A Survey on Multiview Clustering

Guoqing Chao, Shiliang Sun, *Member, IEEE*, and Jinbo Bi, *Member, IEEE*

***Abstract*—Clustering is a machine learning paradigm of dividing sample subjects into a number of groups such that subjects in the same groups are more similar to those in other groups. With advances in information acquisition technologies, samples can frequently be viewed from different angles or in different modalities, generating multiview data. Multiview clustering (MVC), that clusters subjects into subgroups using multiview data, has attracted more and more attentions. Although MVC methods have been developed rapidly, there has not been enough survey to summarize and analyze the current progress. Therefore, we propose a novel taxonomy of the MVC approaches. Similar to other machine learning methods, we categorize them into generative and discriminative classes. In the discriminative class, based on the way of view integration, we split it further into five groups—common eigenvector matrix, common coefficient matrix, common indicator matrix, direct combination, and combination after projection. Furthermore, we relate MVC to other topics: multiview representation, ensemble clustering, multitask clustering, multiview supervised, and semisupervised learning. Several representative real-world applications are elaborated for practitioners. Some benchmark multiview datasets are introduced and representative MVC algorithms from each group are empirically evaluated to analyze how they perform on benchmark datasets. To promote future development of MVC approaches, we point out several open problems that may require further investigation and thorough examination.**

***Impact Statement*—Multiview clustering has gained the success in a variety of applications in the past decade. In order to obtain a comprehensive picture of the MVC development, we provide a new categorization of existing MVC methods and introduce the representative algorithms in each category. At last, we point out open problems that are worth investigating to advance the MVC study. More promising MVC methods to solve these open problems may appear following this review paper from which a large number of applications can benefit.**

***Index Terms*—Canonical correlation analysis (CCA), clustering, data mining, k-means, machine learning, multiview learning, nonnegative matrix factorization (NMF), spectral clustering, subspace clustering, survey.**

I. Introduction

CLUSTERING [1] is a paradigm to divide the subjects into a number of groups such that subjects in the same groups are more similar to other subjects in the same group and dissimilar to the subjects in other groups. It is a fundamental task in machine learning, pattern recognition, and data mining fields and has widespread applications. Once subgroups are obtained by clustering methods, many subsequent analytic tasks can be conducted to achieve different ultimate goals. Traditional methods cluster subjects on the basis of only a single set of features or a single information window of the subjects. When multiple sets of features are available for each individual subject, how these views can be integrated to help identify essential grouping structure is a problem of our concern in this article, which is often referred to as multiview clustering (MVC). A good example to understand the importance of MVC, or multiview learning is "the blind men and the elephant" story where each blind man (a single view of the subject) may not acquire the true picture of the subject [2], thus only collecting multiview data can recover the whole picture of the subject.

Multiview data are very common in real-world applications in the big data era. For instance, a web page can be described by the words appearing on the web page itself and the words underlying the links pointing to the web page from other pages in nature. In multimedia content understanding, multimedia segments can be simultaneously described by their video signals from visual camera and audio signals from voice recorders. The existence of such multiview data raised the interest of multiview learning [3]–[5], which has been extensively studied in the semisupervised learning setting. For unsupervised learning, particularly, MVC, single view-based clustering methods cannot make an effective use of the multiview information in various problems. For instance, an MVC problem may require to identify clusters of subjects that differ in each of the data views. In this case, concatenating features from the different views into a single union followed by a single-view clustering method may not serve the purpose. It has no mechanism to guarantee that the resultant clusters differ in all of the views because the grouping may be biased toward a view (or views) that yields a dominantly large number of features in the feature union. MVC has thus attracted more and more attention in the past two decades, which makes it necessary and beneficial to summarize the state of the art and delineate open problems to guide future advancement.

At first, we give the definition of MVC. MVC is a machine learning paradigm to classify similar subjects into the same

Manuscript received September 11, 2020; revised December 20, 2020, February 2, 2021, and March 4, 2021; accepted March 5, 2021. Date of publication April 5, 2021; date of current version August 20, 2021. This work was supported in part by the National Institutes of Health (NIH) under Grant K02DA043063 and in part by the National Science Foundation (NSF) under Grant IIS-1718738. The work of Jinbo Bi was also supported by NIH under Grants R01DA051922 and R01MH119678. This paper was recommended for publication by Associate Editor C. L. Philip Chen upon evaluation of the reviewers' comments. *(Corresponding author: Jinbo Bi.)*

Guoqing Chao is with the School of Computer Science and Technology, Harbin Institute of Technology, Weihai 264209, China (e-mail: guoqingchao10@gmail.com).

Shiliang Sun is with the School of Computer Science and Technology, East China Normal University, Shanghai 200062, China (e-mail: slsun@cs.ecnu.edu.cn).

Jinbo Bi is with the Department of Computer Science, University of Connecticut, Storrs, CT 06269 USA (e-mail: jinbo.bi@uconn.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TAI.2021.3065894.

Digital Object Identifier 10.1109/TAI.2021.3065894

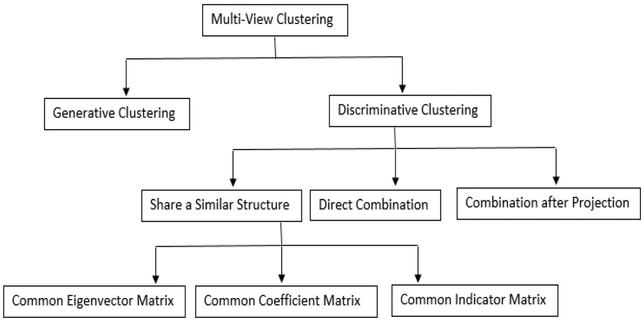

Multi-View Clustering
Generative Clustering
Discriminative Clustering
Share a Similar Structure
Direct Combination
Combination after Projection
Common Eigenvector Matrix
Common Coefficient Matrix
Common Indicator Matrix

Fig. 1. Taxonomy of MVC methods.

group and dissimilar subjects into different groups by combining the available multiview feature information, and to search for the consistent clusterings across different views. Similar to the categorization of clustering algorithms in [1], we divide the existing MVC methods into two categories: generative (or model-based) approaches and discriminative (or similarity-based) approaches. Generative approaches try to learn the fundamental distribution of the data and use generative models to represent the data with each model representing one cluster. Discriminative approaches directly optimize an objective function that involves pairwise similarities to minimize the average similarity within clusters and to maximize the average similarity between clusters. In discriminative clustering family, there are mainly three strategies to combine multiple views—assuming that all views share a similar structure, direct combination of the views, and combination after projection of each view. Due to the different similar structures shared, we further split those MVC methods based on the first strategy into three groups: 1) common eigenvector matrix based (mainly multiview spectral clustering); 2) common coefficient matrix based (mainly multiview subspace clustering); 3) common indicator matrix-based [mainly multiview nonnegative matrix factorization (NMF) clustering]. The complete taxonomy is shown in Fig. 1.

Similarly motivated by the multiview real applications as MVC, multiview representation, multiview supervised, and multiview semisupervised learning methods have an inherently close relation with MVC. Therefore, the similarities and differences of these different learning paradigms are also worth discussing. An obvious commonality between them is that they all learn with multiview information. However, their learning targets are different. Multiview representation methods aim to learn a joint compact representation for subjects from all of the views whereas MVC aims to perform sample partitioning, and MVC is learned without any label information. In contrast, multiview supervised and semisupervised learning methods have access to all or part of the label information. Some of the view combination strategies in these related paradigms can be borrowed and adapted by MVC. In addition, the relationships among MVC, ensemble clustering, multitask clustering are also elaborated in this review.

MVC has been applied to many scientific domains such as computer vision, natural language processing, social multimedia, bioinformatics, and health informatics. Although MVC has permeated into many fields and made great success in practice, there are still some open problems that limit its further advancement. We point out several open problems and hope they can be helpful to promote the development of MVC. With the survey presented in this article, we hope that readers can have a more comprehensive view of the MVC development and what is beyond the current progress.

There has been an earlier MVC survey [6]. We describe the differences between that one and ours which necessitate this survey. First, that work summarized the methods corresponding to a subset of the methods in our discriminative category, but the generative category of methods is a nonnegligible direction. The generative methods assume that each cluster comes from a specific distribution in each view and combine them together to conduct MVC. Since most of them are based on the EM algorithm or convex mixture model, they have some inherent advantages over discriminative methods, such as being capable of dealing with missing values or obtaining global optimal solutions. Second, we discuss the relationship between MVC and several related topics, such as multiview representation learning, ensemble clustering, multitask clustering, and multiview supervised, and semisupervised learning. This discussion helps researchers to position MVC in a scientific context and potentially gain deeper insights into all these topics. Third, we summarize representative applications of the various MVC methods for reference by interested users. Fourth, in Sections II and III, we examine the pros and cons of each class of MVC methods and give the circumstances for which they are suitable. Also, we conduct a comprehensive comparison over the representative MVC algorithm in each group to further analyze and verify the advantages and disadvantages of each group of MVC algorithms. Last but not least, we draw attention to certain open problems with the hope that these directions help further advance MVC.

The remainder of this article is organized as follows. In Section II, we review the existing generative methods for MVC. Section III introduces several classes of discriminative MVC methods. In Section IV, we analyze the relationships between MVC and several related topics. Section V presents the applications of MVC in different areas. In Section VI, we introduce several commonly used MVC datasets and conduct some experiments on them to investigate how they perform. In Section VII, we list several open problems with the aim to help advance the development of MVC. Finally, we make the conclusion in Section VIII.

II. Generative Approaches

Generative approaches aim to learn the generative models each of which is used to generate the data from a cluster. In multiview case, multiple generative models need to be learned and then combined to obtain the final clustering results. In most cases, generative clustering approaches are based on mixture models or constructed via expectation maximization (EM) [7]. Therefore, we first introduce mixture models, EM algorithm and another popular single-view clustering model named convex mixture models (CMMs) [8], and then introduce the multiview variants of these methods.

A. Mixture Models and CMMs

A generative approach assumes that data are sampled independently from a mixture model of multiple probability distributions. The mixture distribution can be written as

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p(\boldsymbol{x}|\boldsymbol{\theta}_k) \tag{1}$$

where π_k is the prior probability of the kth component and satisfies $\pi_k \geq 0$, and $\sum_{k=1}^{K} \pi_k = 1$, $\boldsymbol{\theta}_k$ is the parameter of the kth probability density model, and $\boldsymbol{\theta} = \{(\pi_k, \boldsymbol{\theta}_k), k = 1, 2, \ldots, K\}$ is the parameter set of the mixture model. For instance, $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ for Gaussian mixture model.

EM is a widely used algorithm for parameter estimation of the mixture models. Suppose that the observed data and unobserved data are denoted by $\boldsymbol{X}$ and $\boldsymbol{Z}$, respectively. $\{\boldsymbol{X}, \boldsymbol{Z}\}$ and $\boldsymbol{X}$ are called *complete data* and *incomplete data*, respectively. In the E (expectation) step, the posterior distribution $p(\boldsymbol{Z}|X, \boldsymbol{\theta}^{old})$ of the unobserved data are evaluated with the current parameter values $\boldsymbol{\theta}^{old}$. The E step calculates the expectation of the complete data log likelihood evaluated for some general parameter value $\boldsymbol{\theta}$. The expectation, denoted by $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$, is given by

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) = \sum_{\boldsymbol{Z}} p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{old}) \ln p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}). \tag{2}$$

The first item is the posterior distribution of the latent variables $\boldsymbol{Z}$ and the second one is the complete data log likelihood. According to maximum likelihood estimation, the M (maximization) step updates the parameters by maximizing the function (2)

$$\boldsymbol{\theta} = \arg \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}). \tag{3}$$

Note that for clustering, $\boldsymbol{X}$ can be considered as the observed data while $\boldsymbol{Z}$ is the latent variable whose entry z_{nk} indicates the nth data point comes from the kth component. Also note that the posterior distribution form used to be evaluated in E step and the expectation of the complete data log likelihood used to evaluate the parameters are different for different distribution assumptions. It can adopt Gaussian distribution and any other probability distribution form, which depends on specific applications.

CMMs [8] are simplified mixture models that can probabilistically assign data points to clusters after extracting the representative exemplars from the dataset. By maximizing the log-likelihood, all instances compete to become the "center" (representative exemplar) of the clusters. The instances corresponding to the components that received the highest priors are selected exemplars and then the remaining instances are assigned to the "closest" exemplar. The priors of the components are the only adjustable parameters of a CMM.

Given a dataset $\boldsymbol{X} = \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N \in \mathbb{R}^{d \times N}$, the CMM distribution is $Q(\boldsymbol{x}) = \sum_{j=1}^{N} q_j f_j(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^d$, where $q_j \geq 0$ denotes the prior probability of the jth component that satisfies the constraint $\sum_{j=1}^{N} q_j = 1$, and $f_j(\boldsymbol{x})$ is an exponential family distribution, with its expected parameters equal to the jth data point. Due to the bijection relationship between the exponential families and Bregman divergences [9], the exponential family $f_j(\boldsymbol{x}) = C_\phi(\boldsymbol{x})\text{exp}(-\beta d_\phi(\boldsymbol{x}, \boldsymbol{x}_j))$ where d_ϕ denotes the Bregman divergence that calculates the component distribution, $C_\phi(\boldsymbol{x})$ is independent of $\boldsymbol{x}_j$, and β is a constant controlling the sharpness of the components.

The log-likelihood that needs to be maximized is given as $L(\boldsymbol{X}; \{q_j\}_{j=1}^{N}) = \frac{1}{N} \sum_{i=1}^{N} \log(\sum_{j=1}^{N} q_j f_j(\boldsymbol{x}_i)) = \frac{1}{N} \sum_{i=1}^{N} \log(\sum_{j=1}^{N} q_j e^{-\beta d_\phi(\boldsymbol{x}_i, \boldsymbol{x}_j)})$+const. If the empirical samples are equally drawn, i.e., the prior of drawing each example is $\hat{P} = 1/N$, the log-likelihood can be equivalently expressed in terms of Kullback Leibler (KL) divergence between $\hat{P}$ and $Q(\boldsymbol{x})$ as

$$\begin{aligned} D(\hat{P}|Q) = & -\sum_{i=1}^{N} \hat{P}(\boldsymbol{x}_i) \text{log} Q(\boldsymbol{x}_i) - \mathbb{H}(\hat{P}) \\ = & -L(\boldsymbol{X}; \{q_j\}_{j=1}^{N}) + \text{c} \end{aligned} \tag{4}$$

where $\mathbb{H}(\hat{P})$ is the entropy of the empirical distribution $\hat{P}(\boldsymbol{x})$ which does not depend on the parameter q_j, and c is a constant. Now, the problem is changed into minimizing (4), which is convex and can be solved by an iterative algorithm. In such an algorithm, the updating rule for prior probabilities is given by

$$q_j^{(t+1)} = q_j^{(t)} \sum_{i=1}^{N} \frac{\hat{P}(\boldsymbol{x}_i) f_j(\boldsymbol{x}_i)}{\sum_{j'=1}^{N} q_{j'}^{(t)} f_{j'}(\boldsymbol{x}_i)}. \tag{5}$$

The data points are grouped into K disjoint clusters by requiring the instances with the K highest q_j values to serve as exemplars and then assigning each of the remaining instances to an exemplar with which the instance has the highest posterior probability. Note that the clustering performance is affected by the value of β. In [8] a reference value β_0 is determined using an empirical rule $\beta_0 = N^2 \text{log} N / \sum_{i,j=1}^{N} d_\phi(\boldsymbol{x}_i, \boldsymbol{x}_j)$ to identify a reasonable range of β, which is around β_0. Further details are mentioned in [8].

B. MVC Based on Mixture Models or EM Algorithm

The method in [10] assumes that the two views are independent, and a multinomial distribution is adopted for document clustering problem. It uses the two-view case as an example, and executes the M and E steps on each view and then interchange the posteriors in two separate views in each iteration. The optimization process is terminated if the log-likelihood of observing the data do not reach a new maximum for a fixed number of iterations in each view. Based on different criteria and assumptions, two multiview EM algorithm versions for finite mixture models are proposed in [11].

Specifically, based on the CMMs for single-view clustering, the multiview version proposed in [12] became much attractive because it can locate the global optimum, and thus, avoid the initialization and local optima problems of standard mixture models, which require multiple executions of the EM algorithms.

For multiview CMMs, each $\boldsymbol{x}_i$ with m views is denoted by $\{\boldsymbol{x}_i^1, \boldsymbol{x}_i^2, \ldots, \boldsymbol{x}_i^m\}$, $\boldsymbol{x}_i^v \in \mathbb{R}^{d^v}$, the mixture distribution for each view is given as $Q^v(\boldsymbol{x}^v) = \sum_{j=1}^N q_j f_j^v(\boldsymbol{x}^v) = C_\phi(\boldsymbol{x}^v) \sum_{j=1}^N q_j e^{-\beta^v d_{\phi_v}(\boldsymbol{x}^v, \boldsymbol{x}_j^v)}$. To pursue a common clustering across all views, all $Q^v(\boldsymbol{x}^v)$ share the same priors. In addition, an empirical data set distribution $\hat{P}^v(\boldsymbol{x}^v) = 1/N$, $\boldsymbol{x}^v \in \{\boldsymbol{x}_1^v, \boldsymbol{x}_2^v, \ldots, \boldsymbol{x}_N^v\}$, is associated with each view and the multiview algorithm minimizes the sum of KL divergences between $\hat{P}^v(\boldsymbol{x}^v)$ and $Q^v(\boldsymbol{x}^v)$ across all views with the constraint $\sum_{j=1}^N q_j = 1$. Thus, the formulated optimization problem is

$$\min_{q_1,\ldots,q_N} \sum_{v=1}^m D(\hat{P}^v | Q^v) = \min_{q_1,\ldots,q_N} -\sum_{v=1}^m \sum_{i=1}^N \hat{P}^v(\boldsymbol{x}_i^v) \log Q^v(\boldsymbol{x}_i^v) - \sum_{v=1}^m \mathbb{H}(\hat{P}^v). \quad (6)$$

It is straightforward to see that the optimized objective is convex, hence the global minimum can be found. The prior update rule is given as follows:

$$q_j^{(t+1)} = \frac{q_j^{(t)}}{M} \sum_{v=1}^m \sum_{i=1}^N \frac{\hat{P}^v f_j^v(\boldsymbol{x}_i^v)}{\sum_{j'=1}^N q_{j'}^{(t)} f_{j'}^v(\boldsymbol{x}_i^v)}. \quad (7)$$

The prior q_j associated with the jth instance is a measure of how likely this instance is to be an exemplar, taking all views into account. The appropriate β^v values are identified in the range of an empirically defined β_0^v by $\beta_0^v = N^2 \log N / \sum_{i,j=1}^N d_{\phi_v}(\boldsymbol{x}_i^v, \boldsymbol{x}_j^v)$. From (6), it can be found that all views contribute equally to the sum, without considering their different importance. To overcome this limitation, a weighted version of multiview CMMs was proposed in [13].

1) Summary: For the aforementioned MVC generative methods, we can find that linear combination with different weights to different views is a common way to fuse information. In addition, multiview generative clustering has not attracted enough attention, maybe because the technique is more difficult compared with its discriminative counterpart. It is not easy for generative methods to combine views by sharing a common variable or distribution, but sharing common variable(s) is the most popular way to combine views in the discriminative paradigm. This can limit the development of multiview generative clustering to some extent, but researchers are actively seeking for ways to combine views in multiview generative clustering methods. For example, it is quite reasonable to share some commonality across the distributions of the data views corresponding to the same cluster. Moreover, generative methods have their advantages. First, generative methods are based on data distribution, and if the data do follow the distribution assumed, the method should perform well. Second, given the methods, such as in [12] can get the global optimum, it is quite intriguing. Third, there is no need to prespecify the number of clusters. We believe multiview generative clustering even single-view generative clustering method is an underestimated direction, more efforts can be made along this direction in the future.

III. Discriminative Approaches

Compared with generative approaches, discriminative approaches directly optimize the objective to seek for the best clustering solution rather than first modeling the sample distribution then solving these models to determine clustering result. Directly focusing on the objective of clustering makes discriminative approaches gain more attentions and develop more comprehensively. Up to now, most of existing MVC methods are discriminative approaches. Based on how to combine multiple views, we categorize MVC methods into five main classes and introduce the representative works in each group.

Given the data with N data points and m views, each data point $\boldsymbol{x}_i$ is denoted by $\{\boldsymbol{x}_i^1, \boldsymbol{x}_i^2, \ldots, \boldsymbol{x}_i^m\}$, $\boldsymbol{x}_i^v \in \mathbb{R}^{d^v}$. The aim of MVC is to cluster the N data points into K classes. That is, finally we will get a membership matrix $\boldsymbol{H} \in \mathbb{R}^{N \times K}$ to indicate which data points are in the same group while others in other classes, the sum of each row entries of $\boldsymbol{H}$ should be 1 to make sure each row is a probability distribution. If only one entry of each row is 1 and all others are 0, it is the so-called hard clustering otherwise it is soft clustering. In the following five subsections, we will introduce each class of multiview discriminative clustering methods.

A. Common Eigenvector Matrix (Mainly Multiview Spectral Clustering)

This class of MVC methods are based on a commonly used clustering technique spectral clustering. Since spectral clustering hinges crucially on the construction of the graph Laplacian [14], [15] and the resulting eigenvectors reflect the grouping structure of the data, this class of MVC methods guarantee to get a common clustering result by assuming that all the views share the same or similar eigenvector matrix. There are two representative methods: cotraining spectral clustering [16] and coregularized spectral clustering [17]. Before discussing them, we will introduce spectral clustering [18] first.

1) Spectral Clustering: Spectral clustering is a clustering technique that utilizes the properties of graph Laplacian where the graph edges denote the similarities between data points and solve a relaxation of the normalized min-cut problem on the graph [19]. Compared with other widely used methods such as the k-means algorithm that only fits the spherical shaped clusters, spectral clustering can apply to arbitrary shaped clusters and demonstrate good performance.

Given $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ as a weighted undirected graph with vertex set $\boldsymbol{V} = v_1, \ldots, v_N$. The data adjacency matrix of the graph is defined as $\boldsymbol{W}$ whose entry w_{ij} represents the similarity of two vertices v_i and v_j. If $w_{ij} = 0$, it means that the vertices v_i and v_j are not connected. Apparently $\boldsymbol{W}$ is symmetric since $\boldsymbol{G}$ is an undirected graph. The degree matrix $\boldsymbol{D}$ is defined as the diagonal matrix with the degrees $d_1, \ldots, d_N$ of each vertex on the diagonal, where $d_i = \sum_{j=1}^N w_{ij}$. Generally, the graph Laplacian is $\boldsymbol{D} - \boldsymbol{W}$ and the normalized graph Laplacian is $\tilde{\boldsymbol{L}} = \boldsymbol{D}^{-1/2}(\boldsymbol{D} - \boldsymbol{W})\boldsymbol{D}^{-1/2}$. In many spectral clustering works, e.g., [16]–[18], [20], $\boldsymbol{L} = \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$ is also used to change a minimization problem (9) into a maximization problem (8) since $\boldsymbol{L} = \boldsymbol{I} - \tilde{\boldsymbol{L}}$, where $\boldsymbol{I}$ is the identity matrix. Following the same terminology adopted in [16]–[18], [20], we will name both $\boldsymbol{L}$ and $\tilde{\boldsymbol{L}}$ as normalized graph Laplacians afterward. Now the single-view spectral clustering approach can be formulated as follows:

$$\begin{cases} \max\limits_{\boldsymbol{U}\in\mathbb{R}^{N\times K}} tr(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{U}) \\ \text{s.t.} \quad \boldsymbol{U}^{\mathrm{T}}\boldsymbol{U} = \boldsymbol{I} \end{cases} \tag{8}$$

which is also equivalent to the following problem:

$$\begin{cases} \min\limits_{\boldsymbol{U}\in\mathbb{R}^{N\times K}} tr(\boldsymbol{U}^{\mathrm{T}}\tilde{\boldsymbol{L}}\boldsymbol{U}) \\ \text{s.t.} \quad \boldsymbol{U}^{\mathrm{T}}\boldsymbol{U} = \boldsymbol{I} \end{cases} \tag{9}$$

where tr denotes the trace norm of a matrix. The rows of matrix $\boldsymbol{U}$ are the embeddings of the data points, which can be fed into the k-means to obtain the final clustering results. A version of the Rayleigh–Ritz theorem in [21] shows that the solution of the above optimization problem is given by choosing $\boldsymbol{U}$ as the matrix containing, respectively, the largest or smallest K eigenvectors of $\boldsymbol{L}$ or $\tilde{\boldsymbol{L}}$ as columns. To understand the spectral clustering method better, we outline a commonly used algorithm [18] as follows:

1) construct the adjacency matrix $\boldsymbol{W}$;
2) compute the normalized Laplacian matrix $\boldsymbol{L} = \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$;
3) calculate the eigenvectors of $\boldsymbol{L}$ and stack the top K eigenvectors as the columns to construct a $N \times K$ matrix $\boldsymbol{U}$;
4) normalize each row of $\boldsymbol{U}$ to obtain $\boldsymbol{U}_{\text{sym}}$;
5) run the k-means algorithm to cluster the row vectors of $\boldsymbol{U}_{\text{sym}}$;
6) assign subject i to cluster k if the ith row of $\boldsymbol{U}_{\text{sym}}$ is assigned to cluster k by the k-means algorithm.

Apart from the symmetric normalization operator $\boldsymbol{U}_{\text{sym}}$, another normalization operator $\boldsymbol{U}_{lr} = \boldsymbol{D}^{-1}\boldsymbol{W}$ is also commonly used. The work in [22] can be referred for further details about spectral clustering.

2) Cotraining Multiview Spectral Clustering: For semisupervised learning, cotraining with two views has been a widely recognized idea when both labeled and unlabeled data are available. It assumes that the predictive models constructed in each of the two views will lead to the same labels for the same sample with high probability. There are two main assumptions to guarantee the success of cotraining.

1) *Sufficiency:* Each view is sufficient for sample classification on its own.
2) *Conditional independence:* The views are conditionally independent given the class labels. In the original cotraining algorithm [23], two initial predictive functions f_1 and f_2 are trained in each view using the labeled data, then the following steps are repeatedly performed: the most confident examples predicted by f_1 are added to the labeled set to train f_2 and vice versa, then f_1 and f_2 are retrained on the enlarged labeled datasets. It can be shown that after a number of iterations, f_1 and f_2 will agree with each other on labels.

For cotraining multiview spectral clustering, the motivation is similar: the clustering result in all views should agree. In spectral clustering, the eigenvectors of the graph Laplacian encode the discriminative information of the clustering. Therefore, cotraining multiview spectral clustering [16] uses the eigenvectors of the graph Laplacian in one view to cluster samples and then use the clustering result to modify the graph Laplacian in the other view.

Algorithm 1: Cotraining Multiview Spectral Clustering.

```
Input: Similarity matrices for two views: $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$.
Output: Assignments to $K$ clusters.
Initialize: $\boldsymbol{L}^{(v)} = \boldsymbol{D}^{(v)(-1/2)}\boldsymbol{L}^{(v)}\boldsymbol{D}^{(v)(-1/2)}$ for $v = 1, 2$,
    ${\boldsymbol{U}^{(v)}}^0 = \underset{\boldsymbol{U}\in\mathbb{R}^{N\times K}}{\text{argmax}}\, tr(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{L}^{(v)}\boldsymbol{U})$ s.t. $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U} = I$ for $v = 1, 2$.
for i=1 to t do
    1. $\boldsymbol{S}^{(1)} = sym\ ({\boldsymbol{U}^{(2)}}^{i-1}{{\boldsymbol{U}^{(2)}}^{i-1}}^{\mathrm{T}}\boldsymbol{W}^{(1)})$
    2. $\boldsymbol{S}^{(2)} = sym\ ({\boldsymbol{U}^{(1)}}^{i-1}{{\boldsymbol{U}^{(1)}}^{i-1}}^{\mathrm{T}}\boldsymbol{W}^{(2)})$
    3. Use $\boldsymbol{S}^{(1)}$ and $\boldsymbol{S}^{(2)}$ as the new graph similarities and compute the graph Laplacians. Solve for the largest $K$ eigenvectors to obtain ${\boldsymbol{U}^{(1)}}^i$ and ${\boldsymbol{U}^{(2)}}^i$
end for
4: Normalize each row of ${\boldsymbol{U}^{(1)}}^i$ and ${\boldsymbol{U}^{(2)}}^i$.
5: Form matrix $\boldsymbol{V} = {\boldsymbol{U}^{(v)}}^i$, where $v$ is the most informative view a priori. If there is no prior knowledge on the view informativeness, matrix $\boldsymbol{V}$ can also be set to be column-wise concatenation of the two ${\boldsymbol{U}^{(v)}}^i$s.
6: Assign example $j$ to cluster $K$ if the $j$th row of $\boldsymbol{V}$ is assigned to cluster $K$ by the k-means algorithm.
```

Each column of the similarity matrix (also called the adjacency matrix) $\boldsymbol{W}_{N\times N}$ can be considered as a N-dimensional vector that indicates the similarities of ith point with all the points in the graph. Since the largest K eigenvectors have the discriminative information for clustering, the similarity vectors can be projected along those directions to retain the discriminative information for clustering and throw away the within cluster details that might confuse the clustering. After that, the projected information is projected back to the original N-dimensional space to get the modified graph. Finally, k-means algorithm is conducted on most informative eigenvector matrix to get the final clustering result.

To make the cotraining spectral clustering algorithm clear, we borrow Algorithm 1 from [16]. Note that the symmetrization operator sym on a matrix $\boldsymbol{S}$ is defined as sym $(\boldsymbol{S}) = (\boldsymbol{S} + \boldsymbol{S}^{\mathrm{T}})/2$ in Algorithm 1.

3) Coregularized Multiview Spectral Clustering: Coregularization is an effective technique in semisupervised multiview learning. The core idea of coregularization is to minimize the distinction between the predictor functions of two views acting as one part of the objective function. However, there are no predictor functions in unsupervised learning like clustering, so how to implement the coregularization idea in clustering problem? Coregularized multiview spectral clustering [17] adopted the eigenvectors of graph Laplacian to play the similar role of predictor functions in semisupervised learning scenario and proposed two coregularized clustering approaches.

Let $\boldsymbol{U}^{(s)}$ and $\boldsymbol{U}^{(t)}$ be the eigenvector matrices corresponding to any pair of view graph Laplacians $\boldsymbol{L}^{(s)}$ and $\boldsymbol{L}^{(t)}$ $(1 \leq s, t \leq m, s \neq t)$. The first version uses a pairwise coregularization criteria that enforces $\boldsymbol{U}^{(s)}$ and $\boldsymbol{U}^{(t)}$ as close as possible. The measure of clustering disagreement between the two views s and t is $D(\boldsymbol{U}^{(s)}, \boldsymbol{U}^{(t)}) = \| \frac{\boldsymbol{K}^{(s)}}{\|K^{(s)}\|_F{}^2} - \frac{\boldsymbol{K}^{(t)}}{\|\boldsymbol{K}^{(t)}\|_F{}^2} \|_F^2$, where $\boldsymbol{K}^{(s)} = \boldsymbol{U}^{(s)}\boldsymbol{U}^{(s)^{\mathrm{T}}}$ using linear kernel is the similarity matrix of $\boldsymbol{U}^{(s)}$. Since $\|\boldsymbol{K}^{(s)}\|_F^2 = K$, where K is the number of the clusters, disagreement between the clustering solutions in the two views can be measured by $D(\boldsymbol{U}^{(s)}, \boldsymbol{U}^{(t)}) = -tr(\boldsymbol{U}^{(s)}\boldsymbol{U}^{(s)^{\mathrm{T}}}\boldsymbol{U}^{(t)}\boldsymbol{U}^{(t)^{\mathrm{T}}})$. Integrating the measure of the disagreement between any pair of views into the spectral clustering objective function, the pairwise coregularized multiview spectral clustering can be formed as the following optimization problem:

$$\begin{cases} \max\limits_{\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \ldots, \boldsymbol{U}^{(m)} \in \mathbb{R}^{N \times K}} \sum_{s=1}^{m} \; tr(\boldsymbol{U}^{(s)^{\mathrm{T}}}\boldsymbol{L}^{(s)}\boldsymbol{U}^{(s)}) \\ \quad + \sum_{1 \leq s,t \leq m, s \neq t} \lambda \, tr(\boldsymbol{U}^{(s)}\boldsymbol{U}^{(s)^{\mathrm{T}}}\boldsymbol{U}^{(t)}\boldsymbol{U}^{(t)^{\mathrm{T}}}) \\ \quad \text{s.t.} \quad \boldsymbol{U}^{(s)^{\mathrm{T}}}\boldsymbol{U}^{(s)} = \boldsymbol{I}, \, \forall 1 \leq s \leq m. \end{cases} \tag{10}$$

The hyperparameter λ is used to tradeoff the spectral clustering objectives and the spectral embedding disagreement terms. After the embeddings are obtained, each $\boldsymbol{U}^{s}$ can be fed for k-means clustering method, the final results are marginally different.

The second version named centroid-based coregularization enforces the eigenvector matrix from each view to be similar by regularizing them toward a common consensus eigenvector matrix. The corresponding optimization problem is formulated as

$$\begin{cases} \max\limits_{\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \ldots, \boldsymbol{U}^{(m)}, \boldsymbol{U}^{*} \in \mathbb{R}^{N \times K}} \sum_{s=1}^{m} \; tr(\boldsymbol{U}^{(s)^{\mathrm{T}}}\boldsymbol{L}^{(s)}\boldsymbol{U}^{(s)}) \\ \quad +\lambda_s \sum_{s=1}^{m} tr(\boldsymbol{U}^{(s)}\boldsymbol{U}^{(s)^{\mathrm{T}}}\boldsymbol{U}^{(*)}\boldsymbol{U}^{(*)^{\mathrm{T}}}) \\ \text{s.t.} \quad \boldsymbol{U}^{(s)^{\mathrm{T}}}\boldsymbol{U}^{(s)} = \boldsymbol{I}, \, \forall 1 \leq s \leq m, \quad \boldsymbol{U}^{*\mathrm{T}}\boldsymbol{U}^{*} = \boldsymbol{I}. \end{cases} \tag{11}$$

Compared with pairwise coregularized version, centroid-based MVC does not need to combine the obtained eigenvector matrices of all views to run k-means. However, the centroid-based version possesses one potential drawback: the noisy views could potentially affect the optimal eigenvectors as it depends on all the views.

Cai *et al.* [24] used a common indicator matrix across the views to perform multiview spectral clustering and derived a formulation similar to the centroid-based coregularization method. The main difference is that [24] used $tr((\boldsymbol{U}^{(*)} - \boldsymbol{U}^{(s)})^{\mathrm{T}}(\boldsymbol{U}^{(*)} - \boldsymbol{U}^{(s)}))$ as the disagreement measure between each view eigenvector matrix and the common eigenvector matrix while coregularized multiview spectral clustering [17] adopted $tr(\boldsymbol{U}^{(s)}\boldsymbol{U}^{(s)^{\mathrm{T}}}\boldsymbol{U}^{(*)}\boldsymbol{U}^{(*)^{\mathrm{T}}})$. The optimization problem [24] is formulated as

$$\begin{cases} \max\limits_{\boldsymbol{U}^{(s)}, s=1,2\cdots,m, \boldsymbol{U}^{*}} \sum_{s=1}^{m} \; tr(\boldsymbol{U}^{(s)^{\mathrm{T}}}\boldsymbol{L}^{(s)}\boldsymbol{U}^{(s)}) \\ \quad +\lambda \sum_{s=1}^{m} tr((\boldsymbol{U}^{*} - \boldsymbol{U}^{(s)})^{\mathrm{T}}(\boldsymbol{U}^{*} - \boldsymbol{U}^{(s)})) \\ \quad \text{s.t.} \quad \boldsymbol{U}^{*} \geq 0, \quad \boldsymbol{U}^{*\mathrm{T}}\boldsymbol{U}^{*} = I \end{cases} \tag{12}$$

where $\boldsymbol{U}^{*} \geq 0$ makes $\boldsymbol{U}^{*}$ become the final cluster indicator matrix. Different from general spectral clustering that get eigenvector matrix first and then run clustering (such as k-means that is sensitive to initialization condition) to assign clusters, Cai *et al.* [24] directly solves the final cluster indicator matrix, thus it will be more robust to the initial condition.

4) Others: Besides the two representative multiview spectral clustering methods discussed above, Wang *et al.* [25] enforces a common eigenvector matrix across the views and formulates a multiobjective problem which is then solved using Pareto optimization.

A relaxed kernel k-means can be shown to be equivalent to spectral clustering, as in the following Section III-D2, Ye *et al.* [26] proposes a coregularized kernel k-means for MVC. With a multilayer Grassmann manifold interpretation, Dong *et al.* [27] obtains the same formulation with the pairwise coregularized multiview spectral clustering.

Because the MVC methods based on a shared eigenvector matrix are rooted from the special clustering, they can be applied to data clusters of any shape or any positioning of cluster centers. This merit is inherited from spectral clustering that does not make any assumption about the statistics of the clusters. However, since spectral clustering needs eigen decomposition, this type of MVC methods can be time consuming.

B. Common Coefficient Matrix (Mainly Multiview Subspace Clustering)

In many practical applications, even though the given data are high dimensional, the intrinsic dimension of the problem is often low. For example, the number of pixels in a given image can be large, yet only a few parameters are used to describe the appearance, geometry, and dynamics of a scene. This motivates the development of finding the underlying low dimensional subspace. In practice, the data could be sampled from multiple subspaces. Subspace clustering [28] is the technique to find the underlying subspaces and then cluster the data points correctly according to the identified subspaces.

1) Subspace Clustering: Subspace clustering uses the self-expressiveness property [29] of the data samples, i.e., each sample can be represented by a linear combination of a few other data samples. The classic subspace clustering formulation is given as follows:

$$\boldsymbol{X} = \boldsymbol{X}\boldsymbol{Z} + \boldsymbol{E} \tag{13}$$

where $\boldsymbol{Z} = \{z_1, z_2, \ldots, z_N\} \in \mathbb{R}^{N \times N}$ is the subspace coefficient matrix (representation matrix), and each z_i is the representation of the original data point x_i based on the subspace. $\boldsymbol{E} \in \mathbb{R}^{N \times N}$ is the noise matrix.

The subspace clustering can be formulated as the following optimization problem:

$$\begin{cases} \min\limits_{\boldsymbol{Z}} \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{Z}\|_F^2 \\ \text{s.t.} \quad \boldsymbol{Z}(i,i) = 0, \boldsymbol{Z}^{\mathrm{T}}\boldsymbol{1} = \boldsymbol{1}. \end{cases} \tag{14}$$

The constraint $\boldsymbol{Z}(i,i) = 0$ is used to avoid the case that a data point is represented by itself, while $\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{1} = \boldsymbol{1}$ denotes that the data point lies in a union of affine subspaces. The nonzero elements of z_i correspond to data points from the same subspace.

After getting the subspace representation $\boldsymbol{Z}$, the similarity matrix $\boldsymbol{W} = \frac{|\boldsymbol{Z}|+|\boldsymbol{Z}^{\mathrm{T}}|}{2}$ can be obtained to further construct the graph Laplacian and then run spectral clustering on that graph Laplacian to get the final clustering results.

2) Multiview Subspace Clustering: With multiview information, each subspace representation $\boldsymbol{Z}_v$ can be obtained from each view. To get a consistent clustering result from multiple views, Yin *et al.* [30] shares the common coefficient matrix by enforcing the coefficient matrices from each pair of views as similar as possible. The optimization problem is formulated as

$$\begin{cases} \min\limits_{\boldsymbol{Z}^{(s)},s=1,2,\ldots,m} \sum_{s=1}^{m} \|\boldsymbol{X}^{(s)} - \boldsymbol{X}^{(s)}\boldsymbol{Z}^{(s)}\|_F^2 \\ \quad +\alpha \sum_{s=1}^{m} \|\boldsymbol{Z}^{(s)}\|_1 + \beta \sum_{1\leq s\leq t} \|\boldsymbol{Z}^{(s)} - \boldsymbol{Z}^{(t)}\|_1 \\ \text{s.t.} \quad \mathrm{diag}(\boldsymbol{Z}^{(s)}) = 0, \quad \forall s \in \{1, 2, \ldots, m\}. \end{cases} \tag{15}$$

where $\|\boldsymbol{Z}^{(s)} - \boldsymbol{Z}^{(t)}\|_1$ is the l_1-norm based pairwise coregularization constraint that can alleviate the noise problem. $\|\boldsymbol{Z}\|_1$ is used to enforce sparse solution. diag$(\boldsymbol{Z})$ denotes the diagonal elements of matrix $\boldsymbol{Z}$, and the zero constraint is used to avoid trivial solution (each data point represents by itself).

Maria *et al.* [31] also considered the low rank and sparse representation to conduct multiview subspace clustering. Wang *et al.* [32] enforced the similar idea to combine multi-view information. Apart from that, it adopted a multigraph regularization with each graph Laplacian regularization characterizing the view-dependent nonlinear local data similarity. At the same time, it assumes that the view-dependent representation is low rank and sparse and considers the sparse noise in the data. Wang *et al.* [33] proposed an angular based similarity to measure the correlation consensus in multiple views and obtained a robust subspace clustering for multiview data. Zhang *et al.* [34] adopted linear correlation and neural networks to integrate the representation of each view and proposed two latent subspace MVC methods. To deal with the scenario where each view is unsufficient to discover the latent cluster structure, Huang *et al.* [35] proposed a multiview intact subspace clustering by assuming a latent space and defining a mapping function from the latent space to view representation. Different from the above approaches, the three works [36]–[38] adopted general NMF formulation but shared a common representation matrix for the samples with both views and kept each view representation matrix specific. Zhao *et al.* [39] adopted a deep semi-NMF to perform MVC, and enforced a common coefficient matrix in the last layer to exploit the multiview information. By introducing a label constraint matrix and enforcing representation matrix of each view close to a common one, Cai *et al.* [40] solved the MVC in semisupervised settings.

The MVC methods based on a shared coefficient matrix are applied to multiview subspace clustering, which assumes that the cluster structures can be found by identifying the low dimensional subspaces. This kind of MVC methods has great utility in the computer vision field. Typically, after the final low-dimensional representation is obtained, spectral clustering is conducted on the graph Laplacian constructed from that representation, so this group of methods possesses the same advantages and disadvantages as spectral clustering as discussed in Section III-A.

C. Common Indicator Matrix (Mainly Multiview NMF Clustering)

NMF is commonly used in clustering. It enforces one of the factorized matrix as an indicator matrix whose nonzero entry can indicate which data point belongs to which cluster. Therefore, enforcing the indicator matrix for multiple views be same or similar is a natural way to conduct MVC.

1) Nonnegative Matrix Factorization: For a nonnegative data matrix $\boldsymbol{X} \in \mathbb{R}_+^{d\times N}$, NMF [41] seeks two nonnegative matrix factors $\boldsymbol{U} \in \mathbb{R}_+^{d\times K}$ and $\boldsymbol{V} \in \mathbb{R}_+^{N\times K}$ such that their product is a good approximation of $\boldsymbol{X}$

$$\boldsymbol{X} \approx \boldsymbol{U}\boldsymbol{V}^{\mathrm{T}} \tag{16}$$

where K denotes the desired reduced dimension (for clustering, it is the number of clusters), $\boldsymbol{U}$ is the basis matrix, and $\boldsymbol{V}$ is the indicator matrix.

Due to the nonnegative constraints, a widely known property of NMF is that it can learn a part-based representation. It is intuitive and meaningful in many applications, such as in face recognition [41]. The samples in many of these applications, e.g., information retrieval [41] and pattern recognition [42], [43] can be explained as additive combinations of nonnegative basis vectors. The NMF has been applied successfully to cluster analysis and has shown the state-of-the-art performance [41], [44].

2) MVC Based on NMF: To combine multiview information in the NMF framework, Akata *et al.* [45] enforces a common indicator matrix in the NMF among different views to perform MVC. However, the indicator matrix $\boldsymbol{V}^{(v)}$ might not be comparable at the same scale. In order to keep the clustering solutions across different views meaningful and comparable, Liu *et al.* [46] enforces a constraint to push each view-dependent indicator matrix toward a common indicator matrix, which leads to another normalization constraint inspired by the connection between NMF and probability latent semantic analysis. The final optimization problem is formulated as

$$\begin{cases} \min\limits_{\boldsymbol{U}^{(v)},\boldsymbol{V}^{(v)},v=1,2,\ldots,m} \sum_{v=1}^{m} \|\boldsymbol{X}^{(v)} - \boldsymbol{U}^{(v)}\boldsymbol{V}^{(v)}\|_F^2 \\ \quad + \sum_{v=1}^{m} \lambda_v \|\boldsymbol{V}^{(v)} - \boldsymbol{V}^*\|_F^2 \\ \text{s.t.} \quad \forall 1 \leq k \leq K, \|\boldsymbol{U}_{.,k}^{(v)}\|_1 = 1, \boldsymbol{U}^{(v)}, \boldsymbol{V}^{(v)}, \boldsymbol{V}^{(*)} \geq 0. \end{cases} \tag{17}$$

The constraint $\|\boldsymbol{U}_{.,k}^{(v)}\|_1 = 1$ is used to guarantee $\boldsymbol{V}^{(v)}$ within the same range for different v such that the comparison between the view-dependent indicator matrix $\boldsymbol{V}^{(v)}$ and the consensus indicator matrix $\boldsymbol{V}^{(*)}$ is reasonable. After obtaining the consensus matrix $\boldsymbol{V}^*$, the cluster label of data point i can be computed from $argmax_k \boldsymbol{V}_{i,k}^*$.

3) Multiview K-Means: The k-means clustering method can be formulated using NMF by introducing an indicator matrix $\boldsymbol{U}$.

The NMF formulation of k-means clustering is

$$\begin{cases} \min\limits_{\boldsymbol{U},\boldsymbol{V}} \|\boldsymbol{X}^{\mathrm{T}} - \boldsymbol{U}\boldsymbol{V}^{\mathrm{T}}\|_F^2 \\ \text{s.t.} \quad \boldsymbol{U}_{i,k} \in \{0,1\}, \sum_{k=1}^{K} \boldsymbol{U}_{i,k} = 1, \forall i = 1,2,\ldots,N \end{cases} \tag{18}$$

where the columns of $\boldsymbol{V} \in \mathbb{R}^{d\times K}$ give the cluster centroids.

Because the k-means algorithm has lower computational cost than those requiring eigen-decomposition, it can be a good choice for large scale data clustering. To deal with large scale multiview data, Cai *et al.* [47] proposed a multiview k-means clustering method by adopting a common indicator matrix across different views. The $\ell_{2,1}$ norm has been applied in traditional NMF-based clustering methods with proved performance, such as model sparse and robustness. Herein the Frobenius norm has been replaced by a $\ell_2, 1$ norm, and different views are weighed differently according to their importance. The new optimization problem obtained from (18) is formulated as follows:

$$\begin{cases} \min\limits_{\boldsymbol{V}^{(v)},\alpha^{(v)},\boldsymbol{U}} \sum_{v=1}^{m} (\alpha^{(v)})^{\gamma} \|\boldsymbol{X}^{(v)^{\mathrm{T}}} - \boldsymbol{U}\boldsymbol{V}^{(v)^{\mathrm{T}}}\|_{2,1} \\ \text{s.t.} \quad \boldsymbol{U}_{i,k} \in \{0,1\}, \sum_{k=1}^{K} \boldsymbol{U}_{i,k} = 1, \sum_{v=1}^{m} \alpha^{(v)} = 1 \end{cases} \tag{19}$$

where $\alpha^{(v)}$ is the weight for the vth view and γ is the parameter to control the weight distribution. By learning the weights α for different views, the important views will be emphasized.

Still based on multiview k-means clustering (18), to deal with high dimensional problems in multiple views, Xu *et al.* [48] introduced one projection matrix for data of each view, and then conduct MVC by enforcing the common indicator matrix. Their optimization problem is formulated as

$$\begin{cases} \min\limits_{\boldsymbol{V}^{(v)},\boldsymbol{W}^{(v)},\boldsymbol{U}} \sum_{v=1}^{m} \|\boldsymbol{X}^{(v)^{\mathrm{T}}}\boldsymbol{W}^{(v)} - \boldsymbol{U}\boldsymbol{V}^{(v)^{\mathrm{T}}}\|_F \\ \text{s.t.} \quad \boldsymbol{W}^{(v)^{\mathrm{T}}}\boldsymbol{W}^{(v)} = \boldsymbol{I}, \boldsymbol{U}_{i,k} \in \{0,1\}, \sum_{k=1}^{K} \boldsymbol{U}_{i,k} = 1 \end{cases} \tag{20}$$

where $\boldsymbol{W}^{(v)} \in \mathbb{R}^{D_v \times m_v}$ indicates the projection matrix which embeds the data matrix $\boldsymbol{X}^{(v)}$ from D_v to m_v, $m_v < D_v, \forall v$. Note that to deal with outliers, Frobenious norm (not squared) is adopted. By replacing Frobenious norm with a ℓ_2 norm and considering different importance of each view, a reweighted discriminative embedding k-means method is formulated as

$$\begin{cases} \min\limits_{\boldsymbol{V}^{(v)},\boldsymbol{W}^{(v)},\alpha^{(v)},\boldsymbol{U}} \sum_{v=1}^{m} \alpha^{(v)} \|\boldsymbol{X}^{(v)^{\mathrm{T}}}\boldsymbol{W}^{(v)} - \boldsymbol{U}\boldsymbol{V}^{(v)^{\mathrm{T}}}\|_2 \\ \text{s.t.} \quad \boldsymbol{W}^{(v)^{\mathrm{T}}}\boldsymbol{W}^{(v)} = \boldsymbol{I}, \boldsymbol{U}_{i,k} \in \{0,1\}, \sum_{k=1}^{K} \boldsymbol{U}_{i,k} = 1 \end{cases} \tag{21}$$

where $\alpha^{(v)} = (2\|\boldsymbol{X}^{(v)^{\mathrm{T}}}\boldsymbol{W}^{(v)} - \boldsymbol{U}\boldsymbol{V}^{(v)^{\mathrm{T}}}\|_F)^{-1}$ is the weight for the vth view and is computed by current $\boldsymbol{V}^{(v)}$, $\boldsymbol{W}^{(v)}$ and $\boldsymbol{U}$.

Besides the above multiview NMF clustering methods, Liu and Fu [49] introduced a categorical utility function to measure similarity between the common indicator matrix and the indicator matrix from each view and proposed a consensus based MVC method.

According to [50], when $\boldsymbol{W} = \boldsymbol{H} * \boldsymbol{H}^T$, where $\boldsymbol{W}$ indicates similarity between data points or is a kernel, the above method is equivalent to spectral clustering or kernel k-means clustering. Although the single view methods (NMF, kernel k-means, and spectral clustering) have connections between each other, their multiview versions are less connected because the views need to share some common factors, but there is only one factor $\boldsymbol{H}$, which cannot be used in multiple of the views. However, for the multiview k-means clustering method can be expressed as a multiview NMF-based clustering problem with $\boldsymbol{U}$ indicating the indicator matrix according to formulation (18).

4) Others: As mentioned earlier, there are generally two steps in subspace clustering: find a subspace representation and then run spectral clustering on the graph Laplacian computed from the subspace representation. To identify consistent clusters from different views, Gao *et al.* [51] merged these two steps in subspace clustering and enforced a common indicator matrix across different views. The formulation is given as follows:

$$\begin{cases} \min\limits_{\boldsymbol{Z}^{(v)},\boldsymbol{E}^{(v)},\boldsymbol{U}} \sum_{v=1}^{m} \|\boldsymbol{X}^{(v)} - \boldsymbol{X}^{(v)}\boldsymbol{Z}^{(v)} - \boldsymbol{E}^{(v)}\|_F^2 \\ \quad +\lambda_1 tr(\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{D}^{(v)} - \boldsymbol{W}^{(v)})\boldsymbol{U}) + \lambda_2 \sum_{v=1}^{m} \|\boldsymbol{E}^{(v)}\|_1 \\ \text{s.t.} \quad \boldsymbol{Z}^{(v)^{\mathrm{T}}}, \boldsymbol{Z}^{(v)}(i,i) = \boldsymbol{I}, \boldsymbol{U}^{\mathrm{T}}\boldsymbol{U} = \boldsymbol{I} \end{cases} \tag{22}$$

where $\boldsymbol{Z}^{(v)}$ is the subspace representation matrix of the vth view, $\boldsymbol{W}^{(v)} = \frac{|\boldsymbol{Z}^{(v)}|+|\boldsymbol{Z}^{(v)^{\mathrm{T}}}|}{2}$, $\boldsymbol{D}^{(v)}$ is a diagonal matrix with diagonal elements defined as $d_{v_{i,i}} = \sum_j w_{v_{i,j}}$, and $\boldsymbol{U}$ is the common indicator matrix which indicates a unique cluster assignment for all the views. Although this multiview subspace clustering method is based on subspace clustering, it does not enforce a common coefficient matrix $\boldsymbol{Z}$, but uses a common indicator matrix for different views. We thus categorize it into this group.

Wang *et al.* [52] integrates multiview information via a common indicator matrix and simultaneously selects features for different data clusters by formulating the problem as follows:

$$\begin{cases} \min\limits_{\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}=\boldsymbol{I},\boldsymbol{W}} \|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W} + \boldsymbol{1}_N\boldsymbol{b}^{\mathrm{T}} - \boldsymbol{U}\|_F \\ +\gamma_1\|\boldsymbol{W}\|_{G_1} + \gamma_2\|\boldsymbol{W}\|_{2,1} \end{cases} \tag{23}$$

where $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\} \in \mathbb{R}^{d\times N}$, but here each $\boldsymbol{x}_i$ includes the features from all the m views and each view has d_j features such that $d = \sum_{j=1}^{m} d_j$. The coefficient matrix $\boldsymbol{W} = [w_1^1, \ldots, w_K^1; \ldots, \ldots, \ldots; w_1^m, \ldots, w_K^m] \in \mathbb{R}^{d\times K}$ contains the weights of each feature for K clusters, $\boldsymbol{b} \in \mathbb{R}^{K\times 1}$ is the intercept vector, $\boldsymbol{1}_N$ is N-element constant vector of ones, and $\boldsymbol{U} = [u_1, \ldots, u_N]^{\mathrm{T}} \in \mathbb{R}^{N\times K}$ is the cluster (assignment) indicator matrix. The regularizer $\|\boldsymbol{W}\|_{G_1} = \sum_{i=1}^{K}\sum_{j=1}^{m} \|w_i^j\|_2$ is the group l_1 regularization to evaluate the importance of an entire view's features as a whole for a cluster whereas $\|\boldsymbol{W}\|_{2,1} = \sum_{i=1}^{d} \|w^i\|_2$ is the $l_{2,1}$ norm to select individual features from all views that are important for all clusters.

In [53], a matrix factorization approach was adopted to reconcile the clusters arisen from the individual views. Specifically, a matrix that contains the partition indicator of each individual view is created and then decomposed into two matrices: one showing the contribution of individual groupings to the final MVC, called metaclusters, and the other showing the assignment of instances to the meta-clusters. Tang *et al.* [54] treated MVC as clustering with multiple graphs, each of which is approximated by matrix factorization with two factors: a graph-specific factor and a factor common to all graphs. Qian *et al.* [55] and Zong *et al.* [56] required each view's indicator matrix to be as close as possible to a common indicator matrix and employed the Laplacian regularization to maintain the latent geometric structure of the views simultaneously. After learning indicator matrices of different views, Kang *et al.* [57] learned a common indicator matrix by measuring distance between indicator matrix and considering different impact each view enforces. Also by learning an indicator matrix and maximizing the worst-case performance against single-view case, Tao *et al.* [58] proposed a reliable MVC method. Zhang *et al.* [59] proposed a robust manifold matrix factorization to cluster hyperspectral images. Taking the discriminative information in low dimensional spaces into account, Ma *et al.* [60] extend the work in [59] to MVC by enforcing the same indicator matrix.

Besides using a common indicator matrix, [61]–[63] introduced a weight matrix to indicate whether there are missing entries so that it can tackle the missing value problem. The multiview self-paced clustering method [64] takes the complexities of the samples and views into consideration to alleviate the local minima problem. Tao *et al.* [65] enforces a common indicator matrix and seeks for the consensus clustering among all the views in an ensemble way. Another method that utilizes a common indicator matrix to combine multiple views [66] employed the linear discriminant analysis idea and automatically weighed different views. For graph-based clustering methods, the similarity matrix for each view is obtained, and then by minimizing the differences between a common indicator matrix and each similarity matrix, Nie *et al.* [67] provided one MVC method with multiple graphs.

The MVC methods that use a shared indicator matrix across views include the k-means or NMF. On one side, it can scale to large scale datasets compared with spectral clustering based MVC approaches. On the other side, it can only be applied to data with cluster of spherical shape to cluster center. This is because k-means clustering makes a strong assumption that the data points assigned to a cluster are spherical about the cluster center.

D. Direct Combination (Mainly Multikernel-Based MVC)

Besides the methods that share some structure among different views, direct view combination via a kernel is another common approach to perform MVC. A natural way is to define a kernel for each view and then combine these kernels in a convex combination [68]–[70].

1) Kernel Functions and Kernel Combination Methods: Kernel is a trick to learn nonlinear problem just by linear learning algorithm, since kernel function $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ can directly give the inner products in feature space without explicitly defining the nonlinear transformation ϕ. There are some common kernel functions as follows:

1) linear kernel: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i \cdot \boldsymbol{x}_j)$;
2) polynomial kernel: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i \cdot \boldsymbol{x}_j + 1)^d$;
3) Gaussian kernel (Radial basis kernel): $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\exp(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2})$;
4) sigmoid kernel: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\tanh(\eta \boldsymbol{x}_i \cdot \boldsymbol{x}_j + \nu))$.

Kernel functions in a reproducing kernel Hilbert space (RKHS) can be viewed as similarity functions [71], [72] in a vector space, so we can use a kernel as a non-Euclidean similarity measure in the spectral clustering and kernel k-means methods. There have been some works on multikernel learning for clustering [73]–[76], however, they are all for single-view clustering. If a kernel is derived from each view, and different kernels are combined elaborately to deal with the clustering problem, it will become the multikernel learning method for MVC. Obviously, multikernel learning [77]–[80] can be considered as the most important part in this kind of MVC methods. There are three main categories of methods for combining multiple kernels [81].

1) Linear combination: It includes two basic subcategories: unweighted sum $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{v=1}^{m} k_v(\boldsymbol{x}_i^v, \boldsymbol{x}_j^v)$ and weighted sum $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{v=1}^{m} w_v^q k_v(\boldsymbol{x}_i^v, \boldsymbol{x}_j^v)$ where $w_v \in \mathbb{R}_+$ denotes the kernel weight for the vth view and $\sum_{v=1}^{m} w_v = 1$, q is the hyperparameter to control the distribution of the weights,
2) Nonlinear combination: It uses a nonlinear function in terms of kernels—namely, multiplication, power, and exponentiation,
3) Data-dependent combination: It assigns specific kernel weights for each data instance, which can identify the local distributions in the data and learn proper kernel combination rules for different regions.

2) Kernel K-Means and Spectral Clustering: Kernel k-means [82] and spectral clustering [83] are two kernel-based clustering methods for optimizing the intracluster variance. Let $\phi(\cdot) : \boldsymbol{x} \in \mathcal{X} \rightarrow \boldsymbol{H}$ be a feature mapping which maps $\boldsymbol{x}$ onto an RKHS $\boldsymbol{H}$. The kernel k-means method is formulated as the following optimization problem:

$$\begin{cases} \min\limits_{\boldsymbol{H}} \sum_{i=1}^{N} \sum_{k=1}^{K} H_{ik} \|\phi(\boldsymbol{x}_i) - \boldsymbol{\mu}_k\|_2{}^2 \\ \text{s.t.} \quad \sum_{k=1}^{K} H_{ik} = 1 \end{cases} \tag{24}$$

where $\boldsymbol{H} \in \{0, 1\}^{N \times K}$ is the cluster indicator matrix (also known as cluster assignment matrix), $n_k = \sum_{i=1}^{N} H_{ik}$ and $\boldsymbol{\mu}_k = \frac{1}{n_k} \sum_{i=1}^{N} H_{ik} \phi(\boldsymbol{x}_i)$ are the number of points in the kth cluster and the centroid of the kth cluster. With a kernel matrix $\boldsymbol{K}$ whose (i, j)th entry is $K_{ij} = \phi(\boldsymbol{x}_i)^{\mathrm{T}} \phi(\boldsymbol{x}_j)$, $\boldsymbol{L} = \mathrm{diag}([n_1^{-1}, n_2^{-1}, \ldots, n_K^{-1}])$ and $\mathbf{1}_l \in \mathbb{R}^l$, a column vector of all ones, (24) can be equivalently rewritten as the following matrix-vector form:

$$\begin{cases} \min\limits_{\boldsymbol{H}} \quad tr(\boldsymbol{K}) - tr(\boldsymbol{L}^{\frac{1}{2}} \boldsymbol{H}^{\mathrm{T}} \boldsymbol{K} \boldsymbol{H} \boldsymbol{L}^{\frac{1}{2}}) \\ \text{s.t.} \quad \boldsymbol{H} \mathbf{1}_k = \mathbf{1}_N. \end{cases} \tag{25}$$

For the above kernel k-means matrix-factor form, the matrix $\boldsymbol{H}$ is binary, which makes the optimization problem difficult to solve. By relaxing the matrix $\boldsymbol{H}$ to take arbitrary real values, the above problem can be approximated. Specifically, by defining $\boldsymbol{U} = \boldsymbol{H} \boldsymbol{L}^{\frac{1}{2}}$ and letting $\boldsymbol{U}$ take real values, further considering

$\mathrm{Tr}(\boldsymbol{K})$ is constant, (25) will be relaxed to

$$\begin{cases}\max\limits_{\boldsymbol{U}} \quad tr(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{U}) \\ \text{s.t.} \quad \boldsymbol{U}^{\mathrm{T}}\boldsymbol{U} = \mathbf{1}_K.\end{cases} \tag{26}$$

The fact $\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H} = \boldsymbol{L}^{-1}$ leads to the orthogonality constraint on $\boldsymbol{U}$ which tells us that the optimal $\boldsymbol{U}$ can be obtained by the top K eigenvectors of the kernel matrix $\boldsymbol{K}$. Therefore, (26) can be considered as the generalized optimization formulation of spectral clustering. Note that (26) is equivalent to (8) if the kernel matrix $\boldsymbol{K}$ takes the normalized Gram matrix form.

3) Multikernel-Based MVC: Assume that there are m kernel matrices available, each of which corresponds to one view. To make a full use of all views, the weighted combination $\boldsymbol{K} = \sum_{v=1}^{m} w_v^p \boldsymbol{K}^{(v)}, w_v^p \geq 0, \sum_{v=1}^{m} w_v^p = 1, p \geq 1$ will be used in kernel k-means (26) and spectral clustering (8) to obtain the corresponding multiview kernel k-means and multiview spectral clustering [84]. Using the same nonlinear combination but specifically setting $p = 1$, Guo *et al.* [85] extended the spectral clustering to MVC with kernel alignment. Due to the potential redundance of the selected kernels, Liu *et al.* [86] introduced a matrix-induced regularization to reduce the redundancy and enhance the diversity of the selected kernels to attain the final goal of boosting the clustering performance. By replacing the original Euclidean norm metric in fuzzy c-means with a kernel-induced metric in the data space and adopting the weighted kernel combination, Zhang *et al.* [87] successfully extended the fuzzy c-means to MVC that is robust to noise and outliers. In the case when incomplete multiview dataset exists, by optimizing the alignment of the shared data instances, Shao *et al.* [88] collectively completes the kernel matrices of incomplete datasets. Liu *et al.* [89] integrated imputation and clustering into a unified learning procedure, but the computational and storage complexities of this method is quite high. To overcome these drawbacks, they proposed a late fusion method that effectively and efficiently conduct MVC with a three-step iterative procedure [90]. To overcome the cluster initialization problem associated with kernel k-means, Tzortzis *et al.* [91] proposed a global kernel k-means algorithm, a deterministic and incremental approach that adds one cluster in each stage, through a global search procedure consisting of several executions of kernel k-means from suitable initiations.

4) Others: Besides multikernel-based MVC, there are some other methods that use the direct combination of features to perform MVC like [66], [67], [92]. In [93], two-level weights: view wights and variable wights are assigned to the clustering algorithm for multiview data to identify the importance of the corresponding views and variables. Zhou *et al.* [94] learns an optimal neighborhood Laplacian matrix by searching the neighborhood of both the linear combination of the first-order and high-order base Laplacian matrices simultaneously to conduct multiview spectral clustering finally. To extend fuzzy clustering method to MVC, each view is weighted and the multiview versions of fuzzy c-means and fuzzy k-means are obtained in [95] and [96], respectively.

Direct combination-based MVC can adaptively tune the weights of each view, which is necessary and important when some views are of low quality. The consensus information among different views are not clear in the direct combination based MVC methods because there are no commonality shared between different views.

E. Combination After Projection (Mainly CCA-Based MVC)

For multiview data with all views with the same data type, like categorical or continuous, it is reasonable to directly combine them together. However, in real-world applications, the multiple representations may have different data types, and it is difficult to compare them directly. For instance, in bioinformatics, genetic information can be one view while clinical symptoms can be another view in the cluster analysis of patients [97]. Obviously, the information cannot be combined directly. Moreover, high dimension and noise are also difficult to handle. To solve the above problems, the last yet important combination way is introduced: combination after projection. The most commonly used technique is canonical correlation analysis (CCA) and the kernel version of CCA (KCCA).

1) CCA and KCCA: To better understand this style of view combination, CCA and KCCA are briefly introduced (refer to [98] for more detail). Given two datasets $\boldsymbol{S}_x = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{d_x \times N}$ and $\boldsymbol{S}_y = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N] \in \mathbb{R}^{d_y \times N}$ where each entry $\boldsymbol{x}$ or $\boldsymbol{y}$ has a zero mean, CCA aims to find a projection $\boldsymbol{w}_x \in \mathbb{R}^{d_x}$ for $\boldsymbol{x}$ and another projection $\boldsymbol{w}_y \in \mathbb{R}^{d_y}$ for $\boldsymbol{y}$ such that the correlation between the projection of $\boldsymbol{S}_x$ and $\boldsymbol{S}_y$ on $\boldsymbol{w}_x$ and $\boldsymbol{w}_y$ are maximized

$$\rho = \max_{\mathbf{w_x},\mathbf{w_y}} \frac{\mathbf{w_x}^{\mathrm{T}}\mathbf{C_{xy}}\mathbf{w_y}}{\sqrt{(\mathbf{w_x}^{\mathrm{T}}\mathbf{C_{xx}}\mathbf{w_x})(\mathbf{w_y}^{\mathrm{T}}\mathbf{C_{yy}}\mathbf{w_y})}} \tag{27}$$

where ρ is the correlation and $\mathbf{C_{xy}} = \mathbb{E}[\boldsymbol{x}\boldsymbol{y}^{\mathrm{T}}]$ denotes the covariance matrix of $\boldsymbol{x}$ and $\boldsymbol{y}$ with zero mean. Observing that ρ is not affected by scaling $\mathbf{w_x}$ or $\mathbf{w_y}$ either together or independently, CCA can be reformulated as

$$\begin{cases}\max\limits_{\mathbf{w_x},\mathbf{w_y}} \quad \mathbf{w_x}^{\mathrm{T}}\mathbf{C_{xy}}\mathbf{w_y} \\ \text{s.t.} \quad \mathbf{w_x}^{\mathrm{T}}\mathbf{C_{xx}}\mathbf{w_x} = \mathbf{1} \\ \qquad \mathbf{w_y}^{\mathrm{T}}\mathbf{C_{yy}}\mathbf{w_y} = \mathbf{1}\end{cases} \tag{28}$$

which can be solved using the method of Lagrange multiplier. The two Lagrange multipliers λ_x and λ_y are equal to each other, that is $\lambda_x = \lambda_y = \lambda$. If $\mathbf{C_{yy}}$ is invertible, $\mathbf{w_y}$ can be obtained as $\mathbf{w_y} = \frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{C}_{\mathbf{yy}}{}^{\mathbf{-1}}\boldsymbol{C}_{\mathbf{yx}}\boldsymbol{w}_{\mathbf{x}}$ and $\mathbf{C_{xy}}(\mathbf{C_{yy}})^{\mathbf{-1}}\mathbf{C_{yx}}\mathbf{w_x} = \lambda^{\mathbf{2}}\mathbf{C_{xx}}\mathbf{w_x}$. Hence, $\mathbf{w_x}$ can be obtained by solving an eigen problem. For different eigen values (from large to small), eigen vectors are obtained in a successive process.

The above canonical correlation problem can be transformed into a distance minimization problem. For ease of derivation, the successive formulation of the canonical correlation is replaced by the simultaneous formulation of the canonical correlation. Assume that the number of projections is p, the matrices $\mathbf{W_x}$ and $\mathbf{W_y}$ denote $(\mathbf{w_{x1}}, \mathbf{w_{x2}}, \ldots, \mathbf{w_{xp}})$ and $(\mathbf{w_{y1}}, \mathbf{w_{y2}}, \ldots, \mathbf{w_{yp}})$, respectively. The formulation that simultaneously identifies all the $\mathbf{w}$'s can be written as an optimization problem with p iteration steps

$$\begin{cases} \max\limits_{(\mathbf{w_{x1}},\mathbf{w_{x2}},\ldots,\mathbf{w_{xp}}),(\mathbf{w_{y1}},\mathbf{w_{y2}},\ldots,\mathbf{w_{yp}})} \sum\limits_{i=1}^{p} \mathbf{w_{xi}}^{\mathrm{T}}\mathbf{C_{xy}}\mathbf{w_{yi}} \\ \text{s.t.} \quad \mathbf{w_{xi}}^{\mathrm{T}}\mathbf{C_{xx}}\mathbf{w_{xj}} = \begin{cases} 1 & \text{if } i=j \\ 0 & \text{otherwise,} \end{cases} \\ \quad \mathbf{w_{yi}}^{\mathrm{T}}\mathbf{C_{yy}}\mathbf{w_{yj}} = \begin{cases} 1 & \text{if } i=j \\ 0 & \text{otherwise} \end{cases} \\ \quad i,j=1,2,\ldots,p \\ \quad \mathbf{w_{xi}}^{\mathrm{T}}\mathbf{C_{xy}}\mathbf{w_{yj}} = \mathbf{0} \\ \quad i,j=1,2,\ldots,p, j\neq i. \end{cases} \tag{29}$$

The matrix formulation to the optimization problem (29) is

$$\begin{cases} \max\limits_{\mathbf{W_x},\mathbf{W_y}} \mathrm{Tr}(\mathbf{W_x}^{\mathrm{T}}\mathbf{C_{xy}}\mathbf{W_y}) \\ \text{s.t.} \quad \mathbf{W_x}^{\mathrm{T}}\mathbf{C_{xx}}\mathbf{W_x} = \mathbf{I} \\ \quad \mathbf{W_y}^{\mathrm{T}}\mathbf{C_{yy}}\mathbf{W_y} = \mathbf{I} \\ \quad \mathbf{w_{xi}}^{\mathrm{T}}\mathbf{C_{xy}}\mathbf{w_{yj}} = \mathbf{0} \\ \quad \mathbf{w_{yi}}^{\mathrm{T}}\mathbf{C_{yx}}\mathbf{w_{xj}} = \mathbf{0} \\ \quad i,j=1,\ldots,p,\ j\neq i \end{cases} \tag{30}$$

where $\boldsymbol{I}$ is an identity matrix with size $p \times p$. Maximizing the objective function of (30) can be transformed into the equivalent form as follows:

$$\min_{\mathbf{W_x},\mathbf{W_y}} \left\|\mathbf{W_x}^{\mathrm{T}}\mathbf{S_x} - \mathbf{W_y}^{\mathrm{T}}\mathbf{S_y}\right\|_F \tag{31}$$

which is used widely in many works [36], [38], [99].

KCCA uses the "kernel trick" to maximize the correlation between two nonlinear projected variables. Analogous to (28), the optimization problem for KCCA is formulated as follows:

$$\begin{cases} \max\limits_{\mathbf{w_x},\mathbf{w_y}} \dfrac{\mathbf{w_x}^{\mathrm{T}}\mathbf{K_x}\mathbf{K_y}\mathbf{w_y}}{\sqrt{(\mathbf{w_x}^{\mathrm{T}}\mathbf{K_x^2}\mathbf{w_x})(\mathbf{w_y}^{\mathrm{T}}\mathbf{K_y^2}\mathbf{w_y})}} \\ \text{s.t.} \quad \mathbf{w_x}^{\mathrm{T}}\mathbf{K_x}\mathbf{w_x} = \mathbf{1} \\ \quad \mathbf{w_y}^{\mathrm{T}}\mathbf{K_y}\mathbf{w_y} = \mathbf{1}. \end{cases} \tag{32}$$

In contrast to the linear CCA that works by solving an eigen-decomposition of the covariance matrix, KCCA solves the following eigen-problem:

$$\begin{pmatrix} 0 & \boldsymbol{K}_x\boldsymbol{K}_y \\ \boldsymbol{K}_y\boldsymbol{K}_x & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{w}_x \\ \boldsymbol{w}_y \end{pmatrix} = \lambda \begin{pmatrix} \boldsymbol{K}_x^2 & 0 \\ 0 & \boldsymbol{K}_y^2 \end{pmatrix} \begin{pmatrix} \boldsymbol{w}_x \\ \boldsymbol{w}_y \end{pmatrix}. \tag{33}$$

2) CCA-Based MVC: Since cluster analysis in a high dimensional space is difficult, Chaudhuri *et al.* [100] first projects the data into a lower dimensional space via CCA and then clusters samples in the projected low dimensional space. Under the assumption that multiple views are uncorrelated given the cluster labels, it shows a weaker separation condition required to guarantee the algorithm successful. Blaschko *et al.* [101] projects the data onto the top directions obtained by the KCCA across different views and applies k-means to clustering the projected samples.

For the case of paired views with some class labels, CCA can still be applied by ignoring the class labels. However, the performance can be ineffective. To take an advantage of the class label information, Rasiwasia *et al.* [102] has proposed two solutions with CCA: mean-CCA and cluster-CCA. Consider two datasets each of which is divided into K different but corresponding classes or clusters. Given $\boldsymbol{S}_x = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_K\}$ and $\boldsymbol{S}_y = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_K\}$, where $\boldsymbol{x}_k = \{\boldsymbol{x}_1^k, \boldsymbol{x}_2^k, \ldots, \boldsymbol{x}_{|\boldsymbol{x}_k|}^k\}$ and $\boldsymbol{y}_k = \{\boldsymbol{y}_1^k, \boldsymbol{y}_2^k, \ldots, \boldsymbol{y}_{|\boldsymbol{y}_k|}^k\}$ are the data points in the kth cluster for the first and second views, respectively. The first solution is to establish correspondences between the mean cluster vectors in the two views. Given the cluster means $\boldsymbol{m}_x^k = \frac{1}{|\boldsymbol{x}_k|}\sum_{i=1}^{|\boldsymbol{x}_k|} x_i^k$ and $\boldsymbol{m}_y^k = \frac{1}{|\boldsymbol{y}_k|}\sum_{i=1}^{|\boldsymbol{y}_k|} y_i^k$, mean-CCA is formulated as

$$\rho = \max_{\boldsymbol{w}_x,\boldsymbol{w}_y} \frac{\boldsymbol{w}_x\boldsymbol{V}_{xy}\boldsymbol{w}_y}{\sqrt{(\mathbf{w_x}^{\mathrm{T}}\mathbf{V_{xx}}\mathbf{w_x})(\mathbf{w_y}^{\mathrm{T}}\mathbf{V_{yy}}\mathbf{w_y})}} \tag{34}$$

where $\boldsymbol{V}_{xy} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{m}_x^k \boldsymbol{m}_y^{k\mathrm{T}}$, $\boldsymbol{V}_{xx} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{m}_x^k \boldsymbol{m}_x^{k\mathrm{T}}$, and $\boldsymbol{V}_{yy} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{m}_y^k \boldsymbol{m}_y^{k\mathrm{T}}$. The second solution is to establish a one-to-one correspondence between all pairs of data points in a given cluster across the two views of datasets and then standard CCA is used to learn the projections.

For multiview data with at least one complete view (features for this view are available for all data points), Anusua *et al.* [103] borrowed the idea from Laplacian regularization to complete the incomplete kernel matrix and then applied KCCA to perform MVC. In another method for MVC, multiple data matrices $\boldsymbol{A}^{(v)} \in \mathbb{R}^{N \times K_v}, v = 1, 2, \ldots, K$ each of which corresponds to a view are obtained in an intermediate step and then a consensus data matrix should be learned to approximate each view's data matrix as much as possible. Due to the unsupervised property, however, the data matrices are often not directly comparable. Using the CCA formulation (31), Long *et al.* [104] projects one view's data matrix first before comparing with another view's data matrix.

The same idea can be used to tackle the incomplete view problem (i.e., there are no complete views). For instance, if there are only two views, the methods in [36] and [38] split data into the portion of data with both views and the portion of data with only one view, and then projects each view's data matrix so that it is close to the final indicator matrix. Multiview information is connected by the common indicator matrix corresponding to the projected data from both views. Wang *et al.* [105] provides a MVC method using an extreme learning machine that maps the normalized feature space onto a higher dimensional feature space.

Combination after projection-based MVC methods fit for scenarios where different views cannot be compared directly in original input space. Although the consensus information is used well in this group of MVC methods, the complementary information is not taken into account. This is contrary to direct combination-based MVC approaches. Thus, it is intriguing to explore whether it is possible to fuse this two groups of methods together to make full use of the consensus and complimentary information.

F. Other MVC Methods

In Section III-A, III-B, III-C, we have introduced three classes of similarity structure-based MVC methods. In addition, there are also some methods to share other similar structures to perform MVC. By sharing an indicator vector across views in a singular value decomposition of multiple data matrices, Sun *et al.* [97], [106], [107] extend the biclustering [108] method to the multiview settings. Wang *et al.* [109] chooses the Jaccard similarity to measure the cross-view clustering consistency and simultaneously considers the within-view clustering quality to cluster multiview data. By sharing a shared subspace's bidirectional sparsity, Fan *et al.* [110] proposed an MVC approach which can find an effective subspace dimension and deal with outliers simultaneously.

Apart from the above categorized methods, there are some other MVC methods. Different from exploiting the consensus information of multiview data, Cao *et al.* [111] utilizes a Hilbert Schmidt independence criterion as a diversity term to explore the complementarity of multiview information. It reduces the redundancy of multiview information to improve the clustering performance. Based on the idea of "minimizing disagreement" between clusters from each view, De Sa [112] proposes a two-view spectral clustering that creates a bipartite graph of the views. Zhou *et al.* [113] defines a mixture of Markov chains on similarity graph of each view and generalize spectral clustering to multiple views. In [114], a transition probability matrix is constructed from each single view, and all these transition probability matrices are used to recover a shared low-rank transition probability matrix as a crucial input to the standard Markov chain method for clustering. By fusing the similarity data from different views, Lange *et al.* [115] formulates an NMF problem and adopts an entropy-based mechanism to control the weights of multiview data. Zhu *et al.* [116] enforced a common affinity matrix to conduct MVC in one step. Liu *et al.* [117] chooses tensor to represent multiview data and then performs cluster analysis via tensor methods. Based on an assumption that the exemplar of a cluster in one view is always an exemplar of that cluster in the other views, Zhang *et al.* [118] proposed a multiview and multiexemplar fuzzy clustering method which has a theoretical guarantee on the performance improvement compared with single-view clustering counterpart. In paper [119], via cross-view graph diffusion, a unified graph for multiview data is learned to conduct final clustering.

IV. Relationships to Related Topics

As we mentioned previously, MVC is a learning paradigm for cluster analysis with multiview feature information. It is a basic task in machine learning and thus can be useful for various subsequent analyses. In machine learning and data mining fields, there are several closely related learning topics such as multiview representation learning, ensemble clustering, multitask clustering, and multiview supervised, and semisupervised learning. In the following, we will elaborate the relationships between MVC and a few other topics.

A. Relationship to Multiview Representation

Multiview representation [120] is the problem of learning a more comprehensive or meaningful representation from multiview data. According to [121], representation learning (also known as embedding learning or metric learning) is a way to take advantage of human ingenuity and prior knowledge to extract some useful but far-removed feature representation for the ultimate objective. Thus representation learning does not need to be unsupervised in nature. For instance, metric learning has mainly been studied from the supervised perspective, when class labels are present. Using the class labels, approaches usually form constraints, for example, pairwise or triplet-based constraints. Multiview representation can be considered as a more basic task than MVC, since multiview representation can be useful in broader purpose such as classification or clustering and so on. However, cluster analysis based on multiview representation may not be ideal because the creation of multiview representation is unaware of the final goal of clustering [122], [123].

In an archived survey article [120], multiview representation methods are categorized into mainly two classes: the shallow methods and the deep methods. The shallow methods are mainly based on CCA, which may correspond to Section III-E. For the deep methods, there exist a large number of works [124]–[130] on multiview representation. For multiview deep clustering, there are also many recent works including [131]–[136]. As mentioned above, the sequential way of first multiview representation and then clustering is a natural way to perform MVC, but the ultimate performance is usually not good because of the gap in the two steps. Therefore, how to integrate clustering and multiview representation learning into a simultaneous process is an intriguing direction up to date, especially for deep multiview representation. In addition, although many MVC methods sprung up in recent years, it still has large space to develop, especially compared with the development of multiview deep representation learning.

B. Relationship to Ensemble Clustering

Ensemble clustering [137] (also named consensus clustering or aggregation of clustering) is made up of two steps: generation step and consensus step. Generation step is used to generate several sets of clusterings of the dataset while consensus step is used to combine those sets of clusterings to obtain a consensus clustering. MVC does not need to obtain the final clustering result based on the sets of clusterings from original datasets, the final clustering result can be directly obtained from original datasets. This is the big difference between ensemble clustering and MVC. Certainly, MVC can also conduct clustering from generation step and consensus step when original datasets are multiview and those clusterings obtained in generation step are gotten from each view of the original datasets. Thus if ensemble clustering is applied to clustering with multiple views of data, it becomes a type of MVC method. In this sense, MVC and ensemble clustering have some overlaps. Therefore, some of the ensemble clustering techniques, e.g., [138]–[143] can be applied to MVC. This works in [65], [144], and [145] are representative multiview ensemble clustering methods. Although the idea of ensemble clustering is simple, it has gained good performance in real-world application. Especially in many kaggle competitions held recently, ensemble mechanism is quite popular and performed well. Thus, more exploration in this direction can be done in future. However, it should be noted that MVC does not need to have clear separate generation and consensus steps. More works connect MVC and ensemble clustering can be investigated further.

C. Relationship to Multitask Clustering

Multitask clustering improves the clustering performance of each task by transferring knowledge among the related tasks, such as in [146]–[151]. Between MVC and multitask clustering, there are two big differences. The first one is that multitask cares about the performance of each task, while MVC just cares about a final consensus clustering performance not each view. The second one is that one works on multiple tasks while the other one works on multiple views. Multiple tasks can be based on multiple datasets, while multiple views have to be based on the same dataset (but just different views of this one dataset). If each task corresponds to clustering in a specific view of the same dataset, multiple clustering results will be obtained, and then ensemble clustering methods may be employed to fuse these clustering results. Therefore, multitask clustering, potentially combined with ensemble clustering, can implement MVC in the scenario where each task corresponds to each view of the same data. In addition, multitask clustering and MVC can be conducted simultaneously to improve the clustering performance [152]–[154]. However, we should still distinguish the differences between them, since multi-task clustering cares about the clustering performance of each task. Even if each task corresponds to each view of the dataset, multitask clustering is still not equivalent to MVC. When multitask clustering combines ensemble clustering further, it will achieve MVC. Thus some techniques and ideas in multitask clustering and ensemble clustering can be helpful for MVC

D. Relationship to Multiview Supervised and Semisupervised Learning

The difference between MVC and multiview supervised, semisupervised learning lies in whether to use the label of the data. MVC does not use any label of the data while multiview supervised learning [4], [155] uses the labeled data to learn classifiers (or other inference models), multiview semisupervised learning [3], [4] can learn classifiers with both the labeled and unlabeled data.

The commonality between them lies in the way to combine multiple views. Many widely recognized techniques for combining views in the supervised or semisupervised settings, e.g., cotraining [23], [156], coregularization [157], [158], margin consistency [159], [160] can lend a hand to MVC if there is a mechanism to estimate the initial labels. Thus, the key point to conduct MVC with some techniques in multiview supervised or semisupervised learning is how to estimate the initial labels or get some pseudo labels to play the role of labels in multiview supervised learning or multiview semisupervised learning.

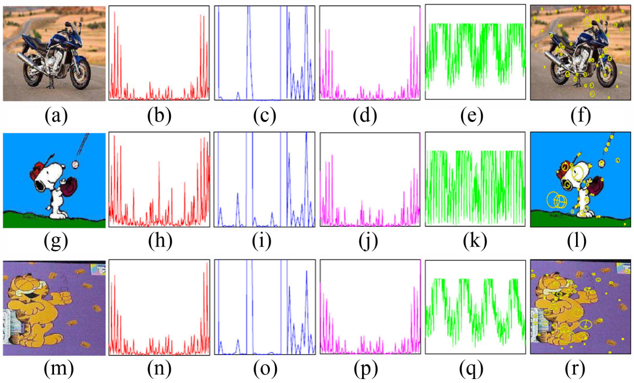


Fig. 2. Five views (CENTRIST, ColorMoment, LBP, HOG, and SIFT) on three sample images from Caltech101.

V. Applications

MVC has been successfully applied to various applications including computer vision, natural language processing, social multimedia, bioinformatics and health informatics, and so on.

A. Computer Vision

MVC has been widely used in image categorization [30], [32], [51], [111], [139], [161], [162] and motion segmentation tasks [45], [163]. Typically, several feature types, e.g., CENTRIST [164], ColorMoment [165], HOG [166], LBP [167], and SIFT [168] can be extracted from the images (see Fig. 2 [51]) prior to cluster analysis. Yin *et al.* [30] proposed a pairwise sparse subspace representation for multiview image clustering, which harnesses the prior information and maximizes the correlation between the representations of different views. Wang *et al.* [32] enforced between-view agreement in an iterative way to perform multiview spectral clustering on images. Gao *et al.* [51] assumed a common low dimensional subspace representation for different views to reach the goal of MVC in computer vision applications. Cao *et al.* [111] adopted Hilbert Schmidt independence criterion as a diversity term to exploit the complementary information of different views and performed well on both image and video face clustering tasks. Jin *et al.* [161] utilized the CCA to perform multiview image clustering for large-scale annotated image collections.

Ozay *et al.* [139] used consensus clustering to fuse image segmentations. Chi *et al.* [169] conducted MVC for web image retrieval ranking. Méndez *et al.* [162] adopted the ensemble way to perform MVC for MRI image segmentation. NMF was adopted in [45] to perform MVC for motion segmentation. Djelouah *et al.* [163] addressed the motion segmentation problem by propagating segmentation coherence information in both space and time. Xin *et al.* [89] successfully applied MVC for person reidentification. Tao *et al.* [170] applied their proposed multiview subspace clustering methods to background subtraction from multiview videos.

Fig. 3. Some photographs from two social events: concerts (top row) and NBA game (bottom row).

B. *Natural Language Processing*

In natural language processing, text documents can be obtained in multiple languages. It is natural to use MVC to conduct document categorization [16], [17], [46], [51], [171]–[173] with each language as one view. Employing the cotraining and coregularization ideas, Kumar *et al.* [16], [17] proposed cotraining MVC and coregularization MVC, respectively. The performance comparison on multilingual data demonstrates the superiority of these two methods over single-view clustering. Liu *et al.* [46] extended NMF to multiview settings for clustering multilingual documents. Kim *et al.* [171] obtained the clustering results from each view and then constructed a consistent data grouping by voting. Jiang *et al.* [172] proposed a collaborative PLSA method that combines individual PLSA models in different views and imports a regularizer to force the clustering results in an agreement across different views. Hussain *et al.* [174] utilized an ensemble way to perform MVC on documents. Zhang *et al.* [43] adopted an MVC method with graph regularization to improve object recognition.

C. *Social Multimedia*

Currently, with the fast development of social multimedia, how to make full use of large quantities of social multimedia data is a challenging problem, especially when matching them to the "real-world concepts" such as the "social event detection." Fig. 3 shows two such events: a concert and an NBA game. The pictures showed there form just one view, and other textural features such as tags and titles form the other view. Such a social event detection problem is a typical MVC problem. Petkos *et al.* [175] adopted a multiview spectral clustering method to detect the social event and additionally utilized some known supervisory signals (the known clustering labels). Samangooei *et al.* [176] performed feature selection first before constructing the similarity matrix and applied a density-based clustering to the fused similarity matrix. Petkos *et al.* [177] proposed a graph-based MVC to cluster the data from social multimedia. MVC has also been applied to grouping multimedia collections [178], news stories [179], and social web videos [180].

D. *Bioinformatics and Health Informatics*

In order to identify genetic variants underlying the risk for substance dependence, Sun *et al.* [97], [106], [107] designed three multiview coclustering methods to refine diagnostic classification to better inform genetic association analyses. Chao *et al.* [181] extended the method in [97] to handle missing values that might appear in each view of the data, and used the method to analyze heroin treatment outcomes. The three views of data for heroin-dependent patients are demonstrated in Fig. 4. Yu *et al.* [182], [183] designed a multikernel combination to fuse different views of information and showed superior performance on disease datasets. In [184], an MVC based on the Grassmann manifold was proposed to deal with gene detection for complex diseases. MVC is also applied to analyze athlete's physical fitness test [185]. Recently, Rappoport and Shamir [186] provided a review on MVC on biomedical omics datasets.

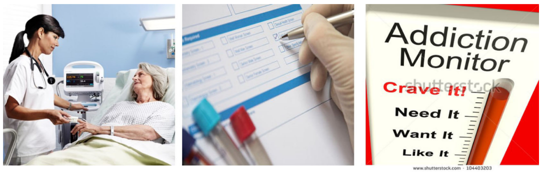


Fig. 4. Three views from health informatics: vital sign (left), urine drug screen (middle), and craving measure (right)).

VI. Datasets and Experiments

To further analyze the advantages and disadvantages of each group of MVC algorithms, we provide several commonly used MVC datasets and conduct empirical evaluation to measure how each group of MVC algorithms performs.

A. *Datasets*

Six benchmark multiview datasets are adopted, and the statistics of these datasets are summarized in Table I.

3 Sources[1] is a news article dataset. These articles are collected from three news sources: BBC, Reuters, and Guardians. In original datasets, there are 948 articles that are reported by at least one of the three sources. Herein, 169 of these articles are included, and the bag-of-word representation is adopted to represent the articles. These 169 articles are dominated by one of the six topical classes: business, entertainment, health, politics, sport, technology.

Reuters [187] includes documents in five languages: English, French, German, Spanish, and Italian. These five language versions constructed five views of these documents, and bag of words is used to represent the features in each view. These documents belong to one of the six categories. 100 documents are randomly sampled from each category to construct a dataset of 600 documents.

Handwritten Digits is available from the UCI repository.[2] It has 2000 examples of handwritten digits (0-9) extracted from Dutch utility maps. There are 200 examples in each class, each represented with six feature sets. Following experiments

[1] [Online]. Available: http://mlg.ucd.ie/datasets/3sources.html

[2] [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Multiple+Features

TABLE I
STATISTICS OF THE MULTIVIEW DATASETS

Dataset	# **Samples**	# **Views**	# **Clusters**	# **Features in each view**	Is entry non-negative
3 Sources	169	3	6	3560, 3631, 3068	No
Reuters	600	5	6	21526, 24892, 34121, 15487, 11539	Yes
Handwritten Digits	2000	3	10	76, 216, 64	Yes
COIL20	1440	3	20	30, 19, 30	Yes
YALE	165	3	15	4096, 3304, 6750	No
Movies	617	2	17	1878, 1398	No

in [188], three feature sets: 76 Fourier coefficients of the character shapes, 216 profile correlations and 64 Karhunen-Love Coefficiens are adopted.

COIL20[3] consists of 1440 images belonging to 20 classes. Three views are represented by 30 isometric projection (ISO), 19 linear discriminant analysis (LDA), and 30 neighborhood preserving embedding (NPE), respectively.

YALE [189] consists of 165 images from 15 subjects, which has 11 images per subject and corresponds to different facial expressions or configurations. Each image is expressed by three heterogeneous feature sets with dimensions of 4096, 3304, and 6750.

Movies[4] includes 617 movies belonging to 17 genres. Each movie is described by two views: 1878 keywords and 1398 actors.

B. Compared Methods and Parameter Settings

In the experiment, six representative MVC algorithms corresponding to each group of MVC approaches are used to compare. To explore how deep MVC algorithms perform, one deep algorithm is chosen to compare. These algorithms are multiview mixture-of-multinomials EM (MVMMEM) [10], co-regularization multiview spectral clustering (Co-Reg) [17], multiview low-rank sparse subspace clustering (MVLRSSC) [31], multiview clustering via joint NMF (MultiNMF) [46], kernel-based weighted multiview clustering (MVKKM) [84], MVC via CCA (MVCCA) [100], and MVC via deep matrix factorization (DeepNMF) [39].

As for the parameter settings, we try our best to set it according to that in their original papers. For MVMMEM, the number of rounds are selected from $\{5, 10, \ldots, 100\}$. For Co-Reg, parameter α is selected from 0.01 to 0.05 with step 0.01. For MVLRSSC, we tune the parameters $\beta_1, \beta_2, \lambda^{(v)}$ according to [31]. For MultiNMF, λ_v is set to 0.01 for all views. According to MVKKM [84], good performance can be obtained with $p = 1.5$, we adopted this setting. For MVCCA, we kept vectors with canonical correlation bigger than 0.01. For DeepNMF, two layers with layer size [100, 50] are designed for all the datasets except COIL20 ([18,9]), parameter $\beta = 0.1$, $\gamma = 0.5$.

To conduct a comprehensive evaluation, all the approaches are compared with six evaluation metrics: normalized mutual information (NMI), accuracy (ACC), adjusted rand index (ARI), F-score, Precision, and Recall. For all these metrics, the higher value indicates better clustering performance. All the algorithms are run 20 times and the mean and standard deviation of each metric is reported.

C. Experiment Results

The results are shown in Table II. On datasets 3 Sources, Reuters and Movies, MVLRSSC performs best. On datasets Handwritten Digits, COIL20, MVKKM outperforms all the other algorithms. On dataset YALE, DeepNMF obtained the best performance. These results are almost consistent on six different metrics except NMI on dataset Handwritten Digits and Recall on dataset Movies. On dataset Handwritten Digits, the performance in NMI for MVKKM and MVLRSSC are very close. On dataset Movies, although the performance in Recall for MVKKM is significantly better than that for MVLRSSC, the precision and comprehensive metric F1 score for MVKKM are worse.

Datasets 3 Sources, Reuters, and Movies consist of text information. Due to special topic properties, low rank and sparsity are important when conducting MVC, thus the algorithm MVLRSSC that take low rank and sparsity into account performs well. Datasets Handwritten Digits, COIL20 and YALE are datasets of images, maybe it is necessary to use nonlinear or deep structure to learn the abstract or meaningful clusters, thus MVKKM and DeepNMF perform better on these datasets. In addition, different views contribute different in final clustering, thus different weights should be given them to promote the performance, thus MVKKM can be a good choice. From the results, we can find that MultiNMF just applied to datasets 3 Sources, YALE, and Movies, that is because MultiNMF can apply to the scenario where all entries of the datasets are nonnegative. This is one limitation of MultiNMF. Results in Table II also shows that algorithms MVMMEM and MVCCA just apply to dataset Movies, that is because they are suitable for the datasets with only two views, thus this is the limitation of these algorithms. MVMMEM performs worse than MVLRSSC, as well as Co-Reg, better than MultiNMF, MVKKM, MVCCA, and DeepNMF. It can be seen that generative algorithms has the potential to be comparable with state of the art discriminative algorithms. Although MultiNMF has nonnegative property, it does not perform well compared with other group of algorithms on the above datasets. This maybe because compared with nonnegative property, other properties like low rank, sparsity, weights difference are more important for performance improvement.

Based on the results in Table II, we can find that multiview subspace clustering group, multikernel MVC, and deep MVC algorithms perform well. Spectral clustering-based MVC, NMF-based MVC, and MVCCA perform worse than the above

[3] [Online]. Available: http://www.cs.columbia.edu/CAVE/software/softlib/coil20.php

[4] [Online]. Available: http://lig-membres.imag.fr/grimal/data.html

TABLE II
PERFORMANCE OF SEVEN ALGORITHMS ON SIX MULTIVIEW DATASETS. THE MEAN AND STANDARD DEVIATION OF 20 RUNS OF THESE ALGORITHMS ARE REPORTED

Dataset	**Method**	**ACC**	**F-score**	**Precision**	**Recall**	**NMI**	**ARI**
3 Sources	MVMMEM	/	/	/	/	/	/
	Co-Reg	0.5434 (0.0097)	0.4648 (0.0100)	0.4985 (0.0124)	0.4373 (0.0100)	0.4894 (0.0086)	0.3161 (0.0131)
	MVLRSSC	**0.6730 (0.0089)**	**0.6350 (0.0101)**	**0.6770 (0.0086)**	**0.6005 (0.0145)**	**0.5855 (0.0054)**	**0.5355 (0.0018)**
	MultiNMF	0.4107 (0.0079)	0.3244 (0.0045)	0.2631 (0.0051)	0.4227 (0.0030)	0.3426 (0.0061)	0.0531 (0.0084)
	MVKKM	0.3550 (0)	0.3621 (0)	0.2298 (0)	0.8430 (0)	0.1131 (0)	-0.0064 (0)
	MVCCA	/	/	/	/	/	/
	DeepNMF	0.6509 (0.0076)	0.5079 (0.0047)	0.5645 (0.0060)	0.4614 (0.0065)	0.4892 (0.0107)	0.3780 (0.0056)
Reuters	MVMMEM	/	/	/	/	/	/
	Co-Reg	0.4800 (0.0068)	0.3699 (0.0030)	0.3386 (0.0041)	0.4091 (0.0047)	0.3000 (0.0038)	0.2308 (0.0041)
	MVLRSSC	**0.5280 (0.0069)**	**0.4174 (0.0030)**	**0.3641 (0.0062)**	**0.4931 (0.0052)**	**0.3782 (0.0032)**	**0.2802 (0.0055)**
	MultiNMF	—	—	—	—	—	—
	MVKKM	0.2283 (0)	0.2870 (0)	0.1737 (0)	0.8261 (0)	0.1909 (0)	0.0191 (0)
	MVCCA	/	/	/	/	/	/
	DeepNMF	0.2977 (0.0020)	0.2217 (0.0018)	0.2138 (0.0034)	0.2303 (0.0049)	0.1053 (0.0014)	0.0607 (0.0032)
Handwritten Digits	MVMMEM	/	/	/	/	/	/
	Co-Reg	0.7571 (0.0064)	0.6842 (0.0087)	0.6656 (0.0087)	0.7042 (0.0087)	0.7298 (0.0076)	0.6481 (0.0097)
	MVLRSSC	0.7699 (0.390)	0.7288 (0.0534)	0.6970 (0.0615)	0.7642 (0.0462)	**0.7799 (0.0333)**	0.6971 (0.0601)
	MultiNMF	—	—	—	—	—	—
	MVKKM	**0.8650 (0)**	**0.7530 (0)**	**0.7411 (0)**	**0.7653 (0)**	0.7740 (0)	**0.7252 (0)**
	MVCCA	/	/	/	/	/	/
	DeepNMF	0.7738 (0.0009)	0.7456 (0.0019)	0.7042 (0.0016)	0.7921 (0.0022)	0.7961 (0.0022)	0.7156 (0.0021)
COIL20	MVMMEM	/	/	/	/	/	/
	Co-Reg	0.9591 (0.0146)	0.9643 (0.0129)	0.9436 (0.0199)	0.9871 (0.0050)	0.9899 (0.0037)	0.9623 (0.0136)
	MVLRSSC	0.9767 (0.0078)	0.9799 (0.0065)	0.9686 (0.0099)	0.9922 (0.0028)	0.9943 (0.0019)	0.9788 (0.0068)
	MultiNMF	—	—	—	—	—	—
	MVKKM	**1 (0)**	**1 (0)**	**1 (0)**	**1 (0)**	**1 (0)**	**1 (0)**
	MVCCA	/	/	/	/	/	/
	DeepNMF	0.3857 (0.0050)	0.2688 (0.0121)	0.2133 (0.0163)	0.3651 (0.0156)	0.5144 (0.0029)	0.2202 (0.0142)
YALE	MVMMEM	/	/	/	/	/	/
	Co-Reg	0.5913 (0.0140)	0.4599 (0.0150)	0.4376 (0.0149)	0.4851 (0.0155)	0.6418 (0.0113)	0.4229 (0.0161)
	MVLRSSC	0.5677 (0.0103)	0.4141 (0.0090)	0.3939 (0.0090)	0.4368 (0.0093)	0.6088 (0.0082)	0.3739 (0.0097)
	MultiNMF	0.5188 (0.0054)	0.3652 (0.0149)	0.3470 (0.0111)	0.3855 (0.0201)	0.5602 (0.0203)	0.3217 (0.0154)
	MVKKM	0.6364 (0)	0.4732 (0)	0.4064 (0)	0.5661 (0)	0.6855 (0)	0.4329 (0)
	MVCCA	/	/	/	/	/	/
	DeepNMF	**0.7446(0.0191)**	**0.5664 (0.0138)**	**0.5522 (0.0168)**	**0.5815 (0.0107)**	**0.7312 (0.0103)**	**0.5375 (0.0149)**
Movies	MVMMEM	0.2592 (0.0163)	0.1538 (0.0134)	0.1460 (0.0135)	0.1627 (0.0150)	0.2529 (0.0144)	0.0955 (0.0144)
	Co-Reg	0.2615 (0.0033)	0.1517 (0.0023)	0.1402 (0.0022)	0.1657 (0.0031)	0.2657 (0.0031)	0.0916 (0.0024)
	MVLRSSC	**0.3180 (0.0053)**	**0.1933 (0.0034)**	**0.1913 (0.0032)**	0.1955 (0.0035)	**0.3184 (0.0029)**	**0.1403 (0.0036)**
	MultiNMF	0.1900 (0.0130)	0.1220 (0.0055)	0.0722 (0.0038)	0.3966 (0.0412)	0.2073 (0.0115)	0.0208 (0.0068)
	MVKKM	0.1005 (0)	0.1146 (0)	0.0611 (0)	**0.9241 (0)**	0.0670 (0)	0 (0)
	MVCCA	0.1295 (0.0036)	0.0607 (0.0013)	0.06175 (0.0014)	0.0596 (0.0014)	0.0854 (0.0058)	0.0007 (0.0015)
	DeepNMF	0.1847 (0.0033)	0.0945 (0.0028)	0.0911 (0.0035)	0.0981 (0.0026)	0.1626 (0.0027)	0.0332 (0.0035)

"—" indicates that this dataset has negative entries, thus MultiNMF cannot apply. "/" indicates this algorithm only applies to two-view case directly but this dataset has more than two views. the best results among seven MVC algorithms on each dataset is shown in bold font.

algorithms on the six commonly used datasets. Generative MVC performs better than many discriminative ones, thus it is worth attracting more attention in future. In this experiment, we focused on clustering performance, a more comprehensive study including time cost factor, and more advanced MVC algorithms that are worth further exploration.

VII. OPEN PROBLEMS

We have identified several problems that are still underexplored in the current body of MVC literature. We discuss these problems in this section.

A. Large Scale Problem (Size and Dimension)

In modern life, large quantities of data are generated every day. For instance, several million posts are shared per minute in Facebook, which include multiple data forms (views): videos, images, and texts. At the same time, a large amount of news are reported in different languages, which can also be considered as multiview data with each language as one view. However, most of the existing MVC methods can only deal with small datasets. It is important to extend these methods to large scale applications. For instance, it is difficult for the existing multiview spectral clustering based methods to work on datasets of massive samples due to the expensive computation of graph construction and eigen-decomposition. Although some previous works such as [190]–[193] attempted to accelerate the spectral clustering method to scale with big data, it is intriguing to extend them effectively to the multiview settings. Recently, Zhang *et al.* [194] proposed an interesting idea to solve large scale problem by encoding multiview image data into a compact common binary code space and then conduct binary clustering.

Another type of big data has high dimensionality. There is a large quantity of single-view clustering methods [195] to deal with this kind of problem, However, there is still one special class of such problem tough to deal with. For instance, in bioinformatics, each person has millions of genetic variants as genetic features where, compared with the problem dimension, the number of samples is low. Using genetic features in a clinical analysis with another view of clinical phenotypes often forms a multiview analytics problem. How to deal with such a clustering problems is tough due to the over-fitting problem.

Although feature selection [196], [197] or feature dimension reduction [198] like PCA is commonly used to alleviate this problem in single-view settings, there are no convincing methods up to now, especially because deep learning cannot cope with it due to the properties: small size and high feature dimension. It may recall new theory to appear to handle this problem.

B. Incomplete Views or Missing Value

MVC has been successfully applied to many applications as shown in Section V. However, there is an underlying problem hidden behind: what if one or more views are incomplete? This is very common in real-world applications. For example, in multilingual documents, many documents may have only one or two language versions; in social multimedia, some sample may miss visual or audio information due to sensor failure; in health informatics, some patients may not take certain lab tests to cause missing views or missing values. Some data entries may be missing at random while others are nonrandom [181]. Simply replacing the missing entries with zero or mean values [199] is a common way to deal with the missing value problem, and multiple imputation [200] is also a popular method in statistical field. The missing entries can be generated by the recently popular generative adversarial networks [201]. However, without considering the differences of random and nonrandom effects in missing data, the clustering performance is not ideal [181].

Up to now, there have already been several multiview works [36]–[38], [61], [63], [88], [103], [202] that attempted to solve the incomplete view problem. Two methods in [61] and [63] introduced a weight matrix $M_{i,j}$ to indicate whether the ith instance present in the jth view. For the two-view case, the method in [36] reorganized the multiview data to include three parts: samples with both views, samples only having view 1, and samples only having view 2 and then analyzed them to handle missing entries. Assuming that there is at least one complete view, Trivedi *et al.* [103] used the graph Laplacian to complete the kernel matrix with missing values based on the kernel matrix computed from the complete view. Shao [88] borrowed the same idea to deal with multiview setting. Instead imputing kernel matrix, Liu [203] imputed each base matrix generated by incomplete views with a learned consensus clustering matrix. It is noted that all these methods deal with incomplete views or missing value with some constraints, but they do not aim to deal with the situation with arbitrarily missing values in any of the views. In other words, this situation is that all views have missing values and the samples just miss a few features in a view. Obviously, the above methods have significant limitations that cannot make full use of the available multiview incomplete information. In addition, all existing methods do not take into consideration the difference between random and nonrandom missing patterns. Therefore, it is worth exploring how to use the mixed types of data in multiview analysis.

C. Initialization and Local Minima

For MVC methods based on k-means, the initial clusters are very important and different initializations may lead to different clustering results. It is still challenging to select the initial clusters effectively in MVC and even in single-view clustering settings.

Most NMF-based methods rely on nonconvex optimization formulations, and thus are prone to the local optimum problem, especially when missing values and outliers exist. By enforcing a consistent clustering result in different view, Zhao *et al.* [173] formulated a jointly convex optimization formulation and additionally using some side information. Self-paced learning [204] is a possible solution, and Xu *et al.* [64] applied it to MVC to alleviate the local minimum problem.

The generative convex clustering method [8] is an interesting approach to avoid the local minimum problem. In [12], a multiview version of the method in [8] is proposed and shows good performance. This kind of generative methods may be another good direction worth further exploring.

D. Deep Learning

Recently, deep learning has demonstrated outstanding performance in many applications such as speech recognition, image segmentation, object detection, and so on. However, compared with the fast growth of supervised deep learning and unsupervised deep representation learning, deep clustering still has a lot of room to develop, especially multiview deep clustering. A natural way to conduct deep clustering or multiview deep clustering is to conduct clustering on the representation obtained from single-view representation learning or multiview representation learning. In fact, there should be many advanced ways to explore how to conduct multiview deep clustering.

Recently, there indeed appeared a number of deep clustering works. For example, the works in [205]–[207] borrowed the supervised deep learning idea to perform supervised clustering. In fact, they can be considered as performing semisupervised learning. So far, there are already several truly deep clustering works [131], [132], [208]. Tian *et al.* [131] proposed a deep clustering algorithm that is based on spectral clustering, but replaced eigenvalue decomposition by a deep auto-encoder. Xie *et al.* [132] proposed a clustering approach using deep neural network which can learn representation and perform clustering simultaneously. It is interesting to explore how to extend them to multiview scenarios.

Besides deep clustering works, there also exist some MVC methods. Huang *et al.* [208] proposed to use multiple layer matrix factorization and shared the same representation matrix across different views to conduct MVC. Experimental results demonstrates the superiority of this deep learning methods to multi-view shallow clustering methods like cotraining clustering, coregularization clustering, and multiview k-means clustering. By using auto-encoder architecture, Zhu *et al.* [135] designed a diverse net and universal net to make full use of the complementary and consensus information among multiple views to implement MVC. To let clustering label to guide the representation learning, Sun *et al.* [136] proposed another deep subspace MVC method in a semisupervised way. Li *et al.* [133] presented a deep MVC approach borrowing the idea and architecture of generative discriminative network (GAN). Inspired by the great success obtained by using attention mechanism in deep learning fields, Zhou *et al.* [134] explored an MVC method by combing GAN and attention mechanisms, and experiments support its effectiveness.

Compared with traditional multiview shallow clustering methods, the aforementioned multiview deep clustering methods demonstrated better performance due to several reasons. First, deep networks adopted in multiview deep clustering methods have better expression ability, maybe it can discover the more real structure of the multiview data. Second, part of them adopt end-to-end multiview deep clustering way. The representation obtained amid can reflect multiview data comprehensively and, at the same time, serve to the final goal clustering well. However, there are still large space to explore and develop in this direction. First, there are more and more novel deep learning architectures; how to extend them to multiview scenarios needs more investigation. Second, although some end-to-end multiview deep clustering methods appeared, more such methods are expected, since multiview deep representation learning developed more sufficiently than multiview deep clustering, and it is simple and natural to run clustering algorithm on the representation obtained from multiview deep representation learning. However, the separate process to deal with multiview deep clustering has its limitations, like being unaware of clustering goal in representation learning. Third, deep learning techniques has its special properties; more ways to combine multiple views can be designed to serve to multiview deep clustering. Fourth, some theoretical investigation should be conducted to unfold how and why multiview deep clustering shows better performance than traditional shallow methods.

E. Mixed Data Types

Multiview data may not necessarily just contain numerical or categorical features. They can also have other types such as symbolic, ordinal, etc. These different types can appear simultaneously in the same view, or in different views. How to integrate different types of data to perform MVC is worthy of careful investigation. Converting all of them to categorical type is a straightforward solution. However, much information will be lost during such processing. For example, the difference of the continuous values categorized into the same category is ignored. The work in [209] proposed a solution to mixed data type problem with vine copulas. It is worth more exploring to make full use of the information within mixed data types in MVC settings.

F. Multiple Solutions

Most of the existing MVC, even single-view clustering, algorithms only output a single clustering solution. However, in real-world applications, data can often be grouped in many different ways, and all these solutions are reasonable and interesting from different perspectives. For example, it is both reasonable to group the fruits apple, banana, and grape according to the fruit type or color. Until now, to the best of our knowledge, there are very few works along this direction [210]–[212]. Cui *et al.* [210] proposed to partition multiview data by projecting the data to a space that is orthogonal to the current solution so that multiple nonredundant solutions were obtained. In another work [211], Hilbert–Schmidt independence criterion was adopted to measure the dependence across different views and then one clustering solution was found in each view. Chang *et al.* [212] automatically learned multiple expert views and the clustering structure corresponding to each view in a Bayesian probabilistic model. MVC algorithms that can produce multiple solutions should attract more attention in the future.

VIII. Conclusion

To sort out existing MVC methods, we proposed a novel taxonomy to introduce them. Similar to machine learning method categorizations, we split MVC methods into two classes: generative methods and discriminative methods. Based on the way to combine multiple views, discriminative methods are further split into five main classes, the first three of which have a commonality: sharing certain structures across the views. The fourth one uses direct combinations of the views, while the fifth one employs view combinations after projections. Compared with discriminative methods, generative methods have developed far less sufficiently. Although it has inherent limitation, it can deal with missing data and get global optima easily, thus it calls for more attention. To better understand MVC, we elaborate on the relationships between MVC and several closely related learning topics. We have also introduced several real-world applications of MVC and, most importantly, we conducted a comprehensive experimental study on representative MVC algorithms of each group to further analyze the advantages and disadvantages of them, and finally pointed out some interesting and challenging directions to guide researchers to advance in future.

Acknowledgment

The authors would like to thank S. Wang for running the experiments.

References

[1] P. Berkhin, “Survey of clustering data mining techniques,” Yahoo, Sunnyvale, CA, USA, Tech. Rep., 2002, doi: 10.1007/3-540-28349-8_2.

[2] J. G. Saxe, *The Blind Men and the Elephant*. Hong Kong: Enrich Spot Limited, 2016.

[3] C. Xu, D. Tao, and C. Xu, “A survey on multi-view learning,” 2013, *arXiv:1304.5634*.

[4] *S. Sun*, “A survey of multi-view machine learning,” *Neural Comput. Appl.*, vol. 23, no. 7/8, pp. 2031–2038, 2014.

[5] J. Zhao, X. Xie, X. Xu, and S. Sun, “Multi-view learning overview: Recent progress and new challenges,” *Inf. Fusion*, vol. 38, pp. 43–54, 2017.

[6] Y. Yang and H. Wang, “Multi-view clustering: A survey,” *Big Data Mining Analytics*, vol. 1, no. 2, pp. 83–107, 2018.

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. Roy. Statist. Soc. Ser. B*, vol. 39, pp. 1–38, 1977.

[8] D. Lashkari and P. Golland, “Convex clustering with exemplar-based models,” in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2008, pp. 825–832.

[9] A. Banerjee, S. Merugu, I. S. Dhillin, and J. Ghosh, “Clustering with Bregman divergences,” *J. Mach. Learn. Res.*, vol. 6, no. 12, pp. 1705–1749, 2005.

[10] S. Bickel and T. Scheffer, “Multi-view clustering,” in *Proc. IEEE Int. Conf. Data Mining*, 2004, pp. 19–26.

[11] X. Yi, Y. Xu, and C. Zhang, “Multi-view em algorithm for finite mixture models,” in *Proc. Int. Conf. Pattern Recognit. Image Anal.*, Aug. 2005, pp. 420–425.

[12] G. Tzortzis and A. Likas, “Convex mixture models for multi-view clustering,” in *Proc. Int. Conf. Artif. Neural Netw.*, Dec. 2009, pp. 205–214.

[13] G. Tzortzis and A. Kikas, “Multiple view clustering using a weighted combination of exemplar-based mixture models,” *IEEE Trans. Neural Netw.*, vol. 21, no. 12, pp. 1925–1938, Dec. 2010.

[14] M. Belkin and P. Niyogi, “Laplacian eigenmaps and spectral techniques for embedding and clustering,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 585–591.

[15] Z. Kang, H. Pan, S. C. H. Hoi, and Z. Xu, “Robust graph learning from noisy data,” *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 1833–1843, May 2020.

[16] A. Kumar and H. DaumeIII, “A co-training approach for multi-view spectral clustering,” in *Proc. 28th Int. Conf. Mach. Learn.*, New York, NY, USA, Jun. 2011, pp. 393–400.

[17] A. Kumar, P. Rai, and H. DaumeIII, “Co-regularized multi-view spectral clustering,” in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2011, pp. 1413–1421.

[18] A. Y. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2001, pp. 849–856.

[19] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans. Pattern Anal. Mach. Learn.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[20] G. Chao, “Discriminative k-means Laplacian clustering,” *Neural Process. Lett.*, vol. 49, pp. 393–405, 2019.

[21] H. Lütkepohl, *Handbook of Mactrices*. Chichester, U.K.: Wiley, 1997, pp. 67–69.

[22] U. Luxburg, “A tutorial on spectral clustering,” *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.

[23] A. Blum and T. Mitchell, “Combining labeled and unlabeled data with co-training,” in *Proc. 11th Annu. Conf. Comput. Learn. Theory*, Jul. 1998, pp. 92–100.

[24] X. Cai, F. Nie, H. Huang, and F. Kamangar, “Heterogeneous image feature integration via multi-modal spectral clustering,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1977–1984.

[25] X. Wang, B. Qian, J. Ye, and I. Davidson, “Multi-objective multi-view spectral clustering via pareto optimization,” in *Proc. SIAM Int. Conf. Data Mining*, May 2013, pp. 234–242.

[26] Y. Ye, X. Liu, J. Yin, and E. Zhu, “Co-regularized kernel k-means for multi-view clustering,” in *Proc. 23rd Int. Conf. Pattern Recognit.*, Aug. 2016, pp. 1583–1588.

[27] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, “Clustering on multi-layer graphs via subspace analysis on Grassmann manifolds,” *IEEE Trans. Signal Process.*, vol. 62, no. 4, pp. 905–918, Feb. 2014.

[28] R. Vidal, “A tutorial on subspace clustering,” *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, Mar. 2011.

[29] E. Elhamifar and R. Vidal, “Sparse subspace clustering: Algorithm, theory, and applications,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.

[30] Q. Yin, S. Wu, R. He, and L. Wang, “Multi-view clustering via pairwise sparse subspace representation,” *Neurocomputing*, vol. 156, no. 5, pp. 12–21, 2015.

[31] M. Brbić and I. Kopriva, “Multi-view low-rank sparse subspace clustering,” *Pattern Recognit.*, vol. 73, pp. 247–258, 2018.

[32] Y. Wang, W. Zhang, L. Wu, X. Lin, M. Fang, and S. Pan, “Iterative views agreement: An iterative low-rank based structured optimization method to multi-view spectral clustering,” in *Proc. 25th Int. Joint Conf. Artif. Intell.*, Jul. 2016, pp. 2153–2159.

[33] Y. Wang, X. Lin, L. Wu, W. Zhang, Q. Zhang, and X. Huang, “Robust subspace clustering for multi-view data by exploiting correlation consensus,” *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3939–3949, Nov. 2015.

[34] C. Zhang *et al.*, “Generalized latent multi-view subspace clustering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 86–99, Jan. 2020.

[35] L. Huang, H.-Y. Chao, and C.-D. Wang, “Multi-view intact space clustering,” *Pattern Recognit.*, vol. 86, pp. 344–353, 2019.

[36] S.-Y. Li, Y. Jiang, and Z.-H. Zhou, “Partial multi-view clustering,” in *Proc. 28th AAAI Conf. Artif. Intell.*, Jul. 2014, pp. 1968–1974.

[37] H. Zhao, H. Liu, and Y. Fu, “Incomplete multi-modal visual data grouping,” in *Proc. 25th Int. Joint Conf. Artif. Intell.*, Jul. 2016, pp. 2392–2398.

[38] Q. Yin, S. Wu, and L. Wang, “Incomplete multi-view clustering via subspace learning,” in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2015, pp. 383–392.

[39] H. Zhao, Z. Ding, and Y. Fu, “Multi-view clustering via deep matrix factorization,” in *Proc. 31st AAAI Conf. Artif. Intell.*, Jun. 2017, pp. 2921–2927.

[40] H. Cai, B. Liu, Y. Xiao, and L. Y. Lin, “Semi-supervised multi-view clustering based on constrained nonnegative matrix factorization,” *Knowl.-Based Syst.*, vol. 182, 2019, Art. no. 104798.

[41] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[42] W. Xu, X. Liu, and Y. Gong, “Document clustering based on non-negative matrix factorization,” in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, Jul. 2003, pp. 267–273.

[43] X. Zhang *et al.*, “Multi-view clustering based on graph-regularized nonnegative matrix factorization for object recognition,” *Inf. Sci.*, vol. 432, pp. 463–478, 2018.

[44] J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov, “Metagenes and molecular pattern discovery using matrix factorization,” *Proc. Nat. Acad. Sci.*, vol. 101, no. 12, pp. 4164–4169, 2004.

[45] Z. Akata, C. Bauckhage, and C. Thurau, “Non-negative matrix factorization in multimodality data for segmentation and label prediction,” in *Proc. 16th Comput. Vis. Winter Workshop*, 2011, pp. 1–8.

[46] J. Liu, C. Wang, J. Gao, and J. Han, “Multi-view clustering via joint nonnegative matrix factorization,” in *Proc. SIAM Int. Conf. Data Mining*, Feb. 2013, pp. 252–260.

[47] X. Cai, F. Nie, and H. Huang, “Multi-view k-means clustering on big data,” in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, Aug. 2013, pp. 2598–2604.

[48] J. Xu, J. Han, F. Nie, and X. Li, “Re-weighted discriminatively embedded k-means for multi-view clustering,” *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 3016–3027, Jun. 2017.

[49] H. Liu and Y. Fu, “Consensus guided multi-view clustering,” *ACM Trans. Knowl. Discov. Data*, vol. 12, no. 4, 2018, Art. no. 42.

[50] C. Ding, X. He, and H. D. Simon, “On the equivalence of nonnegative matrix factorization and spectral clustering,” in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2005, pp. 1–5.

[51] H. Gao, F. Nie, X. Li, and H. Huang, “Multi-view subspace clustering,” in *Proc. IEEE Conf. Comput. Vis.*, Dec. 2015, pp. 4238–4246.

[52] H. Wang, F. Nie, and H. Huang, “Multi-view clustering and feature learning via structured sparsity,” in *Proc. 30th Int. Conf. Mach. Learn.*, Jun. 2013, pp. 352–360.

[53] D. Greene and P. Cunningham, “A matrix factorization approach for integrating multiple data views,” in *Proc. Eur. Conf. Mach. Learn. Knowl. Discov. Databases: Part I*, Sep. 2009, pp. 423–438.

[54] W. Tang, Z. Lu, and I. S. Dhillon, “Clustering with multiple graphs,” in *Proc. 9th IEEE Int. Conf. Data Mining*, Dec. 2009, pp. 1016–1021.

[55] B. Qian, X. Shen, Y. Gu, Z. Tang, and Y. Ding, “Double constrained NMF for partial multi-view clustering,” in *Proc. Int. Conf. Digit. Image Comput.: Techn. Appl.*, 2016, pp. 1–7.

[56] L. Zong, X. Zhang, L. Zhao, H. Yu, and Q. Zhao, “Multi-view clustering via multi-manifold regularized non-negative matrix factorization,” *Neural Netw.*, vol. 88, pp. 74–89, 2017.

[57] Z. Kang *et al.*, “Partition level multiview subspace clustering,” *Neural Netw.*, vol. 122, pp. 279–288, 2020.

[58] H. Tao, C. Hou, X. Liu, D. Yi, and J. Zhu, “Reliable multi-view clustering,” in *Proc. 32nd AAAI Conf. Artif. Intell.*, Jul. 2018, pp. 4123–4130.

[59] L. Zhang, L. Zhang, B. Du, J. You, and D. Tao, “Hyperspectral image unsupervised classification by robust manifold matrix factorization,” *Inf. Sci.*, vol. 485, pp. 154–169, 2019.

[60] J. Ma, L. Zhang, and L. Zhang, “Discriminative subspace matrix factorization for multiview data clustering,” *Pattern Recognit.*, vol. 111, 2021, Art. no. 107676.

[61] W. Shao, L. He, and S. Y. Philip, “Multiple incomplete views clustering via weighted nonnegative matrix factorization with l21 regularization,” in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, Sep. 2015, pp. 318–334.

[62] Y.-M. Xu, C.-D. Wang, and J.-H. Lai, “Weighted multi-view clustering with feature selection,” *Pattern Recognit.*, vol. 53, pp. 25–35, 2016.

[63] W. Shao, L. He, C. ta Lu, and S. Y. Philip, “Online multi-view clustering with incomplete views,” in *Proc. IEEE Int. Conf. Big Data*, Feb. 2016, pp. 1012–1017.

[64] C. Xu, D. Tao, and C. Xu, “Multi-view self-paced learning for clustering,” in *Proc. 24th Int. Joint Conf. Artif. Intell.*, Jul. 2015, pp. 3974–3980.

[65] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, “From ensemble clustering to multi-view clustering,” in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2843–2849.

[66] J. Xu, J. Han, and F. Nie, "Discriminatively embedded k-means for multiview clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Dec. 2016, pp. 5356–5364.

[67] F. Nie, J. Li, and X. Li, "Self-weighted multiview clustering with multiple graphs," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2564–2570.

[68] T. Joachims, N. Cristiani, and J. Shawe-Taylor, "Composite kernels for hypertext categorisation," in *Proc. 18th Int. Conf. Mach. Learn.*, Jul. 2001, pp. 250–257.

[69] T. Zhang, A. Popescul, and B. Dom, "Linear prediction models with graph regularization for web-page categorization," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, Aug. 2006, pp. 821–826.

[70] G. Chao and S. Sun, "Multi-kernel maximum entropy discrimination for multi-view learning," *Intell. Data Anal.*, vol. 20, no. 3, pp. 481–493, 2016.

[71] J. P. Vert, K. Tsuda, and B. Schölkopf, *A Primer on Kernel Methods*. Cambridge, MA, USA: MIT Press, 2004, pp. 35–70.

[72] Z. Kang, C. Peng, and Q. Cheng, "Kernel-driven similarity learning," *Neurocomputing*, vol. 267, pp. 210–219, 2017.

[73] B. Zhao, J. T. Kwok, and C. Zhang, "Multiple kernel clustering," in *Proc. SIAM Int. Conf. Data Mining*, May 2009, pp. 638–649.

[74] H. Zeng and Y. M. Cheung, "Kernel learning for local learning based clustering," in *Proc. Int. Conf. Artif. Neural Netw.*, Sep. 2009, pp. 10–19.

[75] H. Valizadegan and R. Jin, "Generalized maximum margin clustering and unsupervised kernel learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Nov. 2006, pp. 1417–1424.

[76] Z. Kang, L. Wen, W. Chen, and Z. Xu, "Low-rank kernel learning for graph-based clustering," *Knowl.-Based Syst.*, vol. 163, pp. 510–517, 2019.

[77] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and I. Jordan, "Learning the kernel matrix with semidefinite programming," *J. Mach. Learn. Res.*, vol. 5, pp. 27–72, 2004.

[78] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proc. 21st Int. Conf. Mach. Learn.*, Jul. 2004, pp. 41–48.

[79] S. Sonnenburg, G. Räsch, and C. Schäfer, "A general and efficient multiple kernel learning algorithm," in *Proc. 18th Int. Conf. Neural Inf. Process. Syst.*, Dec. 2005, pp. 1273–1280.

[80] M. Gönen and E. Alpaydin, "Localized multiple kernel learning," in *Proc. 25th Int. Conf. Mach. Learn.*, Jul. 2008, pp. 352–359.

[81] M. Gönen and E. Alpaydin, "Multiple kernel learning algorithms," *J. Mach. Learn. Res.*, vol. 12, no. 3, pp. 2211–2268, 2011.

[82] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, 1998.

[83] I. S. Dhillon, Y. Guan, and B. Kulis, "Weighted graph cuts without eigenvectors: A multilevel approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 11, pp. 1944–1957, Nov. 2007.

[84] G. Tzortzis and A. Likas, "Kernel-based weighted multi-view clustering," in *Proc. IEEE 12th Int. Conf. Data Mining*, Dec. 2012, pp. 675–684.

[85] D. Guo, J. Zhang, X. Liu, Y. Cui, and C. Zhao, "Multiple kernel learning based multi-view spectral clustering," in *Proc. 22nd Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 3774–3779.

[86] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, "Multiple kernel k-means clustering with matrix-induced regularization," in *Proc. 30th AAAI Conf. Artif. Intell.*, Feb. 2016, pp. 1888–1894.

[87] D. Q. Zhang and S. C. Chen, "Clustering incomplete data using kernel-based fuzzy c-means algorithm," *Neural Process. Lett.*, vol. 18, pp. 155–162, 2003.

[88] W. Shao, X. Shi, and P. S. Yu, "Clustering on multiple incomplete datasets via collective kernel learning," in *Proc. IEEE 13th Int. Conf. Data Mining*, Dec. 2013, pp. 1181–1186.

[89] X. Xin, J. Wang, R. Xie, S. Zhou, W. Huang, and N. Zheng, "Semi-supervised person re-identification using multi-view clustering," *Pattern Recognit.*, vol. 88, pp. 285–297, 2019.

[90] X. Liu *et al.*, "Late fusion incomplete multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2410–2423, Oct. 2019.

[91] G. F. Tzortzis and A. C. Likas, "The global kernel k-means algorithm for clustering in feature space," *IEEE Trans. Neural Netw.*, vol. 20, no. 7, pp. 1181–1194, Jul. 2009.

[92] Q. Wang, Y. Dou, X. Liu, F. Xia, Q. Lv, and K. Yang, "Local kernel alignment based multi-view clustering using extreme learning machine," *Neurocomputing*, vol. 275, pp. 1099–1111, 2018.

[93] X. Chen, X. Xu, J. Z. Huang, and Y. Ye, "TW-k-means: Automated two-level variable weighting clustering algorithm for multiview data," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 4, pp. 932–944, Apr. 2013.

[94] S. Zhou *et al.*, "Multi-view spectral clustering with optimal neighborhood Laplacian matrix," in *Proc. 33rd AAAI Conf. Artif. Intell.*, Jul. 2020, pp. 6965–6972.

[95] G. Cleuziou, M. Exbrayat, L. Martin, and J.-H. Sublemontier, "Cofkm: A centralized method for multiple-view clustering," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2009, pp. 752–757.

[96] Y. Jiang, F. L. Chuang, S. Wang, Z. Deng, J. Wang, and P. Qian, "Collaborative fuzzy clustering from multiple weighted views," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 688–701, Apr. 2015.

[97] J. Sun, J. Lu, T. Xu, and J. Bi, "Multi-view sparse co-clustering via proximal alternating linearized minimization," in *Proc. 32th Annu. Int. Conf. Mach. Learn.*, Jul. 2015, pp. 757–766.

[98] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learningmethods," *Neural Comput.*, vol. 16, no. 12, pp. 2639–2664, 2004.

[99] G. Chao and S. Sun, "Consensus and complementarity based maximum entropy discrimination for multi-view classification," *Inf. Sci.*, vol. 367, no. 11, pp. 296–310, 2016.

[100] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan, "Multi-view clustering via canonical correlation analysis," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, Jun. 2009, pp. 129–136.

[101] M. B. Blaschko and C. H. Lampert, "Correlational spectral clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[102] N. Rasiwasia, D. Mahajan, V. Mahadevan, and G. Aggarwal, "Cluster canonical correlation analysis," in *Proc. 31th Annu. Int. Conf. Mach. Learn.*, Jun. 2014, pp. 823–831.

[103] A. Trivedi, P. Rai, H. DauméIII, and S. L. DuVall, "Muliview clusterting with incomplete views," in *Proc. Neural Inf. Process. Syst.: Workshop Mach. Learn. Social Comput.*, Whistler, Canada, 2010, pp. 1–8.

[104] B. Long, P. S. Yu, and Z. Zhang, "A general model for multiple view unsupervised learning," in *Proc. 8th SIAM Int. Conf. Data Mining*, Apr. 2008, pp. 822–833.

[105] Q. Wang, Y. Dou, X. Liu, Q. Lv, and S. Li, "Multi-view clustering with extreme learning machine," *Neurocomputing*, vol. 214, pp. 483–494, 2016.

[106] J. Sun, J. Bi, and H. R. Kranzler, "Multi-view biclustering for genotype-phenotype association studies of complex diseases," in *Proc. IEEE Int. Conf. Bioinf. Biomed.*, 2013, pp. 316–321.

[107] J. Sun, J. Bi, and H. R. Kranzler, "Multi-view singular value decomposition for disease subtyping and genetic associations," *BMC Genetics*, vol. 15, no. 73, pp. 1–12, 2014.

[108] M. Lee, H. Shen, J. Z. Huang, and J. S. Marron, "Biclustering via sparse singular value decomposition," *Biometrics*, vol. 66, pp. 1087–1095, 2010.

[109] C.-D. Wang, J.-H. Lai, and P. S. Yu, "Multi-view clustering based on belief propagation," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 4, pp. 1007–1021, Apr. 2016.

[110] R. Fan, T. Luo, W. Zhuge, S. Qiang, and C. Hou, "Multi-view subspace learning via bidirectional sparsity," *Pattern Recognit.*, vol. 108, 2020, Art. no. 107524.

[111] X. Cao, C. Zhang, H. Fu, and H. Zhang, "Diversity-induced multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 586–594.

[112] V. R. DeSa, "Spectral clustering with two views," in *Proc. 22th Annu. Int. Conf. Mach. Learn., Workshop Learn. Multiple Views*, Jun. 2005, pp. 20–27.

[113] D. Zhou and C. J. Burges, "Spectral clustering and transductive learning with multiple views," in *Proc. 24th Int. Conf. Mach. Learn.*, Jul. 2007, pp. 1159–1166.

[114] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *Proc. 28th AAAI Conf. Artif. Intell.*, Jul. 2014, pp. 2149–2155.

[115] T. Lange and J. M. Buhmann, "Fusion of similarity data in clustering," in *Proc. 18th Int. Conf. Neural Inf. Process. Syst.*, Dec. 2005, pp. 723–730.

[116] X. Zhu, S. Zhang, W. He, R. Hu, C. Lei, and P. Zhu, "One-step multi-view spectral clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 10, pp. 2022–2034, Oct. 2019.

[117] X. Liu, S. Ji, W. Glänzel, and B. De Moor, "Multiview partitioning via tensor methods," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 5, pp. 1056–1069, May 2013.

[118] Z. Yuanpeng, F.-L. Chung, and S. Wang, "A multi-view and multi-exemplar fuzzy clustering approach: Theoretical analysis and experimental studies," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 8, pp. 1543–1557, Aug. 2019.

[119] C. Tang *et al.*, "CGD: Multi-view clustering via cross-view graph diffusion," in *Proc. 33rd AAAI Conf. Artif. Intell.*, Jul. 2020, pp. 5924–5931.

[120] Y. Li and M. Y. Z. Zhang, "Multi-view representation learning: A survey from shallow methods to deep methods," 2016, *arXiv:1610.01206v5*.

[121] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Recogn. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[122] W. Zhuge, C. Hou, X. Liu, H. Tao, and D. Yi, "Simultaneous representation learning and clustering for incomplete multi-view data," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Jul. 2019, pp. 4482–4488.

[123] W. Zhuge, H. Tao, T. Luo, C. Hou, and D. Yi, "Joint representation learning and clustering: A framework for grouping partial multiview data," *IEEE Trans. Knowl. Data Eng.*, 2020.

[124] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep Boltzmann machines," *J. Mach. Learn. Res.*, vol. 15, pp. 2949–2980, 2014.

[125] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proc. 28th Int. Conf. Int. Conf. Mach. Learn.*, Jul. 2011, pp. 689–696.

[126] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, "Deep captioning with multimodal recurrent neural networks (m-RNN)," 2015, *arXiv:1412.6632v5*.

[127] F. Feng, X. Wang, R. Li, and I. Ahmad, "Correspondence autoencoders for cross-modal retrieval," in *Proc. ACM Multimedia*, Oct. 2015, pp. 7–16.

[128] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning," in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn.*, Jul. 2015, pp. 1083–1092.

[129] A. Karpathy and F.-F. Li, "Deep visual-semantic alignments for generating image descriptions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 664–676, Apr. 2017.

[130] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 677–691, Apr. 2017.

[131] F. Tian, B. Gao, Q. Cui, E. Chen, and T. Y. Liu, "Learning deep representations for graph clustering," in *Proc. 28th AAAI Conf. Artif. Intell.*, Jul. 2014, pp. 1293–1299.

[132] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. 33rd Int. Conf. Mach. Learn.*, Jun. 2016, pp. 478–487.

[133] Z. Li, Q. Wang, Z. Tao, Q. Gao, and Z. Yang, "Deep adversarial multi-view clustering network," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Jul. 2019, pp. 2952–2958.

[134] R. Zhou and Y.-D. Shen, "End-to-end adversarial-attention network for multi-modal clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 14619–14628.

[135] P. Zhu, B. Hui, C. Zhang, D. Du, L. Wen, and Q. Hu, "Multi-view deep subspace clustering networks," 2019, *arXiv:1908.01978*.

[136] X. Sun, M. Cheng, C. Min, and L. Jing, "Self-supervised deep multi-view subspace clustering," in *Proc. Asian Conf. Mach. Learn.*, 2019, pp. 1001–1016.

[137] S. Vega-Pons and J. Ruiz-Shulcloper, "A survey of clustering ensemble algorithm," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 25, no. 3, pp. 337–372, 2011.

[138] X. Zhang and C. E. Brodley, "Solving cluster ensemble problems by bipartite graph partitioning," in *Proc. 21st Int. Conf. Mach. Learn.*, Jul. 2004, pp. 281–288.

[139] M. Ozay, F. T. Y. Vural, S. R. Kulkarni, and H. V. Poor, "Fusion of image segmentation algorithms using consensus clustering," in *Proc. 20th IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 4049–4053.

[140] E. F. Lock and D. B. Dunson, "Bayesian consensus clustering," *Bioinformatics*, vol. 29, no. 20, pp. 2610–2616, 2013.

[141] Y. Senbabaoğlu, G. Michilidis, and J. Z. Li, "Critical limitations of consensus clustering in class discovery," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 4, no. 1, pp. 1–13, 2014.

[142] H. Liu, J. Wu, T. Liu, D. Tao, and Y. Fu, "Spectral ensemble clustering via weighted k-means: Theoretical and practical evidence," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 5, pp. 1129–1143, May 2017.

[143] S. Saha, S. Mitra, and S. Kramer, "Exploring multiobjective optimization for multiview clustering," *ACM Trans. Knowl. Discov. Data*, vol. 12, no. 4,2018, Art. no. 44.

[144] X. Xie and S. Sun, "Multi-view clustering ensembles," in *Proc. Int. Conf. Mach. Learn. Cybern.*, Sep. 2013, pp. 51–56.

[145] Z. Xue, J. Du, D. Du, and S. Lyu, "Deep low-rank subspace ensemble for multi-view clustering," *Inf. Sci.*, vol. 482, pp. 210–227, 2019.

[146] J. Zhang and C. Zhang, "Multitask bregman clustering," *Neurocomputing*, vol. 74, no. 10, pp. 1720–1734, 2011.

[147] X. Zhang, X. Zhang, and H. Liu, "Smart multitask Bregman clustering and multitask kernel clustering," *ACM Trans. Knowl. Discov. Data*, vol. 10, no. 1, pp. 8: 1–8:29, 2015.

[148] Q. Gu, Z. Li, and J. Han, "Learning a kernel for multi-task clustering," in *Proc. 25th AAAI Conf. Artif. Intell.*, Aug 2011, pp. 368–373.

[149] X.-L. Zhang, "Convex discriminative multi task clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 28–40, Jan. 2015.

[150] X. Zhang, X. Zhang, and H. Liu, "Self-adapted multi-task clustering," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, Jul. 2016, pp. 2357–2363.

[151] Y. Ren, X. Que, D. Yao, and Z. Xu, "Self-paced multi-task clustering," *Neurocomputing*, vol. 350, pp. 212–220, 2019.

[152] X. Zhang, X. Zhang, and H. Liu, "Multi-task multi-view clustering for non-negative data," in *Proc. 24th Int. Conf. Artif. Intell.*, Jul. 2016, pp. 4055–4061.

[153] X. Zhang, X. Zhang, H. Liu, and X. Liu, "Multi-task multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 12, pp. 3324–3338, Dec. 2016.

[154] Y. Ren, X. Yan, Z. Hu, and Z. Xu, "Self-paced multi-task multi-view capped-norm clustering," in *Proc. Int. Conf. Neural Inf. Process.*, Berlin, Germany: Springer, 2018, pp. 205–217.

[155] S. Sun, J. Shawe-Taylor, and L. Mao, "PAC-Bayes analysis of multi-view learning," *Inf. Fusion*, vol. 35, pp. 117–131, 2017.

[156] S. Yu, B. Krishnapuram, R. Rosales, and R. B. Rao, "Bayesian co-training," *J. Mach. Learn. Res.*, vol. 12, pp. 2649–2680, Jan. 2011.

[157] V. Sindhwani and P. Niyogi, "A co-regularized approach to semi-supervised learning with multiple views," in *Proc. Int. Conf. Mach. Learn. Workshop Learn. Multiple Views*, 2005, pp. 824–831.

[158] V. Sindhwani and D. S. Rosenberg, "An RKHS for multi-view learning and manifold co-regularization," in *Proc. 25th Int. Conf. Mach. Learn.*, Jul 2008, pp. 976–983.

[159] S. Sun and G. Chao, "Multi-view maximum entropy discrimination," in *Proc. 23th Int. Joint Conf. Artif. Intell.*, Aug. 2013, pp. 1706–1712.

[160] S. Sun and G. Chao, "Alternative multi-view maximum entropy discrimination," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, pp. 1445–1556, Jun. 2016.

[161] C. Jin, W. Mao, R. Zhang, Y. Zhang, and X. Xue, "Cross-modal image clustering via canonical correlation analysis," in *Proc. 29th AAAI Conf. Artif. Intell.*, Jan. 2015, pp. 151–159.

[162] C. Andrés Méndez, P. Summers, and G. Menegaz1, "Multiview cluster ensembles for multimodal MRI segmentation," *Int. J. Imag. Syst. Technol.*, vol. 25, no. 1, pp. 56–67, 2015.

[163] A. Djelouah, J.-S. Franco, and E. Boyer, "Multi-view object segmentation in space and time," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2640–2647.

[164] J. Wu and J. M. Rehg, "Centrist: A visual descriptor for scene categorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1489–1501, Aug. 2011.

[165] H. Yu, M. Li, H.-J. Zhang, and J. Feng, "Color texture moments for content-based image retrieval," in *Proc. Int. Conf. Image Process.*, Sep. 2002, pp. 929–932.

[166] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 886–893.

[167] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[168] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[169] M. Chi, P. Zhang, Y. Zhao, R. Feng, and X. Xue, "Web image retrieval reranking with multi-view clustering," in *Proc. 18th Int. Conf. World Wide Web*, 2009, pp. 1189–1190.

[170] H. Tao, C. Hou, Y. Qian, J. Zhu, and D. Yi, "Latent complete row space recovery for multi-view subspace clustering," *IEEE Trans. Image Process.*, vol. 29, no. 1, pp. 8083–8096, Jul. 2020.

[171] Y.-M. Kim, M.-R. Amini, C. Goutte, and P. Gallinari, "Multi-view clustering of multilingual documents," in *Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, Jul. 2010, pp. 821–822.

[172] Y. Jiang, J. Liu, Z. Li, and H. Lu, "Collaborative PLSA for multi-view clustering," in *Proc. 21st Int. Conf. Pattern Recognit.*, Nov. 2012, pp. 2997–3000.

[173] P. Zhao, Y. Jiang, and Z.-H. Zhou, "Multi-view matrix completion for clustering with side information," in *Proc. Pacific-Asia Conf. Knowl. Discov. Data Mining*. Berlin, Germany: Springer, 2017, pp. 403–415.

[174] S. F. Hussain, M. Mushtaq, and Z. Halim, "Multi-view document clustering via ensemble method," *J. Intell. Inf. Syst.*, vol. 43, no. 1, pp. 81–99, 2014.

[175] G. Petkos, S. Papadopoulos, and Y. Kompatsiaris, "Social event detection using multimodal clustering and integrating supervisory signals," in *Proc. 2nd ACM Int. Conf. Multimedia Retrieval*, Jun. 2012, pp. 1–8.

[176] S. Samangooei *et al.*, "Social event detection via sparse multi-modal feature selection and incremental density based clustering," in *Proc. MediaEval*, 2013.

[177] G. Petkos, S. Papadopoulos, E. Schinas, and Y. Kompatsiaris, "Graph-based multimodal clustering for social event detection in large collections of images," in *Proc. 20th Anniversary Int. Conf. MultiMedia Model.*, Jan. 2014, pp. 146–158.

[178] R. Bekkerman and J. Jeon, "Multi-modal clustering for multimedia collections," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[179] X. Wu, C.-W. Ngo, and A. G. Hauptmann, "Multimodal news story clustering with pairwise visual near-duplicate constraint," *IEEE Trans. Multimedia*, vol. 10, no. 2, pp. 188–199, Feb. 2008.

[180] V. Mekthanavanh, T. Li, H. Meng, Y. Yang, and J. Hu, "Social web video clustering based on multi-view clustering via nonnegative matrix factorization," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 10, pp. 2779–2790, 2019.

[181] G. Chao *et al.*, "Multi-view cluster analysis with incomplete data to understand treatment effects," *Inf. Sci.*, vol. 494, pp. 278–293, 2019.

[182] S. Yu *et al.*, "Optimized data fusion for kernel k-means clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 1031–1039, May 2012.

[183] S. Yu *et al.*, "Optimized data fusion for k-means Laplacian clustering," *Bioinformatics*, vol. 27, no. 1, pp. 118–126, 2011.

[184] D. Li, L. Wang, Z. Xue, and S. T. C. Wong, "When discriminative k-means meets Grassmann manifold: Disease gene identification via a general multi-view clustering method," in *Proc. IEEE Int. Conf. Biomed. Health Inform.*, Feb. 2016, pp. 364–367.

[185] B. Jiang *et al.*, "Data analysis of soccer athletes physical fitness test based on multi-view clustering," in *Journal of Physics: Conference Series*. Bristol, U.K.: IOP Publishing, 2018, vol. 1060, Art. no. 0 12024.

[186] N. Rappoport and R. Shamir, "Multi-omic and multi-view clustering algorithms: Review and cancer benchmark," *Nucleic Acids Res.*, vol. 46, no. 20, pp. 10546–10562, 2018.

[187] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "RCV1: A new benchmark collection for text categorization research," *J. Mach. Learn. Res.*, vol. 5, no. 4, pp. 361–397, 2004.

[188] C. Lu, S. Yan, and Z. Lin, "Convex sparse spectral clustering: Single-view to multi-view," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2833–2843, Jun. 2016.

[189] D. Cai, X. He, and J. Han, "Using graph model for face analysis," Tech. Rep. UIUCDCS-R-2005-2636, 2005.

[190] D. Cai and X. Chen, "Optimized data fusion for kernel k-means clustering," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1669–1680, May 2014.

[191] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the nyström method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 214–225, Feb. 2004.

[192] D. Yan, L. Huang, and M. I. Jordan, "Fast spectral clustering of data using sequential matrix compression," in *Proc. 17th Eur. Conf. Mach. Learn.*, Sep. 2006, pp. 590–597.

[193] D. Yan, L. Huang, and M. I. Jordan, "Fast approximate spectral clustering," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, Jun. 2009, pp. 907–916.

[194] Z. Zhang, L. Liu, F. Shen, H. T. Shen, and L. Shao, "Binary multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1774–1782, Jul. 2019.

[195] H.-P. Kriegel, P. Kröger, and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *ACM Trans. Knowl. Discovery Data*, vol. 3, no. 1, pp. 1–58, 2009.

[196] J. G. Dy and C. E. Brodley, "Feature selection for unsupervised learning," *J. Mach. Learn. Res.*, vol. 5, pp. 845–889, 2004.

[197] D. M. Witten and R. Tibshirani, "A framework for feature selection in clustering," *J. Amer. Statist. Assoc.*, vol. 105, no. 490, pp. 713–726, 2010.

[198] G. Chao, Y. Luo, and W. Ding, "Recent advances in supervised dimension reduction: A survey," *Mach. Learn. Knowl. Extraction*, vol. 1, no. 1, pp. 341–358, 2019.

[199] Y. Fujikawa and T. B. Ho, "Cluster-based algorithms for dealing with missing values," in *Proc. 6th Pacific-Asia Conf. Adv. Knowl. Discovery Data Mining*, May 2002, pp. 549–554.

[200] Y. S. Su, A. Gelman, J. Hill, and M. Yajima, "Multiple imputation with diagnostics (mi) in R:opening windows into the black box," *J. Statist. Softw.*, vol. 45, pp. 1–31, 2011.

[201] C. Shang, A. Palmer, J. Sun, K.-S. Chen, J. Lu, and J. Bi, "VIGAN: Missing view imputation with generative adversarial networks," in *Proc. IEEE Int. Conf. Big Data*, BigData, Boston, MA, USA, Dec. 2017, pp. 766–775.

[202] N. Rai, S. Negi, S. Chaudhury, and O. Deshmukh, "Partial multi-view clustering using graph regularized NMF," in *Proc. 23rd Int. Conf. Pattern Recognit.*, Dec. 2016, pp. 2192–2197.

[203] L. Xinwang *et al.*, "Efficient and effective incomplete multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020, doi: 10.1109/TPAMI.2020.2974828.

[204] Q. Zhao, D. Meng, L. Jiang, Q. Xie, Z. Xu, and A. G. Hauptmann, "Self-paced learning for matrix factorization," in *Proc. 29th AAAI Conf. Artif. Intell.*, Jan. 2015, pp. 3196–3202.

[205] M. T. Law, R. Urtasun, and R. S. Zemel, "Deep spectral clustering learning," in *Proc. The 34th Int. Conf. Mach. Learn.*, Aug. 2017, pp. 1985–1994.

[206] J. R. Hershey, Z. Chen, J. L. Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. 2016 IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2016, pp. 31–35.

[207] H. O. Song, S. Jegelka, V. Rathod, and K. Murphy, "Deep metric learning via facility location," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5382–5390.

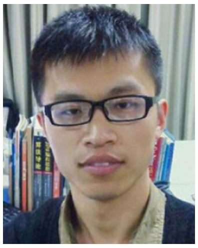
[208] S. Huang, Z. Kang, and Z. Xu, "Auto-weighted multi-view clustering via deep matrix decomposition," *Pattern Recognit.*, vol. 97, 2019, Art. no. 107015.

[209] L. S. Tekumalla, V. Rajan, and C. Bhattacharyya, "Vine copulas for mixed data: Multi-view clustering for mixed data beyond meta-gaussian dependencies," *Mach. Learn.*, vol. 106, no. 9/10, pp. 1331–1357, 2017.

[210] Y. Cui, X. Z. Fern, and J. G. Dy, "Non-redundant multi-view clustering via orthogonalization," in *Proc. 7th IEEE Int. Conf. Data Mining*, Feb. 2007, pp. 133–142.

[211] D. Niu, J. G. Dy, and M. I. Jordan, "Multiple non-redundant spectral clustering views," in *Proc. 27th Int. Conf. Int. Conf. Mach. Learn.*, Jun. 2010, pp. 831–838.

[212] Y. Chang *et al.*, "Multiple clustering views from multiple uncertain experts," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 674–683.



Guoqing Chao received the B.S. degree from Xinyang Normal University, Xinyang, China, in 2009, and the Ph.D. degree from the East China Normal University, Shanghai, China in 2015, both in computer science and technology.

After graduation, he worked as a PostDoc, respectively, at the University of Connecticut, Northwestern University, US, and Singapore Management University. Currently, he works with Harbin Institute of Technology at Weihai, China. His research interests include machine learning, data mining and bioinformatics.

Shiliang Sun (Member, IEEE) received the B.E. degree in automatic control from the Beijing University of Aeronautics and Astronautics in 2002, and the Ph.D. degree in automation from Tsinghua University in 2007.

He is a Professor at the School of Computer Science and Technology and the Head of the Pattern Recognition and Machine Learning Research Group, East China Normal University. His research interests include approximate inference, learning theory, sequential modeling, kernel methods, and their applications.

Dr. Sun is a member of the PASCAL network of excellence, and on the editorial boards of multiple international journals.

Jinbo Bi (Member, IEEE) received the B.S. degree in applied mathematics and the M.S. degree in automation from Beijing Institute of Technology, China, respectively, in 1996 and 1999, and the Ph.D. degree in mathematics from Rensselaer Polytechnic Institute, Troy, NY, USA, in 2003.

She is currently the Frederick H Leonhardt Professor of Computer Science in the University of Connecticut, Storrs, CT, USA. Before joining the university, she worked with Siemens Medical Solutions on computer aided diagnosis research as well as Partners Healthcare on clinical decision support systems. Her research interests include machine learning, data mining, bioinformatics, biomedical informatics, and drug discovery.