International Journal of Machine Learning and Cybernetics (2019) 10:2779–2790
https://doi.org/10.1007/s13042-018-00902-5

ORIGINAL ARTICLE

Social web video clustering based on multi-view clustering via nonnegative matrix factorization

Vinath Mekthanavanh[1] · Tianrui Li[1] · Hua Meng[1] · Yan Yang[1] · Jie Hu[1]

Received: 5 December 2017 / Accepted: 14 December 2018 / Published online: 2 February 2019

Abstract
Social web videos are rich data sources containing valuable information, which have a great potential to improve the performance of social web video clustering. Social web video data usually present a characteristic of multiple views. Multi-view clustering provides a useful way to generate clusters from multi-view data. Previous studies have applied different single-view data to do social web video clustering and classification; however, multi-view data has not been a factor considered in these methods. Therefore, in this paper, we propose a framework based on a novel online multi-view clustering algorithm (called SOMVCS) to cluster social web videos with large-scale possibly incomplete views into meaningful clusters. SOMVCS learns the latent feature matrices from all the views and then drives them towards a common consensus matrix based on nonnegative matrix factorization (NMF). Particularly, we incorporate graph regularization to preserve local structure information in the model. The experimental results show that online multi-view clustering via NMF is a preferable method for social web video clustering. Moreover, we find that using multi-view data with feature types from different feature families to do social web video clustering outperforms that using data with only the feature type from a single family.

Keywords Multi-view clustering · Nonnegative matrix factorization (NMF) · Social web videos mining

1 Introduction

Online video services have surged to an unprecedented level in recent years. The number of users and videos have been increasing constantly, together with the number of video contents, genres, and concepts, which leads to the difficulties of video clustering analysis task. The effective social web video clustering method can automatically organize the video corpus into a meaningful cluster (category) hierarchy for efficient searching and browsing. Social web video clustering is widely studied and much work has been done by using different types of videos' information. The standard approach consists of three main stages: (1) extracting features (videos' contents or the surrounding text information of videos) and representing videos as feature vectors; (2) similarity measurement based on video representations; (3) applying the clustering algorithm to cluster web videos based on similarity measured. However, web video clustering suffers well-known difficulty due to the calculation of the similarity between videos which comprises two problems, i.e., the video similarity which differs from visual features' similarity, and the low quality of text information because videos on video sharing website has been uploaded by unlimited levels of users. Text information, e.g., title, tag, and description, contains many synonyms, noise, ambiguous or is totally absent. Previous studies try to address this problem by using single features or merging them together. Zhang et al. [1] applied spectral clustering (SC) to cluster videos by merging two types of features such as speech transcripts and visual features. Guil et al. [2] used visual features to compare videos efficiently based on the keyframe selection. Hindle et al. [3] proposed a model for web video clustering based on two clustering algorithms, i.e., affinity

✉ Tianrui Li
trli@swjtu.edu.cn

Vinath Mekthanavanh
vinath.mek@gmail.com

Hua Meng
menghua@swjtu.edu.cn

Yan Yang
yyang@swjtu.edu.cn

Jie Hu
jiehu@swjtu.edu.cn

[1] School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China

propagation (AP) and normalization cut (NC) by using text metadata and visual features. Another useful videos' information, i.e., co-watched videos was used together with text information in [4]. Kamie et al. [5] presented a method for web video clustering by using videos' playlist information. Mahmood et al. [6] developed a framework by utilizing textual information and external support from the web to enhance the semantic relation of terms then incorporated must-link constraints into different clustering algorithms and combined clustering results by CSPA to deal with the web video categorization problem. Different types of features of web videos data have been utilized (e.g., textual features, semantic features and visual features) which are regarded as multi-modality for clustering purpose in [7].

Those studies are based on single-view data implemented in the models. However, real data usually comes from multiple sources along with the phenomenon of concept drift [8]. In many real-world applications, data is naturally comprised of different heterogeneous representations or views. For example, video data can be represented by using a collection of heterogeneous features (views), e.g., text-tag-unigrams, audio-spectrogram-stream, and vision-cuboids-histogram, which can be regarded as three views. Similarly, image data is presented by color descriptor, local shape descriptor, and local binary patterns, etc. A metadata of documents can be represented by the title, author, journals' name and etc. Usually, multiple representation or views provide compatible and complementary information for the semantically same data. By exploiting the characteristics of multi-view data, multi-view learning can obtain better performance than that relies only on a single view.

Generally, real-world datasets contain missing values, such as instances with some views missing. Trivedi et al. [9] firstly attempted to deal with incomplete views by using information from one complete view to infer the essential incomplete views. Li et al. [10] tried to solve multi-view clustering with the incomplete views by establishing a latent subspace where the instances corresponding to the same example in different views are close to each other, and similar instances (belonging to different examples) in the same view are well grouped. Shao et al. [11] learned a latent representation for all views and then generated a consensus matrix based on weighted NMF with $L_{2,1}$ regularization to deal with the multiple incomplete views. However, multi-view data in existing methods needs to fit into the memory.

Nonnegative matrix factorization (NMF) is one of the most popular techniques for dimensionality reduction of clustering by extracting latent features from high-dimensional data. NMF has become an imperative tool in conventional data analysis and widely used in the fields of data mining [12], machine learning [13], text mining [14], and so on. Zhou et al. [15, 16] proposed a new algorithm for sentiment classification to enable unsupervised sentiment analysis by constructing the nearest neighbor graphs. The formulation of a joint NMF process with the consistency constraint that pushes the clustering solution of each view towards a common consensus for multi-view data was presented in [17]. A method for feature extraction by the matrix factorization objective function by constructing a nearest neighbor to integrate local geometrical information of each view via NMF was proposed in [18]. The graph regularized NMF for data representation by encoding the geometrical information and seeking a matrix factorization was shown in [19]. Different variations of NMF were proposed to handle very large data in the last few years. Wang et al. presented an algorithm for document clustering based on online NMF by utilizing the second-order Hessian matrix to optimize the objective function [20]. Guan et al. proposed a novel online NMF algorithm with the robust stochastic approximation that accepts one sample or a chunk of samples and updates the bases per step [21]. Shao et al. [22] developed an online algorithm by adding graph regularization to joint NMF to help select discriminative features for multi-view feature selection.

Recently, multi-view clustering has gained increasing attention due to its high performance which performs with the information from multiple views to generate clusters [23, 24]. In order to enforce the indicator matrices from different views, Akata et al. [25] extended the NMF [26] to multi-view settings. To combine multi-view information in the NMF framework, Akata et al. [25] enforced a shared indicator matrix among different views to perform multi-view clustering in NMF framework. In order to keep the clustering solutions across different views meaningful and comparable, Liu et al. [17] enforced a constraint to push each view indicator matrix towards a common indicator matrix and normalization constraint inspired by the connection between NMF and probability latent semantic. A number of NMF-based multi-view algorithms have been proposed, such as factorizing each view as a linear combination of a shared latent representation in [27] or establishing a latent subspace in [10] where the instances corresponding to the same example in different views are close to each other and similar instances in the same view are well grouped. The disagreement between each pair of views is minimized the loss function of NMF in [28] by indicating that the vectors from two different views should be similar if they belong to the same cluster and dissimilar otherwise. A weighted extension of multi-view NMF in [29] imposed consensus constraint on the efficient matrices across different features (views) for image annotation.

In this paper, we tackle the problem of Social Web Video Clustering based on Multi-View Clustering via NMF (SOMVCS). SOMVCS deals with large-scale multi-view data of web videos. The framework of SOMVCS is illustrated in Fig. 1.

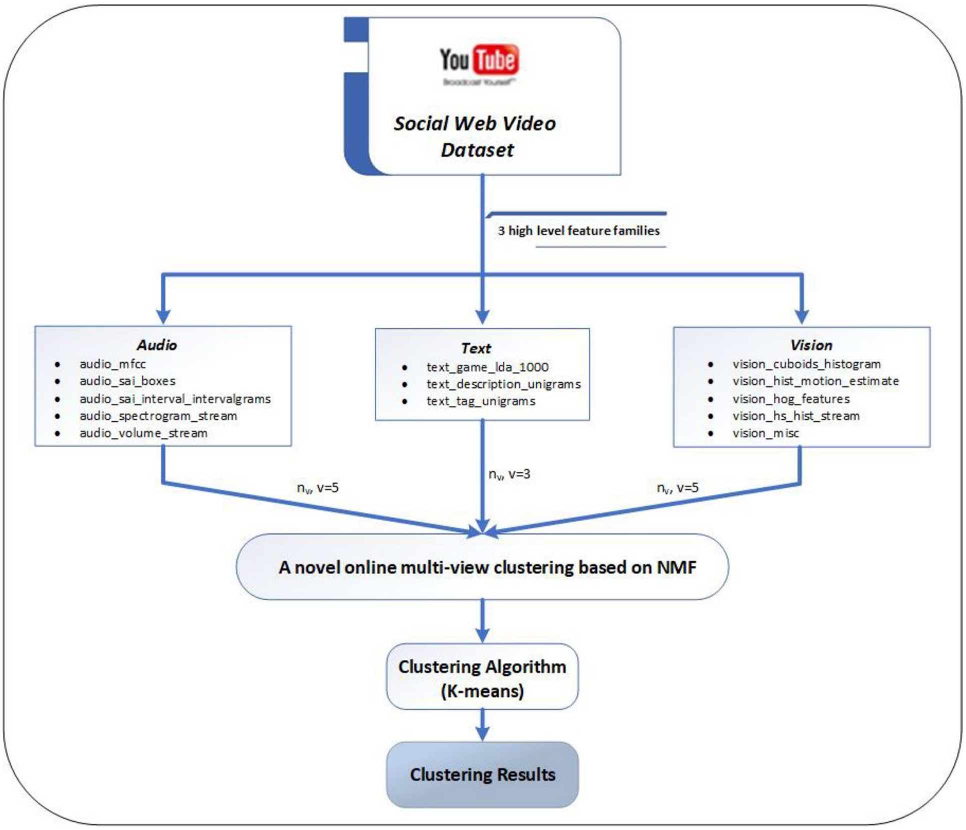

Fig. 1 The proposed framework of SOMVCS

The contributions of this paper are as follows:

- We propose SOMVCS, which models the multi-view clustering as a joint weighted NMF problem, that can capture relation between heterogeneous views.
- SOMVCS incorporates the graph regularization to preserve the local structure of data and processes the multi-view data chunk by chunk, which will be fit into the memory.
- It learns latent feature matrices across all views and drives them towards to a common consensus. By applying clustering algorithm, SOMVCS provides accurate results.

The rest of this paper is organized as follows. In Sect. 2, a brief review of NMF is described. The formulation and the algorithm of SOMVCS are presented in Sect. 3. The experimental study is provided in Sect. 4. The results and discussion are shown in Sect. 5. The paper ends with conclusions and future work in Sect. 6.

2 A brief review of NMF

In this section, we will introduce some knowledge about clustering based NMF. It is well understood that NMF is used for feature extraction in data mining and it is a method to factorize a nonnegative data matrix into two nonnegative matrix factors where the product of these two factors is an approximation of the original matrix. As the decomposition result is nonnegative, the interpretability of NMF outputs makes it fit on large-scale and time-varying datasets.

Given a nonnegative data matrix $X \in R_{+}^{D\times N}$, where each row represents a feature and each column represents an instance. NMF aims to transform the data matrix X into two rank-r nonnegative matrices, that is a basis matrix $U \in R_{+}^{D\times r}$ and a feature matrix $V \in R_{+}^{N\times r}$. U and V are determined by minimizing the cost function as follows:

$$\min_{U,V} L = \left\| X - UV^T \right\|_F^2 \quad s.t.\ U \geq 0,\ \ V \geq 0, \tag{1}$$

where $\|\cdot\|$ is the matrix Frobenius norm.

In general, solving Eq. (1) is difficult as the objective function is not convex for U and V jointly. One of the well-known algorithms for implementing the alternate update rules is Lee and Seung's multiplicative update approach in [26] which updates U and V by

$$U_{i,j} \leftarrow U_{i,j} \frac{(XV)_{i,j}}{(UV^T V)_{i,j}} \tag{2}$$

$$V_{i,j} \leftarrow V_{i,j} \frac{(X^T U)_{i,j}}{(VU^T U)_{i,j}}. \tag{3}$$

However, NMF still does not produce satisfactory results in clustering tasks because it cannot theoretically guarantee the composition results to be sparse. The existing NMF solutions require the entire data matrix to reside in the memory which is problematic when the datasets are very large or streaming. To address this issue, Wang et al. [20] proposed an online NMF algorithm based on a stochastic approximation to efficiently handle streaming data by processing one chunk of data points at a time.

3 A novel online multi-view clustering algorithm

3.1 Formulation

In this section, we present the SOMCVS which processes multi-view data chunk by chunk in a streaming fashion with low storage complexity. In order to deal with incomplete views (missing instances), dynamic weights from online multi-view clustering [30] were used to assign lower weights to less informative estimations and higher weights to more informative estimations. SOMVCS provides us a common consensus matrix with a latent representation of the original data points. Finally, k-means algorithm is applied to generate the result.

Suppose we have a multi-view data set (incomplete) $X = \{X^{(1)}, X^{(2)}, \dots, X^{(n_v)}\}$ denoting the data of all the views, where $X^{(v)} = \{x_1^{(v)}, x_2^{(v)}, \dots, x_n^{(v)}\} \in R^{D_v \times N}$ represents the data of the v-th view. In NMF, the coefficient matrices $V^{(v)}$ is learned from different views which give forward a common consensus matrix V^*. This consensus matrix is considered to reflect simultaneously the latent structure shared by every view [17].

In the study of manifold learning theory, it has shown that respecting the intrinsic geometrical structure is an effective method that can improve clustering quality [19]. Therefore, we introduce additional graph regularization. Given a similarity matrix S, one can define a smoothness penalty term for each view using the following function f:

$$\begin{aligned} f &= \sum_{i,j=1}^{N} \left\| V_{i,:}^{(v)} - V_{j,:}^{(v)} \right\|^2 \times S_{ij}^{(v)} \\ &= Tr\left(V^{(v)^T} D^{(v)} V^{(v)} \right) - Tr\left(V^{(v)^T} S^{(v)} V^{(v)} \right) \\ &= Tr\left(V^{(v)^T} L^{(v)} V^{(v)} \right) \end{aligned} \tag{4}$$

where $Tr(\cdot)$ denotes the trace of a matrix and $D^{(v)}$ is a diagonal matrix such that, $D_{i,i}^{(v)} = \sum_i S_{i,j}^{(v)}$ or $D_{i,i}^{(v)} = \sum_j S_{i,j}^{(v)}$. $L^{(v)} = D^{(v)} - S^{(v)}$ is called graph Laplacian matrix. Therefore, SOMCVS for multi-view clustering is formulated as

$$\begin{aligned} \min_{U^{(v)},V^{(v)},V^{(*)}} L = &\sum_{v=1}^{n_v} \left\| X^{(v)} - U^{(v)} V^{(v)^T} \right\|_F^2 + \sum_{v=1}^{n_v} \alpha_v Tr\left(V^{(v)^T} L^{(v)} V^{(v)} \right) \\ &+ \sum_{v=1}^{n_v} \beta_v \left\| V^{(v)} - V^* \right\|_F^2 \\ &s.t.\ U^{(v)} \geq 0,\ V^{(v)} \geq 0,\ V^* \geq 0,\ v = 1, 2, \dots, n_v \end{aligned} \tag{5}$$

where $U^{(v)} \in R^{D_v \times K}$ and $V^{(v)} \in R^{N \times K}$ are the basis matrix and the latent feature matrix for the v-th view, $V^* \in R^{N \times K}$ is the consensus latent feature matrix across all views, α_v and β_v are the trade-off parameters used to control the contribution of the manifold regularization and reconstruction error between view v and the consensus, and K is the number of clusters.

In order to support incompleteness, the assigning of dynamic weights in [30] such as assigning lower weights to less informative estimations and higher weights to more informative estimations have been used. After adding the weight matrices, Eq. (5) becomes:

$$\begin{aligned} L = &\sum_{v=1}^{n_v} \left\| \omega^{(v)} \left(X^{(v)} - U^{(v)} V^{(v)^T} \right) \right\|_F^2 + \sum_{v=1}^{n_v} \alpha_v Tr\left(V^{(v)^T} L^{(v)} V^{(v)} \right) \\ &+ \sum_{v=1}^{n_v} \beta_v \left\| \omega^{(v)} \left(V^{(v)} - V^* \right) \right\|_F^2 \\ &s.t.\ U^{(v)} \geq 0,\ V^{(v)} \geq 0,\ V^* \geq 0,\ v = 1, 2, \dots, n_v. \end{aligned} \tag{6}$$

Because SOMVCS is for social web videos clustering, the data matrices are too large to fit into the memory. Hence, it is crucial to solving the above optimization in an incremental way. Let $X_t^{(v)} \in R^{z \times D_v}$ denote the data received at time t, where z is the number of instances (size of the data chunk). Equation (6) can be written as follows:

$$\begin{aligned} L = &\sum_{v=1}^{n_v} \sum_{t=1}^{[N/z]} \left\| \omega_t^{(v)} \left(X_t^{(v)} - U^{(v)} V_t^{(v)^T} \right) \right\|_F^2 + \sum_{v=1}^{n_v} \sum_{t=1}^{[N/z]} \alpha_v Tr\left(V_t^{(v)^T} L_t^{(v)} V_t^{(v)} \right) \\ &+ \sum_{v=1}^{n_n} \sum_{t=1}^{[N/z]} \beta_v \left\| \omega_t^{(v)} \left(V_t^{(v)} - V_t^* \right) \right\|_F^2 \\ &s.t.\ U^{(v)} \geq 0,\ V_t^{(v)} \geq 0,\ V_t^* \geq 0,\ v = 1, 2, \dots, n_v \end{aligned} \tag{7}$$

where $X_t^{(v)}$ is a data chunk at time t in the v-th view, $\omega_t^{(t)} \in R^{z \times z}$ is the diagonal weight matrix for data chunk at time t, $V_t^{(v)} \in R^{z \times K}$ is the latent feature matrix for t-th data chunk in the v-th view, and $V_t^{(v)} \in R^{z \times K}$ is the consensus latent feature matrix for t-th data chunk across all the views.

3.2 Algorithm

In this section, we solve the objective function of SOMVCS derived in the above section. From Eq. (7), we can see that for each time t, we need to update $U^{(v)}$, $V^{(v)}$ and V_t^* in an alternate way since the objective function is not jointly convex. The optimization problem of SOMVCS is described as follows:

(1) Optimize $U^{(v)}$ with $V_t^{(v)}$ and V_t^* fixed: To optimize $U^{(v)}$ for v-th view at time t, we only need to minimize the following objective function:

$$T_t\left(U^{(v)}\right) = \sum_{i=1}^{t} \left\| \omega_i^{(v)} \left(X_i^{(v)} - U^{(v)} V_i^{(v)^T} \right) \right\|_F^2 \qquad s.t.\ U^{(v)} \geq 0. \tag{8}$$

Taking the derivative of the T_t with respect to $U^{(v)}$, we have:

$$\frac{\partial T_t}{\partial U^{(v)}} = 2U^{(v)} \sum_{i=1}^{t} V_i^{(v)^T} \tilde{\omega}_i^{(v)} V_i^{(v)} - 2 \sum_{i=1}^{t} X_i^{(v)} \tilde{\omega}_i^{(v)} V_i^{(v)} \tag{9}$$

where $\tilde{\omega}_i^{(v)} = \omega_i^{(v)} \omega_i^{(v)^T} = \omega_i^{(v)^T} \omega_i^{(v)}$.

We introduce two terms $M_t^{(v)}$ and $N_t^{(v)}$ as

$$M_t^{(v)} = \sum_{i=1}^{t} V_i^{(v)^T} \tilde{\omega}_i^{(v)} V_i^{(v)} \tag{10}$$

$$N_t^{(v)} = \sum_{i=1}^{t} X_i^{(v)} \tilde{\omega}_i^{(v)} V_i^{(v)}. \tag{11}$$

Using the Karush–Kuhn–Tucker condition for the nonnegativity constraint on $U^{(v)}$, the update rule for $U^{(v)}$ is:

$$U_{i,k}^{(v)} \leftarrow U_{i,k}^{(v)} \sqrt{\frac{\left(N_t^{(v)}\right)_{i,k}}{\left(U^{(v)}\left(M_t^{(v)}\right)\right)_{i,k}}}. \tag{12}$$

(2) Optimize $V_t^{(v)}$ with V_t^* and $U^{(v)}$ fixed: Using the same steps as before, then let ψ_{ij} be the Lagrange multiplier for nonnegative constraint $V_t^{(v)}$ and $\Psi = [\psi_{ij}]$, and using the Kuhn–Tucker condition $\psi_{ij}(V_t^{(v)})_{ij} = 0$. The objective function becomes

$$T\left(V_t^{(v)}\right) = \left\| \omega_t^{(v)} \left(X_t^{(v)} - U^{(v)} V_t^{(v)^T} \right) \right\|_F^2 + \alpha_v Tr\left(V_t^{(v)^T} L_t^{(v)} V_t^{(v)} \right) + \beta_v \left\| \omega_t^{(v)} \left(V_t^{(v)} - V_t^* \right) \right\|_F^2 + Tr\left(\Psi V_t^{(v)^T} \right). \tag{13}$$

Taking the partial derivative of T with respect to $V_t^{(v)}$, we have

$$\frac{\partial T}{\partial V_t^{(v)}} = 2\tilde{\omega}_t^{(v)} U^{(v)^T} V_t^{(v)} U^{(v)} - 2\tilde{\omega}_t^{(v)} X_t^{(v)^T} U^{(v)} + 2\alpha_v L_t^{(v)} V_t^{(v)} + 2\beta_v \tilde{\omega}_t^{(v)} V_t^{(v)} - 2\beta_v \tilde{\omega}_t^{(v)} V_t^* + \Psi. \tag{14}$$

Following the steps as above, the update rule for $V_t^{(v)}$ is as following Eq. (15) by giving the graph Laplacian matrix $L_t^{(v)} = D_t^{(v)} - S_t^{(v)}$:

$$\left(V_t^{(v)}\right)_{j,k} \leftarrow \left(V_t^{(v)}\right)_{j,k} \sqrt{\frac{\left(\tilde{\omega}_t^{(v)} X_t^{(v)^T} U^{(v)} + \alpha_v S_t^{(v)} V_t^{(v)} + \beta_v \tilde{\omega}_t^{(v)} V_t^*\right)_{j,k}}{\left(\tilde{\omega}_t^{(v)} U^{(v)^T} V_t^{(v)} U^{(v)} + \alpha_v D_t^{(v)} V_t^{(v)} + \beta_v \tilde{\omega}_t^{(v)} V_t^{(v)}\right)_{j,k}}}. \tag{15}$$

(3) Optimize V_t^* with $V_t^{(v)}$ and $U^{(v)}$ fixed: To optimize the consensus V_t^*, it requires minimizing the following objective function:

$$T(V_t^*) = \sum_{v=1}^{n_v} \beta_v \left\| \omega_t^{(v)} \left(V_t^{(v)} - V_t^* \right) \right\|_F^2 \qquad s.t.\ V_t^* \geq 0. \tag{16}$$

Taking the derivative of the objective function $T(V_t^*)$, we have

$$\frac{\partial T}{\partial V_t^*} = \sum_{v=1}^{n_v} \beta_v \tilde{\omega}_t^{(v)} \left(-2V_t^{(v)} + 2V_t^* \right) = 0. \tag{17}$$

Therefore, we have a closed form solution:

$$V_t^* = \frac{\sum_{v=1}^{n_v} \beta_v \tilde{\omega}_t^{(v)} V_t^{(v)}}{\sum_{v=1}^{n_v} \beta_v \tilde{\omega}_t^{(v)}} = \sum_{v=1}^{n_v} \beta_v \tilde{\omega}_t^{(v)} V_t^{(v)}. \tag{18}$$

Table 1 A description of the dataset

Dataset	# Instances	Feature family	# Views	# Classes
DS1	12,177	Audio, text, vision	3(2000, 1000, 512)	31
DS2	11,958	Text	3(12,183,626, 1000, 12,178,696)	31
DS3	12,177	Audio, vision	3(7168, 512, 838)	31
DS4	12,177	Audio, text, vision	5(7168, 4096, 1024, 1000, 838)	31
DS5	12,177	Audio, text, vision	7(2000, 7168, 4096, 1024, 1000, 512, 838)	31
DS6	11,958	Audio, text, vision	9(2000, 7168, 4096, 1024, 12,183,626, 1000, 12,178,696, 512, 838)	31
DS7	98,500	Audio, text, vision	3(2000, 1000, 838)	31
DS8	98,500	Audio, text, vision	3(7168, 1000, 512)	31

```
Algorithm 1. (The SOMVCS Algorithm)
Input: Web videos multi-view data matrices of all (possibly incomplete) views, $\{X^{(1)}, X^{(2)}, \ldots, X^{(n_v)}\}$ parameters $\{\alpha_v, \beta_v\}$, number of cluster $K$, the batch size (data points) $z$.
Output: Basis Matrices $\{U^{(1)}, U^{(2)}, \ldots, U^{(v)}\}$, representation matrices $\{V_t^{(1)}, V_t^{(2)}, \ldots, V_t^{(v)}\}$, consensus matrix $\{V_t^*\}$, and clustering results.
Begin
    for $t = 1: [N/z]$ do
        Normalize each view $X_t^{(v)}$and construct the graph Laplacian matrix $L_t^{(v)}$.
        Split $X_t^{(v)}$(i.e., $z$ data points) for all the views.
        Infuse missing instances and the weights.
        repeat
            for $v = 1: n_v$ do
                Fixing $V_t^{(v)}$and $V_t^*$, update $U^{(v)}$ by Eq. (12).
                Fixing $U^{(v)}$ and $V_t^*$, update $V_t^{(v)}$ by Eq. (15).
            end
            Fixing $U^{(v)}$ and $V_t^{(v)}$, update $V_t^*$ by Eq. (18).
        until Convergence;
        $M_t^{(v)} = M_{t-1}^{(v)} + V_t^{(v)^T} \tilde{\omega}_t^{(v)} V_t^{(v)}$
        $N_t^{(v)} = N_{t-1}^{(v)} + X_t^{(v)} \tilde{\omega}_t^{(v)} V_t^{(v)}$
        Apply K-means to $V_t^*$ to get the clustering results.
    end
end
```

4 Experimental study

4.1 Dataset

In this paper, we use a publicly available real-world large-scale dataset (YouTube[1]) that is named YouTube Multiview Video Games. This dataset consists of feature values and class labels about 120,000 videos. Each video is described by up to 13 feature types from 3 high-level feature families such as auditory, textual and visual features. There are 31 class labels corresponding to 30 popular video games and the remaining correspond to other games [31]. The data is very valuable to do web videos clustering due to their quality of the feature representation. The subset of the dataset used in the experiments is shown in Table 1.

[1] http://archive.ics.uci.edu/ml/datasets/youtube+multiview+video+games+dataset.

4.2 Performance evaluation

In our experiments, the clustering performance is measured by using two popular evaluation metrices. The first metric is the ACCuracy (ACC) [32, 33]. Given a video z_i, let l_i and c_i be the cluster label and the label provided by the data corpus, respectively. The ACC is defined as follows:

$$ACC = \frac{\sum_{i=1}^{N} \delta(c_i, map(l_i))}{N} \tag{19}$$

where N is the total number of videos in the test, $\delta(a, b)$ is the delta function that equals to one if $a = b$ and equals to zero otherwise, and $map(l_i)$ is the permutation mapping function that maps each cluster label l_i to the equivalent label from the data corpus. The best mapping can be found by using the Kuhn–Munkres algorithm [34].

The second metric is normalized mutual information (NMI) [32, 33]. Let C is the set of clusters obtained from the ground truth and C' is the clusters obtained from our algorithm. Their mutual information metric $MI(C, C')$ is defined as follows:

$$MI(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, {c'}_j) \cdot \log_2 \frac{p(c_i, {c'}_j)}{p(c_i) \cdot p({c'}_j)} \tag{20}$$

where $p(c_i)$ and $p(c'_j)$ are the probabilities that a video arbitrarily selected from the corpus belongs the clusters c_i and c'_j, respectively, and $p(c_i, c'_j)$ is the joint probability that the arbitrarily selected video belongs to the clusters c_i as well as c'_j at the same time. In our experiments, we use the NMI as follows:

$$NMI(C, C') = \frac{MI(C, C')}{\max(H(C), H(C'))} \tag{21}$$

where $H(C)$ and $H(C')$ are the entropies of C and C', respectively. $NMI(C, C')$ takes values between 0 and 1. $NMI = 1$ if the two sets of clusters are identical, and $NMI = 0$ if the two sets of clusters are independent.

The dataset used in our experiments is so big to fit in the memory that it is impossible to keep this data in the memory while running the model. We compare the results of the proposed SOMVCS with ONMF [20] and OMVC [30]. The other existing methods such as GMNMF [18], MIC [11] and MultiNMF [17] are the off-line methods based on NMF which can not apply to the above dataset since these methods are designed for keeping the data in the memory. ONMF is proposed for solving the problem of single-view. In order to apply ONMF to the above dataset, we concatenate views into one single-view. After the concatenation of views of the original datasets, data becomes so large that it will take too much time to process. Therefore, in the experiments, we omit DS5 and DS6 since both of them comprise 7 and 9 views, respectively.

5 Results and discussion

5.1 Results

To show the clustering performance of the proposed SOMVCS, we set the chunk size z to be 2500 for the datasets DS1–DS6 and 3000 for the datasets DS7 and DS8. In our experiments, we notice that applying multiple passes is better than applying one pass. This is because, with the first input few data point the performance of clustering may not be satisfactory, and if multiple passes are feasible, the performance of clustering may get a chance to improve. We report the ACC and NMI in Table 2. In Table 2, based on our introduce of additional manifold regularization term, we give the results with $\alpha = 0$ as a baseline.

From Table 2, we can observe that the performance of the proposed SOMVCS outperforms the compared method for all the datasets in terms of ACC and NMI. According to the different significance of each feature type, some of them have a high performance while some have a low performance as reported by Madani et al. [31] in UCI dataset. The performance on DS2 is worse than the others because it consists of feature types from only the feature family such as “text”. The performance becomes better when the dataset includes feature types from up to two or three feature families together (e.g., DS1 and DS3). Therefore, in this experiment, we design the dataset comprising feature types from variant numbers of feature families. It means that when the data consists of feature types from more feature families, it will perform better for clustering purpose. The maximum number of feature type we include in the dataset in our experiments is 9 (DS6); Otherwise, the datasets would be too large for batch learning.

The results demonstrate that the proposed method can learn a better feature representation. It is good for learning with the large-scale data, especially the data from real-world applications as YouTube[2]. In SOMVSC, the performance improves when the number of passes increases. We set the number of passes to be 10 because after the several passes the result has already improved and then varies a little up and down. The graph for comparing the performance between SOMVCS, ONMF, OMVC is shown in Figs. 2, 3 (ACC) and Figs. 4, 5 (NMI).

From these figures, we can observe that the proposed SOMVCS gets close performance within the first pass and continue to improve after several passes. The advantages

[2] http://www.youtube.com.

Table 2 Accuracy and NMI of ONMF, OMVC, and SOMVCS

Method	DS1		DS2		DS3		DS4	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
SOMVCS (α = 0)								
Min	0.0458	0.0119	0.0197	0.0101	0.0551	0.0129	0.0572	0.0204
Max	0.0525	0.0136	0.0354	0.0120	0.0576	0.0130	0.0620	0.0217
Avg	0.0491	0.0127	0.0275	0.0110	0.0563	0.0129	0.0596	0.0210
ONMF								
Min	0.0544	0.0128	0.0203	0.0102	0.0550	0.0107	0.0677	0.0525
Max	0.0669	0.0146	0.0425	0.0125	0.0599	0.0194	0.0783	0.0656
Avg	0.0606	0.0137	0.0314	0.0113	0.0574	0.0150	0.0730	0.0590
OMVC								
Min	0.0654	0.0149	0.0300	0.0113	0.0676	0.0129	0.0766	0.0669
Max	0.0895	0.0156	0.0510	0.0145	0.0701	0.0221	0.0820	0.0682
Avg	0.0774	0.0156	0.0405	0.0129	0.0688	0.0175	0.0793	0.0675
SOMVCS								
Min	0.0702	0.0349	0.0401	0.0129	0.0576	0.0203	0.1023	0.0735
Max	0.0925	0.0362	0.0599	0.0182	0.0820	0.0299	0.1215	0.0837
Avg	0.0813	0.0355	0.0500	0.0155	0.0698	0.0251	0.1119	0.0786
Method	DS5		DS6		DS7		DS8	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
SOMVCS (α = 0)								
Min	0.1000	0.0427	0.1053	0.0370	0.0594	0.0180	0.0527	0.0179
Max	0.1006	0.0475	0.1068	0.0372	0.0656	0.0195	0.0635	0.0286
Avg	0.1003	0.0451	0.1060	0.0371	0.0625	0.0187	0.0581	0.0232
ONMF								
Min	–	–	–	–	0.0674	0.0184	0.0724	0.0186
Max	–	–	–	–	0.0790	0.0210	0.0881	0.0294
Avg					0.0732	0.0197	0.0802	0.0240
OMVC								
Min	0.1117	0.0876	0.1143	0.0903	0.0903	0.0222	0.0876	0.0265
Max	0.1262	0.0978	0.1285	0.0985	0.1068	0.0246	0.0978	0.0329
Avg	0.1189	0.0927	0.1214	0.0944	0.0985	0.0234	0.0927	0.0297
SOMVCS								
Min	0.1204	0.1031	0.1259	0.1035	0.1000	0.0214	0.1003	0.0349
Max	0.1470	0.1051	0.1524	0.1058	0.1078	0.0258	0.1099	0.0389
Avg	0.1337	0.1041	0.1391	0.1046	0.1039	0.0236	0.1051	0.0369

of the SOMVCS is that it utilizes a weight matrix for each view (possibly incomplete view), the intrinsic geometry of the data distribution and sparsity of the latent features, while OMVC and ONMF does not consider the intrinsic geometry, and also ONMF does not consider the possibly incomplete data and sparsity of the data.

5.2 Parameter study

The parameters selection aims to provide a practical guidance of the proposed method. There are two essential parameters, $\{\alpha_v\}$ and $\{\beta_v\}$ in SOMVCS. Basically, we set α_v to be the same for all the views and also set β_v the same as α_v. We run SOMVCS with different values for $\{\alpha_v\}$ and $\{\beta_v\}$ on DS1. We only show the results in ACC and NMI on DS1 since we have similar observation on the other datasets.

From Fig. 6, assume that $\alpha_v = \alpha$ and $\beta_v = \beta$ we can see that the proposed SOMVCS method is sensitive to the parameters α and β. Parameter α controls the contribution of the manifold regularization of the views. Parameter β controls the co-regularization between views and the consensus. We can observe that when β becomes small, the consensus has more contributed to the learning of each view, meanwhile when it is too small, the performance will not be stable

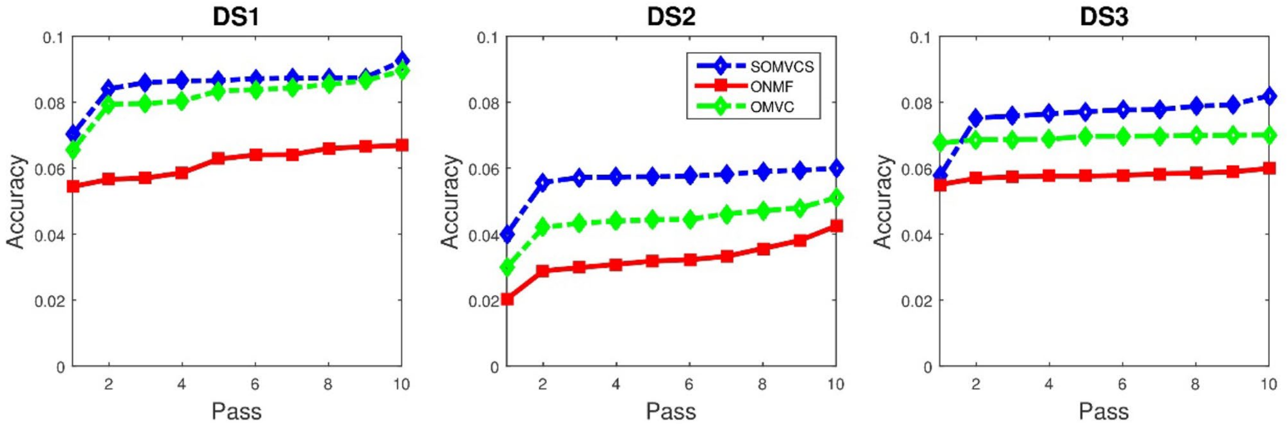

Fig. 2 Accuracy on DS-1, 2 and 3

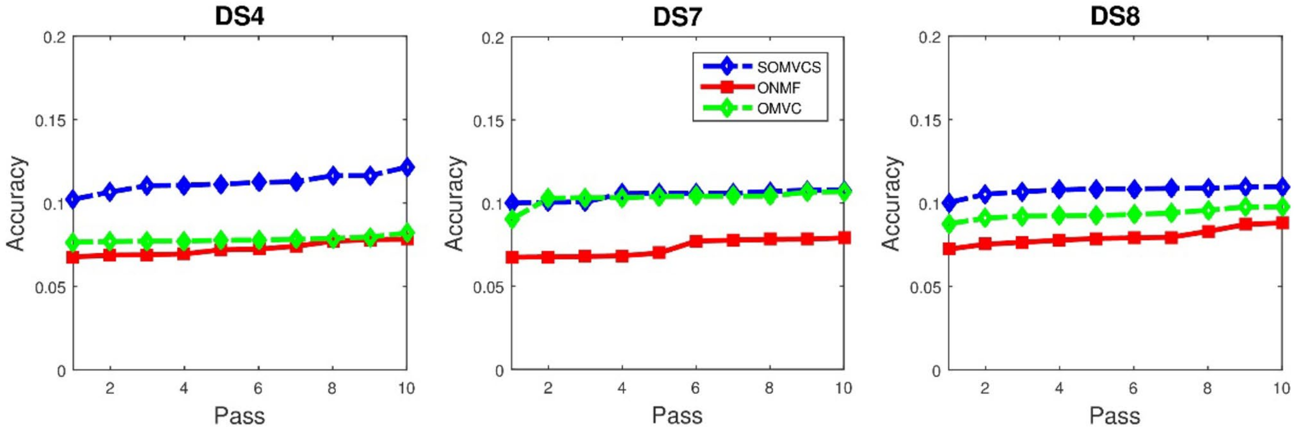

Fig. 3 Accuracy on DS-4, 7 and 8

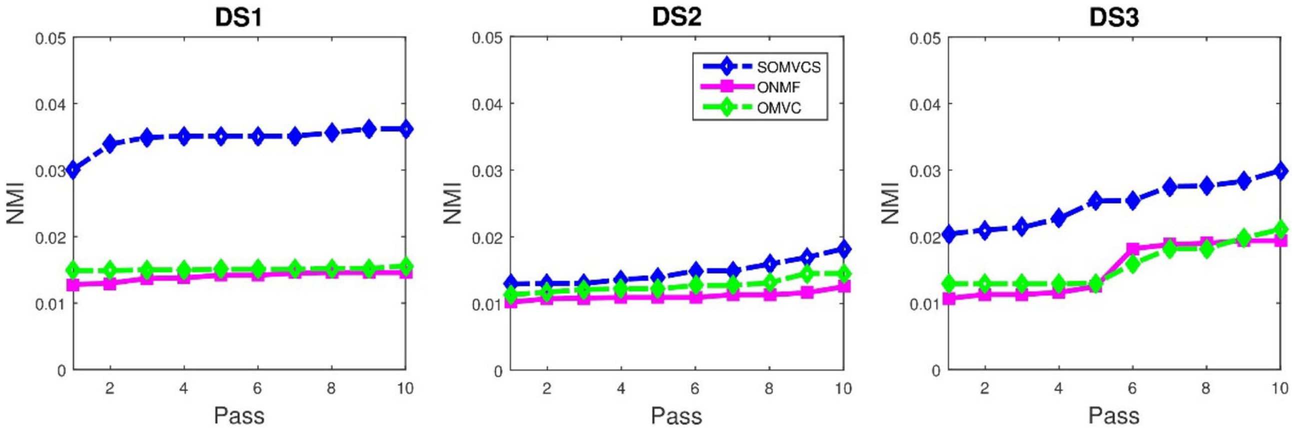

Fig. 4 NMI on DS-1, 2 and 3

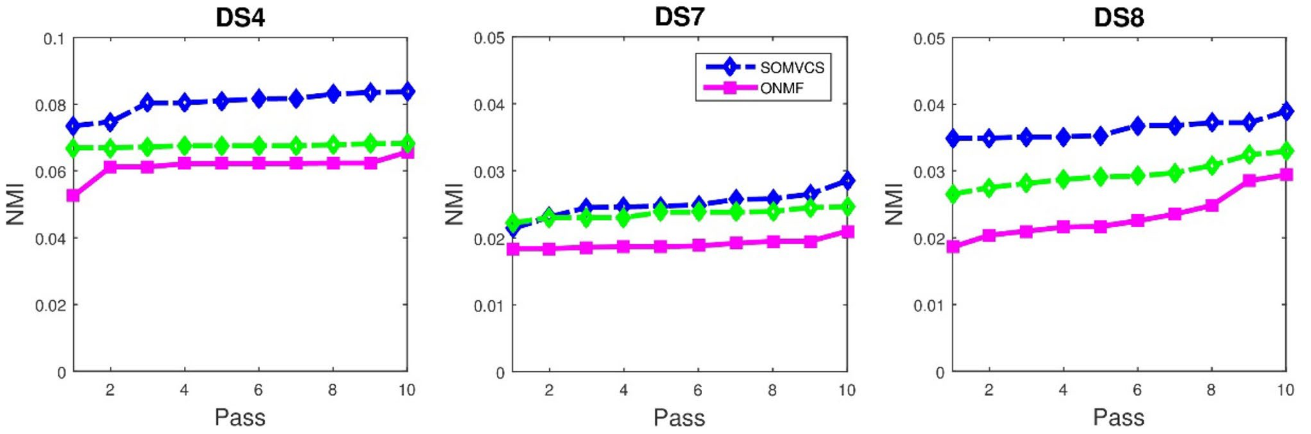

Fig. 5 NMI on DS-4, 7 and 8

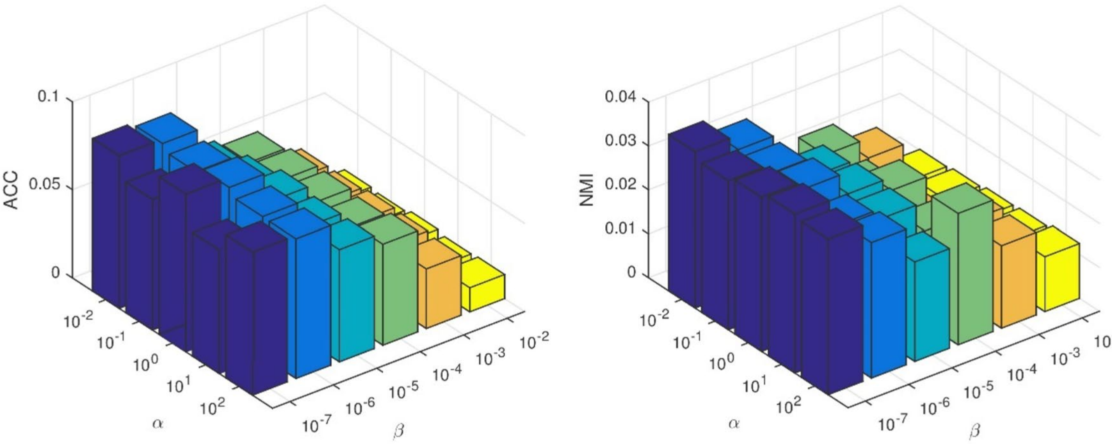

Fig. 6 ACC and NMI of SOMVCS with different α_v, β_v on DS1

(ACC) and finally will decline. When the value of parameter β increases, the consensus has more influence to each view and the performance will decrease. SOMVCS achieves stably good performance when β is in $[10^{-6}, 10^{-7}]$ (NMI). From these results, the selection of an appropriate value of α and β shows that a combination of these two parameters affects the performance of the SOMVCS.

5.3 Results discussion

In summary, we have the following conclusions for our proposed method:

(a) The idea of utilizing the multi-view clustering via NMF technique for social web video clustering has shown a promising performance.

(b) Multi-view data with multi-view learning can achieve better performance than relying on just single-view or either multi-view data from the same feature family (Videos).

(c) The results of multi-view data comprising more feature types from different feature families can achieve better results (e.g., the result of DS6 is better than that of DS2).

(d) The performance improves when multiple passes are feasible.

(e) The proposed SOMVCS has outperformed the state-of-the-art in most cases. The ACC and NMI are not high since the method does not hold all data in the memory but split the data into the chunks. The results are obtained at the time *t* according to the number of passes.

6 Conclusion

In this paper, we presented a novel online multi-view clustering algorithm based on NMF called (SOMVCS), which processes the incoming data chunk by chunk at a time. SOMVCS does not require to hold the whole data matrix in the memory, which can reduce the storage complexity. It learns the latent feature matrices for each view (possibly incomplete view) and merges them into a common consensus. Graph Laplacian regularization enables SOMVCS to exploit the intrinsic geometry of the data distribution in each incomplete view. Our method can scale up to large-scale video datasets. The experimental results demonstrated the effectiveness of the proposed SOMVCS. As the SOMVCS algorithm is designed for dealing with the data in high-dimensional space, we will looking for other real-world datasets of social media as the benchmark and try to extend our algorithm with other clustering algorithms to further improve its performance in the future work.

Acknowledgements This work was supported by the National Science Foundation of China (nos. 61573292, 61572407, 61603313) and the Fundamental Research Funds for the Central Universities (No. 2682017CX097).

References

1. Zhang DQ, Lin CY, Chang SF, Smith JR (2004) Semantic video clustering across sources using bipartite spectral clustering. In: Proceedings of international conference on multimedia and expo (ICME), vol 1. IEEE, pp 117–120
2. Guil N, González-Linares JM, Cózar JR, Zapata EL (2007) A clustering technique for video copy detection. In: Proceedings of the 3rd Iberian conference on pattern recognition and image analysis, Part I, pp 451–458
3. Hindle A, Shao J, Lin D, Lu J, Zhang R (2011) Clustering web video search results based on integration of multiple features. World Wide Web 14(1):53–73
4. Gargi U, Lu W, Mirrokni VS, Yoon S (2011) Large-scale community detection on YouTube for topic discovery and exploration. In: Proceedings of 5th international conference on weblogs and social media. The AAAI Press, Palo Alto, pp 486–489
5. Kamie M, Hashimoto T, Kitagawa H (2012) Effective web video clustering using playlist information. In: Proceedings of the 27th annual ACM symposium on applied computing. ACM, New York, pp 949–956
6. Mahmood A, Li T, Yang Y, Wang H, Afzal M (2015) Semi-supervised evolutionary ensembles for web video categorization. Knowl Based Syst 76:53–66
7. Mekthanavanh V, Li T (2016) Social web videos clustering based on ensemble technique. In: Proceedings of international joint conference on rough sets. Springer, Berlin, pp 449–458
8. Wang H, Fan W, Yu PS, Han J (2003) Mining concept drifting data streams using ensemble classifiers. In: Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining, pp 226–235
9. Trivedi A, Rai P, Daume H, DuVall SL (2010) Multiview clustering with incomplete views. In: NIPS workshop on machine learning for social computing, Whistler
10. Li SY, Jiang Y, Zhou ZH (2014) Partial multi-view clustering. In: Proceedings of the 28th AAAI conference on artificial intelligence, pp 1968–1974
11. Shao W, He L, Philip SY (2015) Multiple incomplete views clustering via weighted nonnegative matrix factorization with L2;1 regularization. In: Proceedings of the joint European conference on machine learning and knowledge discovery in databases. Springer, Berlin, pp 318–334
12. Cichocki A, Zdunek R, Phan AH, Amari S (2009) Nonnegative matrix and tensor factorizations: applications to exploratory multiway data analysis and blind source separation. Wiley, New York
13. Berry MW, Browne M, Langville AN, Pauca VP, Plemmons RJ (2007) Algorithms and applications for approximate nonnegative matrix factorization. Comput Stat Data Anal 52(1):155–173
14. Ding C, Li T, Peng W, Park H (2006) Orthogonal nonnegative matrix t-factorizations for clustering. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 126–135
15. Zhou G, Zhao J, Zeng D (2014) Sentiment classification with graph co-regularization. In: Proceedings of COLING 2014, the 25th international conference on computational linguistics, pp 1331–1340
16. Zhou G, He T, Wu W, Hu XT (2015) Linking heterogeneous input features with pivots for domain adaptation. In: Proceedings of the 24th international joint conference on artificial intelligence, pp 1419–1425
17. Liu J, Wang C, Gao J, Han J (2013) Multi-view clustering via joint nonnegative matrix factorization. In: Proceedings of the 2013 SIAM international conference on data mining, pp 252–260
18. Wang Z, Kong X, Fu H, Li M, Zhang Y (2015) Feature extraction via multi-view non-negative matrix factorization with local graph regularization. In: 2015 IEEE international conference on image processing (ICIP), pp 3500–3504
19. Cai D, He X, Han J, Huang TS (2011) Graph regularized nonnegative matrix factorization for data representation. IEEE Trans Pattern Anal Mach Intell 33(8):1548–1560
20. Wang F, Tan C, Li P, Konig AC (2011) Efficient document clustering via online nonnegative matrix factorizations. In: Proceedings of the 2011 SIAM international conference on data mining. Society for industrial and applied mathematics, pp 908–919
21. Guan N, Tao D, Luo Z, Yuan B (2012) Online nonnegative matrix factorization with robust stochastic approximation. IEEE Trans Neural Netw Learn Syst 23(7):1087–1099
22. Shao W, He L, Lu CT, Wei X, Philip SY (2016) Online unsupervised multi-view features selection. In: Proceedings of the IEEE 16th international conference on data mining, pp 1203–1208
23. Bickel S, Scheffer T (2004) Multi-view clustering. In: Proceedings of international conference on data mining (ICDM), vol 4, pp 19–26
24. Sun S (2013) A survey of multi-view machine learning. Neural Comput Appl 23(7–8):2031–2038
25. Akata Z, Bauckhage C, Thurau C (2011) Non-negative matrix factorization in multimodality data for segmentation and label prediction. In: Proceedings of the 16th computer vision winter workshop, pp 1–8
26. Lee DD, Seung HS (1999) Learning the parts of objects by nonnegative matrix factorization. Nature 40(6755):788–791
27. Greene D, Cunningham P (2009) A matrix factorization approach for integrating multiple data views. In: Proceedings of joint European conference on machine learning and knowledge discovery in databases. Springer, Berlin, Heidelberg, pp 423–438
28. Zhang X, Zong L, Liu X, Yu H (2015) Constrained NMF-based multi-view clustering on unmapped data. In: Proceedings of the 29th AAAI conference on artificial intelligence, pp 3174–3180
29. Kalayeh MM, Idrees H, Shah M (2014) NMF-KNN: image annotation using weighted multi-view nonnegative matrix

factorization. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 184–191

30. Shao W, He L, Lu CT, Philip SY (2016) Online multiview clustering with incomplete views. In: Proceedings of the IEEE international conference on big data, pp 1012–1017
31. Madani O, Georg M, Ross DA (2012) On using nearly independent feature families for high precision and confidence. In: Proceedings of Asian conference on machine learning, pp 269–284
32. Cai D, He X, Han J (2005) Document clustering using locality preserving indexing. IEEE Trans Knowl Data Eng 17(12):1624–1637
33. Xu W, Liu X, Gong Y (2003) Document clustering based on non-negative matrix factorization. In: Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval, pp 267–273
34. Lovàsz L, Plummer M (1986) Matching theory. Mathematics studies, vol 121. North Holland, Amsterdam