Incomplete Multi-view Clustering

Hang Gao[✉], Yuxing Peng, and Songlei Jian

Science and Technology on Parallel and Distributed Processing Laboratory,
National University of Defense Technology, Changsha 410073, China
{gao.hang,pengyuxing,jiansonglei}@aliyun.com

Abstract. Real data often consists of multiple views (or representations). By exploiting complementary and consensus grouping information of multiple views, multi-view clustering becomes a successful practice for boosting clustering accuracy in the past decades. Recently, researchers have begun paying attention to the problem of incomplete view. Generally, they assume at least there is one complete view or only focus on two view problems. However, above assumption is often broken in real tasks. In this work, we propose an IVC algorithm for clustering with more than two incomplete views. Compared with existing works, our proposed algorithm (1) does not require any view to be complete, (2) does not limit the number of incomplete views, and (3) can handle similarity data as well as feature data. The proposed algorithm is based on the spectral graph theory and the kernel alignment principle. By aligning projections of individual views with the projection integration of all views, IVC exchanges the complementary grouping information of incomplete views. Consequently, projections of individual views are made complete and thereby resulting the consensus with accurate grouping information. Experiments on synthetic and real datasets demonstrate the effectiveness of IVC.

Keywords: Multi-view clustering · Incomplete view clustering · Spectral clustering

1 Introduction

Many datasets in real world are naturally comprised of heterogeneous views (or representations). Clustering with such type of data is commonly referred to as multi-view Clustering. With the assumption of complementary data representation and consensus decision of clusterings, multi-view clustering has the potential to dramatically increase the learning accuracy over single view clustering [1]. The main problem in multi-view clustering is how to integrate grouping information of individual views. Existing works can be roughly classified into three categories. (1) Multi-kernel learning based approach. The most representative work of this category is Multi-kernel Kmeans [2]. It first uses kernel representation for each

view, and then it incorporates different views by seeking optimal combination of multiple kernels of different views. (2) Subspace learning based approach. It obtains a latent consensus subspace shared by multiple views and cluster the instances on the latent subspace. There are many research works in this category, including CCA-based methods [3], spectral graph based methods [4–6], matrix factorization based methods [7,8]. (3) Ensemble learning based approach. [9] takes a decision in each individual view separately and then combines all decisions of distinct views to establish a consensus decision by determining cluster agreements/disagreements.

Traditional research assumes data are complete in all views. However, in many real applications, parts of instances are not available in some views. For example, in a news story clustering task, articles are collected from different on-line news sources. Only a part of news are reported in all views. No single source includes all news. Another example is image clustering. Images are based on multiple visual and textual features. Some images have only a fraction of visual or textual feature sets.

Recently, a few attempts have been made for multi-view clustering with incomplete views. The first work to deal with incomplete view clustering was proposed in [10]. It uses one view's kernel representation as the similarity matrix and complete the incomplete view's kernel using Laplacian regularization. However, this approach requires that there exists at least one complete view containing all the instances. Shao et al. [11] relax the above constraint. They collectively complete the kernel matrices of incomplete datasets by optimizing the alignment of shared instances of the datasets. Furthermore, a clustering algorithm is proposed based on the kernel canonical correlation analysis. However, this approach focus on two view problem. It can not exploit relation among more than two views. Li et al. proposed a Partial view clustering algorithm (PVC) [12]. Based on non-negative matrix factorization (NMF), PVC works by establishing a latent subspace where the instances corresponding to the same example in different views are close to each other. PVC concentrates on two views problem. Extending PVC to more views suffers from computational problem. In most recently, Shao et al. developed an incomplete view clustering algorithm (MIC) [13]. MIC handles the situation of more than two incomplete views. With joint weighted non-negative matrix factorization, it learns a $L_{2,1}$ regularized latent subspace for multiple views. With mean value imputation initialization, MIC gives lower weights to the incomplete instances than the complete instances. During optimization, MIC pushes multiple views towards a consensus matrix iteratively. But, there are some limitations about MIC. It converges slowly and contains too many parameters, which makes it difficult to operate. Moreover, both PVC and MIC are NMF based method. Both of them inherit the limitations of NMF: (1) It cannot well deal with data with negative feature values. while in many real applications, the non-negative constraints can not be satisfied. (2) It is essentially linear, and thus cannot disclose non-linear structures hidden in data, which limits its learning ability. (3) It only deals with feature values, while in some applications we know the similarities (relationships) of instances

while the detailed feature values are unavailable. Yin et al. [14] proposed a subspace learning algorithm. It utilizes a regression-like objective to learn a latent consensus representation. Besides, it explores the inter-view and intra-view relationship of the data examples by a graph regularization. However, it converges too slowly. It achieves optimal results with about one hundred iterations. This make it difficult to extend to more than two views.

In this paper, we focus on the problem of incomplete view clustering with more than two views. We propose a novel incomplete multi-view clustering (IVC) algorithm. Aiming at completing incomplete views, IVC first integrate individual views by collective spectral decomposition. Then, IVC aligns each individual with the integration respectively. In this way, complementary grouping information is shared among views and missing values of incomplete views are estimated. With estimated individual views, IVC constructs the latent consensus space. At last, clustering solution is obtained by applying the standard spectral clustering on the consensus space. As compared with previous works, the proposed algorithm has several advantages: (1) It does not require any view to be complete. (2) It does not limit the number of incomplete views. (3) It can handle similarity data (or kernel data) as well as feature data. (4) Since it has few parameters to be set, it is easy-implemented. (5) Due to the non-iterative optimization, it is efficient than most iterative algorithms such as MIC. Moreover, it shows better performance. We demonstrate it in the experiment.

The rest of this paper is organized as follows: In Sect. 2, we give a brief review of the spectral clustering and the kernel alignment principle which is our basis. Section 3 presents details of the proposed algorithm. In Sect. 4, we validate the proposed algorithm. Section 5 concludes the paper.

2   Preliminary

In this section, we give a brief review of the spectral clustering and the kernel alignment principle, which provide the necessary background and pave the way to the proposed algorithm.

2.1   Spectral Clustering

Spectral clustering is a theoretically sound and empirically successful clustering algorithm. It treats clustering as a graph partitioning problem. By making use of the spectrum graph theory, it project original data in a low-dimensional space that contains more discriminative grouping information. Algorithm 1 briefly describe the spectral clustering algorithm [15] which is the basis of our work.

The equivalent optimization formular of Algorithm 1 is Eq. 1.

$$\max_{\mathbf{U} \in \Re^{N \times M}} Trace\left(\mathbf{U}^T \mathbf{L} \mathbf{U}\right), \qquad s.t. \mathbf{U}^T \mathbf{U} = \mathbf{I} \qquad (1)$$

Algorithm 1. Normalized Spectral Clustering

Require:
    Similarity matrix: $\mathbf{S} \in \Re^{N \times N}$,   Number of clusters: K;
 1: Compute the symmetrical normalized lapacian matrix $\mathbf{L} = \mathbf{D}^{(-1/2)} \mathbf{S} \mathbf{D}^{(-1/2)}$.
 2: Compute the first K largest eigenvectors $\boldsymbol{u_1}, \boldsymbol{u_2}, ... \boldsymbol{u_K}$ of $\mathbf{L}$.
 3: Let $\mathbf{U} \in \Re^{N \times K}$ be the matrix containing the first k eigenvectors $\boldsymbol{u_1}, \boldsymbol{u_2}, ... \boldsymbol{u_K}$.
 4: Normalize the rows of matrix $\mathbf{U}$ by scaling the norm to be 1.
 5: For $i = 1, 2, ..., n$, let $\boldsymbol{y_i} \in \Re^K$ be the vector corresponding to the i-th row of $\mathbf{U}$.
 6: Cluster the points $\{\boldsymbol{y_i}\}_{i=1,2,...,N}$ with K-means algorithm into cluster $C_1, C_2, ... C_K$.
 7: **return** $\{C_i\}, i = 1, 2, ... K$;

2.2   Kernel Alignment

Kernel alignment is a measurement of similarity (or dissimilarity) between different kernels. Let $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$ be two positive definite kernel matrices such that $\|\mathbf{S}^{(1)}\|_F \neq 0$ and $\|\mathbf{S}^{(2)}\|_F \neq 0$. Then, the dissimilarity between $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$ is defined by Eq. (2) [16], where $\langle \mathbf{S}^{(1)}, \mathbf{S}^{(2)} \rangle_F = \sum_{i=1}^{N} \sum_{j=1}^{N} \mathbf{S}_{i,j}^{(1)} \mathbf{S}_{i,j}^{(2)}$.

$$\rho(\mathbf{S}^{(1)}, \mathbf{S}^{(2)}) = \frac{\langle \mathbf{S}^{(1)}, \mathbf{S}^{(2)} \rangle_F}{\|\mathbf{S}^{(1)}\|_F \|\mathbf{S}^{(2)}\|_F}. \tag{2}$$

3   Proposed Methods

In this section, we present the detail of the incomplete view clustering (IVC) algorithm. We first describe the IVC framework and present its objectives, and then describe the optimization procedures.

3.1   Model Description

Given V incomplete views and the similarity matrices are $\mathbf{S}^{(i)}, i = 1, 2, ..., V$. The cluster number is K. Incomplete views contain different numbers of observed values. In order to make these kernel matrices co-operable (or with the same size N), we initialize incomplete kernels by filling missing entries with the corresponding average of the column (i.e. early estimation).

   First, we exploit the discriminative grouping information of each individual view by spectral decomposition on its similarity matrix $\mathbf{S}^{(i)}, i = 1, 2, ..., V$.

$$\max_{\mathbf{U}^{(i)} \in \Re^{N \times K}} Trace \left(\mathbf{U}^{(i)^T} \mathbf{L}^{(i)} \mathbf{U}^{(i)} \right), \qquad s.t. \, \mathbf{U}^{(i)^T} \mathbf{U}^{(i)} = \mathbf{I} \tag{3}$$

Note that $\mathbf{U}^{(i)}$ is a recasted matrix of the original feature matrix. Each row of $\mathbf{U}^{(i)}$ is a new representation of an instance with lower dimension and more discriminative grouping information.

Next, in order to make different views consistent, we push them towards a latent consensus matrix $\mathbf{U}^*$. Because $\mathbf{U}^{(i)}$ is a projection of original feature matrix, $\mathbf{S}^{(i)} = \mathbf{U}^{(i)}\mathbf{U}^{(i)^T}$ can be seen as a new kernel representation. Similarly, the latent consensus kernel can be decomposed as $\mathbf{S}^* = \mathbf{U}^*\mathbf{U}^{*T}$, where $\mathbf{U}^*$ is the latent projected matrix. Note that $\mathbf{U}^{(i)}$s are derieved by kernels with early estimation. We call $\mathbf{U}^*$ as early consensus projection.

Borrowing the idea from kernel alignment, we measure the dissimilarity between early consensus and each view by Eq. (4).

$$\rho(\mathbf{U}^*, \mathbf{U}^{(i)}) = \|\frac{\mathbf{U}^*\mathbf{U}^{*T}}{\|\mathbf{U}^*\mathbf{U}^{*T}\|_F^2} - \frac{\mathbf{U}^{(i)}\mathbf{U}^{(i)^T}}{\|\mathbf{U}^{(i)}\mathbf{U}^{(i)^T}\|_F^2}\|_F^2 \tag{4}$$

Minimizing the sum of dissimilarities between early consensus and all individuals, we get objective function (5), where λ_i is the tradeoff between different views and expresses the importance of view i in clustering.

$$\max_{\mathbf{U}^* \in \Re^{N \times K}} \quad \sum_i \lambda_i \rho(\mathbf{U}^*, \mathbf{U}^{(i)}), \tag{5}$$

Since that $\|\mathbf{U}^{(i)}\mathbf{U}^{(i)^T}\|_F^2 = K$, $\|\mathbf{U}^*\mathbf{U}^{*T}\|_F^2 = K$, by ignoring constant factors and trace property $(trace(\mathbf{A}\mathbf{A}^T) = \|\mathbf{A}\|_F^2)$, we rewrite the objective function (5) as follows.

$$\max_{\mathbf{U}^* \in \Re^{N \times K}} \sum_i \lambda_i Trace(\mathbf{U}^{(i)}\mathbf{U}^{(i)^T}\mathbf{U}^*\mathbf{U}^{*T}) \tag{6}$$

Now, we retransmit the early consensus back to individuals. Specifically, we reorder each individual view as $\mathbf{U}^{(i)} = \begin{bmatrix} \mathbf{U}_a^{(i)} \\ \mathbf{U}_e^{(i)} \end{bmatrix}$, where $\mathbf{U}_a^{(i)}$ is the part derived by available (observed) values, while $\mathbf{U}_e^{(i)}$ is the part derived by estimated (or missing) values. Correspondingly, we reorder $\mathbf{U}^*$ as $\begin{bmatrix} \mathbf{U}_a^* \\ \mathbf{U}_e^* \end{bmatrix}$. Then, we update each $\mathbf{U}_e^{(i)}$ by aligning $\mathbf{U}^{(i)}$ with $\mathbf{U}^{(*)}$. According to Eq. (4), we get the objective function (7).

$$\max_{\mathbf{U}_e^{(i)}} \quad Trace(\begin{bmatrix} \mathbf{U}_a^* \\ \mathbf{U}_e^* \end{bmatrix} \begin{bmatrix} \mathbf{U}_a^* \\ \mathbf{U}_e^* \end{bmatrix}^T \begin{bmatrix} \mathbf{U}_a^{(i)} \\ \mathbf{U}_e^{(i)} \end{bmatrix} \begin{bmatrix} \mathbf{U}_a^{(i)} \\ \mathbf{U}_e^{(i)} \end{bmatrix}^T) \tag{7}$$

In this way, complementary grouping information is exchanged among incomplete individuals. With updated $\mathbf{U}^{(i)}$s, we construct the final consensus U_f^* by Eq. (6). U_f^* contains more accurate grouping information than U^*. At last, we apply standard K-means clustering on U_f^* to get the final decision.

3.2   Model Training

In this subsection, we demonstrate how does IVC optimizes Eqs. (6) and (7).

By the cyclic property of the trace, we transform optimization problem (6) into (8), which is equivalent to a standard spectral clustering with graph laplacian $\sum_v \lambda_v \mathbf{U}^{(v)}.\mathbf{U}^{(v)T}$. The solution of $\mathbf{U}^{(*)}$ is just the optimal consensus eigen vectors of all individual views.

$$\max_{\mathbf{U}^* \in \Re^{n \times K}} Trace(\mathbf{U}^{*T}(\sum_v \lambda_i \mathbf{U}^{(i)}\mathbf{U}^{(i)T})\mathbf{U}^*)$$
$$s.t.\, \mathbf{U}^{*T}\mathbf{U}^* = \mathbf{I} \tag{8}$$

Transforming and expanding Eq. (7) as Eq. (9), then, taking its derivative w.r.t. $\mathbf{U}_e^{(i)}$ and setting it to zero, we get the solution as in Eq. (10). To the ends, $\mathbf{U}_e^{(i)}$ is calculated.

$$\max_{\mathbf{U}_e^{(i)}} Trace(\begin{bmatrix} \mathbf{U}_a^*\mathbf{U}_a^{*T} & \mathbf{U}_a^*\mathbf{U}_e^{*T} \\ \mathbf{U}_e^*\mathbf{U}_a^{*T} & \mathbf{U}_e^*\mathbf{U}_e^{*T} \end{bmatrix} \begin{bmatrix} \mathbf{U}_a^{(i)}\mathbf{U}_a^{(i)T} & \mathbf{U}_a^{(i)}\mathbf{U}_e^{(i)T} \\ \mathbf{U}_e^{(i)}\mathbf{U}_a^{(i)T} & \mathbf{U}_e^{(i)}\mathbf{U}_e^{(i)T} \end{bmatrix}) \tag{9}$$

$$\mathbf{U}_e^{(i)} = -(\mathbf{U}_a^*\mathbf{U}_e^{*T} + \mathbf{U}_e^*\mathbf{U}_a^{*T} + 2\mathbf{U}_e^*\mathbf{U}_e^{*T})^{-1}$$
$$\times (\mathbf{U}_e^*\mathbf{U}_a^{*T}\mathbf{U}_a^{(i)} + \mathbf{U}_a^*\mathbf{U}_a^{*T}\mathbf{U}_a^{(i)} + \mathbf{U}_e^*\mathbf{U}_e^{*T}\mathbf{U}_a^{(i)} + \mathbf{U}_e^*\mathbf{U}_a^{*T}\mathbf{U}_a^{(i)}) \tag{10}$$

The specific procedure of IVC is summarized in Algorithm 2. IVC first initializes incomplete kernels with early estimation. Then, it projects each individual view into a more discriminative space by spectral decomposition. Next, IVC establishes the early consensus projection, and thereby updating individual projections. With these updated individual projections, IVC constructs the final consensus projection.

Algorithm 2. The Proposed algorithm

Require:
   Similarity matrices: $\mathbf{S}^{(i)}, i = 1, 2, ...V$; Number of clusters: K;
 1: Initialize all individual $\mathbf{K}^{(i)}$s with early estimation;
 2: Do spectral decomposition for all $\mathbf{S}^{(i)}$s by Equation (3);
 3: Calculate consensus eigenvectors $\mathbf{U}^{(*)}$ by Equation (8);
 4: Update each individual $\mathbf{U}_e^{(i)}$ by Equation (10);
 5: Construct $\mathbf{U}_f^*$ with updated $\mathbf{U}^{(i)}$s by Equation (8);
 6: Do K-means clustering with final consensus projection $\mathbf{U}_f^*$;
 7: **return** $\{C_j\}, j = 1, 2, ...K$;

---

4   Experiment

4.1   Comparison Methods

We compare the proposed IVC with several state-of-art methods. The details of comparison methods are as follows:

IVC: IVC is the proposed approach in this work. We set equal default value for λ_i to be 1. Without prior knowledge, we treat all views equally.

MIC: Multiple incomplete view clustering [13] is one of the most recent work. It applies weighted joint non-negative matrix with $L_{2,1}$ regularization. The default co-regularization parameter set α_i and the robust parameter set β_i are all set to be 0.01 for all the views as in the original paper.

MVSpec: MVSpec is a weighted multi-kernel learning and specrtal graph theory based algorithm for multi-view clustering. It represents views through kernel matrices and optimize the intra-cluster variance function. We set the parameter p and initial weights as its original paper [2].

KADD: Integrating multiple kernels by adding them, and then running standard spectral clustering on the corresponding Laplacian. As suggested in earlier findings [17], even this seemingly simple approach often result in near optimal clustering as compared to more sophisticated approaches.

Concat: Feature concatenation is the most simple and intuitive way to integrate all the views. It concatenates features of all views and runs K-means clustering on the concatenated feature set.

  We also report the best performance of complete single view. Note that the compared methods such as **KADD**, **Concat**, and **MVSpec** cannot directly deal with incomplete views. Therefore, we pre-process the incomplete views by mean imputation for these methods. We evaluate above methods by the normalized mutual information (**NMI**). Besides, we use k-means to get the clustering solution at the end, we run k-means 10 times and report the average performance.

4.2   Datasets

In this paper, we use one synthetic dataset and three real-world datasets to evaluate the comparison methods. The details of four datasets are as follows. Table 1 presents the statistics of the datasets.

Synthetic Dataset: This dataset contains three views. For each view, we sample points from a two component Gaussian mixture model as instances. There are two clusters (i.e. cluster A and cluster B). Both the features and views are correlated. Specifically, the cluster means and the covariances for the three views are listed below.

$$\begin{array}{lll}
\mu_A^{(1)} = (2,2), & \Sigma_A^{(1)} = \begin{bmatrix} 1, 0.5 \\ 0.5, 2 \end{bmatrix}, & \Sigma_B^{(1)} = \begin{bmatrix} 0.3, 0.2 \\ 0.2, 0.8 \end{bmatrix} \\
\mu_B^{(1)} = (4,4), & & \\
\mu_A^{(2)} = (1,1), & \Sigma_A^{(2)} = \begin{bmatrix} 1.5, 0.2 \\ 0.2, 1 \end{bmatrix}, & \Sigma_B^{(2)} = \begin{bmatrix} 0.3, 0.2 \\ 0.2, 0.8 \end{bmatrix} \\
\mu_B^{(2)} = (3,3), & & \\
\mu_A^{(3)} = (1,2), & \Sigma_A^{(3)} = \begin{bmatrix} 1, -0.3 \\ -0.3, 1 \end{bmatrix}, & \Sigma_B^{(3)} = \begin{bmatrix} 0.5, 0.2 \\ 0.2, 0.5 \end{bmatrix}
\end{array} \tag{11}$$

Oxford Flowers Dataset (Flowers17)[1]: This dataset is composed of 17 flower categories, with 80 images for each category. Each image is described by different visual features using color, shape, and texture. χ^2 distance matrices for different flower features (color, shape, texture) are used as three different views.

Reuters Multilingual Dataset (Reuters)[2]: This dataset contains six samples of 1200 documents, balanced over the 6 labels (E21, CCAT, M11, GCAT, C15, ECAT). Each sample is made of 5 views (EN, FR, GR, IT, SP) on the same documents. The documents were initially in English, and the FR, GR, IT, and SP views corresponds to the words of their traductions respectively in French, German, Italian and Spanish.

Multi-feature digit Dataset (Mfeat) [18]: This dataset consists of features of handwritten numerals ('0'–'9') extracted from a collection of Dutch utility maps. 200 patterns per class (for a total of 2,000 patterns) have been digitized in binary images. These digits are represented in terms of the following five feature sets (files): mfeat-fou, mfeat-fac, mfeat-kar, mfeat-pix, and mfeat-zer.

Table 1. Details of the datasets

Dataset	# Instance	# Views	# Clusters
Synth	1000	3	2
Flowers	1360	3	17
Reuters	1200	5	6
Mfeat	2000	5	10

All original datasets are complete. We simulate incomplete views for them. In specific, we set incomplete ratio from 0 % to 90 % with 10 % as interval. Incomplete instances are distributed evenly in all views. Note that for each instance, it is available in at least one view.

4.3   Results

The NMIs of four datasets are plotted in Fig. 1. For synthetic data, IVC shows the best NMI. IVC, MIC and Concate preform stable even when the incomplete ratio is close to 90 %. While the NMIs of other methods drops sharply as incomplete ratio rises.

For Flowers17, all methods present the downward trends as incomplete ratio increasing. IVC shows relatively better NMI than others. MvSpec is the second best method. Note that MIC shows worst performance. The possible reason is that NMF-based method is not suitable for similarity data. (we apply MIC on kernel data of Flower17 as in original paper [13]).

As Flowers17, similar results for Reuters. IVC demonstrates slight advantage over MIC and more obvious advantage over others.

For Mfeat, in case of low incomplete ratio (i.e. when incomplete ratio is below 20 %), all methods except Concate show close NMIs. As the incomplete ratio arises, IVC shows more and more obvious superiority over others.

It can be summarized that although views are incomplete, their integration can still be more useful than single complete view. Among above multi-view methods, IVC achieves most accurate clustering for incomplete views in most cases.
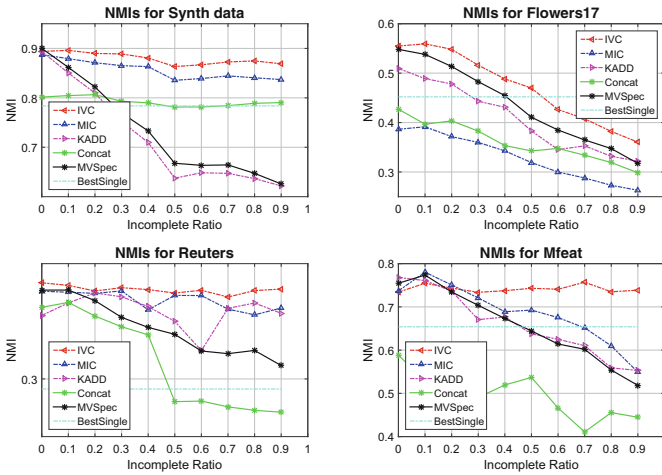

Fig. 1. NMIs

5   Conclusion

In this paper, we propose the IVC algorithm for multiple incomplete view clustering. IVC initializes incomplete views with early estimation. Based on the spectral graph theory, IVC projects original data into a new space with more discriminative grouping information. Then, individual projections are integrated. By

aligning individual projections with the projection integration, estimated part of individual projections are updated to be more accurate. With those updated individual projections, final consensus is established and thereby standard K-Means is applied on. Compared with existing works, our proposed algorithm (1) does not require any view to be complete, (2) does not limit the number of incomplete views, and (3) can handle similarity data as well as feature data. Experimental results validate the effectiveness of the IVC algorithm.

References

1. Bickel, S., Scheffer, T.: Multi-view clustering. In: ICDM, vol. 4, pp. 19–26 (2004)
2. Tzortzis, G., Likas, A.: Kernel-based weighted multi-view clustering. In: 2012 IEEE 12th International Conference on Data Mining (ICDM), pp. 675–684. IEEE (2012)
3. Chaudhuri, K., Kakade, S.M., Livescu, K., Sridharan, K.: Multi-view clustering via canonical correlation analysis. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 129–136. ACM (2009)
4. de Sa, V.R.: Spectral clustering with two views. In: ICML Workshop on Learning with Multiple Views, pp. 20–27 (2005)
5. Kumar, A., Daumé, H.: A co-training approach for multi-view spectral clustering. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 393–400 (2011)
6. Kumar, A., Rai, P., Daume, H.: Co-regularized multi-view spectral clustering. In: Advances in Neural Information Processing Systems, pp. 1413–1421 (2011)
7. Liu, J., Wang, C., Gao, J., Han, J.: Multi-view clustering via joint nonnegative matrix factorization. In: Proceedings of SDM, vol. 13, pp. 252–260. SIAM (2013)
8. Guo, Y.: Convex subspace representation learning from multi-view data. In: AAAI, vol. 1, p. 2 (2013)
9. Bruno, E., Marchand-Maillet, S.: Multiview clustering: a late fusion approach using latent models. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 736–737. ACM (2009)
10. Trivedi, A., Rai, P., Hal Daumé, I.I.I., DuVall, S.L.: Multiview clustering with incomplete views. In: NIPS Workshop (2010)
11. Shao, W., Shi, X., Yu, P.S.: Clustering on multiple incomplete datasets via collective kernel learning. In: 2013 IEEE 13th International Conference on Data Mining (ICDM), pp. 1181–1186. IEEE (2013)
12. Li, S.-Y., Jiang, Y., Zhou, Z.-H.: Partial multi-view clustering. In: Twenty-Eighth AAAI Conference on Artificial Intelligence (2014)
13. Shao, W., He, L., Yu, P.S.: Multiple incomplete views clustering via weighted nonnegative matrix factorization with $L_{2,1}$ regularization. In: Appice, A., Rodrigues, P.P., Santos Costa, V., Soares, C., Gama, J., Jorge, A. (eds.) ECML PKDD 2015. LNCS (LNAI), vol. 9284, pp. 318–334. Springer, Heidelberg (2015). doi:10.1007/978-3-319-23528-8_20
14. Yin, Q., Shu, W., Wang, L.: Incomplete multi-view clustering via subspace learning. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 383–392. ACM (2015)

15. Ng, A.Y., Jordan, M.I., Weiss, Y., et al.: On spectral clustering: analysis and an algorithm. In: Advances in Neural Information Processing Systems, vol. 2, pp. 849–856 (2002)
16. Igel, C., Glasmachers, T., Mersch, B., Pfeifer, N., Meinicke, P.: Gradient-based optimization of kernel-target alignment for sequence kernels applied to bacterial gene start detection. IEEE/ACM Trans. Comput. Biol. Bioinform. **4**(2), 216–226 (2007)
17. Cortes, C., Mohri, M., Rostamizadeh, A.: Learning non-linear combinations of kernels. In: Advances in Neural Information Processing Systems, pp. 396–404 (2009)
18. Lichman, M.: UCI machine learning repository (2013)