Wireless Networks

Anchor-based Sparse Subspace Incomplete Multi-view Clustering

--Manuscript Draft--

Manuscript Number:		
Full Title:	Anchor-based Sparse Subspace Incomplete Multi-view Clustering	
Article Type:	Manuscript	
Keywords:	incomplete multi-view clustering, anchor graph, sparse subspace clustering, spectral clustering	
Corresponding Author:	Zhuo Wang Harbin University of Science and Technology Harbin, CHINA	
Corresponding Author Secondary Information:		
Corresponding Author's Institution:	Harbin University of Science and Technology	
Corresponding Author's Secondary Institution:		
First Author:	Ao Li, PhD	
First Author Secondary Information:		
Order of Authors:	Ao Li, PhD	
	Cong Feng, M.D.	
	Zhuo Wang, Master Degree	
	Yuegong Sun, Master Degree	
	Zizhen Wang, Master Degree	
Order of Authors Secondary Information:		
Funding Information:	National Natural Science Foundation of China (62071157)	Professor Ao Li
	Natural Science Foundation of Heilongjiang (YQ2019F011)	Professor Ao Li

Anchor-based Sparse Subspace Incomplete Multi-view Clustering

Ao Li[1], Cong Feng[1], Zhuo Wang[1*], Yuegong Sun[1], Zizhen Wang[1] and Ling Sun[2]

[1*]School of Computer Science and Technology, Harbin University of Science and Technology, No. 52 Xuefu Road, Harbin, 150080, Heilongjiang Province, China.

[2]School of Computer Science and Technology, Heilongjiang Collage of Business and Technology, No. 33 Qunying Street, Harbin, 150025, Heilongjiang Province, China.

*Corresponding author(s). E-mail(s): wz1997sapphire@163.com;

Abstract

In recent decades, multi-view clustering has received a lot of attention. The majority of previous research has assumed that all instances have complete views or at least one view that includes all instances. However, the incomplete multi-view clustering issue arises because real-world data frequently lack instances in each view. We propose a novel Anchor-based Sparse Subspace Incomplete Multi-view Clustering solution to this issue. Through a unified sparse subspace learning framework, the proposed method learns inter-view anchor-to-anchor and intra-view anchor-to-incomplete affinities and fuses them into a consensus sparse anchor graph, which yields a unified clustering result. Our method outperforms other incomplete multi-view clustering methods in three important ways:1) it uses a small number of hyperparameters to learn a sparse consensus graph from the data; 2) Because of the anchor-based graph construction, it can process large datasets; 3) It is naturally capable of handling both negative entries and multiple views. Last but not least, extensive experiments show that the proposed method is effective, supporting the claim that it consistently outperforms current clustering methods.

Keywords: incomplete multi-view clustering, anchor graph, sparse subspace clustering, spectral clustering

1 Introduction

Multi-view data are all over the world and have received a lot of attention from researchers over the past few decades [1, 2]. To take advantage of the power of multi-view data, numerous cutting-edge multi-view clustering algorithms have been proposed, and their clustering capabilities have steadily improved [3, 4]. However, the performance of existing multi-view clustering algorithms is seriously harmed by the fact that real-world data frequently exhibit incompleteness for a variety of reasons. Some languages may not have translations in the machine translation corpus, for instance. In multi-view data, incompleteness can roughly be divided into two categories. One is at the feature level, where specific instances lack particular features. At the instance level, there are instances that are absent entirely, which is the other case. The second scenario, which is typically referred to as *incomplete multi-view clustering* (IMVC), is the focus of this work.

The IMVC task has been the subject of numerous efforts. Non-negative matrix factorization and the l_1-norm are pioneering techniques used in PVC [5] to learn complete latent representations. The current IMVC methods can roughly be divided

into four categories based on the kinds of technologies that underpin them: NMF-based, graph-based, subspace-based, and neural network-based. The following is a brief synopsis.

NMF-based methods like MIC, ANIMC, DAIMC, UEAF, and GIMC-FLSD use weight matrices and non-negative matrix factorization to create a common latent space for clustering and reduce the impact of missing instances. A weighted semi-NMF model with $l_2, 1$-norm regularized regression on basis matrices is proposed by DAIMC[6]. Based on the weighted NMF framework that was suggested by MIC, UEAF [7] unifies embedding alignment and missing view inferring. ANIMC is based on a soft auto-weighted regularized adaptive semi-NMF model, [8]. A graph-regularized matrix factorization model is proposed by GIMC-FLSD [9] to preserve local geometric information. However, the NMF-based methods typically only take into account linear relationships between instances, ignoring potentially useful nonlinear data.

Graph-based methods [10–12] endeavor to capture latent consensus affinities for spectral clustering by constructing a complete common graph from incomplete multi-view data. The IMVC problem can be solved by combining adaptive graph learning and spectral clustering, as proposed by IMSC-AGL in [11]. An adaptive graph completion framework AGC-IMC is made by [10] to recover the similarity graphs of all views. The anchor graph technique is used to reduce complexity with just a few anchors due to the high computational complexity of eigendecomposition. Instances with all views are used as anchors to connect instances without views in APMC, a ground-breaking work that applies anchor graphs to the IMVC problem. However, APMC's performance is hampered by its use of a hand-crafted kernel function for graph construction and a straightforward weighted average for feature fusion.

Subspace-based methods [13, 14] learn a similarity graph for spectral clustering by assuming that data points are in a union of subspaces and utilizing the self-representation property. To improve clustering performance, the similarity graph can be subjected to a variety of constraints, such as the sparse [15] and the low-rank [16]. After projecting incomplete multi-view data to a complete latent representation, incomplete multi-view subspace clustering [13] learns the subspace representation from the latent representation. To make use of both unique and consistent information from incomplete multi-view data, V3H [14] incorporates view heredity and variation into subspace learning. Subspace-based methods also suffer from the high complexity brought on by spectral clustering, just like graph-based methods.

Neural network-based methods [17–19] project incomplete multi-view data into a latent semantic space by utilizing highly non-linear functions learned by contemporary neural networks. To directly generate clustering results, DIMC-net [17] combines autoencoders, the graph embedding layer, and the clustering layer into an end-to-end network. Deep Canonical Correlation Analysis, self-expression layer, $l_1, 2$-norm regularization, and autoencoder are all part of the unified framework that iCmSC [18] uses to learn more discriminative representation. Utilizing weighted fusion to obtain a complete latent representation in the face of incomplete instances is a common strategy employed by these techniques, as shown in [17, 18]. Utilizing contrastive learning [19] or generative adversarial networks [20] to generate the missing instances is yet another approach. However, it has long been known that neural network-based methods are difficult to interpret and prone to hyperparameter tuning.

The potential of anchor graphs as a promising method for modeling non-linear relationships among large-scale incomplete multi-view data has been overlooked by the majority of previous IMVC methods. In both multi-view clustering [21] and semi-supervised learning [22], the anchor graph has been shown to be a powerful and effective approximation of the prohibitive large-scale graph. Anchor-based Sparse Subspace Incomplete Multi-view Clustering is a novel approach to the IMVC problem that combines sparse subspace clustering [15] and anchor graph [21]. As depicted in Figure 1, the proposed method primarily combines inter-view consensus sparse subspace learning and intra-view anchor-based sparse subspace learning into a single objective function to produce a complete consensus anchor graph. On this graph, we apply the effective Fast Spectral Clustering [23] to produce a single clustering result. The following is a summary of our major contributions:

1. Using a conceptually simple and empirically effective sparse subspace learning framework, we propose the acquisition of inter-view anchor-to-anchor and intra-view anchor-to-incomplete affinities.
2. As our preliminary experiments demonstrate, our anchor-based graph learning approach is capable of processing large-scale incomplete multi-view datasets. By nature, it can also handle multiple views and negative entries.
3. Our method's superior performance in comparison to the most recent IMVC techniques demonstrates its usefulness for IMVC.

2 Related Work

sparse subspace clustering and the anchor graph are the most closely related aspects of our work. A brief overview of related approaches is provided below.

2.1 Anchor Graph

An effective method for constructing large-scale graph matrices is the anchor graph [21, 22] $\mathbf{A} = [a_1, a_2, \cdots, a_M] \in R^{M \times d}$ is used to select a subset of M samples from the original dataset using either k-means clustering or random sampling. Each anchor resembles the local geometric structure and serves as a landmark. The following steps are followed in order to construct a sparse affinity matrix between the anchors and the original data:

$$\mathbf{Z}_{i,j} = \begin{cases} \frac{K(x_i, a_j)}{\sum_{j' \in <i>} K(x_i, a_{j'})}, & j \in \langle i \rangle \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where $K(\cdot, \cdot)$ is a kernel function like the Gaussian kernel and $\langle i \rangle$ is an index set of k nearest anchors around x_i. After that, the full $N \times N$ affinity matrix $\mathbf{S}$ can be approximated by $\mathbf{S} = \bar{\mathbf{Z}}\bar{\mathbf{Z}}^T$, where $\mathbf{\Lambda}$ is a diagonal matrix with entries $\mathbf{\Lambda}_{i,i} = \sum_{j=1}^{N} \mathbf{Z}_{j,i}$.

The singular value matrix, left singular vector matrix, and right singular vector matrix, respectively, can be written as follows for the Singular Value Decomposition (SVD) of $\bar{\mathbf{Z}}$: $\bar{\mathbf{Z}} = \mathbf{P}\mathbf{\Sigma}\mathbf{Q}^T$. It is demonstrated that:

$$\begin{aligned} \mathbf{S} &= \bar{\mathbf{Z}}\bar{\mathbf{Z}}^T \\ &= \left(\mathbf{P}\mathbf{\Sigma}\mathbf{Q}^T\right)\left(\mathbf{P}\mathbf{\Sigma}\mathbf{Q}^T\right)^T \\ &= \mathbf{P}\mathbf{\Sigma}\mathbf{Q}^T\mathbf{Q}\mathbf{\Sigma}^T\mathbf{P}^T \\ &= \mathbf{P}\left(\mathbf{\Sigma}\mathbf{\Sigma}^T\right)\mathbf{P}^T. \end{aligned} \tag{2}$$

Since $\mathbf{\Sigma}$ is the singular value matrix of $\bar{\mathbf{Z}}$, $\mathbf{\Sigma}\mathbf{\Sigma}^T$ is the eigenvalue matrix of $\mathbf{S}$. Fast Spectral Clustering[23] has shown that the left singular vectors of $\bar{\mathbf{Z}}$ are the same as the eigenvectors of $\mathbf{S}$. As a result, we use SVD on $\bar{\mathbf{Z}}$ to obtain spectral embeddings $\mathbf{Q}$ instead of eigendecomposition on $\mathbf{S}$.

2.2 Sparse Subspace Clustering

Subspace clustering [15, 16, 24] minimizes reconstruction loss to obtain combination coefficients and represents each sample as a linear combination of other samples. After that, these coefficients are regarded as affinities between samples that correspond to one another. By making use of *sparse* coefficients and the sparse prior [25], sparse subspace clustering builds a similarity graph that is more resistant to data nuisances and performs better. Sparse subspace clustering solves the following issue when presented with data $\mathbf{X} \in \mathbb{R}^{N \times d}$ containing d features and N samples:

$$\min \|\mathbf{S}\mathbf{X} - \mathbf{X}\|_F^2 + \alpha \|\mathbf{S}\|_1 \text{ s.t. } diag(\mathbf{S}) = \mathbf{0}, \tag{3}$$

where $diag(\mathbf{S}) = \mathbf{0}$ prevents trivial solution, $\mathbf{S}$ is an $N \times N$ affinity matrix, $\alpha > 0$ is a balance parameter, and $\|\mathbf{S}\|_1$ is the l_1-norm regularizer. The l_1-norm regularizer on $\mathbf{S}$ is removed in order to reduce hyper-parameters, as in [26], and the equivalent sparsity constraints are added as follows:

$$\begin{aligned} &\min \|\mathbf{S}\mathbf{X} - \mathbf{X}\|_F^2 \\ &\text{s.t. } diag(\mathbf{S}) = \mathbf{0}, \mathbf{S}\mathbf{1} = \mathbf{1}, \mathbf{S} \geq 0, \end{aligned} \tag{4}$$

where the hyperparameter α is eliminated and $\mathbf{1}$ is an all-one and compatible vector.

3 Proposed Approach

Our Anchor-based Sparse Subspace Incomplete Multi-view Clustering is detailed in this section. The anchor-based affinities are broken down into multiple intra-view anchor-to-incomplete parts and an inter-view anchor-to-anchor part. After

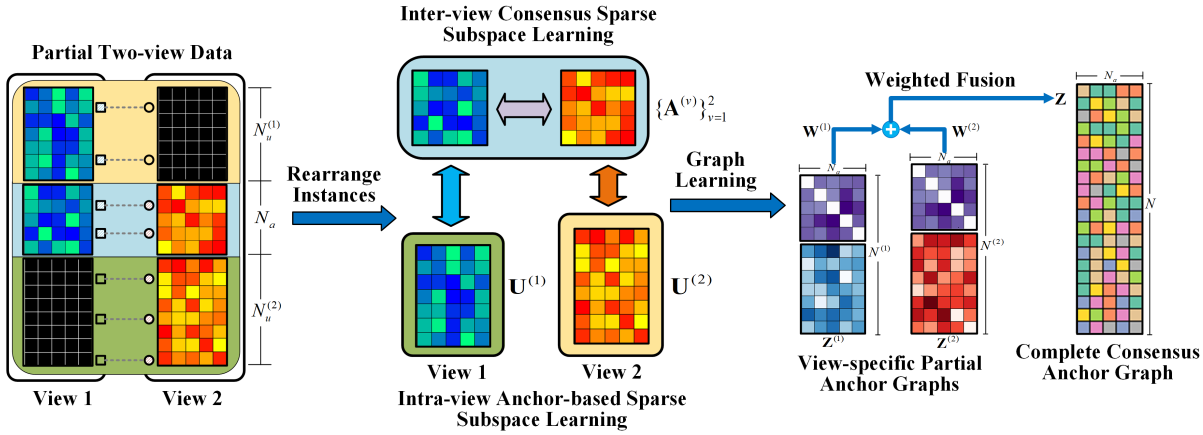


Fig. 1: Illustration of the proposed method: Anchor-based Sparse Subspace Incomplete Multi-view Clustering.

presenting two aspects of our model, we present a unified objective function for putting the proposed method into action.

3.1 Notation and Problem Definition

Scalars are represented by non-bold italic letters $(k, K, \cdots)$, except in a few specific cases. Vectors are represented by the bold lowercase letters $(x, \cdots)$, and matrices by the bold uppercase letters $(\mathbf{X}, \cdots)$. An appropriate-sized identity matrix is $\mathbf{I}$. $\mathbf{1}$ and $\mathbf{0}$ are, depending on the context, all-one or all-zero vectors of compatible size. The diagonal elements of a matrix are returned as vectors by calling $diag(\cdot)$.

Given a dataset with $\{\mathbf{X}^{(v)}, v = 1, 2, \cdots, V\}$, where $\mathbf{X}^{(v)} \in \mathbb{R}^{N \times d_v}$ is the dataset's v-th view and there are N instances, C clusters, and V views, For incomplete multiview data, we use [20] to denote fully-paired instances (or paired instances) as $\mathbf{A} = \{\mathbf{A}^{(v)}, v = 1, 2, \cdots, V\}$, and incompletely-paired instances (or incomplete instances) as $\mathbf{U} = \{\mathbf{U}^{(v)}, v = 1, 2, \cdots, V\}$. The number of paired instances (N_a) is the same for each view. $N^{(v)} = N_a + N_u^{(v)}, v = 1, 2, \cdots, V$ denotes the total number of instances presented for each view, with zeros representing the views that are missing from incomplete instances. Each instance is represented by a row in the feature matrix. The goal of incomplete multi-view clustering is to group all instances into C clusters, including incomplete and paired instances.

The missing samples are removed, and the original view matrix $\mathbf{X}^{(v)} \in \mathbb{R}^{N \times d_v}$ is changed to a new matrix $\bar{\mathbf{X}}^{(v)} = [\mathbf{A}^{(v)}; \mathbf{U}^{(v)}] \in \mathbb{R}^{N^{(v)} \times d_v}$, where $N^{(v)} < N$ is the number of samples presented in each view. This ensures that the paired instances are aligned rows. We define the weight matrices $\mathbf{W}^{(v)} \in \{0, 1\}^{N \times N^{(v)}}, v = 1, 2, \cdots, V$ to indicate the update as follows:

$$\mathbf{W}_{i,j}^{(v)} = \begin{cases} 1, & \text{if the } i\text{-th sample is the } j\text{-th} \\ & \text{presented sample in the } v\text{-th view;} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

By left-multiplying the corresponding $\mathbf{W}^{(v)}$, any representation learned from $\bar{\mathbf{X}}^{(v)}$ can be transformed back to the instance order of $\mathbf{X}^{(v)}$. This is possible when $\mathbf{X}^{(v)} = \mathbf{W}^{(v)} \bar{\mathbf{X}}^{(v)}$.

3.2 Inter-view Consensus Sparse Subspace Learning

With paired instances $\mathbf{A} = \{\mathbf{A}^{(v)}, v = 1, 2, \cdots, V\}$, a square affinity matrix $\mathbf{Z}_a^*$ [1] can be established to model the pairwise relationships among the N_a instances. Moreover, the row-wise alignment condition makes it convenient to apply any off-the-shelf multi-view learning method and

[1] Although the symbol $\mathbf{S}$ is more conventional for denoting square affinity matrices, we choose to use the symbol $\mathbf{Z}$ for terminology coherence since the anchor-to-anchor graph $\mathbf{Z}_a$ is a subset of a larger anchor graph.

capture inter-view semantic consistency from $\mathbf{A}$. However, adapting an existing multi-view learning method to incomplete multi-view data is non-trivial. Fortunately, the anchor-to-anchor affinity graph $\mathbf{Z}_a^* \in \mathbb{R}^{N_a \times N_a}$ is a subgraph of the view-specific full anchor graph $\mathbf{Z}^{(v)} \in \mathbb{R}^{N^{(v)} \times N_a}$ with no edges between $\mathbf{Z}_a^*$ and the rest of $\mathbf{Z}^{(v)}$, which means $\mathbf{Z}_a^*$ can be separately learned and then integrated into $\mathbf{Z}^{(v)}$ with a simple row-wise matrix concatenation. This fact makes it possible to directly learn a consensus sparse subspace model over the paired instances to obtain the anchor-to-anchor affinities $\mathbf{Z}_a^*$ with the following proposed model:

$$\begin{aligned} \min & \sum_{v=1}^{V} \left\| \mathbf{Z}_a^{(v)} \mathbf{A}^{(v)} - \mathbf{A}^{(v)} \right\|_F^2 + \lambda \left\| \mathbf{Z}_a^{(v)} - \mathbf{Z}_a^* \right\|_F^2 \\ \text{s.t. } & \mathbf{Z}_a^{(v)} \geq 0, \mathbf{Z}_a^{(v)} \mathbf{1} = \mathbf{1}, diag(\mathbf{Z}_a^{(v)}) = \mathbf{0} \\ & \mathbf{Z}_a^* \geq 0, \mathbf{Z}_a^* \mathbf{1} = \mathbf{1}, diag(\mathbf{Z}_a^*) = \mathbf{0} \end{aligned} \tag{6}$$

Remark 1 As aforementioned, the non-negative constraints $\mathbf{Z}_a^{(v)} \geq 0, \mathbf{Z}_a^* \geq 0$ and the row-normalized constraints $\mathbf{Z}_a^{(v)} \mathbf{1} = \mathbf{1}, \mathbf{Z}_a^* \mathbf{1} = \mathbf{1}$ are the equivalent to the sparsity-inducing l_1-norm regularizer, and this saves us a hyper-parameter. The diagonal constraints $diag(\mathbf{Z}_a^{(v)}) = \mathbf{0}, diag(\mathbf{Z}_a^*) = \mathbf{0}$ are necessary for preventing trivial solutions, i.e., $\mathbf{Z}_a^* = \mathbf{Z}_a^{(v)} = \mathbf{I}$. λ is a tunable parameter balancing the view-specific subspace learning term $\left\| \mathbf{Z}_a^{(v)} \mathbf{A}^{(v)} - \mathbf{A}^{(v)} \right\|_F^2$ and the consistency-enforcing term $\left\| \mathbf{Z}_a^{(v)} - \mathbf{Z}_a^* \right\|_F^2$.

Remark 2 In the past few decades, numerous effective multi-view subspace clustering techniques have been proposed. Equation 6 is not always the most sophisticated. However, its advantages include being empirically effective and conceptually straightforward.

3.3 Intra-view Anchor-based Sparse Subspace Learning

The core idea of anchor-based subspace clustering is to replace the traditional self-reconstruction term $\|\mathbf{S}\mathbf{X} - \mathbf{X}\|_F^2$ with the anchor-reconstruction term $\|\mathbf{Z}\mathbf{A} - \mathbf{X}\|_F^2$ and learn an $N \times M$ anchor graph $\mathbf{Z}$ that is more efficient than the $N \times N$ graph $\mathbf{S}$. It is effective to learn an anchor graph $\mathbf{Z}$ by treating anchors as a basis to represent other data points in semi-supervised learning [22] as well as large-scale multi-view clustering [21]. Further, the anchor-based sparse subspace model fits nicely in our framework, since the anchor-to-incomplete affinities can be learned with the same *sparse subspace learning* technique as the anchor-to-anchor affinities, yielding a unified model to obtain both kinds of affinities.

To this end, we propose to learn the intra-view anchor-to-incomplete affinities $\mathbf{Z}_u^{(v)} \in \mathbb{R}^{N_u^{(v)} \times N_a}$ by reconstructing incomplete instances $\mathbf{U}^{(v)} \in \mathbb{R}^{N_u^{(v)} \times d_v}$ with anchors (paired instances) $\mathbf{A}^{(v)} \in \mathbb{R}^{N_a \times d_v}$ as follows:

$$\begin{aligned} \min & \sum_{v=1}^{V} \left\| \mathbf{Z}_u^{(v)} \mathbf{A}^{(v)} - \mathbf{U}^{(v)} \right\|_F^2 \\ \text{s.t. } & \mathbf{Z}_u^{(v)} \geq 0, \mathbf{Z}_u^{(v)} \mathbf{1} = \mathbf{1} \end{aligned} \tag{7}$$

Remark 3 Diagonal constraints are unnecessary for Equation 7 since the set of anchors and the set of incomplete instances are disjoint and trivial solutions do not exist. This forces the model to discover the set of most relevant anchors for each incomplete instance.

Remark 4 Equation 7 differs from [21] in the following ways: 1) we eliminate the l_2-norm regularizer on $\mathbf{Z}_u^{(v)}$ because our primary focus is on the sparsity prior; 2) $\mathbf{Z}_u^{(v)}$ cannot be fused directly because [21] primarily works with large-scale complete multi-view data while we handle incomplete multi-view data. In addition, our model differs from [22], which is a single-view model with k-means anchor generation while treating all paired instances as anchors. [2]

3.4 Unified Objective Function

Combining the objective for inter-view consensus sparse subspace learning and intra-view anchor-based sparse subspace learning, we optimize the following objective function:

$$\begin{aligned} \min & \sum_{v=1}^{V} \left\| \mathbf{Z}_a^{(v)} \mathbf{A}^{(v)} - \mathbf{A}^{(v)} \right\|_F^2 + \lambda \left\| \mathbf{Z}_a^{(v)} - \mathbf{Z}_a^* \right\|_F^2 \\ & + \left\| \mathbf{Z}_u^{(v)} \mathbf{A}^{(v)} - \mathbf{U}^{(v)} \right\|_F^2 \\ \text{s.t. } & \mathbf{Z}_a^{(v)} \geq 0, \mathbf{Z}_a^{(v)} \mathbf{1} = \mathbf{1}, diag(\mathbf{Z}_a^{(v)}) = \mathbf{0} \end{aligned} \tag{8}$$

[2] We can also use k-means or random sampling to generate anchors, but for simplicity, we treat all the paired instances as anchors in this work.

$$\mathbf{Z}_a^* \geq 0, \mathbf{Z}_a^* \mathbf{1} = \mathbf{1}, diag(\mathbf{Z}_a^*) = \mathbf{0}$$
$$\mathbf{Z}_u^{(v)} \geq 0, \mathbf{Z}_u^{(v)} \mathbf{1} = \mathbf{1}$$

By resolving the Equation 8, we obtain the inter-view anchor-to-anchor affinity matrix $\mathbf{Z}_a^*$ and the intra-view anchor-to-incomplete affinity matrix $\mathbf{Z}_u^{(v)}, v = 1, 2, \cdots, V$. The view-specific anchor-to-instance affinity matrix $\mathbf{Z}^{(v)}$ is then obtained by concatenating $\mathbf{Z}_a$ and $\mathbf{Z}_u^{(v)}$. In the end, we rearrange and fuse these $\mathbf{Z}^{(v)}$s by equal-weighted average to obtain the consensus anchor graph $\mathbf{Z}$, on which we perform Fast Spectral Clustering to obtain the final clustering results [23].

3.5 Optimization

The objective function in Equation 8 is not convex over all variables $\mathbf{Z}_a, \{\mathbf{Z}_a^{(v)}, \mathbf{Z}_u^{(v)}\}_{v=1}^V$ and can be solved by Alternating Direction Method of Multipliers (ADMM). However, it can also be solved by gradient descent-based methods, which enjoy hardware acceleration offered by modern tensor libraries such as PyTorch and TensorFlow. Thus, we implement the proposed method in PyTorch to speed up experiments. We first transform Equation 8 into an unconstrained loss function and then minimize it with the Adam[27] optimizer.

Specifically, the diagonal constraints are implemented as a mask matrix $\mathbf{M} = (\mathbf{1} - \mathbf{I})$ that sets diagonal elements to zeros. The non-negative and row-normalized constraints are implemented as the softmax activation function. To solve Equation 8, The following loss function is minimized:

$$\begin{aligned} &\mathcal{L}\left(\mathbf{\Theta}_a^{(v)}, \mathbf{\Theta}_u^{(v)}, \mathbf{\Theta}_a^*\right) \\ &= \sum_{v=1}^{V} \left\| \sigma(\mathbf{M}\mathbf{\Theta}_a^{(v)})\mathbf{A}^{(v)} - \mathbf{A}^{(v)} \right\|_F^2 \\ &+ \left\| \sigma(\mathbf{\Theta}_u^{(v)})\mathbf{A}^{(v)} - \mathbf{U}^{(v)} \right\|_F^2 + \lambda \left\| \mathbf{\Theta}_a^{(v)} - \mathbf{\Theta}_a^* \right\|_F^2 \end{aligned} \tag{9}$$

where $\{\mathbf{\Theta}_a^{(v)}, \mathbf{\Theta}_u^{(v)}, \mathbf{\Theta}_a^*\}$ are the unconstrained learnable parameters, $\sigma(\cdot)$ is the softmax activation function, $\mathbf{M} \in \mathbb{R}^{N_a \times N_a}$ is the diagonal mask matrix. Algorithm 1 provides a summary of the entire procedure.

Algorithm 1 Anchor-based Sparse Subspace Incomplete Multi-view Clustering

Input: Incomplete multi-view data matrices $\bar{\mathbf{X}}^{(v)} = [\mathbf{A}^{(v)}; \mathbf{U}^{(v)}]$, weight matrices $\mathbf{W}^{(v)}$, $v = 1, 2, \cdots, V$, hyper-parameter λ, number of clusters C, learning rate α.

1: Initialize $\mathbf{\Theta}_a^{(v)} \in \mathbb{R}^{N_a \times N_a}$, $\mathbf{\Theta}_u^{(v)} \in \mathbb{R}^{N_u^{(v)} \times N_a}$ and $\mathbf{\Theta}_a^* \in \mathbb{R}^{N_a \times N_a}$, $v = 1, 2, \cdots, V$.
2: Compute $\{\mathbf{\Theta}_a^{(v)}, \mathbf{\Theta}_u^{(v)}, \mathbf{\Theta}_a^*\}$ by optimizing loss function 9 with learning rate α.
3: Compute $\mathbf{Z}_a^* = \sigma(\mathbf{M}\mathbf{\Theta}_a^*)$.
4: Compute $\mathbf{Z}_u^{(v)} = \sigma(\mathbf{\Theta}_u^{(v)}), v = 1, 2, \cdots, V$.
5: Compute $\mathbf{Z}^{(v)} = [\mathbf{Z}_a^*; \mathbf{Z}_u^{(v)}], v = 1, 2, \cdots, V$.
6: Compute $\mathbf{Z} = \frac{1}{V} \sum_{v=1}^{V} \mathbf{W}^{(v)} \mathbf{Z}^{(v)}$.
7: Perform Fast Spectral Clustering on $\mathbf{Z}$ to get clustering results.

Output: Consensus anchor graph $\mathbf{Z}$ and clustering results.

3.6 Complexity Analysis

The two main stages of our proposed approach's computational complexity are now the focus of our investigation. The following is an examination of the associated computational complexities.

The construction of $\mathbf{Z}_a^*$ and $\{\mathbf{Z}_v^{(v)}\}_{v=1}^V$ requires $\mathcal{O}(M^3 d)$ and $\mathcal{O}\left((N-M)M^2 d\right)$, respectively, in the first stage of anchor-based subspace learning. In this case, N represents the total number of samples, M represents the number of anchors, and $d = \sum_{v=1}^{V} d_v$ are sparse matrices, the fusion of $\mathbf{Z}^{(v)}$ to obtain $\mathbf{Z}$ requires only $\mathcal{O}(NMV)$, whereas the construction of $\mathbf{W}^{(v)}$ requires $\mathcal{O}(NMV)$. As a result, the first stage's complexity is $\mathcal{O}(NM^2 d)$.

The complexity of calculating spectral embeddings $\mathbf{Q}$ by performing SVD on $\bar{\mathbf{Z}} = \mathbf{Z}\mathbf{\Lambda}^{-1/2} \in \mathbb{R}^{N \times M}$ in the second stage of spectral clustering on $\mathbf{Z}$ is $\mathcal{O}\left(\min\{NM^2, N^2 M\}\right)$. Since we only require the C largest singular values, the complexity can be reduced to $\mathcal{O}(NC^2)$ [23]. Lastly, in order to obtain C clusters, we require $\mathcal{O}(tNC^2)$ to perform k-means clustering for t iterations on $\mathbf{Q} \in \mathbb{R}^{N \times C}$. As a result, the second stage's complexity is $\mathcal{O}(tNC^2)$.

As a result, the proposed method as a whole has a computational complexity of $\mathcal{O}(NM^2 d + tNC^2)$, which is not strictly linear in N because we currently treat all paired instances as anchors.

However, the selection of a predetermined number of anchors can easily lead to linear complexity; this is a topic for additional research.

4 Experiments

4.1 Datasets

On four actual datasets, we assess the proposed method's capacity for clustering. A brief introduction is provided below, and important dataset statistics are listed in Table 1.

Table 1: Description of the used datasets

Dataset	#Samples	#Views	#Clusters	#Features
Digits	2000	2	10	76/216
ORL	400	3	40	4096/3304/6750
COIL-20	1440	3	20	1024/944/576
USPS-MNIST	10000	2	10	784/256

Digits [3] is a collection of 2000 handwritten numbers (from 0 to 9) taken from Dutch utility maps. Six different kinds of features taken from a sample's binary image are used to represent it. There are 200 samples in each class. The two views used in this work are profile correlations and Fourier coefficients, two types of features.
ORL [4] is made up of 400 images of the faces of forty people of different genders. There are ten images of each person's face taken from various angles. To represent these face images, we extract three kinds of features from each sample: gray intensity, local binary pattern, and Gabor.
COIL-20 [5] The image dataset COIL-20 has 20 clusters across 1440 samples. There are three perspectives available for each image: intensity has 1024 features, a 3304-feature local binary pattern, and 6750-feature Gabor.
USPS-MNIST is a ten thousand-sample, large-scale, two-view dataset. It is a combination of two well-known handwritten datasets: USPS[28] and MNIST[29], respectively. In ten classes, USPS has 9298 images with a size of 16×16, while MNIST has 70000 images with a size of 28×28. Two views can be used to describe the same class of digits in two datasets. From each dataset, we randomly select 1000 images for each digit class to create a two-view dataset with 10000 samples.

[3] http://archive.ics.uci.edu/ml/datasets.html.
[4] http://cam-orl.co.uk/facedatabase.html/.
[5] https://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php.

4.2 Comparison Methods

The following cutting-edge approaches are contrasted with the proposed method.

BSV (Best Single View) We are unable to directly apply k-means clustering to the original data because there are missing samples in each view. Mean-filling, or filling in all missing instances with the average values for each view, is the first step we take following [30]. The best outcome is then reported after k-means clustering is carried out on each view.
Concat Our second baseline uses feature concatenation as a straightforward method for utilizing multi-view data. Similar to BSV, we combine features from all views into one by first filling in all the missing data with the average features for each view.
SC-A (Spectral Clustering with Average Graph) Before carrying out mean-filling, we first employ a Gaussian kernel to generate an instance-to-instance affinity graph for each view. The average of these graphs is used to fuse them, and then spectral clustering is applied to the fused graph.
SC-C (Spectral Clustering with Concatenated Features) Following preprocessing, we combine the features of every instance from multiple perspectives into a single feature vector. The concatenated features are then used for spectral clustering and instance-to-instance affinity graph computation.
DAIMC [6] applies k-means to the learned representation to obtain clustering results after using the weighted semi-NMF to learn a common representation for all views.
APMC [12] constructs a unified anchor graph by employing the paired samples as anchors, then applies spectral clustering to it to obtain clustering results.
UEAF [7] Based on missing view inferring and reverse graph regularization, UEAF acquires a single representation for each view. The results of the clustering are then obtained by applying k-means to the learned representation.

GIMC-FLSD [9] acquires a graph-regularized matrix factorization model that produces a unified discriminative representation and adaptively determines the significance of various viewpoints. *ANIMC* [8] To deal with noisy and incomplete multi-view data, ANIMC learns an auto-weighted regularized semi-NMF model and creates a single representation.

4.3 Settings

We use the following settings for our experiments.

Data construction: We follow [20] to simulate incomplete multi-view data because the datasets used in our experiments were originally complete multi-view datasets. In particular, we remove some views from each of the N_u instances and select them as incomplete samples at random from a total of N instances.N_u/N refers to the Partial Example Rate (PER), which has a stride of 0.1 and ranges from 0.1 to 0.9.
Data preprocessing: For the proposed method, we standardize each view $\{\bar{\mathbf{X}}^{(v)}\}_{v=1}^{V}$ so that the mean and standard deviation of each feature dimension are both zero. We standardize BSV, Concat, SC-A, and SC-C in the same way that we standardize the proposed method. We use their preprocessing pipelines for the other baselines.
Evaluation metrics: Accuracy (ACC), Normalized Mutual Information (NMI), Purity (PUR), and F1-score (F1) are the four clustering metrics we use to conduct a comprehensive analysis of our proposed method. To determine the average and standard deviation, each experiment is repeated five times.
Experimental Platform: The GPU-accelerated tensor library PyTorch is utilized in the implementation of our approach. [6] PyTorch is used to implement the BSV, Concat, SC-A, and SC-C baseline methods as well. We use the public packages released by the corresponding authors for other baseline methods. Centos 7.6 and NVIDIA GeForce graphics processing units are used in all experiments. We set the learning rate to 0.1 for all experiments because we empirically find that it only affects the speed of convergence.

[6] https://pytorch.org/

4.4 Results and Analysis

Figure 2 and Table 2 demonstrate the outcomes of four distinct metrics on one large-scale dataset and three middle-scale datasets with distinct PERs, respectively. The following observations and discussion are based on these experimental results.

- The performances of all algorithms decrease as PER rises. Our method outperforms DAIMC, UEAF, APMC, and other comparison methods more and more. When the dataset is finished, all methods perform fairly well. The fact that incomplete multi-view clustering is more difficult than complete multi-view clustering is demonstrated by this.
- The following method groups {DAIMC, UEAF, ANIMC, GIMC-FLSD}, {Concat, BSV}, and {SC-C, SC-A, APMC} share similar trends for metric values with varying PER: The mechanisms of their models can explain this. A model of doubly-aligned matrix factorization (MF) is the foundation of DAIMC. By combining an embedding alignment and missing view inferring with a reverse graph-regularized model, UEAF improves DAIMC. Both ANIMC and GIMC-FLSD go into detail about some MF model variations. As a result, these MF-based methods exhibit comparable trends. Concat and BSV apply k-means clustering to the raw features directly. APMC, SC-C, and SC-A all use spectral clustering as their foundation, but their primary distinction lies in how affinity graphs are constructed.
- The fact that the proposed method performs significantly better than two straightforward baselines, BSV and Concat, further demonstrates its effectiveness. Instead of simple anchor-based spectral clustering, the performance boost comes from the anchor-based affinity graph construction with sparse subspace learning. In the meantime, our approach achieves a significant performance advantage over APMC, which is also based on anchor graphs. This demonstrates that the graph learned through sparse subspace learning is more efficient than a hand-crafted Gaussian kernel.
- Our proposed approach consistently outperforms other baseline methods in all PER cases on the massive USPS-MNIST dataset. This

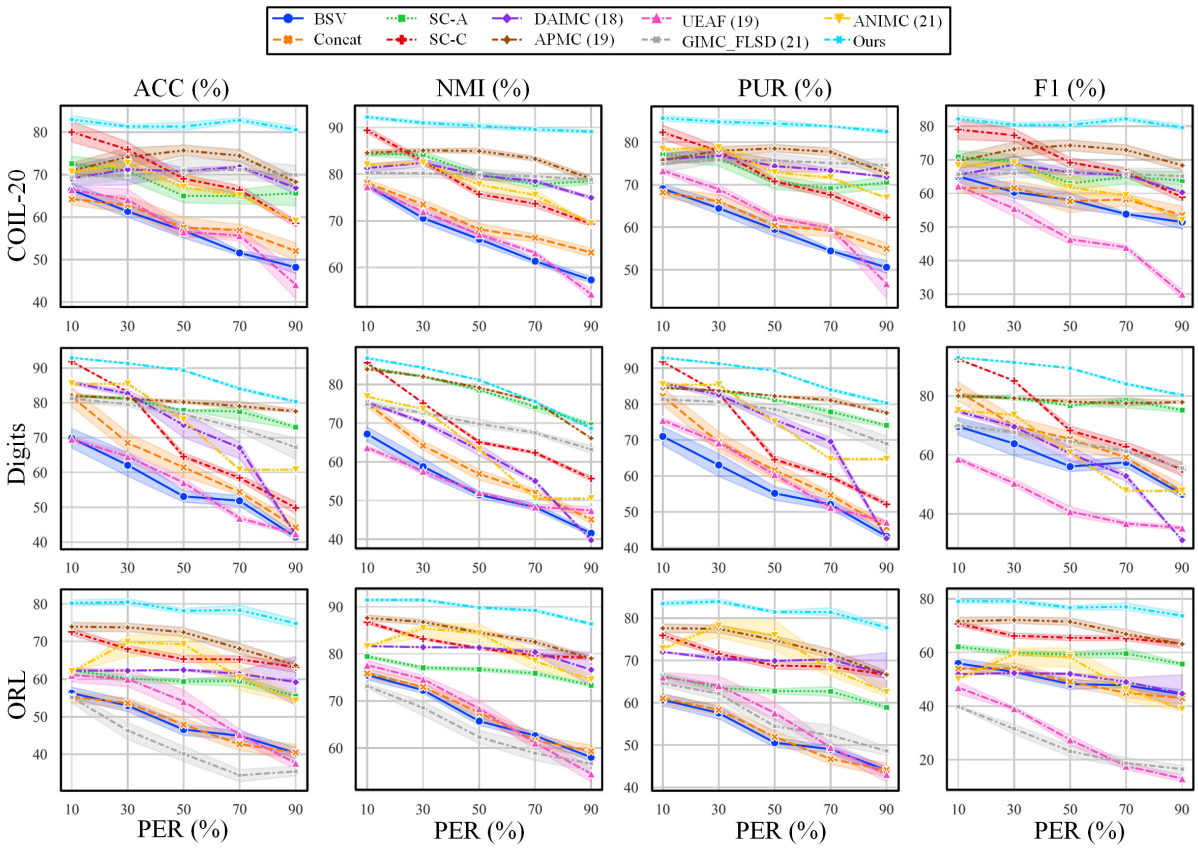


Fig. 2: Clustering results of different methods on three real-world datasets with different PERs.

demonstrates the method's promising performance when applied to large datasets. The methods not depicted in Table 2 are skipped because it takes too long or runs out of memory.

The following is an analysis of the proposed method's superiority. The alignment information and consistent affinities between the paired samples are effectively captured when the consensus sparse subspace model is applied. Under IMVC conditions, this takes advantage of the versatility of multi-view data and boosts performance when PER decreases as data become more complete. In the meantime, the complete anchor-based affinity graph is recovered from incomplete data using paired samples as anchors. This naturally closes the gap between paired and unpaired samples and boosts performance as PER rises. An empirically powerful model that is more resistant to incomplete multi-view data is produced when anchor-to-anchor and anchor-to-incomplete affinities are combined with a sparse prior.

4.5 Visualization

We visually compare the spectral embeddings or consensus features learned by various methods—namely, BSV, Concat, SC-A, SC-C, APMC, and Ours with t-SNE [31] to further demonstrate the reasons why the clustering results of the proposed method are superior to those of other baselines. To show the block-diagonal structure that was discovered by a variety of methods, we also plot the affinity matrices of the consensus features or spectral embeddings.

On the Digits dataset, we visualize the features or spectral embeddings obtained by various methods using t-SNE with a PER of 50% in Figure 3. When samples from the same class naturally gather together and the gap between different groups is obvious, we can see that the spectral

Table 2: On the USPS-MNIST dataset, the averages and standard deviations of four distinct metrics for various methods with various PER (%). It is important to note that some baselines are not included because they are unable to produce results on such a large dataset.

PER (%)	Method	ACC (%)	NMI (%)	PUR (%)	F1 (%)
10	BSV	46.87 ± 0.47	42.64 ± 1.24	49.45 ± 0.23	46.33 ± 0.63
	Concat	73.85 ± 0.17	68.69 ± 0.15	73.87 ± 0.16	73.08 ± 0.35
	SC-A	97.33 ± 0.01	95.25 ± 0.02	98.33 ± 0.01	98.33 ± 0.01
	SC-C	43.41 ± 7.94	47.37 ± 4.22	43.96 ± 7.14	33.79 ± 9.80
	APMC (19)	98.34 ± 0.00	95.26 ± 0.00	98.34 ± 0.00	98.34 ± 0.00
	UEAF (19)	87.10 ± 0.34	75.08 ± 0.51	87.10 ± 0.34	76.48 ± 0.54
	Ours	**98.94 ± 0.39**	**96.81 ± 0.70**	**98.94 ± 0.39**	**98.93 ± 0.40**
30	BSV	43.83 ± 0.99	37.41 ± 0.12	46.03 ± 1.31	43.77 ± 1.17
	Concat	59.68 ± 2.61	53.76 ± 1.53	60.85 ± 1.91	59.25 ± 3.12
	SC-A	97.33 ± 0.01	92.90 ± 0.01	97.33 ± 0.01	97.33 ± 0.01
	SC-C	36.25 ± 7.52	38.13 ± 10.53	36.71 ± 7.33	27.04 ± 7.31
	APMC (19)	97.12 ± 0.00	92.39 ± 0.00	97.12 ± 0.00	97.12 ± 0.00
	UEAF (19)	70.89 ± 0.45	60.20 ± 0.29	71.47 ± 0.83	57.81 ± 0.21
	Ours	**97.36 ± 0.50**	**93.28 ± 0.89**	**97.36 ± 0.50**	**97.35 ± 0.51**
50	BSV	38.56 ± 0.13	32.76 ± 0.29	40.27 ± 0.35	39.52 ± 0.42
	Concat	47.10 ± 4.00	40.94 ± 2.81	48.02 ± 3.81	47.98 ± 3.52
	SC-A	95.17 ± 0.01	88.06 ± 0.01	95.17 ± 0.01	95.16 ± 0.01
	SC-C	33.57 ± 3.76	32.74 ± 5.71	33.61 ± 3.77	24.10 ± 3.54
	APMC (19)	94.37 ± 0.00	86.73 ± 0.00	94.37 ± 0.00	94.35 ± 0.00
	UEAF (19)	56.43 ± 7.96	46.79 ± 5.06	58.37 ± 9.18	41.19 ± 7.32
	Ours	**95.54 ± 0.08**	**89.10 ± 0.14**	**95.54 ± 0.08**	**95.54 ± 0.08**
70	BSV	34.73 ± 0.60	28.69 ± 0.56	35.06 ± 0.82	36.94 ± 0.62
	Concat	37.64 ± 1.15	30.49 ± 1.19	38.92 ± 1.46	38.74 ± 0.71
	SC-A	76.85 ± 0.11	74.65 ± 0.02	76.87 ± 0.10	77.04 ± 0.12
	SC-C	34.24 ± 5.25	35.51 ± 6.97	35.45 ± 6.38	25.96 ± 6.35
	APMC (19)	91.45 ± 0.00	81.36 ± 0.00	91.45 ± 0.00	91.41 ± 0.00
	UEAF (19)	45.38 ± 0.44	39.36 ± 0.96	48.56 ± 2.15	33.41 ± 1.05
	Ours	**94.00 ± 0.15**	**85.78 ± 0.31**	**94.00 ± 0.15**	**94.00 ± 0.15**
90	BSV	30.65 ± 0.46	25.22 ± 0.77	31.53 ± 0.86	32.89 ± 0.45
	Concat	31.87 ± 1.15	26.67 ± 1.44	33.84 ± 1.13	32.53 ± 1.60
	SC-A	67.77 ± 4.11	67.04 ± 1.59	68.80 ± 2.15	68.57 ± 4.46
	SC-C	27.14 ± 3.05	24.15 ± 4.26	27.75 ± 3.25	20.62 ± 3.06
	APMC (19)	83.89 ± 0.01	69.25 ± 0.01	83.89 ± 0.01	83.77 ± 0.01
	UEAF (19)	37.61 ± 0.26	35.58 ± 1.91	43.67 ± 1.74	29.56 ± 1.31
	Ours	**87.85 ± 0.09**	**75.84 ± 0.12**	**87.85 ± 0.09**	**87.81 ± 0.09**

embeddings obtained by the proposed method, i.e., (d), show the best separability for different classes. This demonstrates how well our approach to incomplete multi-view clustering works.

In addition, we visually compare the affinity matrices of the consensus features or spectral embeddings learned from the ORL dataset to confirm the proposed method's efficacy in Figure 4. Our affinity matrix has a more clear block diagonal structure than APMC [12], which is the comparison algorithm with the best clustering performance.

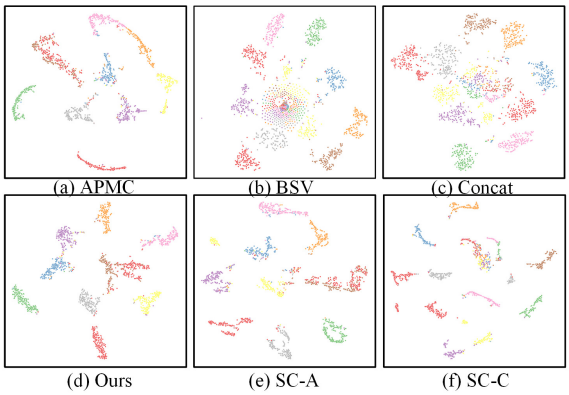


Fig. 3: t-SNE visualization of (a) APMC spectral embeddings, (b) best view features, (c) concatenation features, (d) the proposed method's spectral embedding, (e) SC-A and SC-C spectral embeddings, respectively, on a Digits dataset with a PER of 50%.

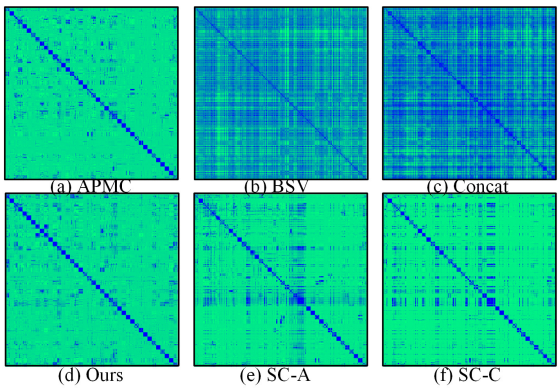
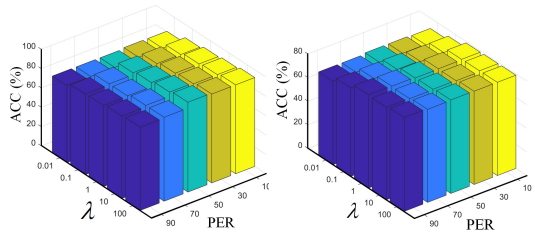


Fig. 4: On an ORL dataset with a PER of 50% percent, an affinity matrix depicts (a) the spectral embeddings of APMC, (b) the features of the best view, (c) the concatenation features, (d) the spectral embedding of the proposed method, (e) the spectral embeddings of SC-A, and (f) the spectral embeddings of SC-C.

4.6 Parameter Analysis

In terms of parameter analysis, our proposed method only requires the setting of one parameter λ to fine-tune its clustering performance. We do this by varying λ across a range of $\{0.01, 0.1, 1, 10, 100\}$ on the Digits and ORL dataset, and similar patterns can be observed across other datasets. As depicted in Figure 5, the proposed method is not affected by λ over a broad range.

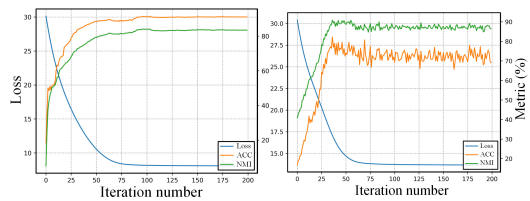


(a) Digits (b) ORL

Fig. 5: ACC (%) of the proposed method v.s. parameter λ and PER on (a) Digits and (b) ORL datasets.

4.7 Convergence Analysis

On Digits and ORL datasets with a PER of 50%, Figure 6 displays two metrics (ACCs and NMIs) and loss versus the number of optimization iterations. We can see from this figure that ACC and NMI rise after a few iterations before stabilizing. After some iterations, the loss also tends to stay the same. This suggests that our gradient descent-based optimization method can converge to a local minimum that is relatively sound after only a few dozen of iterations.



(a) Digits (b) ORL

Fig. 6: ACC (%), NMI (%) and loss v.s. iteration number on (a) Digits and (b) ORL datasets with a PER of 50%.

5 Conclusion

For incomplete multi-view clustering, we present a novel Anchor-based Sparse Subspace Incomplete

Multi-view Clustering approach. Learning inter-view anchor-to-anchor and intra-view anchor-to-incomplete affinities within a conceptually straightforward and empirically potent sparse subspace framework forms the core of our approach. To accomplish this, we use an anchor-based sparse subspace model to capture intra-view anchor-to-incomplete relationships and then fuse them into a complete consensus anchor graph. Additionally, we employ a consensus sparse subspace model to learn the inter-view anchor-to-anchor relationships. Our approach has the capability of processing large-scale datasets and naturally bridges the paired and incomplete instances. Lastly, the results of the experiment demonstrate that the proposed method is superior to other outstanding methods.

To model anchor-to-anchor relationships, more advanced multi-view subspace methods [16, 24] will be interesting for future research. To achieve linear complexity, a more effective anchor selection scheme will also be useful. Finally, it would be beneficial to incorporate the spectral clustering objective into our model so that we can directly generate clustering results, just like [24].

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China under Grant 62071157, and in part by the Natural Science Foundation of Heilongjiang Province under Grant YQ2019F011.

References

[1] Lahat, D., Adali, T., Jutten, C.: Multimodal data fusion: An overview of methods, challenges, and prospects. Proc. IEEE **103**(9), 1449–1477 (2015). https://doi.org/10.1109/JPROC.2015.2460697

[2] Zhao, J., Xie, X., Xu, X., Sun, S.: Multi-view learning overview: Recent progress and new challenges. Inf. Fusion **38**, 43–54 (2017). https://doi.org/10.1016/j.inffus.2017.02.007

[3] Chao, G., Sun, S., Bi, J.: A Survey on Multiview Clustering. IEEE Transactions on Artificial Intelligence **2**(2), 146–168 (2021). https://doi.org/10.1109/tai.2021.3065894

[4] Yang, Y., Wang, H.: Multi-view clustering: A survey. Big Data Mining and Analytics **1**(2), 83–107 (2018). https://doi.org/10.26599/BDMA.2018.9020003

[5] Li, S., Jiang, Y., Zhou, Z.: Partial multi-view clustering. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada, pp. 1968–1974 (2014). http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8241

[6] Hu, M., Chen, S.: Doubly aligned incomplete multi-view clustering. IJCAI International Joint Conference on Artificial Intelligence **2018-July**, 2262–2268 (2018). https://doi.org/10.24963/ijcai.2018/313

[7] Wen, J., Zhang, Z., Xu, Y., Zhang, B., Fei, L., Liu, H.: Unified embedding alignment with missing views inferring for incomplete multi-view clustering. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI, Honolulu, Hawaii, USA, pp. 5393–5400 (2019). https://doi.org/10.1609/aaai.v33i01.33015393

[8] Fang, X., Hu, Y., Zhou, P., Wu, D.: Animc: A soft approach for autoweighted noisy and incomplete multiview clustering. IEEE Transactions on Artificial Intelligence **3**(2), 192–206 (2022). https://doi.org/10.1109/TAI.2021.3116546

[9] Wen, J., Zhang, Z., Zhang, Z., Fei, L., Wang, M.: Generalized Incomplete Multiview Clustering With Flexible Locality Structure Diffusion. IEEE Transactions on Cybernetics **51**(1), 101–114 (2021). https://doi.org/10.1109/TCYB.2020.2987164

[10] Wen, J., Yan, K., Zhang, Z., Xu, Y., Wang, J., Fei, L., Zhang, B.: Adaptive Graph Completion Based Incomplete Multi-View Clustering. IEEE Transactions on Multimedia **23**, 2493–2504 (2021). https://doi.org/10.1109/TMM.2020.3013408

[11] Wen, J., Xu, Y., Liu, H.: Incomplete Multiview Spectral Clustering With Adaptive Graph Learning. IEEE Transactions on Cybernetics **50**(4), 1418–1429 (2020). https:

//doi.org/10.1109/TCYB.2018.2884715

[12] Guo, J., Ye, J.: Anchors bring ease: An embarrassingly simple approach to partial multi-view clustering. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 118–125 (2019). https://doi.org/10.1609/aaai.v33i01.3301118

[13] Xu, N., Guo, Y., Zheng, X., Wang, Q., Luo, X.: Partial multi-view subspace clustering. In: 2018 ACM Multimedia Conference on Multimedia Conference, MM 2018, Seoul, Republic of Korea, October 22-26, 2018, pp. 1794–1801 (2018). https://doi.org/10.1145/3240508.3240679

[14] Fang, X., Hu, Y., Zhou, P., Wu, D.O.: V^3h: View variation and view heredity for incomplete multiview clustering. IEEE Transactions on Artificial Intelligence **1**(3), 233–247 (2021). https://doi.org/10.1109/tai.2021.3052425

[15] Elhamifar, E., Vidal, R.: Sparse Subspace Clustering: Algorithm, Theory, and Applications. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(11), 2765–2781 (2013). https://doi.org/10.1109/TPAMI.2013.57

[16] Brbić, M., Kopriva, I.: Multi-view low-rank sparse subspace clustering. Pattern Recognition **73**, 247–258 (2018). https://doi.org/10.1016/j.patcog.2017.08.024

[17] Wen, J., Zhang, Z., Zhang, Z., Wu, Z., Fei, L., Xu, Y., Zhang, B.: DIMC-net: Deep Incomplete Multi-view Clustering Network. In: Proceedings of the 28th ACM International Conference on Multimedia, Seattle WA USA, pp. 3753–3761 (2020). https://doi.org/10.1145/3394171.3413807

[18] Wang, Q., Lian, H., Sun, G., Gao, Q., Jiao, L.: iCmSC: Incomplete Cross-Modal Subspace Clustering. IEEE Transactions on Image Processing **30**, 305–317 (2021). https://doi.org/10.1109/TIP.2020.3036717

[19] Lin, Y., Gou, Y., Liu, Z., Li, B., Lv, J., Peng, X.: COMPLETER: incomplete multi-view clustering via contrastive prediction. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, Virtual, June 19-25, 2021, pp. 11174–11183 (2021). https://doi.org/10.1109/CVPR46437.2021.01102

[20] Xu, C., Guan, Z., Zhao, W., Wu, H., Niu, Y., Ling, B.: Adversarial Incomplete Multi-view Clustering. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China, pp. 3933–3939 (2019). https://doi.org/10.24963/ijcai.2019/546

[21] Kang, Z., Zhou, W., Zhao, Z., Shao, J., Han, M., Xu, Z.: Large-Scale Multi-View Subspace Clustering in Linear Time. Proceedings of the AAAI Conference on Artificial Intelligence **34**(04), 4412–4419 (2020). https://doi.org/10.1609/aaai.v34i04.5867

[22] Liu, W., He, J., Chang, S.: Large graph construction for scalable semi-supervised learning. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel, pp. 679–686 (2010). https://icml.cc/Conferences/2010/papers/16.pdf

[23] Wang, R., Nie, F., Yu, W.: Fast spectral clustering with anchor graph for large hyperspectral images. IEEE Geosci. Remote. Sens. Lett. **14**(11), 2003–2007 (2017). https://doi.org/10.1109/LGRS.2017.2746625

[24] Gao, H., Nie, F., Li, X., Huang, H.: Multi-view subspace clustering. In: 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015, pp. 4238–4246 (2015). https://doi.org/10.1109/ICCV.2015.482

[25] Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological) **58**(1), 267–288 (1996). https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

[26] Wang, H., Yang, Y., Liu, B.: GMC: Graph-Based Multi-View Clustering. IEEE Transactions on Knowledge and Data Engineering

32(6), 1116–1129 (2020). https://doi.org/10.1109/TKDE.2019.2903810

[27] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015). http://arxiv.org/abs/1412.6980

[28] Hull, J.J.: A database for handwritten text recognition research. IEEE Trans. Pattern Anal. Mach. Intell. **16**(5), 550–554 (1994). https://doi.org/10.1109/34.291440

[29] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791

[30] Shao, W., He, L., Yu, P.S.: Multiple incomplete views clustering via weighted nonnegative matrix factorization with l_{21} regularization. In: Machine Learning and Knowledge Discovery in Databases, Cham, pp. 318–334 (2015). https://doi.org/10.1007/978-3-319-23528-8_20

[31] van der Maaten, L., Hinton, G.E.: Visualizing data using t-SNE. Journal of Machine Learning Research **9**, 2579–2605 (2008). https://doi.org/10.1.1.457.7213