Telecom SudParis, Institut Polytechnique de Paris, Palaiseau 91120, France*
[d]*School of Computing, Mathematics, and Engineering, Charles Sturt University, Albury, NSW 2640, Australia*

Abstract

Multi-view data obtained from different perspectives are becoming increasingly available. As such, researchers can use this data to explore complementary information. However, such real-world data are often incomplete. Existing algorithms for incomplete multi-view clustering (IMC) have some limitations, such as the ineffective use of valuable information hidden in the data, oversensitivity to model parameters, and ineffective handling of samples with incomplete views. To overcome these limitations, we present a novel algorithm for incomplete multi-view clustering using Non-negative matrix factorization and a low-rank tensor (IMC-NLT). In particular, IMC-NLT first uses a low-rank tensor to retain view features with a unified dimension. Using a consistency measure, IMC-NLT captures a consistent representation across multiple views. Finally, IMC-NLT incorporates multiple learning into a unified model such that hidden information can be extracted effectively from incomplete views. We conducted comprehensive experiments on five real-world datasets to validate the performance of IMC-NLT. The overall experimental results demonstrate that the proposed IMC-NLT performs better than several baseline methods, yielding stable and promising results.

Keywords: Incomplete multi-view clustering, low-rank tensor, consistent representation

1. Introduction

With the development of data acquisition technology, sources and types of data are becoming more diverse. The collected data were characterized by multiple views. Complex data features can be better understood by fusing information from multi-view data. A common approach is multi-view clustering. The basic idea of this type of approach is because multi-view data are strongly related and complementary to each other in different ways(Wang et al., 2015a; Li et al., 2017; Zhang et al., 2021; Shi et al., 2022; Si et al., 2022; El Hajjar et al., 2022; Hu et al., 2021; Wong et al., 2019). However, not all data will contain complete multi-view data. We consider land data as an example, often consisting of data with different views, such as symbols, text, and graphics. In reality, some land data

*Corresponding author

Email addresses: liuzhenjiao@mail.dlut.edu.cn (Zhenjiao Liu), zkchen@dlut.edu.cn (Zhikui Chen), liyue123@mail.dlut.edu.cn (Yue Li), liangzhao@dlut.edu.cn (Liang Zhao), bruceyang@mail.dlut.edu.cn (Tao Yang), reza.farahbakhsh@it-sudparis.eu (Reza Farahbakhsh), noel.crespi@mines-telecom.fr (Noel Crespi), xhuang@csu.edu.au (Xiaodi Huang)

may have the format of graphics, but lack text or symbols. As such, incomplete multi-view data were formed (Wang et al., 2018). That is, a dataset has an arbitrary loss of view with only some instances containing all views.

Specifically, incomplete multi-view clustering methods can be roughly grouped into four categories. Based on a filling strategy, the methods of the first category (Shao et al., 2015; Xu & Tao, 2015; Liu et al., 2019; Shao et al., 2016) usually start by selecting a suitable padding algorithm and then applying existing multi-view learning methods to incomplete multi-view datasets populated by padding algorithms. For example, Shao et al. (Shao et al., 2015) first filled incomplete samples by averaging the eigenvalues and using regularised weighted non-negative matrix decomposition to learn a subspace. The methods in the second category (Tao et al., 2019; Yin & Sun, 2022; Zhao et al., 2016; Quanz, 2012; Yin et al., 2017) ignore incomplete view information in the process of learning potential representations. Quanz et al. (Quanz, 2012) pushed clustering solutions for different views from the same example to the standard membership matrix to simultaneously generate the underlying geometric structure of the views. Learning from a single view, the third method (Yuan et al., 2012; Eaton et al., 2010) attempts to learn a unified model for all views. Yuan et al. (Yuan et al., 2012) built a separate classifier for each data source. This model learns different base classifiers for different data sources. Based on a classifier, the method uses a single column of prediction scores to represent each source and then combines the estimated incomplete prediction scores with the available prediction scores. Thus, a multi-source fusion model was built. The methods of the last category (Xie et al., 2021; Cai et al., 2018; Zhao et al., 2018; Xu et al., 2019) use deep models for incomplete multiview clustering. For example, Tran et al. (Tran, 2017) proposed a cascaded residual autoencoder (CRA) to complement the incomplete multi-view. By stacking the residual autoencoders, the residuals between the current prediction and the original data were obtained by iterative simulation of the algorithm.

Although many methods have been proposed to address the incomplete problem of multi-view clustering, they have some limitations. For example, as mentioned by (Wen et al., 2020), the first limitation is that most approaches do not make full use of the information of observation instances inside and between views, resulting in missing valuable information. Second, some models are sensitive to the choice of parameters and are less robust (Liu et al., 2012), which restricts their availability in real-world scenarios. Third, existing methods cannot effectively handle samples with incomplete views, which inevitably reduces the performance of the IMC (Wen et al., 2018a).

To overcome these limitations, in this paper, we propose a unified framework for incomplete multiview clustering using non-negative matrix factorization (NMF) and the low rank tensor, called IMC-NLT. Specifically, IMC-NLT first utilises NMF to learn a low-dimensional representation for all the views. In this way, not only is the dimensionality of the data reduced, but also non-negative numerical effects with strong explanatory power can be obtained. Using the prior information of a low-rank tensor, IMC-NLT can capture the higher-order and complementary information embedded in the multi-view data. Finally, a new cost function is introduced to measure consistent information across views, using a linear kernel that measures similarities. The contributions of this study are summarised as follows.

- We propose a novel and efficient incomplete multi-view clustering model called IMC-NLT to handle incomplete view data. To the best of our knowledge, IMC-NLT is the first incomplete multi-view clustering method that combines the low-dimensional representation generated by the fast and effective dimension-reduction method with a low-rank tensor model.

- To populate missing multi-view data in various cases, IMC-NLT uses a low-rank constraint and a tensor model constructed from incomplete multi-view data. As such, it can capture the correlations among instances within and between views well.

- The proposed method is robust to globally extracted consistency information. In particular, consistent representation learning can effectively measure the disagreement between consistent information obtained from different perspectives. IMC-NLT can effectively filter out noisy data under these views, producing an accurate multi-view consistent representation.

- We conducted comprehensive experiments on multi-view benchmark datasets collected in different application fields to evaluate the effectiveness of the IMC-NLT. The results showed that IMC-NLT is superior to baseline methods. Furthermore, IMC-NLT has low sensitivity to its parameters, which demonstrates excellent generalization performance for incomplete multi-view clustering.

The remainder of this paper is organised as follows: Section 2 describes the related work and background. Section 3 describes the proposed IMC-NLT algorithm. Sections 4 and 5 present the theoretical analysis of IMC-NLT and evaluation results of the performance of IMC-NLT, respectively. In Section 6, we report the experiments on the parameters and convergence of IMC-NLT. Finally, Section 7 concludes the paper.

2. Related Work and background

In this section, we present two studies closely related to the proposed method. First, we present the basics of low-rank tensor-based models for multi-view learning. Here, we review incomplete multi-view clustering based on matrix factorization.

2.1. Multi-view learning based on low-rank tensor

A low-rank tensor can effectively capture hidden information as a valid technique for analyzing high-dimensional data. There are many examples of low-rank tensors used for multi-view clustering(Zhang et al., 2015; Xu et al., 2020). Zhang et al. (Zhang et al., 2015) proposed a subspace clustering method called low-rank tensor-constrained multi-view subspace clustering (LT-MSC). LT-MSC can be described by the following equation:

$$\begin{aligned} &\min_{Z^{(v)},E^{(v)}} \|\mathcal{Z}\|_* + \lambda\|E\|_{2,1} \\ &s.t.\ X^{(v)} = X^{(v)}Z^{(v)} + E^{(v)}, v = 1, 2, ..., V, \\ &\mathcal{Z} = \Psi(Z^{(1)}, Z^{(2)}, ..., Z^{(v)}), \\ &E = [E^{(1)}; E^{(2)}; ...; E^{(v)}], \end{aligned} \tag{1}$$

where $X^{(v)}$ is the v-th view of data. Note that $X^{(v)} = X^{(v)}Z^{(v)} + E^{(v)}$. The formula can learn the subspace representation matrix $Z^{(v)}$ by exploiting the self-expressive properties of the data; λ is a positive penalty parameter, and $E^{(v)}$ is the reconstruction error matrix. Imposing the $L_{2,1}$ regularizer on $E^{(v)}$ can urge the data in each column of matrix $E^{(v)}$ to be close to 0. $\|\mathcal{Z}\|_*$ is the tensor kernel norm constraint added to $\mathcal{Z}$. The tensor $\mathcal{Z}$ consists of subspace representation $Z^{(v)}$. The model obtains the low-rank tensor using a self-representation multi-view.

Similar to previous research methods, Xu et al. (Xu et al., 2020) proposed a method called low-rank tensor-constrained co-regularised multi-view spectral clustering(LTCSPC). The objective function of the LTCSPC is as follows:

$$\min_{F^{(v)}\in\mathbb{R}^{n\times c}} \sum_{v=1}^{m} \alpha^{(v)} Tr(F^{(v)^T} L^{(v)} F^{(v)}) + \|\mathcal{F}\|_{\omega,\circledast} \tag{2}$$

where we have

$$\alpha^{(v)} = 1 \Big/ (2\sqrt{Tr(F^{(v)^T} L^{(v)} F^{(v)})}) \tag{3}$$

In Eq.(3), $\alpha^{(v)}$ is the weight of each view and ω is the singular value-weighted coefficient. LTCSPC calculates $F^{(v)}$ according to the standard spectral clustering and data $X^{(v)}$ with m views. To use the high-order structure and complement, we denote the indicator matrix as $F^{(v)} \in \mathbb{R}^{n\times c}$, where n is the number of samples, and c is the number of categories. The slice of tensor $\mathcal{F}$ consists of the indicator matrix $F^{(v)}$. $L^{(v)}$ is the Laplacian matrix. $\|\mathcal{F}\|_{\omega,\circledast}$ is the weighted nuclear norm constraint added to $\mathcal{F} \in \mathbb{R}^{n\times m\times c}$, which is defined as:

$$\|\mathcal{F}\|_{\omega,\circledast} = \sum_{i=1}^{c} \|\bar{F}^{(i)}\|_{\omega,*} = \sum_{i=1}^{c} \sum_{j=1}^{min(n,m)} \omega_j * \sigma_j(\bar{F}^{(i)}) \tag{4}$$

where $\bar{F}^{(i)} \in \mathbb{R}^{n\times m}$, $\sigma_j(\bar{F}^{(i)})$ is the j-th largest singular value of $\bar{F}^{(i)}$, and ω_j is the j-th element of the vector ω. Although these methods can elegantly model different views, they improve the clustering accuracy while reducing the redundancy of the learning subspace representation. In addition, they are only applicable to complete multi-view data, and cannot handle incomplete cases.

2.2. Incomplete multi-view clustering based on matrix factorization

In this section, we review two classical methods: one-pass incomplete multi-view clustering(OPIMC) (Hu & Chen, 2019) and partial multi-view clustering(PVC) (Li et al., 2014).

OPIMC applies regularized matrix factorization (RMF) (Gunasekar et al., 2017) and weighted matrix factorization (WMF) (Kim & Choi, 2009) to produce multi-view clustering results. The objective function of the OPIMC is written as:

$$\begin{aligned} J &= \sum_{v=1}^{n_v}\{\sum_{t=1}^{[N/s]} \left\| \ (X_t^{(v)} \ - \ U^{(v)} \ V_t^T)W_t^{(v)}\right\|_F^2 + \alpha \left\|U^{(v)}\right\|_F^2\} \\ s.t.\ V_{ik} &\in \{0,1\}, \sum_{k=1}^{K} V_{ik} = 1, \forall i = 1,2,...,N \end{aligned} \tag{5}$$

where $X_t^{(v)}$ denotes the t-th data chunk in the v-th view. Assume that each view is composed of blocks of size s. $V \in \mathbb{R}^{N\times K}$ is a clustering indicator matrix, and K represents the number of categories of the data. If the instance belongs to one category, it is marked as 1; otherwise, it is 0. $U^{(v)}$ and V_t are the low-rank regularized factor matrix and the clustering indicator matrix for the t-th data chunk, respectively. α is a non-negative parameter. Where N denotes the number of instances. Furthermore, to achieve a high calculation cost when the number of instances and categories are large, the model applies a 1-of-K coding constraint to V, where $W_t^{(v)}$ is a diagonally weighted matrix of the t-th data chunk. OPIMC can directly obtain the clustering results at the end of the iteration.

PVC is an incomplete multi-view clustering method that uses NMF. It constructs a specific latent space for unaligned instances and a shared latent space for aligned instances. These shared and specific representations were used together for clustering. PVC was formalized as follows:

$$\begin{aligned} \min_{P_{(c)},\hat{P}^{(1)},\hat{P}^{(2)},U^{(1)},U^{(2)}} & \left\|\begin{bmatrix} {X_c}^{(1)} \\ \hat{X}^{(1)} \end{bmatrix} - \begin{bmatrix} P_c \\ \hat{P}^{(1)} \end{bmatrix} U^{(1)}\right\|_F^2 \\ & + \left\|\begin{bmatrix} {X_c}^{(2)} \\ \hat{X}^{(2)} \end{bmatrix} - \begin{bmatrix} P_c \\ \hat{P}^{(2)} \end{bmatrix} U^{(2)}\right\|_F^2 \\ & + \lambda \left\| \ \bar{P}^{(1)} \ \right\|_1 + \lambda \left\| \ \bar{P}^{(2)} \ \right\|_1 \\ & s.t. U^{(1)} \geq 0, U^{(2)} \geq 0, \bar{P}^{(1)} \geq 0 \\ & \bar{P}^{(2)} \geq 0 \end{aligned} \tag{6}$$

In Eq.(6), ${X_c}^{(1)}$ and ${X_c}^{(2)}$ represent instance data that exists in both views, $\hat{X}^{(1)}$ represents instance data that exists only in the first view, and $\hat{X}^{(2)}$ represents instance data that exists only in the second view. P_c represents the low-dimensional representation of the common views after matrix decomposition. $\hat{P}^{(1)}$ and $\hat{P}^{(2)}$ represent individual potential representation parts of each view. $U^{(v)}$ is the basis matrix of the view, $\bar{P}^{(1)} = [P_c; \hat{P}^{(1)}]$ and $\bar{P}^{(2)} = [P_c \hat{P}^{(2)}]$ are the latent representation of instances in the latent space. λ is the positive tradeoff parameter. The grouping result was obtained by establishing a potential subspace.

3. Proposed IMC-NLT

To cluster incomplete multi-views, we present a new incomplete multi-view clustering model called IMC-NLT. It consists of three parts: incomplete multi-view data filling and decomposition, multi-view fusion of low-rank tensor, and consensus representation learning. The framework for the IMC-NLT is illustrated in Fig. 1.

3.1. Incomplete multi-view data filling and decomposition

To align the dimensions of multi-view data in a way that better reflects the structure of incomplete data, we used NMF to build a multi-view data model with a unified dimension and maintain information about the data space and feature space. The model is as follows:

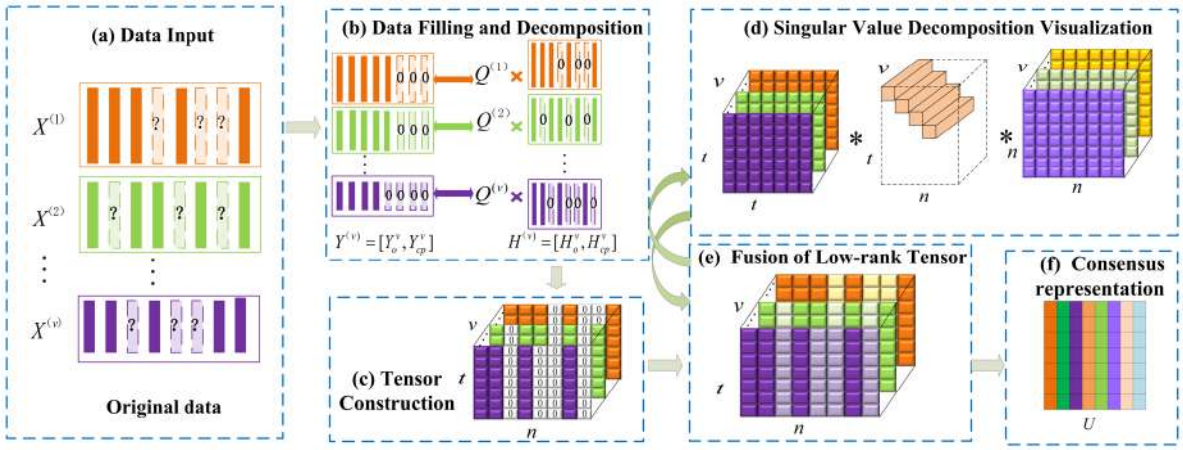

Fig. 1. Overview of IMC-NLT. IMC-NLT consists of six major components: (a) Incomplete multi-view data input; (b) Pre-filing and dimensionality reduction of incomplete view data; (c) Construction of a tensor structure with unified view dimensions. (d) Singular value decomposition visualization; (e) The Low-rank tensor fusion implements data filling; and (f) Consensus representation.

$$\min_{H^{(v)}} \sum_{v=1}^{V} \sum_{i=1}^{n_o^{(v)}+n_{cp}^{(v)}} \left\| \begin{bmatrix} Y_{o_i}^{(v)} \\ Y_{cp_i}^{(v)} \end{bmatrix} - \begin{bmatrix} H_{o_i}^{(v)} \\ H_{cp_i}^{(v)} \end{bmatrix} Q_i^{(v)} \right\|_F^2 \tag{7}$$
$$s.t.\ H^{(v)} \geq 0, Q^{(v)} \geq 0, H^{(v)} = T \begin{bmatrix} H_o^{(v)}; H_{cp}^{(v)} \end{bmatrix}$$

An example is shown in Fig. 2, where matrix $Y_o^{(v)} \in \mathbb{R}^{n_o^{(v)} \times d^{(v)}}$ is the matrix of complete instances selected from the original view $X^{(v)} \in \mathbb{R}^{n^{(v)} \times d^{(v)}}$. $n^{(v)}$ is the number of instances when the view is complete, $n_o^{(v)}$ is the number of instances in each view that are not missing, and $d^{(v)}$ represents the original feature dimension of multi-view data. $Y_{cp}^{(v)} \in \mathbb{R}^{n_{cp}^{(v)} \times d^{(v)}}$ is a matrix of incomplete instances selected from the original view. $n_{cp}^{(v)}$ is the number of instances incomplete in the multi-view data. To better reflect the structure of incomplete data and better fill in incomplete values, we filled the incomplete view data instances matrix $Y_{cp}^{(v)}$ with 0. The main purpose of dividing the available part $Y_o^{(v)}$ and missing part $Y_{cp}^{(v)}$ is to emphasize the filling of the missing part of the data using the proposed low-rank tensor model. The zero matrices represent the missing part of the data, whereas the change in the zero matrices reflects the data recovery function of the model proposed in this study. By introducing the decomposition model, we constructed the following low-dimensional representation structure: $H_o^{(v)} \in \mathbb{R}^{n_o^{(v)} \times t}$ is the low-dimensional representation matrix formed by the complete instances, where t represents the unified dimension number after the dimension reduction. The selection of the feature dimension t depends on the existing algorithms for multi-view data-sharing features based on NMF (Zong et al., 2017). $H_{cp}^{(v)} \in \mathbb{R}^{n_{cp}^{(v)} \times t}$ is a low-dimensional representation matrix formed by the incomplete instances. $Q^{(v)} \in \mathbb{R}^{t \times d^{(v)}}$ is the coefficient matrix after non-negative factorization. T is a reconstruction operation of the modal matrix after the low-dimensional representation of each data view. The purpose of this is to arrange incomplete modal data after dimensionality reduction in the original order of the instance arrangement.

3.2. Multi-view collaborative fusion of low-rank tensor

In the previous section, we completed the unified transformation of the data dimensions between views and pre-filling of missing data. In this section, we introduce a low-rank tensor-filling model that can better capture high-order correlations between viewing data. This process consists of two parts:1) construction of the tensor, and 2) addition of low-rank constraints to the tensor.

An intuitive example is shown in Fig. 3, where $H^{(v)}$ is the low-dimensional representation matrix after incomplete multi-view data-filling and decomposition. A third-order tensor model was constructed using the enumeration method. Its construction is as follows:

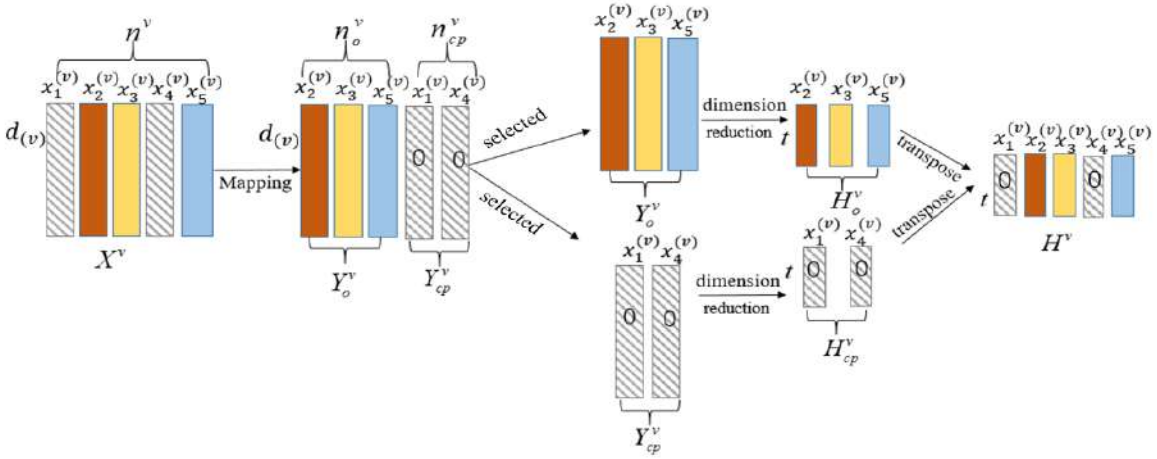

Fig. 2. Example of incomplete multi-view data filling and decomposition.

$$\mathcal{H} = \Psi(H^{(1)}, H^{(2)}, ..., H^{(v)}) \tag{8}$$

operation Ψ represents listing the reduced-dimensional second-order structure matrix $H^{(v)}$ in view order to construct a third-order tensor $\mathcal{H}$.

After obtaining the tensor model, we use tensor kernel parametrization to approximate the tensor low-rank representation. The optimization model is as follows:

$$\min_{\mathcal{H}} \|\mathcal{H}\|_* \tag{9}$$

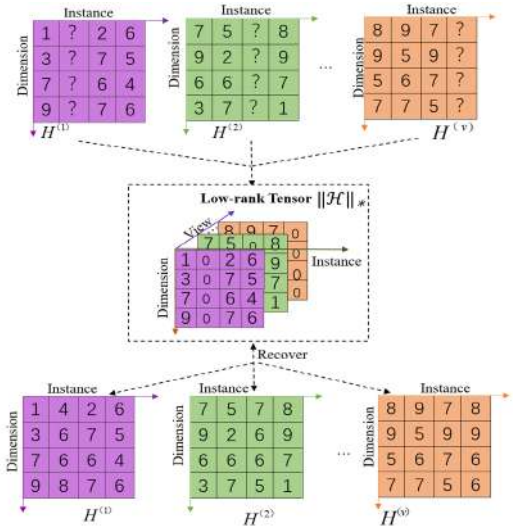

Fig. 3. Example of restoring incomplete data based on a low-rank tensor.

To make the objective function separable, we introduced the auxiliary variable K_m to solve the optimization problem in Eq. (9). The formula used is as follows:

$$\begin{aligned} &\min_{H^{(v)},K_m} \sum_{m=1}^{M} \sigma_m \|K_m\|_* \\ &s.t. k_m = P_m h, m = 1, 2, ...,M, \\ &\mathcal{H} = \Psi(H^{(1)}, H^{(2)}, ...,H^{(v)}) \end{aligned} \tag{10}$$

where σ_m represents the strength of the low-rank tensor constraint and k_m is the vectorization of the matrix K_m. P_m is the alignment matrix used to align the corresponding elements between $H_{(m)}$ and K_m, and $H_{(m)}$ is the matrix obtained by unfolding tensor $\mathcal{H}$ along the m-th mode, defined as $\text{unfold}_m(\mathcal{H}) = H_{(m)} \in \mathbb{R}^{I_m \times (I_1 \times ... I_{m-1} \times I_{m+1} ... I_M)}$. The vectorization of tensor $\mathcal{H}$ is denoted as h. This process can recover incomplete data based on effective association information.

3.3. Consensus representation learning

Generally, an incomplete multi-view clustering algorithm based on the fusion strategy gives the clustering results a more explicit physical meaning. However, existing incomplete multi-view clustering algorithms based on fusion strategies still suffer from several shortcomings:1) during the construction of the model, the fusion results are obtained separately from each view, and the similarity features between views are ignored, resulting in incomplete extraction of essential features for multi-view data; and 2) the semantic consistency features between the fused views are not fully considered. Therefore, we introduce a consistent representation model for our algorithm, which seeks a consensus representation from different perspectives, as follows:

$$\begin{aligned} J = \min \lambda_1 &\sum_{v=1}^{V} \sum_{i=1}^{n_o+n_{cp}} \left\| \begin{bmatrix} Y_{o_i}^{(v)} \\ Y_{cp_i}^{(v)} \end{bmatrix} - \begin{bmatrix} H_{o_i}^{(v)} \\ H_{cp_i}^{(v)} \end{bmatrix} Q_i^{(v)} \right\|_F^2 + \\ &\sigma_m \|K_m\|_* + \lambda_2 \Lambda(H^{(v)}, U) \\ s.t. P_m h = k_m, m = 1, 2, ...,M, &H^{(v)} = T \begin{bmatrix} H_o^{(v)}; H_{cp}^{(v)} \end{bmatrix}, \\ &\mathfrak{R}_\Omega(Y^{(v)}) = \mathfrak{R}_\Omega(X^{(v)}), {H^{(v)}}^T H^{(v)} = I, \\ &\mathcal{H} = \Psi(H^{(1)}, H^{(2)}, ...,H^{(v)}), UU^{(T)} = I, \\ &H^{(v)} \geq 0, Q^{(v)} \geq 0 \end{aligned} \tag{11}$$

where U denotes the learned consensus representation. This representation can effectively fuse each instance of information from all views. λ_1 and λ_2 are positive penalty parameters for adjusting the impact of each term in all objective functions. The function Λ produces a consistent representation matrix, and $H^{(v)}$ is the incomplete view data processed by filling the low-rank tensor model. $\mathfrak{R}_\Omega$ represents a mapping operation, which maps the complete instance part of the view to matrix $Y_o^{(v)}$ and the incomplete part of the view to matrix $Y_{cp}^{(v)}$.

To form a consistent representation of U, we utilised the measurement formula $\Lambda(H^{(v)}, U)$ to measure the degree of inconsistency between U and $H^{(v)}$ (Kumar, 2011):

$$\Lambda(H^{(v)}, U) = \left\| \frac{S_U}{\|S_U\|_F^2} - \frac{S_{H^{(v)}}}{\|S_{H^{(v)}}\|_F^2} \right\|_F^2 \tag{12}$$

where S_U is the similarity matrix of U and $S_{H^{(v)}}$ is the similarity matrix of $H^{(v)}$. This function can produce a better fusion representation by minimizing the gap between multi-view and consistent representations. Furthermore, we use the linear kernel, that is, $S_U = UU^T$, which is the standard of a similarity measurement (Kumar, 2011). Based on the fact that $\|S_U\|_F^2 = c$, and $\|S_{H^{(v)}}\|_F^2 = c$, the value of c in the formula is equal to the number of categories of multi-view data. Most of the algorithm settings were the same, and their effectiveness has been verified (Kumar, 2011). we rewrite Eq.(12) as:

$$\Lambda(H^{(v)}, U) = \frac{2(c - Tr({H^{(v)}}{H^{(v)}}^T UU^T))}{c^2} \tag{13}$$

Because c is a constant, it can be omitted when calculating to obtain the final consistency expression in Eq. (14).

$$\Lambda(H^{(v)}, U) = -Tr({H^{(v)}}{H^{(v)}}^T UU^T) \tag{14}$$

3.4. Objective function and optimization

Putting Eqs. (7), (10) and (14) together, we can obtain the following objective function:

$$\begin{aligned} J = \min \lambda_1 \sum_{v=1}^{V} \sum_{i=1}^{n_o+n_{cp}} \left\| \begin{bmatrix} Y_{o_i}^{(v)} \\ Y_{cp_i}^{(v)} \end{bmatrix} - \begin{bmatrix} H_{o_i}^{(v)} \\ H_{cp_i}^{(v)} \end{bmatrix} Q_i^{(v)} \right\|_F^2 + \\ \sigma_m \|K_m\|_* - \lambda_2 \sum_{v=1}^{V} Tr(H^{(v)} {H^{(v)}}^T U U^T) \\ s.t. P_m h = k_m, m = 1, 2, ..., M, H^{(v)} = T \begin{bmatrix} H_o^{(v)}; H_{cp}^{(v)} \end{bmatrix} \\ \mathfrak{R}_\Omega(Y^{(v)}) = \mathfrak{R}_\Omega(X^{(v)}), {H^{(v)}}^T H^{(v)} = I, \\ \mathcal{H} = \Psi(H^{(1)}, H^{(2)}, ..., H^{(v)}), U U^{(T)} = I, \\ H^{(v)} \geq 0, Q^{(v)} \geq 0 \end{aligned} \tag{15}$$

This objective function is a non-convex function; therefore, it cannot be optimized directly. As such, we can minimize the objective function by iteratively solving the following subproblems: The specific process is as follows.

$H^{(v)}$-sub-problem: By fixing all the other variables, we set the derivative to $H^{(v)}$ as follows:

$$\begin{aligned} D(H^{(v)}) = -2\lambda_1 (Y^{(v)} - HQ) Q^{(T)} - 2\lambda_2 (U U^{(T)} H^{(v)}) + \\ \sum_{m=1}^{M} B_m^{(v)} - \sum_{m=1}^{M} A_m^{(v)} (MI)^{(-1)} \end{aligned} \tag{16}$$

$$A_m^{(v)} = \Omega^{(v)}(\alpha_m), B_m^{(v)} = \Omega^{(v)}(k_m). \tag{17}$$

It is difficult to use the KKT condition because of the complexity of $D(H^{(v)})$. We chose the traditional gradient descent method to update the data.

$$H^{(v)^{t+1}} = H^{(v)^t} - \tau(D(H^{(v)})). \tag{18}$$

where τ denotes the step size. In terms of targeting, there are several M ways to expand the M-order tensor. Our model pair uses unfolding of the three modalities. α_m is the Lagrange multiplier corresponding to the constraint $P_m h = k_m$. Where I is the identity matrix. Operator $\Omega^{(v)}(.)$ only selects $N \times N$ elements corresponding to the v-th views and reshapes them to the $N \times N$ dimensional matrices $A_m^{(v)}$ and $B_m^{(v)}$ corresponding to $H^{(v)}$.

h-sub-problem: We update h directly for each element in $H^{(v)}$ by replacing it directly:

$$h_* \leftarrow H^{(v)}. \tag{19}$$

$Q^{(v)}$-sub-problem: When other variables are fixed, the subproblem process for updating $Q^{(v)}$ is as follows:

$$\min \sum_{v=1}^{V} \sum_{i=1}^{n_0} \left\| Y_{o_i}^{(v)} - H_{o_i}^{(v)} Q^{(v)} \right\|_F^2 \tag{20}$$

$$Q^{(v)} = Q^{(v)} \frac{{H_o^{(v)}}^T Y_o^{(v)}}{{H_o^{(v)}}^T H_o^{(v)} Q^{(v)}}. \tag{21}$$

U-sub-problem: When other variables are fixed, the subproblem process for updating U is as follows:

$$\min_{U^T U = I} -\lambda_2 \sum_{v=1}^{V} Tr(H^{(v)} {H^{(v)}}^T U U^T) \Leftrightarrow \tag{22}$$

$$\max_{U^T U = I} Tr(\sum_{v=1}^{V} U^T (H^{(v)} {H^{(v)}}^T) U). \tag{23}$$

These problems can be computed simply using eigenvalue decomposition. The best solution for variable U is the eigenvector set corresponding to the first c largest eigenvalues of the matrix ($\sum_{v=1}^{V} H^{(v)} H^{(v)^T}$).

K_m-sub-problem: The formula for the sub-problem K_m is as follows:

$$\begin{aligned} K_m^* &= \underset{G_m}{\operatorname{argmin}}\, \sigma_m \|K_m\|_* + \mu\Phi(\alpha_m, P_m h - k_m) \\ &= prox_{\beta_m}^{tr}(\Omega_{(m)}(P_m h + \alpha_m)). \end{aligned} \tag{24}$$

Here, we define $\Phi(\alpha_m, P_m h - k_m) = \frac{1}{2}\|P_m h - k_m\|_F^2 + \langle \alpha_m, P_m h - k_m \rangle$, where $\langle ., . \rangle$ is the inner product of a matrix, and μ is a positive penalty parameter. The $\Omega_{(m)}(P_m h + \alpha_m)$ operator converts the vector $P_m h + \alpha_m$ into a matrix with the corresponding modal expansion. $\beta_m = (\sigma_m/\mu)$ is the threshold value for the soft-threshold operation of the spectrum. $prox_{\beta_m}^{tr}(L) = U\max(S - \beta_m, 0)V^T$ with $L = UST^T$ is the singular value decomposition (SVD) of the matrix L, and the max operation is performed element-wise. Intuitively, the solution is truncated according to the subspace representation tensor $\mathcal{H}$.

k_m-sub-problem: We update k_m by K_m:

$$k_m^* \leftarrow K_m. \tag{25}$$

α_m-subproblem: The variable α_m is updated by:

$$\alpha_m^* = \alpha_m + (P_m h - k_m). \tag{26}$$

4. Theoretical Analysis of IMC-NLT

The complete procedure of IMC-NLT is summarized in Algorithm 1. The values of variables $H^{(v)}$, $Q^{(v)}$, K_m, k_m, α_m, μ, and U are updated iteratively until the number of iterations reaches the maximum, or the difference between the target values in two consecutive steps is less than the set threshold ε. In the following, we analyze the computational cost and the convergence properties.

4.1. Computational complexity

We calculated complexity in five steps. In the first step of mechanical filling, the time cost was $O(\sum_{v=1}^{V} n_o^v d_v)$. In the second step of the SVT operation on the tensor, we use Lemma 1 for the complexity calculation:

Lemma 1 (Oh et al., 2015). Let $A = QB \in \mathbb{R}^{m\times n}$, where $Q \in \mathbb{R}^{m\times n}$ has orthonormal columns. Then, we have:

$$\mathbb{S}_\tau(A) = Q\mathbb{S}_\tau(B), \tag{27}$$

where $\mathbb{S}_\tau(.)$ is the SVT operator with time cost $O(n^3)$. In order to reduce the expensive cost, we perform the SVT operation on the smaller matrix $B \in \mathbb{R}^{n\times n}$ instead, if the matrix $Q \in \mathbb{R}^{m\times n}$ is available($m \geq n$). In the third step of recomposing the tensors into vectors, the time cost is $O(\sum_{v=1}^{V} n^v c)$. In the fourth step of derivative gradient descent, the vector reconstitution tensors can be approximated as $O(n^2 c + cnd_v)$. The time cost is $O(n^3)$ for the fifth step of calculating eigenvalues and eigenvectors.

Therefore, the computational complexity of IMC-NLT proposed in this study is $O(\sum_{v=1}^{V} n_o^v d_v + n^v c + l(n^3 + n^2 c + cnd_v + n^3))$, where l is the iteration number. Because c is usually much smaller than d_v and n, the overall complexity can be approximated as $O(ln^3)$. To ensure that the algorithm achieves faster convergence and better experimental results, we obtained $H^{(v)}$ and $Q^{(v)}$ through the non-negative matrix factorization of $Y^{(v)}$.

4.2. Convergence Analysis

In this section, we demonstrate the convergence of the IMC-NLT iterative algorithm.

Theorem 1. The objective function of the IMC-NLT: $J(H^{(v)}, Q^{(v)}, K_m, U, \alpha_m)$ in (15) is bounded. The proposed optimization algorithm monotonically reduced the value of the objective function.

Proof. Because this is the sum of norms with positive penalty parameters, problem (15) is bounded from below: ($H^{(v)}$, $Q^{(v)}$, U) obtained by formulas (18), (21), and (23) are the minimum points corresponding to subproblems (16),

Algorithm 1 IMC-NLT

Require: Incomplete multi-view dataset $X^{(v)}$, parameters σ_m, λ_1, λ_2, μ, maximum number of iterations t_{max}, the threshold $\varepsilon = 10^{-7}$.

Ensure: The resulting clusters.

Initialize $\rho = 1.1, max_\mu = 10^{10}$;
Initialize $H^{(1)}, H^{(2)} ..., H^{(v)}$;
Initialize $Q^{(1)}, Q^{(2)} ..., Q^{(v)}$;
Initialize $K_1 = 0, ..., K_M = 0$;
Initialize $\alpha_1 = \alpha_2 ..., \alpha_M = 0$;
while not converge **do**
  for $v = 1$ to V **do**
    Update $H^{(v)}$ via Eq. (18);
    Update $Q^{(v)}$ by solving Eq. (21);
  end for
  Update K_m via Eq. (24);
  Update k_m by solving Eq. (25);
  Update α_m via Eq. (26);
  Update μ by $\mu = \min(\rho\mu; \max_\mu)$;
  Update U via Eq. (23);
  Check the convergence conditions:
  $\left|\frac{J_{t+1} - J_t}{J_t}\right| < \varepsilon$;
  Until (15) reaches the maximum number of iterations t_{max} or convergence.
end while
Apply K-means to U for producing the resulting clusters.

(20), and (22). For the subproblem of solving K_m, the fast method of formula (24) is used for the approximate matrix inversion calculation. h and k_m are updated by directly substituting the corresponding elements. Intuitively, multiplier α_m is updated according to the updating rule of multipliers.

Let $\left\{H^{(v)}, Q^{(v)}, K_m, U, \alpha_m\right\}_{t=1}$ be a bounded monotonically decreasing sequence determined by Theorem 1. According to the bounded monotone convergence theorem (Rudin et al., 1976), the objective function monotonically decreases and is bounded.

Theorem 2. Algorithm 1 converges to a minimum under the updating of the value of $J(H^{(v)}, Q^{(v)}, K_m, U, \alpha_m)$ in each optimization step.

Proof. Suppose that $\left\{H^{(v)}, Q^{(v)}, K_m, U, \alpha_m\right\}_t$ and $\left\{H^{(v)}, Q^{(v)}, K_m, U, \alpha_m\right\}_{t+1}$ represent the iterative sequence of the (t) and $(t+1)$ times of problem (15), respectively. According to the previous sub-problem optimization steps, we can conclude that these sub-problems are not only convex optimization problems but also have closed solutions. By solving the above sub-problems one by one, we can get the following formula:

$$\begin{aligned} & J(H_t^{(v)}, Q_t^{(v)}, (K_m)_t, U_t, (\alpha_m)_t) \geq J(H_{t+1}^{(v)}, Q_t^{(v)}, (K_m)_t, U_t, (\alpha_m)_t) \\ \geq & J(H_{t+1}^{(v)}, Q_{t+1}^{(v)}, (K_m)_t, U_t, (\alpha_m)_t) \geq J(H_{t+1}^{(v)}, Q_{t+1}^{(v)}, (K_m)_{t+1}, U_t, (\alpha_m)_t) \\ \geq & J(H_{t+1}^{(v)}, Q_{t+1}^{(v)}, (K_m)_{t+1}, U_{t+1}, (\alpha_m)_t) \geq J(H_{t+1}^{(v)}, Q_{t+1}^{(v)}, (K_m)_{t+1}, U_{t+1}, (\alpha_m)_{t+1}) \end{aligned} \tag{28}$$

Eq. (28) proves that the decrement of the objective function is achieved through the iterative updating of variables in sequence $\left\{H^{(v)}, Q^{(v)}, K_m, U, \alpha_m\right\}_t$. So we have completed the proof of Theorem 2.

The above two theorems ensure that by using the proposed optimization method, the objective function is monotonically decreasing and bounded. Meanwhile, the sequence is continuously optimized and can converge to the minimum value.

5. Experimental Evaluation

In this section, we evaluate the effectiveness of the IMC-NLT in comparison with the experimental results of the baseline algorithm on five incomplete multi-view datasets.

5.1. Datasets

Table 1 lists the five datasets widely used in our experiments.

Table 1: Statistics of the datasets

Dataset	Clusters	Views	Instances	Features
SensIT300	3	2	300	50/50
Statlog	7	2	2310	9/10
Wisconsin	5	2	265	1703/265
WebKB	2	2	1051	1840/3000
Caltech101-7	7	6	1474	48/40/254/1984/512/928

- **SensIT300**[1]**:** Data collected from distributed sensors in an intelligent transportation system. A total of 300 instances were divided into three categories, which corresponded to three types of transportation in real life. Each data instance has two information views: sound information recorded by a sensor, and vibration information, in which each view contains 50-dimensional characteristic attributes.
- **Statlog**[2]**:** An image segmentation dataset was randomly selected from a database of images from seven categories. The images were manually segmented to create classification for each pixel. Collected by the vision group at the University of Massachusetts, this dataset contains 2310 instances with corresponding categories under two views. The characteristic dimension of one view is nine, whereas the characteristic dimension of the other view is 10.
- **Wisconsin** [3]**:** A set of webpages collected from the University of Wisconsin website. The five types of webpages are student, project, course, staff, and faculty. Each has two views: the content view and the reference view. In the content view, each webpage consists of 1703 words. The reference view is described by the reference relationships between a page and other pages.
- **WebKB** [4]**:** A set of course and non-course documents. Each document has two representations: the text content of the webpage and the anchor text with links to other webpages pointing to the webpage. Based on the page representation, 3000 features were selected. For linked representations, 1840 features were generated.
- **Caltech101-7**[5]**:** Caltech101-7 is a subset of the real dataset Caltech101 with seven categories, which are from various categories such as football, camera, and chair. This dataset contains six views: Gabor, WM, Centrist, HOG, GIST, and LBP.

Specifically, we used the SensIT300, Starlog, Wisconsin, and WebKB datasets to build incomplete multi-view data. In our experiments, approximately 10%, 30%, and 50% of the instances were randomly deleted from each view of the four databases. For the accuracy of the results, we perform standard validation on these datasets (Rodriguez et al., 2010), and produced the results of the average calculation. All of the experimental codes were developed using MATLAB 2015a running on an Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz with 16-GB RAM with the Win 10 system.

[1]https://github.com/Liuzhenjiao123/multiview-data-sets/blob/master/sensIT300.mat
[2]https://github.com/Liuzhenjiao123/multiview-data-sets/tree/master
[3]https://lig-membres.imag.fr/grimal/data.html
[4]https://github.com/Liuzhenjiao123/dataset4
[5]https://github.com/Liuzhenjiao123/data5

5.2. Baseline methods

Against the five datasets, IMC-NLT was compared with five IMC methods: IMSC-AGL (Wen et al., 2018a), DAIMC (Hu & Chen, 2018), UEAF (Wen et al., 2019), IMC-GRMF(Wen et al., 2018b), and HCP-IMSC(Li et al., 2022).

- **IMSC-AGL** An algorithm first exploits low-rank representations for multi-view adaptive learning of graphs and then uses spectral constraints to obtain better low-dimensional representations.
- **DAIMC** Incomplete multi-view clustering algorithm based on weighted semi-non-negative matrix factorization (semi-NMF). It exploits a weight matrix to adapt to a variety of incomplete cases and uses $L_{2,1}$ -norm regularization to obtain a cluster-friendly basis matrix shared by views.
- **UEAF** A unified and robust embedding alignment model for incomplete multi-view clustering. To maintain the consistency of the local semantics of the view and infer incomplete information, it learns the local structure shared by the views by reversing the graph regularization.
- **IMC-GRMF** Incomplete multi-view clustering method based on matrix factorization. For better integration, IMC-GRMF uses the local information of each view to facilitate fusion of the complementary information of views to obtain a shared representation. Orthogonal constraints can effectively handle out-of-sample problems.
- **HCP-IMSC** This is an incomplete multi-view clustering method that is based on hypergraph induction and tensor decomposition. It effectively uses the correlation of high-order information to recover missing data and combines the affinity matrix, tensor decomposition, and missing-view recovery into one framework.

5.3. Evaluation metrics

The resulting clusters by the algorithms in the experiments are evaluated by normalized mutual information (NMI) (Estévez et al., 2009), clustering accuracy (ACC) (Cai et al., 2005), Adjusted Rand index (ARI) (Romano et al., 2016), and F1 Score (F1) (Wang et al., 2015b). A higher value of these metrics indicates a higher cluster quality. The NMI is defined as follows:

$$NMI = \frac{\sum_{i=1}^{C}\sum_{j=1}^{C} N_{i,j} \ln \frac{N_{i,j}}{N_i \hat{N}_j}}{\sqrt{(\sum_{i=1}^{C} N_i \ln \frac{N_i}{N})(\sum_{j=1}^{C} \hat{N}_j \ln \frac{\hat{N}_j}{N})}} \tag{29}$$

where N is the number of instances in a complete view, N_i is the number of instances in the i-th cluster, $\hat{N}_j$ is the number of instances of the j-th label, and $N_{i,j}$ is the number of samples that exist in both the i-th cluster and j-th clusters. The ACC measures the quality of clusters as follows:

$$ACC = \frac{\sum_{i=1}^{N} \delta(map(r_i), l_i)}{N} \tag{30}$$

where r_i is the cluster label of x_i, l_i is the exact class label, and N is the number of samples. When x=y, $\delta(x, y)$ is equal to 1; otherwise, it is 0. Map(r_i) is the optimal permutation mapping function obtained. The ARI is formulated as:

$$ARI = \frac{RI - E[RI]}{(max(RI) - E[RI])} \tag{31}$$

where $E[RI]$ resents the expected value of RI and RI is a random index used to measure the similarity between two clusters. It is defined as follows:

$$RI = \frac{TP + TN}{TP + FP + FN + TN} \tag{32}$$

where TP is the true positive, TN is the true negative, FP is false positive, and FN is false negative. The F1 is defined as:

Table 2: Mean NMIs, ACCs, ARIs and F1 of different methods on SensIT300 , Statlog and Wisconsin datasets

Dataset	Method \ PER	NMI			ACC			ARI			F1		
		0.1	0.3	0.5	0.1	0.3	0.5	0.1	0.3	0.5	0.1	0.3	0.5
SensIT300	IMSC-AGL	0.23	0.21	0.16	0.66	0.63	0.61	0.25	0.21	0.17	0.51	0.48	0.45
SensIT300	DAIMC	0.21	0.18	0.16	0.64	0.61	0.59	0.23	0.18	0.16	0.49	0.46	0.45
SensIT300	UEAF	0.21	0.18	0.16	0.65	0.61	0.59	0.24	0.18	0.16	0.51	0.45	0.44
SensIT300	IMC-GRMF	0.16	0.08	0.06	0.61	0.51	0.47	0.17	0.08	0.06	0.45	0.39	0.37
SensIT300	HCP-IMSC	0.32	**0.30**	0.21	**0.72**	0.65	0.57	**0.35**	**0.30**	**0.22**	**0.56**	0.50	0.48
SensIT300	IMC-NLT	**0.32**	0.28	**0.25**	0.69	**0.66**	**0.61**	0.31	0.24	0.19	0.54	**0.51**	**0.49**
Statlog	IMSC-AGL	0.44	0.42	0.38	0.55	0.54	0.48	0.79	0.31	0.29	0.43	0.41	0.41
Statlog	DAIMC	0.47	0.41	0.34	0.57	0.51	0.45	0.71	0.27	0.19	0.46	0.39	0.33
Statlog	UEAF	0.49	0.38	0.35	0.48	0.46	0.44	**0.81**	0.24	0.18	0.41	0.36	0.32
Statlog	IMC-GRMF	0.11	0.21	0.14	0.28	0.38	0.31	0.68	0.15	0.08	0.22	0.28	0.22
Statlog	HCP-IMSC	0.51	0.46	0.40	0.58	0.54	0.49	0.38	0.32	0.26	0.48	0.44	0.39
Statlog	IMC-NLT	**0.63**	**0.56**	**0.48**	**0.68**	**0.62**	**0.58**	0.54	**0.45**	**0.35**	**0.61**	**0.53**	**0.45**
Wisconsin	IMSC-AGL	0.21	0.19	0.14	0.43	0.39	0.34	0.17	0.11	0.08	0.51	0.33	0.32
Wisconsin	DAIMC	0.31	0.27	0.24	0.51	0.44	0.46	0.25	0.17	0.17	0.49	0.39	0.39
Wisconsin	UEAF	0.36	**0.41**	0.34	0.61	0.57	0.51	0.35	0.34	0.25	0.51	0.51	0.44
Wisconsin	IMC-GRMF	0.26	0.19	0.11	0.44	0.37	0.33	0.15	0.11	0.05	0.45	0.34	0.31
Wisconsin	HCP-IMSC	0.27	0.24	0.26	0.50	0.40	0.49	0.21	0.17	0.20	0.42	0.39	0.40
Wisconsin	IMC-NLT	**0.48**	0.39	**0.37**	**0.74**	**0.68**	**0.69**	**0.48**	**0.38**	**0.32**	**0.67**	**0.61**	**0.59**
WebKB	IMSC-AGL	0.66	0.31	0.5	0.95	0.83	0.91	0.79	0.41	0.68	0.93	0.77	0.88
WebKB	DAIMC	0.61	0.52	0.42	0.93	0.90	0.85	0.70	0.64	0.51	0.91	0.88	0.84
WebKB	UEAF	0.68	**0.71**	**0.65**	0.95	**0.96**	**0.95**	0.81	**0.82**	**0.79**	0.93	**0.93**	**0.93**
WebKB	IMC-GRMF	0.52	0.34	0.03	0.92	0.92	0.61	0.68	0.52	0.04	0.89	0.84	0.59
WebKB	HCP-IMSC	0.71	0.68	0.61	0.95	0.93	0.92	0.82	0.79	0.77	0.93	0.92	0.91
WebKB	IMC-NLT	**0.71**	0.65	0.53	**0.96**	0.94	0.92	**0.83**	0.77	0.66	**0.94**	**0.93**	0.90

(a) **(b)**

Fig. 4. Cluster structure illustration on two incomplete multi-view datasets. (a) Statlog with 10% incomplete instances of each view, (b) visualization on WebKB with 10% incomplete

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{33}$$

where $precision = \frac{TP}{TP + FP}$, and $recall = \frac{TP}{TP + FN}$.

5.4. Evaluations on clustering performance and discussion

In our experiment, we selected 10%, 30%, and 50% of the total number of instances to randomly delete them from each view of the five datasets. Table 2 shows the average performance of NMI, ACC, ARI, and F1 for the five

Table 3: Mean NMIs, ACCs, ARIs and F1 of different clustering methods on SensIT300 , Statlog and Wisconsin datasets

Dataset	Method \ PER	NMI			ACC			ARI			F1		
		0.1	0.3	0.5	0.1	0.3	0.5	0.1	0.3	0.5	0.1	0.3	0.5
SensIT300	K-means	0.32	0.28	0.25	0.69	0.66	0.61	0.31	0.24	0.19	0.54	0.51	0.49
SensIT300	Fuzzy	0.32	0.27	0.25	0.68	0.61	**0.64**	0.28	0.21	**0.22**	0.54	0.49	0.49
SensIT300	Spectral	**0.35**	**0.31**	**0.27**	**0.71**	**0.68**	0.62	**0.33**	**0.31**	0.21	**0.57**	**0.56**	**0.51**
Statlog	K-means	**0.63**	**0.56**	**0.48**	**0.68**	**0.62**	**0.58**	**0.54**	**0.45**	**0.35**	**0.61**	**0.53**	**0.45**
Statlog	Fuzzy	0.43	0.27	0.25	0.51	0.36	0.36	0.31	0.13	0.14	0.41	0.29	0.29
Statlog	Spectral	0.62	0.44	0.36	0.54	0.45	0.33	0.38	0.28	0.09	0.51	0.38	0.29
Wisconsin	K-means	**0.48**	**0.39**	**0.37**	**0.74**	**0.68**	**0.69**	**0.48**	0.38	**0.32**	**0.67**	**0.61**	**0.59**
Wisconsin	Fuzzy	0.42	0.35	0.31	0.59	0.45	0.46	0.39	0.28	0.21	0.57	0.42	0.42
Wisconsin	Spectral	0.40	0.39	0.33	0.58	0.66	0.59	0.36	**0.43**	0.32	0.54	0.61	0.51
WebKB	K-means	**0.71**	**0.65**	0.53	**0.96**	**0.94**	0.92	**0.83**	**0.77**	0.66	**0.94**	**0.93**	0.90
WebKB	Fuzzy	0.69	0.55	0.54	0.96	0.93	**0.93**	0.83	0.72	**0.71**	0.94	0.91	**0.91**
WebKB	Spectral	0.68	0.54	**0.89**	0.96	0.92	0.92	0.82	0.66	0.66	0.94	0.90	0.89

Table 4: Two incomplete multi-view clustering methods based on tensor models show different performances in terms of ACC, F1, Running time (seconds), and computational complexity with 90% incomplete instances of each view on the Caltech101-7 dataset

Dataset	Method	ACC	F1	Running time (seconds)	computational complexity
Caltech101-7	HCP-IMSC	0.37	0.40	106.04	$O(Vn^3 + V(n-n_o)^3 + cnVlog(V) + cn^2V)$
Caltech101-7	IMC-NLT	**0.43**	**0.45**	**92.75**	$O(n^3)$

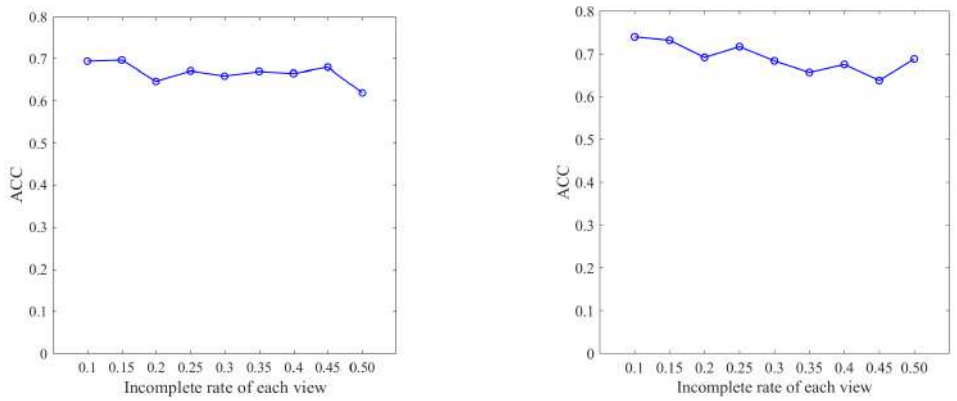

Fig. 5. Robust performance on two incomplete multi-view datasets: (a) SensIT300 (b) Wisconsin

different methods on the five incomplete multi-view datasets. From this table, we can observe the following results.

1) The performance of HCP-IMSC was better than that of other incomplete multi-view clustering methods on most datasets. This indicates that the tensor-based model can achieve incomplete multi-view clustering, which preserves the high-order correlation advantage. In addition, compared to HCP-IMSC, our IMC-NLT can produce good results from most datasets with high missing rates. This verifies that building a tensor model directly from prepopulated modal data can better recover missing data and reduce the impact of noise.

2) Compared with the other five algorithms, the IMC-NLT has obvious advantages. For example, on the statlog and Wisconsin datasets, according to various clustering indicators, IMC-NLT performs the best. When the missing rate of the Wisconsin dataset was set to 10%, our algorithm achieved an ACC score that was approximately 13%

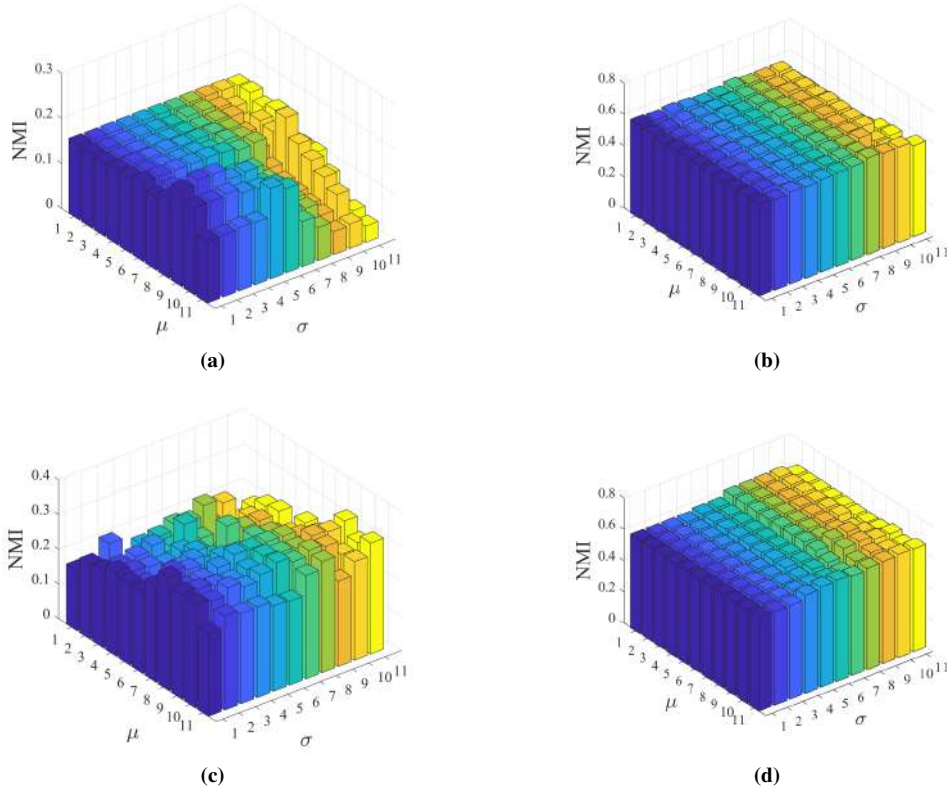

Fig. 6. NMI versus parameters σ and μ of IMC-NLT on different datasets with various percentages of incomplete instances of each view (a) SensIT300 with 50%, (b) Statlog with 10%, (c) Wisconsin with 50%, and (d) WebKB with 10% .

higher than that of the second-best method. Although the advantages of our algorithm on SensIT300 and Webkb are not as prominent as those of the above datasets, the difference between our method and other superior algorithms is not evident in most cases. The IMC-NLT is relatively stable for datasets with different missing rates.

3) From table 2, we can see that our method is superior to other views based on recovery methods such as UEAF. This shows that IMC-NLT not only effectively utilises the specific information of each view but also builds a unified structure to effectively maintain the semantic relationships among the different views. Therefore, IMC-NLT can capture useful information with complex interactions between views to recover the missing data.

4) The proposed IMC-NLT method was more robust than the DAIMC, IMSC-AGL, and IMC-GRMF methods. Our method neither indirectly obtains consensus representations from individual expressions of all viewpoints, such as DAIMC, nor is it constrained by the existing incomplete data. At the same time, it prevents the IMSC-AGL from extracting hidden information from incomplete and complex data. Our method is suitable for various complex missing situations, and ensures the filling effect of complex missing situations through an effective filling mechanism.

5.5. Clustering Performance on Three Clustering Methods

Table 3 reports the performance of the fusion effect on three clustering algorithms. Compared with other incomplete multi-view clustering algorithms, the overall experimental result of applying K-means is the best, followed by Fuzzy clustering and spectral clustering. Again, our algorithm has demonstrated the excellent performance in incomplete multi-view data filling and later fusion.

5.6. Visualization of Clustering Results

To further demonstrate the advantages of the IMC-NLT algorithm, we visualised the distributions of experimental clustering results. For example, on the Statlog and WebKB datasets, the IMC-NLT algorithm can produce different

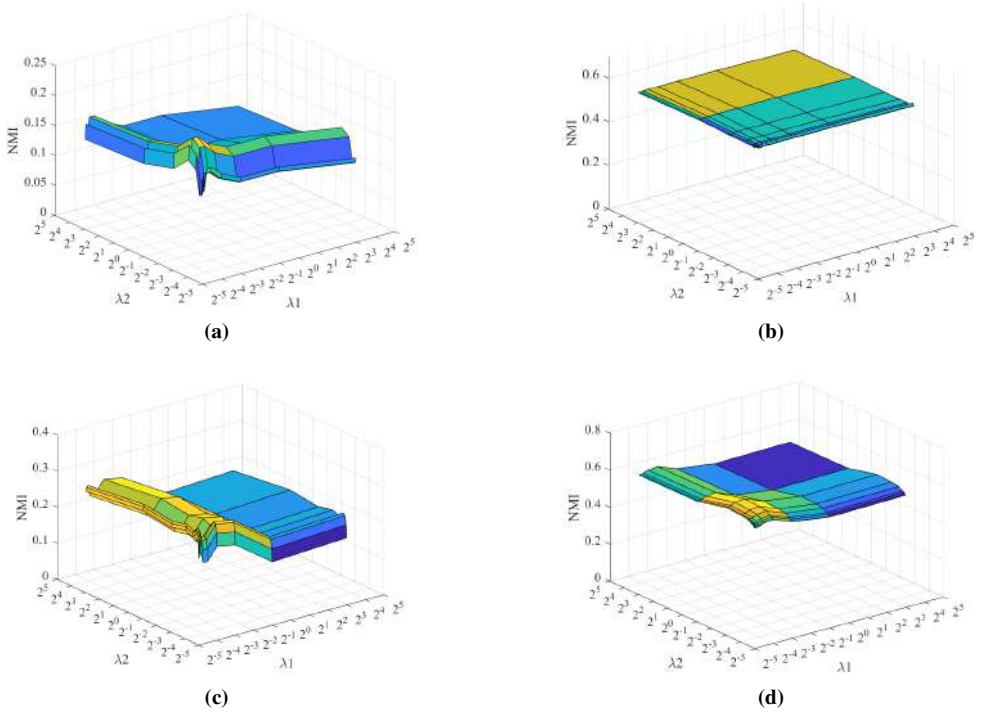

Fig. 7. NMI versus parameters λ_1 and λ_2 of IMC-NLT on the four datasets with various percentages of incomplete instances with each view, respectively. (a) SensIT300 with 50%, (b) Statlog with 10%, (c) Wisconsin with 50%. and (d) WebKB with 10%.

colour clusters with obvious grouping structures, as illustrated in Fig. 4.

5.7. The effectiveness of IMC-NLT on large-scale datasets

The performances of the two methods for incomplete multi-view clustering based on the tensor model are presented in Table 4. From this table, we observe that our method demonstrates certain advantages in terms of clustering indicators, running time, and complexity. In addition, the proposed method performed well on larger datasets.

5.8. Robustness Analysis

To investigate the robustness of the proposed algorithm, we examined the clustering results of the IMC-NLT algorithm on SensIT300 and Wisconsin datasets with different missing rate intervals of 5%, as shown in Fig. 5. In terms of accuracy, our method remains relatively robust as the missing rate of data points increases. This shows the capacity of the IMC-NLT to handle missing data filling.

6. Parameters and Convergence of IMC-NLT

6.1. Parameter sensitivity analysis

In this section, we present several comparative experiments to demonstrate the effects of different IMC-NLT parameter values. We focus mainly on the following parameters of IMC-NLT: the non-negative matrix decomposition control parameter λ_1, feature orthogonal constraints to limit the control parameter λ_2, and low-rank force coding. We set the M parameters inside to be equal, that is, $\sigma_1 = .. = \sigma_M = \sigma$, and accordingly, tune the parameter σ and Lagrange operator control parameter μ.

1)Parameters σ and μ: We show the NMI versus the two parameters σ and μ on the datasets of SensIT300, Statlog, Wisconsin, and WebKB with different incomplete-view rates in Fig. 6. For example, on the Statlog dataset,

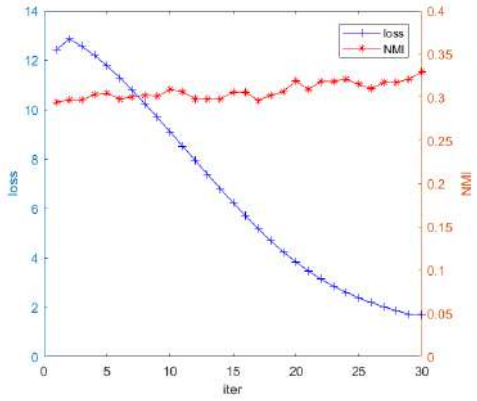

Fig. 8. The objective function loss and NMI v.s. iterations on (a) SensIT300 with 10% incomplete instances of each view.

the experimental results showed that the best performance was achieved when σ ranges between {0,9} and μ between {1,11}. In this study, we find that our algorithm achieves the best clustering when σ takes values in the range of {1,6} and μ in the range of {8,10} by using a latticework search. Meanwhile, on the Statlog and WebKB datasets, IMC-NLT exhibited low sensitivity to its parameters.

2) Parameters λ_1 and λ_2: From Fig. 7, we can observe that parameters λ_1 and λ_2 are insensitive to the Statlog and WebKB datasets. For SensIT300, the algorithm performs well when the parameters λ_1 and λ_2 are between $\{2^0,2^1\}$ and $\{2^{-1}, 2^1\}$, respectively. Similarly, there is a certain parameter sensitivity in the Wisconsin dataset. When the parameters λ_1 and λ_2 are between $\{2^{-2},2^0\}$ and $\{2^1, 2^2\}$, respectively, IMC-NLT achieves good performance.

The adaptive selection of various parameters for different datasets to reach their optimal values is problematic. Determine the most suitable parameters for the proposed IMC-NLT model. We solve this problem by choosing a combination strategy to find the optimal parameters. Specifically, we first fixed the insensitive parameters to set λ_1 and λ_2 with a fixed value range, and then ran IMC-NLT with different values of σ and μ. As such, the optimal parameter value set of the algorithm is obtained, experiments are conducted, and the results are compared and reported.

6.2. Convergence analysis

To better deal with the complex objective function, we split the objective function into several subproblems and use an iterative optimization algorithm to monotonically decrease it until convergence. The objective function values are plotted in Fig. 8 according to the corresponding NMI with the number of iterations (within 30 iterations). As shown, the loss of the objective function decreases monotonically and converges to a stationary point. This ensures the convergence of the proposed optimization method.

7. Conclusion

In this paper, we have presented a novel algorithm for incomplete multi-view clustering called IMC-NLT based on NMF and low-rank tensor fusion. IMC-NLT relies on both the modal unified dimensional structure and low-rank tensor. IMC-NLT can effectively integrate the information hidden in datasets with the same view and between views. Moreover, IMC-NLT not only deals effectively with various incomplete data but also has low sensitivity to its parameters. We have carried out comprehensive experiments on the five representative data sets by comparing IMC-NLT with state-of-the-art algorithms. The experimental results have shown that our method can achieve good clustering results with stability.

In the future, we intend to extend the low-rank tensor filling model to the deep model to obtain more efficient view representations of incomplete multi-view clustering.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 62076047) and the Dalian Innovation Fund (No. 2021JJ12SN44).

References

Cai, D., He, X., & Han, J. (2005). Document clustering using locality preserving indexing. *IEEE Transactions on Knowledge and Data Engineering*, *17*, 1624–1637.

Cai, L., Wang, Z., Gao, H., Shen, D., & Ji, S. (2018). Deep adversarial learning for multi-modality missing data completion. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1158–1166).

Eaton, E., Desjardins, M., & Jacob, S. (2010). Multi-view clustering with constraint propagation for learning with an incomplete mapping between views. In *Proceedings of the 19th ACM international conference on Information and knowledge management* (pp. 389–398).

El Hajjar, S., Dornaika, F., Abdallah, F., & Barrena, N. (2022). Consensus graph and spectral representation for one-step multi-view kernel based clustering. *Knowledge-Based Systems*, *241*, Article 108250.

Estévez, P. A., Tesmer, M., Perez, C. A., & Zurada, J. M. (2009). Normalized mutual information feature selection. *IEEE Transactions on neural networks*, *20*, 189–201.

Gunasekar, S., Woodworth, B. E., Bhojanapalli, S., Neyshabur, B., & Srebro, N. (2017). Implicit regularization in matrix factorization. *Advances in Neural Information Processing Systems*, *30*, 1968–1974.

Hu, M., & Chen, S. (2018). Doubly aligned incomplete multi-view clustering. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI* (pp. 2262–2268).

Hu, M., & Chen, S. (2019). One-pass incomplete multi-view clustering. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*, 3838–3845.

Hu, Y., Song, Z., Wang, B., Gao, J., Sun, Y., & Yin, B. (2021). Akm 3 c: Adaptive k-multiple-means for multi-view clustering. *IEEE Transactions on Circuits and Systems for Video Technology*, *31*, 4214–4226.

Kim, Y.-D., & Choi, S. (2009). Weighted nonnegative matrix factorization. In *2009 IEEE international conference on acoustics, speech and signal processing* (pp. 1541–1544).

Kumar, A. e. R. (2011). Co-regularized multi-view spectral clustering. *Advances in Neural Information Processing Systems*, *24*, 1413–1421.

Li, J., Yue, W., Zhao, J., & Ke, L. (2017). Low-rank discriminant embedding for multiview learning. *IEEE Transactions on Cybernetics*, *47*, 3516–3529.

Li, S.-Y., Jiang, Y., & Zhou, Z.-H. (2014). Partial multi-view clustering. In *Proceedings of the AAAI conference on artificial intelligence* (p. 6151–6159).

Li, Z., Tang, C., Zheng, X., Liu, X., Zhang, W., & Zhu, E. (2022). High-order correlation preserved incomplete multi-view subspace clustering. *IEEE Transactions on Image Processing*, *31*, 2067–2080.

Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., & Ma, Y. (2012). Robust recovery of subspace structures by low-rank representation. *IEEE transactions on pattern analysis and machine intelligence*, *35*, 171–184.

Liu, X., Zhu, X., Li, M., Wang, L., Zhu, E., Liu, T., Kloft, M., Shen, D., Yin, J., & Gao, W. (2019). Multiple kernel *k*-means with incomplete kernels. *IEEE transactions on pattern analysis and machine intelligence*, *42*, 1191–1204.

Oh, T.-H., Matsushita, Y., Tai, Y.-W., & So Kweon, I. (2015). Fast randomized singular value thresholding for nuclear norm minimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4484–4493).

Quanz, J., Brian & Huan (2012). Conet: Feature generation for multi-view semi-supervised learning with partially observed views. In *Proceedings of the 21st ACM international conference on Information and knowledge management* (pp. 1273–1282).

Rodriguez, J. D., Perez, A., & Lozano, J. A. (2010). Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE Trans Pattern Anal Mach Intell*, *32*, 569–575.

Romano, S., Vinh, N. X., Bailey, J., & Verspoor, K. (2016). Adjusting for chance clustering comparison measures. *The Journal of Machine Learning Research*, *17*, 4635–4666.

Rudin, W. et al. (1976). *Principles of mathematical analysis* volume 3.

Shao, W., He, L., Lu, C.-t., & Philip, S. Y. (2016). Online multi-view clustering with incomplete views. In *2016 IEEE International Conference on Big Data (Big Data)* (pp. 1012–1017).

Shao, W., He, L., & Yu, P. S. (2015). Multiple incomplete views clustering via weighted nonnegative matrix factorization with $ł_{2,1}$ regularization. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 318–334).

Shi, S., Nie, F., Wang, R., & Li, X. (2022). Self-weighting multi-view spectral clustering based on nuclear norm. *Pattern Recognition*, *124*, Article 108429.

Si, X., Yin, Q., Zhao, X., & Yao, L. (2022). Consistent and diverse multi-view subspace clustering with structure constraint. *Pattern Recognition*, *121*, Article 108196.

Tao, H., Hou, C., Yi, D., Zhu, J., & Hu, D. (2019). Joint embedding learning and low-rank approximation: A framework for incomplete multiview learning. *IEEE transactions on cybernetics*, *51*, 1690–1703.

Tran, X. e. Z., Luan et Liu (2017). Imputation of missing categories via cascading residual autoencoder. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1405–1414).

Wang, C.-D., Lai, J.-H., & Philip, S. Y. (2015a). Multi-view clustering based on belief propagation. *IEEE Transactions on Knowledge and Data Engineering*, *28*, 1007–1021.

Wang, Q., Ding, Z., Tao, Z., Gao, Q., & Fu, Y. (2018). Partial multi-view clustering via consistent gan. In *2018 IEEE International Conference on Data Mining (ICDM)* (pp. 1290–1295).

Wang, Y., Li, J., Li, Y., Wang, R., & Yang, X. (2015b). Confidence interval for f1 measure of algorithm performance based on blocked 3×2 cross-validation. *IEEE Transactions on Knowledge and Data Engineering*, *27*, 651–659.

Wen, J., Xu, Y., & Liu, H. (2018a). Incomplete multiview spectral clustering with adaptive graph learning. *IEEE transactions on cybernetics*, *50*, 1418–1429.

Wen, J., Yan, K., Zhang, Z., Xu, Y., Wang, J., Fei, L., & Zhang, B. (2020). Adaptive graph completion based incomplete multi-view clustering. *IEEE Transactions on Multimedia*, *23*, 2493–2504.

Wen, J., Zhang, Z., Xu, Y., Zhang, B., Fei, L., & Liu, H. (2019). Unified embedding alignment with missing views inferring for incomplete multi-view clustering. In *The Thirty-Third AAAI Conference on Artificial Intelligence* (pp. 5393–5400).

Wen, J., Zhang, Z., Xu, Y., & Zhong, Z. (2018b). Incomplete multi-view clustering via graph regularized matrix factorization. In *Proceedings of the European conference on computer vision (ECCV) workshops* (pp. 593–608).

Wong, W. K., Han, N., Fang, X., Zhan, S., & Wen, J. (2019). Clustering structure-induced robust multi-view graph recovery. *IEEE Transactions on Circuits and Systems for Video Technology*, *30*, 3584–3597.

Xie, M., Ye, Z., Pan, G., & Liu, X. (2021). Incomplete multi-view subspace clustering with adaptive instance-sample mapping and deep feature fusion. *Applied Intelligence*, *51*, 5584–5597.

Xu, C., Guan, Z., Zhao, W., Wu, H., & Niu, B., Yunfei & Ling (2019). Adversarial incomplete multi-view clustering. In *International Joint Conference on Artificial Intelligence* (pp. 3933–3939).

Xu, C., & Tao, C., Dacheng & Xu (2015). Multi-view learning with incomplete views. *IEEE Transactions on Image Processing*, *24*, 5812–5825.

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

Yin, J., & Sun, S. (2022). Incomplete multi-view clustering with cosine similarity. *Pattern Recognition*, *123*, Article 108371.

Yin, Q., Wu, S., & Wang, L. (2017). Unified subspace learning for incomplete and unlabeled multi-view data. *Pattern Recognition*, *67*, 313–327.

Yuan, L., Wang, Y., Thompson, P. M., Narayan, V. A., & Ye, J. (2012). Multi-source learning for joint analysis of incomplete multi-modality neuroimaging data. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1149–1157).

Zhang, C., Fu, H., Liu, S., Liu, G., & Cao, X. (2015). Low-rank tensor constrained multiview subspace clustering. In *Proceedings of the IEEE international conference on computer vision* (pp. 1582–1590).

Zhang, G.-Y., Zhou, Y.-R., Wang, C.-D., Huang, D., & He, X.-Y. (2021). Joint representation learning for multi-view subspace clustering. *Expert Systems with Applications*, *166*, Article 113913.

Zhao, H., Liu, H., & Fu, Y. (2016). Incomplete multi-modal visual data grouping. In *International Joint Conference on Artificial Intelligence* (pp. 2392–2398).

Zhao, L., Chen, Z., Yang, Y., & Wang, V. C., Z Jane & Leung (2018). Incomplete multi-view clustering via deep semantic mapping. *Neurocomputing*, *275*, 1053–1062.

Zong, L., Zhang, X., Zhao, L., Yu, H., & Zhao, Q. (2017). Multi-view clustering via multi-manifold regularized non-negative matrix factorization. *Neural Networks*, *88*, 74–89.