Semantic Weighted Multi-view Clustering for Web Content

XIAOLONG GONG, LINPENG HUANG, TIANCHENG LUO, ZHIYI MA

1Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
2Advanced Institute of Information Technology, Peking University, Hangzhou 311215, China
3School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China

Corresponding author: Linpeng Huang (e-mail: lphuang@sjtu.edu.cn).

This work is supported by the National Natural Science Foundation of China (No. 61672046).

ABSTRACT Clustering is a long-standing important research problem. However, it remains challenging when handling large-scale web data from different types of information resources such as user profile, comments, user preferences and so on. All these aspects can be seen as different views and often admit the same underlying clustering of the data. In this paper, we present a novel Semantic Weighted Non-negative Matrix Factorization (SWNMF) multi-view clustering framework, which can provide an efficient weighted matrix factorization framework, dexterously manipulate multi-view web content, and easily explore the sparseness problem in semantic space of data. Specifically, each view of dataset forming a huge sparse matrix, which results in the non-robust characteristic during the matrix decomposition process, and further influences the accuracy of clustering results. To address above problem, we attempt to use some preference information (e.g. rating values) given by the users as latent semantic information to handle those features that are unobserved in each data point so as to resolve the sparseness problem in all views matrices. To combine multiple views in our large corpus, the overall objective of our proposed SWNMF is to minimize the loss function of weighted non-negative matrix factorization (NMF) under the $\ell_{2,1}$-norm and the co-regularized constraint under the $F$-norm. Extensive experiments on our large-scale multi-view web datasets demonstrate the competitive performance of our solution.

INDEX TERMS Multi-view clustering, Semantic information, Non-negative matrix factorization

I. INTRODUCTION

Clustering is the process of grouping similar objects together, which is one of the most fundamental tasks to explore the underlying structure of data [1]. The goal of clustering is to categorize data items with similar structures or patterns into the same group for reducing data complexity and facilitating interpretation. With the unprecedentedly explosive growth in the volume of web data, how to effectively cluster large-scale web data becomes an interesting but challenging problem. Moreover, multi-view data are very common in real-world applications due to the innate properties, and usually characterized by various heterogeneous sources of information [2]– [4]. Web pages, multimedia documents, and user profiles in many web communities like sport, music, movie and social network are examples of data that can be organized into multi-graphs or decomposed onto multi-view signals. However, existing single- and multi-view clustering methods fail to efficiently cluster those multi-view web data due to the sparsity challenge.

Different from the traditional data with a single view, multi-view datasets have multiple representations with heterogeneous views. For example, in web clustering, different types of textual information such as plot, reviews, related content, and QAs can be taken into consideration as they are different views of web items. In multi-view data, they commonly have the following properties: 1) Each single view of one item has its own feature sets. For instance, a web item (e.g. a movie in web movie community) has a large number of short comments that consists of abundant words as component feature in comment view. 2) One item under different views should share same consistency in semantic information. Just like a new story may be reported by different news resources, the underlying content should not be changed. 3) The sparseness problem exists in each view of one item. With rapid growth of user generated contents in web community, each available view generally has an intrinsic dimension that is significantly smaller than their ambient dimension. For instance, each item vector refers
to a high-dimensional latent feature space in view matrix, but many features are unobserved, i.e., feature \( w \) has not appeared in item \( i \) which means the relevant element value in view matrix equals to 0. Our intuition is that since observed features in an item are too few to tell us what the item is about, unobserved features can tell us what the item is not about. We assume that the semantic space of both observed and unobserved features make up the complete semantics profile of the item. We demonstrate the sparseness rates of view matrices in relevant large datasets in Fig. 1.

According to the above properties, multi-view clustering by exploiting heterogeneous features of data has attracted considerable attention [2], [3], [5]. Compared to single-view clustering, multi-view clustering normally can explore the intrinsic structural information hidden in the multi-view data and handle large numbers of unlabeled data. Conventional methods either directly concatenate these features as a long representation or treat each view independently for multi-view clustering. However, neither of the strategies is physically meaningful since they fail to solve the sparseness problem due to the high dimensionality. For example, when it comes to textual web content, we can get different types of information for a web item, including long reviews, news reports and topic discussions, each of which is taken as a distinct view of the web item. As items themselves yield intrinsic features, how to integrate the two extrinsic data sources derived from textual content is an important consideration. Recently, many multi-view clustering methods have been proposed [4], [6]–[11]. However, such methods suffer from two challenges: 1) They do not provide a preferable solution to solve the consistency on the intrinsic similarity of each item pair. For instance, if item \( i \) is similar to item \( j \), then the similarity of \( i \) and \( j \) should stay consistent after mapped to a new vector space in each single view. 2) For some specific views of data, they may have high dimensionality which leads to high sparsity, indicating existence of an amount of unobserved features in latent semantic representation, i.e., item \( i \) has no feature \( w \) means the value of \( i \)-th row and \( w \)-th column in item-feature co-occurrence matrix equals to 0. Previous works don’t make full use of the manual information from multiple views, like semantic feedback (e.g. user rates, number of likes) from users (as shown in Fig. 2).

We assume that the semantic spaces of both the observed and unobserved features make up the complete semantics profile of an item. To make full use of these multiple views information to boost clustering accuracy, those observed and unobserved features can be analyzed with a simplified model by means of semantic feedback from users, which can help us know the feature significance in semantic space.

In this paper, we introduce a novel framework for large-scale multi-view clustering, dubbed Semantic Weighted Non-negative Matrix Factorization (SWNMF), which can simultaneously handle the sparseness of the view matrices by integrating the joint semantic weighted non-negative matrix factorization and perform multi-view data clustering task. An objective function is proposed, which takes into consideration both of the similar pair-wise co-regularization constraint and semantic weighted scheme in the process of data clustering. Although, most recently proposed multi-view clustering methods [12], [13] presented algorithm to deal with the sparseness using the Frobenius norm based objectives, in our work, \( l_{2,1} \)-norm minimization constraint is added into the objective function to reduce the noisy features, which guarantees data matrix sparse in rows and obtains a robust cluster structure [4], [14], [15]. The contributions of this work mainly include:

1) The proposed SWNMF simultaneously learn the latent clustering label matrix for data from multiple views and capture similar pair-wise structural information among different views in a joint framework. To the best of our knowledge, SWNMF is the first effort of constructing semantic weighted matrix to address the sparseness problem on many web datasets. We attempt to integrate user subjective preference (rating value) with associated view matrix, which we expect to enhance latent semantic information and weaken the sparseness of original view matrix. Meanwhile, we propose an adaptive semantic weighted strategy, which can directly handle those view content with or without user preference information.

2) An alternating optimization algorithm with rigorous convergence proof is developed to solve the proposed non-smooth and non-convex objective function. By virtue of the proposed effective optimization algorithm, SWNMF on both our constructed three large web data sets and three benchmark data sets show clearly superior multi-view clustering performance in comparison with several state-of-the-art methods.
The rest of this paper is organized as follows. Section II briefly overviews previous works on multi-view data clustering. Problem formulation and the details of the proposed SWNMF framework are given in Section III. Our theory analysis and optimization framework are presented in Section IV. To demonstrate the performance of our algorithm, we have conducted extensive experiments, the experimental results of which are reported in Section V, and followed by the conclusion in Section VI.

II. RELATED WORK

With advances in information acquisition technologies, multi-view data become ubiquitous. Due to the widespread use of multi-view datasets in practice, many realistic applications are accomplished by multi-view learning methods, such as community detection in social networks, image annotation in computer vision, and cross-domain user modeling in recommendation systems [16]. Multi-view clustering is a machine learning paradigm to classify similar subjects into the same group and dissimilar subjects into different groups by combining the available multi-view feature information, and to search for consistent clustering across different views. Meanwhile, based on the seminal work of Bickel and Schaffer [17], plenty of multi-view clustering methods have been proposed and applied to many scientific domains such as computer vision, natural language processing, social multimedia and health informatics. A multi-view clustering strategy via canonical correlation analysis (CCA) is presented in [1]. This method assumes that the views are uncorrelated given the cluster label, and based on the assumption that multiple views are independent of each other. In fact, the real multi-view data is not entirely uncorrelated and they usually share certain inner relationship instead. Another strategy named multiple view semi-supervised dimensionality reduction is devised and employed to multi-view clustering [18]. A consensus pattern is learned from multiple embeddings of multi-view data. However both this strategy and CCA-based multi-view clustering need part of labeled data which is usually unavailable in real application. Several previous multi-view clustering algorithms have been proposed [19]–[21]. However, these techniques assume that the dimensions of the features in multiple views are the same, limiting their applicability to the homogeneous settings. Some methods concentrate on the clustering of only two-views situation so that it might be hard to extend [17]. In natural language processing scenarios, text documents can be obtained in multiple languages. It is natural to use multi-view clustering to conduct document categorization and some works are based on a commonly used spectral clustering technique [7], [22], [23]. Also multi-view clustering plays an important role in helping automatically organizing web resources for content providers and in diversifying search results in web document ranking [24]. Improved multi-view clustering of web community resources also helps to automatically generate more meaningful tags [4]. For clustering multi-view or multi-source datasets, some algorithms have been proposed recently which take different factors into consideration, e.g. the differences and relationships between data from various views. They make use of the nonnegative matrix factorization with different regularization constraints [25]. [9] aims...
to find a unified low-dimensional space to fuse the multi-view representations in order to well explore the common latent structure shared by multi-views. A few clustering algorithms have been proposed to apply NMF on multi-view data before work [26] showed the connection between NMF and clustering methods. [7], [27] are among the first algorithms that seek groupings that are consistent across different views which gives some pair-wise co-regularization constraints on the jointly factorizing matrices. A weighted extension of multi-view NMF is presented in [29] to address the image annotation problem. There are two weight matrices are introduced in this method. One is toward improved reconstruction for rare labels. The other one gives more weight to images containing rare labels. Another weighted extension has also been developed in [30] to handle those elements that are unobserved in each data instance so as to resolve the sparseness problem. Furthermore, there are many views suffer from missing of some data samples resulting in many partial examples, Shao et al. [31] proposed a multiple incomplete views clustering method and [32] also designed to deal with the incompleteness of the views. In this paper, we propose a multi-view clustering framework based on semantic weighted non-negative matrix factorization methods. Our formulations model the multi-view web data and seek a joint prior user preference information by weighted matrix analysis.

III. SEMANTIC WEIGHTED NON-NEGATIVE MATRIX FACTORIZATION FRAMEWORK

To make this paper clear, we summarize some notations used in this paper in Table I. We firstly discuss some necessary preliminaries, and then introduce the background knowledge on multi-view nonnegative matrix factorization. We further describe our SWNMF algorithm and present semantic weighted strategy to model our sparse views of data.

TABLE 1. Summary of the Notations

<table>
<thead>
<tr>
<th>Notations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m )</td>
<td>Total number of items</td>
</tr>
<tr>
<td>( n_v )</td>
<td>Total number of views</td>
</tr>
<tr>
<td>( X )</td>
<td>The whole dataset</td>
</tr>
<tr>
<td>( X^{(l)} )</td>
<td>Data matrix for the ( l )-th view</td>
</tr>
<tr>
<td>( x^{(l)}_i )</td>
<td>The ( i )-th item of the ( l )-th view dataset</td>
</tr>
<tr>
<td>( d^{(l)} )</td>
<td>Dimension of features in the ( l )-th view</td>
</tr>
<tr>
<td>( W^{(l)} )</td>
<td>Semantic weighted matrix for the ( l )-th view</td>
</tr>
<tr>
<td>( U^{(l)} )</td>
<td>Class indicator matrix for the ( l )-th view</td>
</tr>
<tr>
<td>( V^{(l)} )</td>
<td>The basis matrix for the ( l )-th view</td>
</tr>
<tr>
<td>( \lambda_{is} )</td>
<td>Weight parameter for constraint</td>
</tr>
</tbody>
</table>

A. PROBLEM STATEMENT

Consider a dataset consisting of \( m \) items represented by \( n_v \) views, let \( X = [x_1, x_2, \ldots, x_m]^T \in \mathbb{R}_{+}^{m \times d} \) be the original item matrix of non-negative elements, where each row vector \( x^T_i \) \((1 \leq i \leq m)\) denotes an item and each column represents one feature. Nonnegative matrix factorization aims to find two factors with nonnegative elements, the factorization is formulated as \( X \approx U V^T \), where \( U \in \mathbb{R}_{+}^{m \times K} \) represents the class indicators, indicating the final clustering result. \( V \in \mathbb{R}_{+}^{d \times K} \) is termed the basis matrix. \( K \) denotes the desired reduced dimension and represents a particular cluster number in multi-view clustering task. Further, let \( \{X^{(1)}, X^{(2)}, \ldots, X^{(n_v)}\} \) denote the data of \( n_v \) views, and each view \( X^{(l)} \) is factorized as \( X^{(l)} \approx U^{(l)}(V^{(l)})^T \). Here for different views, they have the same number of items but allow for different number of features, which means \( U^{(l)} \) are with the same dimension m-by-K for all views, while \( V^{(l)} \) are of dimension K-by-d \((l)\) for per view. The fundamental multi-view based on NMF function tries to minimize the joint problem over \( U^{(l)}, V^{(l)} \):

\[
\sum_{l=1}^{n_v} \|X^{(l)} - U^{(l)}(V^{(l)})^T\|_F^2, \quad s.t. U^{(l)}, V^{(l)} \geq 0 \quad (1)
\]

In general, the objective function above is not convex with \( U^{(l)} \) and \( V^{(l)} \) jointly, so it is infeasible to find the global minima. A common approach is to use an alternating way to update \( U^{(l)} \) and \( V^{(l)} \) \( [33] \), which iteratively updates \( U^{(l)} \) and \( V^{(l)} \) by

\[
U^{(l)}_{i,k} \leftarrow U^{(l)}_{i,k} \left( \frac{X^{(l)} V^{(l)T}_{i,k} (U^{(l)} V^{(l)T})_{i,k}}{(U^{(l)} V^{(l)T})_{i,k}} \right), \quad V^{(l)}_{j,k} \leftarrow V^{(l)}_{j,k} \left( \frac{X^{(l)T} U^{(l)}_{i,k}}{(V^{(l)} U^{(l)T})_{j,k}} \right) \quad (2)
\]

Unlike other matrix factorization algorithms such as PCA and SVD, the nonnegative property of NMF makes the reduced space easy to interpret. In clustering problems, the latent feature matrix \( U^{(l)} \) is used to extract the class indicators (clustering results). Specifically, each element \( U_{i,k}^{(l)} \) of matrix \( U^{(l)} \) indicates the degree of association of item \( i \) with cluster \( k \). For example, we can constrain \( \sum_j U_{i,j}^{(l)} = 1 \) for every row vector \( u^{(l)}_i \in \mathbb{R}_{+}^{1 \times K} \), and then we just need to take the largest value of row vector \( u^{(l)}_i \) as the cluster assignment of item \( i \).

B. MULTI-VIEW NMF MODEL

Multi-view NMF [9] aims to search for a factorization that gives compatible clustering solutions across multiple views. The key idea is to formulate joint matrix factorization process and construct a softly regularized constraint between coefficient matrices of different views and common consensus matrix. To incorporate the regularized constraint with consensus matrix \( U^* \) for individual views, the final form of Multi-view NMF algorithm can be formulated as follows:

\[
\sum_{l=1}^{n_v} \|X^{(l)} - U^{(l)}(V^{(l)})^T\|_F^2 + \sum_{l=1}^{n_v} \lambda_l \|U^{(l)} - U^*\|_F^2 \quad (3)
\]

\[
s.t. \forall 1 \leq k \leq K, \|V^{(l)}_{j,k}\| = 1 \quad \text{and} \quad U^{(l)}, V^{(l)}, U^* \geq 0
\]

where \( \lambda_l \) is the only parameter tuning the relative weight among different views. The goal of multi-view clustering
is to cluster the $m$ items into $K$ clusters according to their intrinsic properties.

**C. GLOBAL OBJECTIVE FUNCTION**

In what follows, we will describe in detail the proposed global objective function, which takes into consideration both the semantic weighted strategy and the consistent similarity constraint in the process of multi-view data clustering. Different semantic weighted strategies are designed, which respectively weight the observed features and unobserved features in each view. Our solution in finding a principled method to combine views adopts the semantic weighted NMF technique.

1) **SWNMF algorithm**

Throughout this paper, matrices are written as capital letters and vectors are denoted as boldface lowercase letters. For an arbitrary matrix $M \in \mathbb{R}^{m \times d}$, $M_{i,j}$ indicates the $j$-th element of vector $m_i$. $\|M\|_F$ is the Frobenius norm of $M$ and $Tr(M)$ is the trace of square matrix $M$, $Tr(A^T B)$ denotes the inner product between two matrices. The $l_{2,1}$-norm is defined as $\|M\|_{2,1} = \sum_{i=1}^{m} \|m_i\| = \sum_{i=1}^{m} \sqrt{\sum_{j=1}^{d} M_{i,j}^2}$. The hypothesis behind multi-view clustering is that different views should admit the same underlying clustering of the data. Formally, given $n_v$ views denoting as $\{X^{(1)}, \ldots, X^{(n_v)}\}$, where $X^{(l)} \approx U^{(l)}(V^{(l)})^T$ and each $U^{(l)}$ is with same dimension $m \times K$. Fig. 3 shows that our framework of the proposed SWNMF.

In our SWNMF approach, let $X^{(l)} \in \mathbb{R}^{m \times d^{(l)}}$ be the view $l$ data matrix with each row $x_i \in \mathbb{R}^{1 \times d^{(l)}}$ being a data item. In order to address the sparseness issue for our clustering task, we introduce a novel weighting strategy for the weight matrix and further present a completely NMF formulation with manifold regularization. For smooth combine the semantic weight with each item vector, an intuitive idea is that use the $l_{2,1}$-norm in NMF objective function, which makes the model robust to these data items. Suppose that we want to cluster $X$ into $K$ clusters under the nonnegative matrix factorization framework and solve:

$$
\min_{U,V} \sum_{l=1}^{n_v} \|X^{(l)} - U^{(l)}(V^{(l)})^T\|_2,1, \quad s.t. \quad U^{(l)}, V^{(l)} \geq 0
$$

(4)

and Eq 4 can be reformulated as following:

$$
\min_{U,V} \sum_{l=1}^{n_v} \sum_{i=1}^{m} \|x^{(l)}_i - u^{(l)}_i(V^{(l)})^T\|_2, \quad s.t. \quad U^{(l)}, V^{(l)} \geq 0
$$

(5)

Meanwhile, the orthogonal constraint on $U^{(l)}$ is necessary in our framework, which often has been used to guarantee the uniqueness of the solution. And we let $\hat{w}^{(l)}_i$ be a semantic weighted value for item $i$ in view $l$. Generally speaking, the objective function of SWNMF is formulated as:

$$
J = \min_{U,V} \sum_{l=1}^{n_v} \sum_{i=1}^{m} \hat{w}^{(l)}_i \|x^{(l)}_i - u^{(l)}_i(V^{(l)})^T\|_2 + R_{IS}
$$

s.t. $U^{(l)}, V^{(l)} \geq 0, \quad U^{(l)}(V^{(l)})^T = I$

(6)

where $R_{IS}$ is our similar pair-wise co-regularization function that enforces similarity constraints on multiple views. SWNMF is a general framework as different regularization schemes and similarity measures can be used to implement the co-regularization function $R_{IS}$. By encouraging heterogeneous content of views, the algorithm finally learns different cluster class indicator matrix $U^{(l)}$ for each view $l$. The final aggregated representation $U$ can be obtained by combining all $U^{(l)}$: $U = [U^{(1)}, U^{(2)}, \ldots, U^{(n_v)}] \in \mathbb{R}^{m \times (n_v \times K)}$.

2) **Inter-view pair-wise constraint on similarity**

we will first present simple pair-wise constraint, where we expect that the class indicator matrices learned from different views indicate the same class label for one item. Let $X^{(l)} = \{x^{(l)}_1, x^{(l)}_2, \ldots, x^{(l)}_m\}$ denotes the set of $m$ items in view $l$. We should note that $x^{(s)}_i$ and $x^{(s)}_i$ (1 $\leq i \leq m$) represent the same item, which means the true class labels for $x^{(l)}_i$ and $x^{(s)}_i$ ($l \neq s$) should be the same. To implement the hypothesis of multi-view clustering, an intuitive method is to regularize the coefficient matrices of the different views towards a common consensus matrix (see Eq 3). And this consensus matrix is used for clustering results. This approach performs well when views are largely homogeneous and of roughly the same quality. However, different views may be generated heterogeneously and may wide discrepancies in quality. Therefore, the simple pair-wise co-regularization relaxes above constraints, and captures the difference between two indicator matrices of two views and force the representations from different views to be similar, which can be formulated as:

$$
R_1 = \sum_{l=1}^{n_v} \sum_{s=1}^{n_v} \lambda_{ls}\|U^{(l)} - U^{(s)}\|^2_F = \sum_{l,s} \lambda_{ls}\|U^{(l)} - U^{(s)}\|^2_F
$$

(7)

where $\lambda_{ls}$ is the regularization parameter controlling the importance of constraint among different indicator class matrices. But this constraint ignores the similarity of each data point in intra-view. Our intuition is intra structure among different views should be kept consistency, which means the similarity information between any two items can complement with each other during the factorization process. It should yield a better latent space and be more effective for clustering results. So we propose item similarity pair-wise co-regularize constraint for further refinement, which been defined as follows,

$$
R_{IS} = \sum_{l=1}^{n_v} \sum_{s=1}^{n_v} \lambda_{ls}\|U^{(l)}(U^{(l)})^T - U^{(s)}(U^{(s)})^T\|^2_F
$$

= \sum_{l,s} \lambda_{ls}\|P^{(l)} - P^{(s)}\|^2_F
$$

(8)

1In this paper, we use ‘item’ and ‘instance’ exchangeable.
where $\mathbf{P}^{(l)} = \mathbf{U}^{(l)T} \mathbf{U}^{(l)}$ denotes a similarity matrix between each two items in view $l$. Meanwhile, we should note that the column vector of the coefficient matrix $\mathbf{U}^{(l)}$ represents a cluster, and each term $(\mathbf{U}^{(l)})^T \mathbf{U}^{(l)}$ demonstrates that the cosine similarity between two clusters. We let $\mathbf{C}^{(l)} = (\mathbf{U}^{(l)})^T \mathbf{U}^{(l)}$ be the pair-wise cluster similarity matrix. Thus, our natural definition for our cluster similarity pair-wise co-regularize constraint as follows,

$$R_{CS} = \sum_{l=1}^{n_v} \sum_{s=1}^{d(l)} \lambda_s \| (\mathbf{U}^{(l)})^T \mathbf{U}^{(l)} - (\mathbf{U}^{(s)})^T \mathbf{U}^{(s)} \|_F^2$$

$$= \sum_{l,s} \lambda_s \| C^{(l)} - C^{(s)} \|_F^2. \quad (9)$$

3) Semantic weighted scheme

Now we give the detailed definition of $w^{(l)}_{i,j}$. For many multi-view datasets, parts of view types in web content data are associated with user semantic information. But for some views content they do not refer to semantic feedback information (e.g. commenting users, recommendation list). Most importantly, semantic information only has been presented after user generate content on most of web communities and has different appearance forms in different textual views. For example, there is an overall rating score for the view of plot summary and there are numbers of thumbs up for the view of short comments, long reviews, reports, etc. In the meantime, user view or other preference views may not have relevant semantic information, thus we need give out two different semantic weighted strategies for those different types of views respectively. We should demonstrate that $\mathbf{W}^{(l)} = [w^{(l)}_1, \cdots, w^{(l)}_{d(l)}]^T \in \mathbb{R}^{m \times d(l)}$ is our semantic weighted matrix for view $l$ and $\mathbf{W}^{(l)}$ defines different weights for each cell in $\mathbf{X}^{(l)}$, where $w_{i,j}^{(l)} = [\mathbf{W}_i^{(l)}]_{j} \in \mathbb{R}^{d(l) \times 1}$ is the vector of data instance $i$. If each view content in dataset has the corresponding rating feature, we use symbol $\mathbf{W}^{(l)}_{i,j}$ to represent relevant semantic weighted of each cell in view $l$ matrix. We use $SI$ to represent “semantic information”. Therefore, our weight scheme for the view of textual content type is defined as

$$W_{i,j}^{(l)} = \begin{cases} \frac{1}{\epsilon} & \text{if } X_{i,j}^{(l)} \neq 0 , l \text{ with } SI \\ \frac{1}{\epsilon} & \text{if } X_{i,j}^{(l)} = 0 \end{cases} \quad (10)$$

and our relevant weight scheme for the view of user type can be defined as

$$W_{i,j}^{(s)} = \begin{cases} 1 & \text{if } X_{i,j}^{(s)} \neq 0 , s \text{ without } SI \\ \frac{1}{\epsilon} & \text{if } X_{i,j}^{(s)} = 0 \end{cases} \quad (11)$$

where $\epsilon$ is a very small value for those unobserved features in view matrices. The intuition of a small $\epsilon$ when $X_{i,j}^{(l)} = 0$ is to diminish the sparse influence where the $j$-th feature is unobserved in the item $i$. $N_{l}(i)$ indicates the number of text content(i.e. summary, short comments, long reviews) in view $l$, which refers to one item. In other words, there are many short comments and long reviews from a wide array of users for one item, and all these happen in many real web communities. In fact, some features may present in some text content, which may achieve a good score or many thumbs up.

Therefore, for item $i$ of view $l$, we use $f_{l,p}$ to represent rating score or total counts of thumbs up from text content $p$ where the $j$-th feature involves in.

According to above description, we have different feature numbers in different text content view matrices. We believe a same feature may has varied significance among those
different views. Thus, our definition of semantic weighted value for item $i$ in view $l$ that can be formulated as

$$\hat{w}_{i,l} = \sum_j (X_{i,j} - W_{i,l})$$

(12)

Our strategy can be applied to different types of web resources in many web applications, particularly those comment-based corpus.

IV. OPTIMIZATION ALGORITHM

In this section, we first indicate that our initialization and normalization strategy during each iteration process, then we show how to update the two variables and give a specific algorithm flow.

A. INITIALIZATION

According to previous matrix factorization research, the proper initialization plays an important role in the performance of matrix factorization and indicates that matrix factorization algorithms are sensitive to the initialization (e.g., NMF algorithms [34]).

We attempt to use k-means clustering algorithms in our initialization process, which are reasonable and efficient. Our purpose is to yield two factor matrices: class indicators for each item ($U$) and basis matrix ($V$). Running k-means yields two outputs: the cluster assignment of each instance and the centroid of each cluster. Thus, we propose to use these outputs to initialize our two factor matrices. We first combine our textual content view matrices in relevant datasets, then we normalize each view matrix and concatenate all view matrices in the same level before we use k-means method. Specifically, we let $x_{i,l}$ be an item vector in view $l$ matrix, and our combined vector is $x_i = \sqrt{\frac{1}{n_v} (x_{i,1}^{(1)}, \ldots, x_{i,1}^{(n_v)}, \ldots, x_{i,n_v}^{(n_v)})}$. Using k-means method to initialize $U$ and $V$ place them in the same space initially, which is more meaningful than random initialization.

B. NORMALIZATION

Inserting a normalization step is important during our iteration process, the solution is guaranteed to minimize the objective function with local minima, but they do not lead to meaningful results based on real data. There are two possible reasons: (1) the class indicator matrix $U^{(l)}$ might not be comparable at the same scale; (2) there are too much noise information in our data matrix, and a good solution would be affected by the initial results. We can solve both problems by normalizing the $U^{(l)}$ matrix to make them comparable to each other. We should note that each column of $U^{(l)}$ represents a cluster indicator, whose elements give the strength of association of the items to this cluster. Therefore, normalizing the column vectors of $U^{(l)}$ makes the clustering assignments of different views comparable. Formally, let $H^{(l)}$ be the diagonal matrix with nonzero elements equal to the vector based $L_2$ norm, which means $H^{(l)}_{i,j} = \sqrt{\sum_i U^{(l)}_{i,j}^2}$. Then our normalization strategy works as follows:

$$U^{(l)} \leftarrow U^{(l)} H^{(l)-1}, V^{(l)} \leftarrow V^{(l)} H^{(l)}$$

(13)

Note that this normalization does not change the value of Eq. 6 and our co-regularization term becomes meaningful because they are comparable.

C. ALGORITHM AND OPTIMIZATION

The objective function in Eq. 6 is separately convex w.r.t each of $U^{(l)}$ and $V^{(l)}$. Therefore, we propose an alternating optimization algorithm that guarantees each subproblem converges to the local minima under non-negative condition. The optimization works as follows: (1) fix the value of $U^{(l)}$ while minimizing $J$ over $V^{(l)}$; then (2) fix the value of $V^{(l)}$ while minimizing $J$ over $U^{(l)}$. We execute these two steps until convergence, or until a set number of iterations is exceeded. Our SWNMF is summarized in Algorithm 1. We implement similarity constraint by coupling the factorization of the views through similar pairwise co-regularization. Therefore, the final objective function Eq. 6 is equivalent to

Algorithm 1: SWNMF algorithm

Input: Multi-view datasets

$$\{X^{(1)}, X^{(2)}, \ldots, X^{(n_v)}\};$$ Weighting matrices

$$\{W^{(1)}, W^{(2)}, \ldots, W^{(n_v)}\};$$ Number of clusters $K$; Parameters $\lambda_{ls}$;

Output: Class indicator matrices

$$\{U^{(1)}, U^{(2)}, \ldots, U^{(n_v)}\};$$ Basis matrices

$$\{V^{(1)}, V^{(2)}, \ldots, V^{(n_v)}\};$$

1. Initialize $U^{(l)}$ and $V^{(l)}$ using the traditional k-means;

2. Normalize each view matrix $X^{(l)}$ such that $\|X^{(l)}\| = 1$

3. Compute $W^{(l)}$ by Eq. 10 and Eq. 11;

4. repeat

5. for $l$ from 1 to $n_v$ do

6. Fixing $U^{(l)}$, update $V^{(l)}$ by Eq. 18

7. Normalize $U^{(l)}$ and $V^{(l)}$ as in Eq. 13

8. Fixing $V^{(l)}$, update $U^{(l)}$ by Eq. 38

9. end

10. return $U^{(l)}$ and $V^{(l)}$

11. until Eq. 14 is converged;

the following:

$$J = \min_{U,V} \sum_{l=1}^{n_v} \sum_{i=1}^{n} \hat{w}_{i,l} \| x_i - u_i (V^{(l)})^T \|^2_2$$

$$+ \sum_{l=1}^{n_v} \sum_{s=1}^{n} \lambda_{ls} \| P^{(l)} - P^{(s)} \|^2_F$$

(14)

s.t. $U^{(l)}, V^{(l)} \succeq 0$, $U^{(l)^T} U^{(l)} = I$
and this equation can be rewritten as:
\[
J = \min_{U,V} \sum_{l=1}^{n_v} Tr((X(l)^T D(l) X(l) - 2(U(l))^T D(l) X(l) V(l) + V(l) (U(l)^T D(l) U(l)(V(l)^T) + \sum_{l,s} \lambda_{ls} Tr(U(l)(U(l)^T D(l) U(l)(V(l)^T) + U(s)(U(s)^T D(s) U(s))^T + 2U(l)(U(l)^T D(s) U(s))^T + U(s)(U(s)^T D(s) U(s))^T )
\]
where \(D(l)\) is a diagonal matrix with the \(i\)-th diagonal element \(D_{i,i} = w_i\). Here, \(|\|A\|_F^2 = Tr(A^T A)\) and \(Tr(AB) = Tr(BA)\) are used in the derivation. To enforce the orthogonal and non-negative constraints, we need three Lagrange multipliers for three constraints respectively. Thus, our Lagrange \(J(U,V)\) is
\[
J(U,V) = J + \sum_{l=1}^{n_v} Tr(\gamma(l) ((U(l)^T U(l) - I))
+ \sum_{l=1}^{n_v} Tr(\alpha(l)(U(l)^T U(l)) + \sum_{l=1}^{n_v} Tr(\beta(l)(V(l)^T)
\]
where \(\alpha(l), \beta(l), \gamma(l)\) are the Lagrange matrices.

1) Fixing \(U(l)\), then minimize \(V(l)\)

Similar to the known solution for NMF, now we can adopt alternation optimization to minimize the objective function when \(D(l)\) is fixed. We should know that the second part in \(J\), the term \(\sum_{l=1}^{n_v} Tr(\gamma(l)((U(l)^T U(l) - I))\) and the term \(\sum_{l=1}^{n_v} Tr(\alpha(l)(U(l)^T U(l))\) are all constants, so the derivative of \(J(U,V)\) with respect to \(V(l)\) is
\[
\frac{\partial J(U,V)}{\partial V(l)} = -2(U(l)^T D(l) U(l) + 2V(l)(U(l)^T D(l) U(l) + \beta(l)
\]
Using the Karush-Kuhn-Tucker (KKT) condition that \(\beta(l) V(l) = 0\), and we have \(\frac{\partial J(U,V)}{\partial V(l)} \odot V(l) = 0\). So the update solution of \(V(l)\) is
\[
V(l) \leftarrow V(l) \odot \frac{(X(l)^T D(l) U(l))}{V(l)(U(l)^T D(l) U(l))}
\]

2) Fixing \(V(l)\), then minimize \(U(l)\)

Now, we analyze the stationary point \(U(l)\) in the second subproblem, which can be written as:
\[
J_1(U,V) = \min_{U,V} \sum_{l=1}^{n_v} Tr((X(l)^T D(l) X(l) - 2(U(l))^T D(l) X(l) V(l) + V(l) (U(l)^T D(l) U(l)(V(l)^T) + \sum_{l,s} \lambda_{ls} Tr(U(l)(U(l)^T D(l) U(l)(V(l)^T) + U(s)(U(s)^T D(s) U(s))^T + 2U(l)(U(l)^T D(s) U(s))^T + U(s)(U(s)^T D(s) U(s))^T )
\]
and we ignore two irrelevant terms here. Then the derivative of \(J_1(U,V)\) with respect to \(U(l)\) is
\[
\frac{\partial J_1(U,V)}{\partial U(l)} = -2D(l) X(l)V(l) + 2D(l) U(l)(V(l)^T U(l) + \sum_{s} \lambda_{ls} (4U(l)(U(l)^T U(l) - 4U(s)(U(s)^T U(l)) + 2U(l)(U(l)^T U(l) - 2U(l)\gamma(l)
\]
Setting the derivative of \(J_1(U,V)\) to be 0 with respect to \(U(l)\), we have
\[
\alpha(l) = 2D(l) X(l)V(l) + 4 \sum_{s} \lambda_{ls} (U(s)^T U(l)) + 2D(l) U(l)(V(l)^T U(l) - 2U(l)\gamma(l)
\]
Following the KKT condition for the nonnegative of \(U(l)\), we have the following equation:
\[
(2D(l) X(l)V(l) + 4 \sum_{s} \lambda_{ls} U(s)(U(s)^T U(l)) + 2D(l) U(l)(V(l)^T U(l) - 2U(l)\gamma(l))_{i,j}
\]
According to the theorem from [35], we get
\[
(D(l) X(l)V(l) + 2 \sum_{s} \lambda_{ls} U(s)(U(s)^T U(l)) u_{i,j} =
(D(l) U(l)(V(l)^T U(l) + 2 \sum_{s} \lambda_{ls} U(l)(U(l)^T U(l) + U(l)\gamma(l))_{i,j}
\]
It makes sure \(D(l)^{T} V(l)^{T} U(l)^{T} + 2 \sum_{s} \lambda_{ls} U(l)(U(s)^T U(l)) + \gamma(l)_{i,j} \geq 0\), summing over index \(i\), we have
\[
(-U(l)^{T} (D(l) X(l)V(l) + 2 \sum_{s} \lambda_{ls} U(s)(U(s)^T U(l))) + D(l) (V(l)^T U(l) + 2 \sum_{s} \lambda_{ls} U(l)(U(l)^T U(l) + \gamma(l))_{i,i} = 0
\]
Therefore we obtain the diagonal elements of the Lagrangian multipliers
\[
\gamma(l)_{i,i} = (U(l)^{T} (D(l) X(l)V(l) + 2 \sum_{s} \lambda_{ls} U(s)(U(s)^T U(l)) - D(l) (V(l)^T U(l) - 2 \sum_{s} \lambda_{ls} U(l)(U(l)^T U(l)))_{i,i}
\]
The off-diagonal elements of the Lagrangian multipliers are approximately obtained by \(\frac{\partial J_1(U,V)}{\partial \gamma(l)_{i,j}} = 0\), so our final solution for \(\gamma(l)_{i,j}\) equals Eq. [25]. We firstly give a successive update rule of \(U(l)\) is:
\[
U(l) \leftarrow \frac{U(l) \sqrt{D(l)^{T} X(l)V(l) + 2 \sum_{s} \lambda_{ls} U(s)(U(s)^T U(l))}}{D(l)^{T} U(l)(V(l)^T U(l) - 2 \sum_{s} \lambda_{ls} U(l)(U(l)^T U(l))}}
\]
The correctness of the updating rule Eq. [26] can be guaranteed by the following theorem.

\[\hat{\lambda} \in matrix denote element-wise multiplication.\]
Theorem 1. If the updating rule of $U^{(1)}$ converges, then the final solution satisfies the KKT optimality condition.

Proof. For each $U_{ij}^{(1)}$, we have the following equation:
\[
(2D^{(1)}X^{(1)}V^{(1)})' + 4 \sum_{a} \lambda_{ia}(U^{(a)}(U^{(a)})')^{T}U^{(1)} - 2D^{(1)}U^{(1)}(V^{(1)})'V^{(1)}
\]
\[- 4 \sum_{a} \lambda_{ia}(U^{(1)}(U^{(1)})')^{T}U^{(1)} - 2U^{(1)}(V^{(1)})'v_{ij}^{(1)}y_{ij}^{(1)} = 0
\]
which is equivalent to Eq. (22). Therefore, $U^{(1)}$ can get convergence when per iteration uses the updating rule
\[
U^{(l+1)} \leftarrow U^{(l)} - 4 \sum_{a} \lambda_{ia}(U^{(a)}(U^{(a)})')^{T}U^{(1)} - 2D^{(1)}U^{(1)}(V^{(1)})'V^{(1)}
\]
\[- 4 \sum_{a} \lambda_{ia}(U^{(1)}(U^{(1)})')^{T}U^{(1)} - 2U^{(1)}(V^{(1)})'v_{ij}^{(1)}y_{ij}^{(1)}
\]

Our purpose is that we need to achieve an exact analytical local-minimum solution for the minimization of $J_{1}(U,V)$ when $V^{(1)}$ is fixed. To this end, we must investigate the convergence of $J_{1}(U,V)$ for the above stationary point $U^{(1)}$. In other words, we just need to prove that $J_{1}(U,V)$ keeps non-increasing when updating the above stationary point $U^{(1)}$. We present the complete proof of the non-increasing property of stationary point $U^{(1)}$ using the auxiliary function method [33]. Here we first introduce the definition of auxiliary function and three relevant lemmas.

Definition 1. [33] $F(U, U')$ is an auxiliary function of $J_{1}(U,V)$ if the conditions
\[
F(U, U') \geq J_{1}(U,V), \quad F(U, U) = J_{1}(U,V)
\]
are satisfied.

Lemma 1. [33] If $F$ is an auxiliary function for $J$ then $J$ is non-increasing under the update
\[
U^{(l+1)} = \arg \min_{U} F(U, U^{(l)})
\]
Proof. $J_{1}(U^{(l+1)}, V) \leq F(U^{(l+1)}, U^{(l)}) \leq F(U^{(l)}, U^{(l)}) = F(U^{(l)})$

Lemma 2. [33] For any matrices $A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{r \times r}, Q \in \mathbb{R}^{n \times q}, Q' \in \mathbb{R}^{q \times r}$, with $A$ and $B$ symmetric, the following inequality holds:
\[
\text{Tr}(Q^{T}AQB) \leq \sum_{i=1}^{n} \sum_{p=1}^{r} (A_{qp}B_{ip})Q_{ip}^{2}
\]

Lemma 3. [36] For any matrices $A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{r \times r}, Q \in \mathbb{R}^{n \times q}, Q' \in \mathbb{R}^{q \times r}$, with $A$ and $B$ symmetric, the following inequality holds:
\[
\text{Tr}(QAQ'QBQ'T) \leq \sum_{i=1}^{n} \sum_{p=1}^{r} (Q'A_{qp}Q'B_{ip}Q'TA_{ip})Q_{ip}^{2}
\]

Now we show that the updating rules for stationary point $U^{(1)}$ with a proper auxiliary function. For brevity, we let $X, U, V, D$ represent $X^{(1)}, U^{(1)}, V^{(1)}, D^{(1)}$ and an appropriate auxiliary $F(U, U')$ of $J_{1}(U,V)$ is defined as
\[
F(U, U') = -2 \sum_{ip} (DXV)_{ip}U_{ip}'(1 + \log \frac{U_{ip}}{U_{ip}'})
\]
\[+ \sum_{ip} (DU'V^{T}V + U'\gamma)_{ip}U_{ip}^{2}
\]
\[+ \sum_{ls} \lambda_{ls}(U'L_{ip}U_{ip}' + U'L_{ip}U_{ip}')
\]
\[- 2 \sum_{ip}(U^{(a)}(U^{(a)})')^{T}pqU_{ip}'U_{ip}'(1 + \log \frac{U_{ip}U_{ip}'}{U_{ip}'U_{ip}'})
\]

We find upper bounds for each of the three positive terms by the above lemmas. And then we obtain lower bounds for the remaining terms, we use the inequality $z \geq 1 + \log z$ for all $z > 0$ and have
\[
\text{Tr}(U^{T}DXV) \geq \sum_{ip} (DXV)_{ip}U_{ip}'(1 + \log \frac{U_{ip}}{U_{ip}'})
\]
\[\text{Tr}(UU^{T}U^{(a)}(U^{(a)})^{T}) \geq \sum_{ipq} (U^{(a)}(U^{(a)})^{T})_{pq}U_{ip}'U_{ip}'(1 + \log \frac{U_{ip}U_{ip}'}{U_{ip}'U_{ip}'})
\]

Due to the lemma 2 and the inequality $2ab \leq a^2 + b^2$, we can reformulate the second term of $F(U, U')$, which is bounded by
\[
\sum_{ip} (DU'V^{T}V + U'\gamma)_{ip}U_{ip}^{2} \leq \sum_{ip} (DU'V^{T}V + U'\gamma)_{ip}U_{ip}^{2} + U_{ip}^{4}
\]
\[\leq \sum_{ip} (U^{(a)}(U^{(a)})^{T})_{pq}U_{ip}'U_{ip}'(1 + \log \frac{U_{ip}U_{ip}'}{U_{ip}'U_{ip}'})
\]

Therefore, we have the final formulation of our auxiliary function:
\[
F(U, U') = -2 \sum_{ip} (DXV)_{ip}U_{ip}'(1 + \log \frac{U_{ip}}{U_{ip}'})
\]
\[+ \sum_{ip} (DU'V^{T}V + U'\gamma)_{ip}U_{ip}^{2} + U_{ip}^{4}
\]
\[2U_{ip}^{3}
\]
\[\sum_{ip} \lambda_{ls}(U'L_{ip}U_{ip}' + U'L_{ip}U_{ip}')
\]
\[- 2 \sum_{ip}(U^{(a)}(U^{(a)})^{T})_{pq}U_{ip}'U_{ip}'(1 + \log \frac{U_{ip}U_{ip}'}{U_{ip}'U_{ip}'})
\]

To find stationary point of $F(U, U')$, we should take the derivative of $F(U, U')$ with respect to $U$ and fix $U'$ according to lemma 1
\[
\frac{\partial F(U, U')}{\partial U_{ip}} = -2(DXV)_{ip}U_{ip}' + 2(DU'V^{T}V + U'\gamma)_{ip}U_{ip}^{3}
\]
\[+ \sum_{s} \lambda_{ls}(4(U'L_{ip}U_{ip}' + U'L_{ip}U_{ip}'))_{ip}U_{ip}^{3}
\]
\[- 4(U^{(a)}(U^{(a)})^{T})_{ip}U_{ip}' - 4(U^{(a)}(U^{(a)})^{T})_{ip}U_{ip}'
\]
Now we should find the minimum of $F(U, U')$, we set 
\[
\frac{\partial F(U, U')}{\partial U_{ip}} = 0
\]
to zero and substituting $\gamma = \text{thus we can get the}
\]
stationary point of $U_{ip}$ and the updating rule is
\[
U_{ip} \leftarrow U_{ip} \frac{\sqrt{(DVX)'_{ip} + 2 \sum_s \lambda_s (U'((s)')'U)'_{ip}}}{(D(U'V)V)'_{ip} + 2 \sum_s \lambda_s (U(U'V)V)'_{ip}}
\]

As further proof, we take the second derivative with respect to $U$, we get a positive semidefinite Hessian matrix:
\[
\frac{\partial^2 F(U, U')}{\partial U_{ip} \partial U'_{jq}} = (2(DVX)'_{ip} U_{ip}''_{ip} + 6(D(U'V)V')_{ip} U_{ip}''_{ip} +
\sum_s \lambda_s [12(U'U'V'U')_{ip} U_{ip}''_{ip} + 4(U'((s)')'U')_{ip} U_{ip}''_{ip}])
\]
\[
\[
\]
Obviously, above Hessian matrix is a positive semidefinite diagonal matrix, which means $F(U, U')$ is a convex function of $U$. Therefore, we can obtain the global minimum of $F(U, U')$ by setting the value of Eq. 37 equals 0, which gives rise to the updating rule in Eq. 38. And the objective function values of $J_1(U,V)$ will be non-increasing by following Lemma 4.

3) Convergence analysis

According to the alternating optimization steps described in above, since we can find the optimal solution of each subproblem, Algorithm 1 will converge. Now we give convergence study of our semantic weighted framework without any constraints by the updating rules in Algorithm 1.

Theorem 2. In each iteration of Algorithm 1 the updated target $U$ will monotonically decrease the objective problem Eq. 6 without any constraints by the updating rules in Algorithm 1.

We need the following lemma:

Lemma 4. For any positive number $u$ and $v$, the following inequality holds:
\[
u - \frac{u^2}{2v} \leq v - \frac{v^2}{2v}
\]

Proof. Let $\hat{U}(l)$ denote the updated $U(l)$ in each iteration. And we have
\[
\hat{U}(l) = \arg \min_{U(l), V(l) \geq 0} \sum_{l=1}^{n} \emph{w}(l) \|X(l) - U(l)'V(l)\|^2_{2,1}
\]
we replace $u$ and $v$ with $x_i - u_i^{l+1}V^T$ and $x_i - u_i^lV^T$, respectively. We can derive:
\[
\|x_i - u_i^{l+1}V^T\|^2_2 - \frac{\|x_i - u_i^{l+1}V^T\|^2_2}{\|x_i - u_i^lV^T\|^2_2} \leq \|x_i - u_i^lV^T\|^2_2 - \frac{\|x_i - u_i^lV^T\|^2_2}{\|x_i - u_i^lV^T\|^2_2}
\]
which is further equivalent to:
\[
\|x_i - u_i^{l+1}V^T\|^2_2 - \|x_i - u_i^lV^T\|^2_2 \leq \hat{w}(l)\frac{\|x_i - u_i^{l+1}V^T\|^2_2}{\|x_i - u_i^lV^T\|^2_2} \leq \hat{w}(l)\frac{\|x_i - u_i^{l+1}V^T\|^2_2}{\|x_i - u_i^lV^T\|^2_2}
\]
where $\frac{\|x_i - u_i^{l+1}V^T\|^2}{\|x_i - u_i^lV^T\|^2} < 1 \leq \hat{w}(l)$. Summing Eq. 41 and Eq. 43 we can arrive at
\[
\hat{w}(l)\|x_i - u_i^{l+1}V^T\|^2_2 \leq \hat{w}(l)\|x_i - u_i^lV^T\|^2_2
\]
Thus, the iterative optimization will monotonically decrease the objective of the Eq. 6 in each iteration until it converges. \(\square\)

Following the same process of optimization on the above, we could also get the following update rules for objective function with constraint $R_{CS}$ instead of $R_{IDS}$:
\[
\hat{U}(l) \leftarrow \frac{(X(l)'^{T}D(l)'^{T}U(l))}{\sqrt{(D(l)'^{T}(U(l))'^{T}D(l)'^{T}(U(l))))}}
\]
and update rules for objective function with constraint $R_1$:
\[
\hat{U}(l) \leftarrow \frac{(X(l)'^{T}D(l)'^{T}U(l))}{\sqrt{(D(l)'^{T}(U(l))'^{T}D(l)'^{T}(U(l))))}} + \sum_s \lambda_s U'(s)
\]

V. EXPERIMENTS

In this section, we evaluate the effectiveness of the proposed SWVMF for those large-scale and regular multi-view clustering tasks from the aspect of clustering performance. All the experiments are implemented using Python3.7 on a standard Windows PC with an Intel 3.2-GHz CPU and 64-GB RAM.

A. DATASETS

Six datasets are employed in the evaluation: MT-COM\(^3\), D-MOV\(^4\), D-BOOK\(^5\), Last.fm\(^6\), Yelp\(^7\), 3-Sources\(^8\). Specifically, MT-COM is a large-scale review data consists of 101,214 movie items associated with five views which we call “summary”, “short comment”, “long review”, “report” and “users” respectively. In this data, there are 90,340 movie items and rest of it contains episode, opera, etc. There are totally 9 million comments and reviews combine with 8 million users. MT-COM lists 43 item types such as “action”, “love”, “bloopers”, etc. We retain only items tagged to a single type. D-MOV is composed of 30,098

\(^3\)http://www.mtime.com/link/
\(^4\)https://github.com/gxl121438/SWVMF-dataset
\(^5\)https://developers.douban.com/wiki
\(^6\)http://www.last.fm/api
\(^7\)http://www.yelp.com/dataset_challenge
\(^8\)http://mlg.ucd.ie/datasets

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2939334, IEEE Access
items from 39 classes. For each item, we collect four views. In total, our D-MOV includes 30,098 summaries, 2,444,402 comments, 443,153 reviews and 456,308 users. D-MOV lists 39 movie types and tags one type for each item similar to MT-COM. D-BOOK is comprised of 17,572 books, and we extracted 4 views - Review, Comment, Summary, and User - from a popular book review website. Each book contains a summary, 1~180 reviews, and 1~20 comments. There are 19,121 users in total participate in the interactions of at least one book. Each book has one of 39 tags, such as literature, comic, history, etc. Last.fm contains 9,694 items (artists) including 9,694 bio description, 455,4457 users and 2,993,222 comments. Each item was annotated with one of the 21 music genres. All the relevant textual information can be achieved by API of Last.fm. Yelp is a subset of the Yelp Challenge (YDC) dataset, which contains 11,537 items. We randomly sample the equivalent amount of items from every category, and it also has three views just like Last.fm. 3-Sources is a textual content data that was collected from three well-known online news sources: BBC, Reuters, and The Guardian. In total, it consists of 416 distinct news manually categorized into 6 topical labels. Among them, there are 169 stories reported in all three sources, which are used as three views in our experiments. Overall, our three constructed web datasets (MT-COM, D-MOV, D-BOOK) have relevant semantic information, while other three (Last.fm, Yelp, 3-Sources) don’t (see Table 2).

Table 2. Description of the multi-view datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Samples</th>
<th># Views</th>
<th># Classes</th>
<th># Semantic information</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT-COM</td>
<td>54890</td>
<td>5</td>
<td>35</td>
<td>yes</td>
</tr>
<tr>
<td>D-MOV</td>
<td>27685</td>
<td>4</td>
<td>20</td>
<td>yes</td>
</tr>
<tr>
<td>D-BOOK</td>
<td>15324</td>
<td>4</td>
<td>20</td>
<td>yes</td>
</tr>
<tr>
<td>Last.fm</td>
<td>9694</td>
<td>3</td>
<td>21</td>
<td>no</td>
</tr>
<tr>
<td>Yelp</td>
<td>2624</td>
<td>3</td>
<td>7</td>
<td>no</td>
</tr>
<tr>
<td>3-Sources</td>
<td>169</td>
<td>3</td>
<td>6</td>
<td>no</td>
</tr>
</tbody>
</table>

B. DATA BALANCE

Table 2 summarizes the final characteristics of those data sets used in experiments. Data imbalance problem is recognized as one of the major problems in the most real web data mining applications since labeling data do not have exactly equal number of items in each class, especially on large-scale datasets. We take D-MOV and D-BOOK as our two examples and Fig. 4 shows the statistics of number of items in top 20 categories on both data. As can be seen, our original data is imbalance and the distribution is very skewed: the top category takes 19.7% & 16.2% items and the top three categories take 46.8% & 40.0% items, respectively. Such an imbalance data influences the clustering evaluation greatly. To overcome these challenges, one commonly used strategy is called resampling, such as random undersampling for the large categories [9, 27] and our final MT-COM consists of 54,890 items from 35 categories, D-MOV consists of 27,685 items from 20 categories, D-BOOK has 15,324 items from 20 categories either and our final Yelp dataset consists of 2,624 items from 7 categories: Health & Medical, Active Life, Local Services, Pets, Nightlife, Home Services and Arts & Entertainment.

C. EXPERIMENTAL SETTINGS

To evaluate the performance of the proposed method, we compare our method with the following algorithms representative unsupervised clustering algorithms.

1) SVD. We run SVD on the original data matrix, using the objective latent number of dimensions as K, then cluster the reduced space using k-means. This is a typical SVD workflow for clustering [25].

2) CoRe [27]. It proposed the objective functions to co-regularize the eigenvectors of all views’ Laplacian matrices.

3) MultitNMF [9]. This method developed a solution on consensus-based regularization for NMF to group the multi-view data.

4) MVFS [37]. MVFS generates pseudo labels by multi-view spectral analysis to guide the feature selection for each view.

5) PcoNMF [10]. This is a recent pair-wise co-regularization method for clustering the whole mapped data. Through the pair-wise co-regularization, it is expected that the coefficient matrices learned from two views can complement with each other during the factorization process.
6) **CMVNMF** [11]. This approach proposed a novel small number of constraints on must-link sets and cannot-link sets based on the NMF framework.

7) **RMFS** [38]. This recent work applies robust multi-view k-means to obtain the robust and high quality pseudo labels for sparse feature selection in an efficient way.

Our framework is devised for incorporating with constraints: $R_1$, $R_{IS}$ and $R_{CS}$. Such a setting provides a clear view of effects on different constraints. We use the parameter setting for the baseline methods as suggested in the sensitive analysis section of the original papers. In this work, $\lambda_{IS}$ determines the weight of the constraint in co-regularization, and we set $\lambda_{IS} = 1$ for each pair of view in all experiments. We also set our reduced dimension $K$ equals to the number of classes, empirical $\epsilon = 0.001$ and we calculate each weighting matrix $W^{(l)}$ for different views based on our weighting scheme. We run K-means 100 times and select the best clustering result to initialize all the NMF methods, and we iterate algorithm 20 rounds to achieve final average results. For all the used text datasets, we apply the TF-IDF transformation on all the item-word frequency matrices and normalize the TF-IDF vector to unit length. It should be noted that our item-user matrix is a zero-one matrix and it has no rating feature. We adopt the widely used clustering accuracy (ACC) and normalized mutual information (NMI) [39] as our evaluation metrics. The details about the two metrics are described as the following:

1) **Clustering Accuracy (ACC)** is a widely used metric to evaluate the clustering performance. It is defined as following:

   \[
   ACC = \frac{\sum_{i=1}^{n} I(\delta(p_i), l_i)}{n}
   \]  

   where $I(a,b)$ is an indicator function that $I(a,b) = 1$ if $a = b$, and $I(a,b) = 0$ otherwise, $p_i$ and $l_i$ are the clustering label and true label respectively. $\delta(p_i)$ is a mapping function that matches the clustering label to the best true label. A large ACC value indicates a better clustering result.

2) **Normalized Mutual Information (NMI)** is another usually used metric, and it is defined as:

   \[
   NMI = \frac{MI(C, C')}{{max}(H(C) + H(C'))}
   \]

   where $H(C)$ and $H(C')$ are the entropy of $C$ and $C'$, $MI(C, C')$ is the mutual information between the true label $C$ and the clustering label $C'$. Higher value of NMI indicates better quality of clustering.

### D. SINGLE-VIEW CLUSTERING

Running clustering performance on the single views establishes a baseline for comparison against multi-view clustering. We can compare our method with other different single view clustering algorithms such as k-means, SVD and NMF. Since k-means often converges to local minima, we repeat each experiment 20 times and report the average performance. We should note that our $SWNMF$ without any regularizers if we implement single-view clustering performance. Therefore, our $SWNMF$ represents $W \|X^{(l)} - U^{(l)}(V^{(l)})^T \|_F$ in doing single-view clustering.

Fig. 5 shows all the single-view clustering results on our four large datasets, and Table 3 and Table 4 demonstrate the performance of single-view clustering on two small datasets. The best performing algorithm’s results are bolded. For all the figures and tables, our $SWNMF$ achieves best performance on each dataset. In Fig. 5 (a)-(f), the performance variation across different views is consistent in top three large-scale datasets: the users view preforms worst, and review view or report view preforms best. Because in our three constructed corpus, user view is the most sparseness view of all content views. The only one real reason for this situation is that many users would like to comment those items that are very popular. But for those unpopular web items, only a few users choose to give their comments. However, the situation goes a little difference in Fig. 5 (g)-(h). Because our Last.fm is not larger than our constructed corpus, that is the reason why the extent of user view sparseness is lower than other text content view. And user view performs better than the text content view - Description. SVD method maps the data into orthogonal bases, which may lead to negative values in factor matrix, the clusters information are difficult to interpret naturally [25]. For our small data Yelp, the textual content view preforms also better than the user view. In general, our framework shows a better performance than the standard single-view NMF clustering method if we incorporate the semantic information strategically.

Meanwhile, we demonstrate our $SWNMF$ performance that combines with different semantic weighted strategies on single view clustering in Fig. 6 and Fig. 7. For comparison’s purpose, we take D-MOV and D-BOOK as our experimental data since they have the same number of views. $SWNMF-S1$ means our framework utilizes semantic weighted scheme that with user preference information (by Eq. 10) and $SWNMF-S2$ represents our method combines

---

**TABLE 3. Single-view clustering ACC on small datasets.**

<table>
<thead>
<tr>
<th>View</th>
<th>Yelp 20</th>
<th>3-Sources</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Des</td>
<td>Comment</td>
<td>User</td>
</tr>
<tr>
<td>SVD</td>
<td>38.7</td>
<td>42.8</td>
<td>22.4</td>
</tr>
<tr>
<td>k-means</td>
<td>28.4</td>
<td>52.6</td>
<td>30.1</td>
</tr>
<tr>
<td>NMF</td>
<td>39.8</td>
<td>59.8</td>
<td>28.5</td>
</tr>
<tr>
<td>SWNMF</td>
<td>48.8</td>
<td>67.3</td>
<td>43.4</td>
</tr>
</tbody>
</table>

**TABLE 4. Single-view clustering NMI on small datasets.**

<table>
<thead>
<tr>
<th>View</th>
<th>Yelp 20</th>
<th>3-Sources</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Des</td>
<td>Comment</td>
<td>User</td>
</tr>
<tr>
<td>SVD</td>
<td>36.8</td>
<td>40.4</td>
<td>16.1</td>
</tr>
<tr>
<td>k-means</td>
<td>25.2</td>
<td>48.6</td>
<td>29.3</td>
</tr>
<tr>
<td>NMF</td>
<td>38.7</td>
<td>55.2</td>
<td>28.4</td>
</tr>
<tr>
<td>SWNMF</td>
<td>46.3</td>
<td>66.4</td>
<td>41.9</td>
</tr>
</tbody>
</table>
with semantic scheme without those preference information (by Eq. [1]). Specifically, those view matrices that have user preference information, which usually belongs to the textual content type. However, in our setting, user view forms a zero-one matrix without relevant semantic information. Therefore, we can see our single-view clustering result is same on user view matrix. Obviously, our semantic weighted scheme shows a good performance on two datasets in compare with standard method NMF, especially for those large sparse data. The reason why our $SWNMF - S1$ performs better than $SWNMF - S2$ on the D-MOV is that D-MOV is much sparser than D-BOOK.

**E. MULTI-VIEW CLUSTERING**

In Table 5 and Table 6, we present results of all methods measured by ACC and NMI for each dataset. Both mean value and standard deviation are reported, and best results are formatted in bold, while second best result are underlined. Overall, it can be seen that our framework shows a good performance on all multi-view datasets. And our $SWNMF-R_{1S}$ is very competitive, always better than the other baselines. From the experimental comparisons, we observe that: 1)The semantic weighted NMF framework usually outperforms the standard NMF framework-based methods. This may indicate that those standard NMF-based algorithms are usually used for non-sparseness learning and ignore the sparsity of data structure, especially on our large-scale datasets. 2)$SWNMF-R_{1S}$ outperforms the second best standard NMF-based algorithm in terms of ACC/NMI as 14.4%/20.4% on MT-COM, 12.2%/13.3% on D-MOV, 14.0%/15.6% on D-BOOK, 10.1%/11.0% on Last.fm, 9.4%/15.1% on Yelp, and 5.0%/7.9% on 3-Sources. 3)Among the multi-view clustering method with different co-regularization constraint, the similar pair-wise constraint
Table 5. ACC on six real-world datasets (%)

<table>
<thead>
<tr>
<th>Method</th>
<th>MT-COM</th>
<th>D-MOV</th>
<th>D-BOOK</th>
<th>Last</th>
<th>Yell</th>
<th>3-Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVD</td>
<td>(34.1)</td>
<td>(38.5)</td>
<td>(27.7)</td>
<td>(4.1)</td>
<td>(53.7)</td>
<td>(58.1)</td>
</tr>
<tr>
<td>CoRe</td>
<td>(35.2)</td>
<td>42.8</td>
<td>43.1</td>
<td>51.7</td>
<td>60.8</td>
<td>47.9</td>
</tr>
<tr>
<td>MultiNMF</td>
<td>(39.7)</td>
<td>42.1</td>
<td>48.5</td>
<td>45.5</td>
<td>54.2</td>
<td>68.4</td>
</tr>
<tr>
<td>MVFS</td>
<td>(33.7)</td>
<td>38.8</td>
<td>41.9</td>
<td>46.8</td>
<td>53.8</td>
<td>62.8</td>
</tr>
<tr>
<td>PoNMF</td>
<td>(40.2)</td>
<td>46.3</td>
<td>49.5</td>
<td>51.8</td>
<td>67.6</td>
<td>73.3</td>
</tr>
<tr>
<td>CMVNMF</td>
<td>(43.3)</td>
<td>50.4</td>
<td>55.7</td>
<td>60.4</td>
<td>68.2</td>
<td>74.9</td>
</tr>
<tr>
<td>RMFS</td>
<td>(35.8)</td>
<td>40.2</td>
<td>46.8</td>
<td>52.7</td>
<td>59.9</td>
<td>64.4</td>
</tr>
<tr>
<td>SWNMF</td>
<td>(42.7)</td>
<td>48.9</td>
<td>52.1</td>
<td>58.2</td>
<td>63.1</td>
<td>69.8</td>
</tr>
<tr>
<td>SWNMF-RCs</td>
<td>(44.3)</td>
<td>(4.2)</td>
<td>(4.1)</td>
<td>(2.2)</td>
<td>(2.5)</td>
<td>(1.7)</td>
</tr>
<tr>
<td>SWNMF-R</td>
<td>(43.5)</td>
<td>51.8</td>
<td>58.8</td>
<td>62.6</td>
<td>68.8</td>
<td>73.6</td>
</tr>
<tr>
<td>SWNMF-Rs</td>
<td>(50.6)</td>
<td>57.4</td>
<td>64.8</td>
<td>67.2</td>
<td>75.3</td>
<td>78.8</td>
</tr>
</tbody>
</table>

Table 6. NMI on six real-world datasets (%)

<table>
<thead>
<tr>
<th>Method</th>
<th>MT-COM</th>
<th>D-MOV</th>
<th>D-BOOK</th>
<th>Last</th>
<th>Yell</th>
<th>3-Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVD</td>
<td>(31.6)</td>
<td>36.7</td>
<td>39.8</td>
<td>(2.2)</td>
<td>(1.9)</td>
<td>(0.4)</td>
</tr>
<tr>
<td>CoRe</td>
<td>34.4</td>
<td>37.1</td>
<td>41.4</td>
<td>48.6</td>
<td>57.8</td>
<td>43.6</td>
</tr>
<tr>
<td>MultiNMF</td>
<td>(36.3)</td>
<td>36.1</td>
<td>44.6</td>
<td>42.4</td>
<td>53.7</td>
<td>60.2</td>
</tr>
<tr>
<td>MVFS</td>
<td>33.2</td>
<td>35.5</td>
<td>41.1</td>
<td>45.7</td>
<td>53.1</td>
<td>59.7</td>
</tr>
<tr>
<td>PoNMF</td>
<td>(38.7)</td>
<td>44.8</td>
<td>48.8</td>
<td>47.6</td>
<td>64.4</td>
<td>72.8</td>
</tr>
<tr>
<td>CMVNMF</td>
<td>(40.1)</td>
<td>49.6</td>
<td>52.9</td>
<td>59.2</td>
<td>64.8</td>
<td>71.3</td>
</tr>
<tr>
<td>RMFS</td>
<td>(32.6)</td>
<td>37.9</td>
<td>42.9</td>
<td>52.3</td>
<td>58.8</td>
<td>63.2</td>
</tr>
<tr>
<td>SWNMF</td>
<td>41.8</td>
<td>47.6</td>
<td>51.2</td>
<td>53.3</td>
<td>67.1</td>
<td>73.4</td>
</tr>
<tr>
<td>SWNMF-RCs</td>
<td>(4.4)</td>
<td>(4.4)</td>
<td>(4.4)</td>
<td>(2.4)</td>
<td>(3.8)</td>
<td>(1.6)</td>
</tr>
<tr>
<td>SWNMF-R</td>
<td>42.4</td>
<td>49.7</td>
<td>58.5</td>
<td>61.4</td>
<td>67.7</td>
<td>72.1</td>
</tr>
<tr>
<td>SWNMF-Rs</td>
<td>(5.8)</td>
<td>(4.3)</td>
<td>(3.6)</td>
<td>(2.7)</td>
<td>(3.2)</td>
<td>(2.7)</td>
</tr>
</tbody>
</table>

(SWNMF-RcS) performs better than the simple pair-wise constraint (SWNMF-Rc), and SWNMF-Rc slightly performs better than our SWNMF-RcS, which validates that the algorithm based on our proposed co-regularization constraint framework might be a better way to capture the difference in intrinsic connection between every two data points.

As our baseline method, CMVNMF also performs a good result on our web datasets. A plausible explanation is that method designs the disagreement between each pair of views by the fact that indicator vectors of two items from two different views should be similar if they belong to the same cluster and dissimilar otherwise. They use the disagreement between the views to guide the factorization of the matrices, which captures the fundamental structure diversity. And we can observe those feature selection-based methods underperform the multi-view clustering. A reasonable explanation is that feature selection would result in loss of large amount of feature information, especially for those sparse web data. Due to the loss of feature information, more intrinsic structure information between different views are hard to precisely explore.

F. PARAMETER SENSITIVITY ANALYSIS

In the proposed framework, there are three parameters to be tuned, i.e. $\lambda_s$, $K$, $\epsilon$. In the experiments, we first employ the grid search strategy to find the best choices for all parameters on our constructed web datasets, i.e. MT-COM, D-MOV, D-BOOK. When applying to another dataset, we only fine-tune these parameters on randomly selected 10k data instances from this dataset, and then the optimal parameters are exploited for clustering the whole dataset. In our SWNMF framework, there is only one regularization parameter for each pair of views: $\lambda_s$. Relative $\lambda_s$ determines the weight of the consistency on pair’s similarity in co-regularization.

We take other three standard NMF-based frameworks as our baseline methods. In Fig. 9, Fig. 10 and Fig. 11, they explicitly show the multi-view clustering variation curves w.r.t different parameters on three constructed web datasets, respectively.

For those datasets, our semantic weighted-based framework is relatively stable across a wide spectrum of settings. We can see that the best clustering results are established when the values of $\lambda_s$ located in a range from 0.7 to 1.4. Through these accuracy curves, a better understanding of the sparseness problem of large view matrices, resulting from similarity pair-wise constraints. That is why the slope of curves on MT-COM is more higher than the D-MOV. In contrast, our view matrices in D-BOOK data is not sparseness, which results in a smooth change. According to the above words, our optimization parameter $\lambda_s$ can be set to 1.

Performance vs. Number of clusters All the above experiments are evaluated by the exact number of clusters in data, i.e. $K$, but how does the performance change w.r.t different numbers of clusters? To this end, we also perform experiments on all datasets to evaluate the stabilities of SWNMF w.r.t number of clusters. Fig. 8(a) & (b) illustrates the clustering performance changes with increasing number of clusters from 25 to 45 with an interval 2. Fig. 8(c) - (h) illustrates the clustering performance when the number of clusters in a range from 10 to 30 with an interval 2, and Fig. 8(i) - (l) demonstrates a range from 1 to 21 with interval 2. Intuitively, the best clustering performance of SWNMF-RcS is obtained when the number of cluster is around of the original data category (see Table 2). Obviously, the less sparse data is, the smoother performance value with variation of $K$.

Performance vs. $\epsilon$ In our two semantic weighted strategy, we also explore the sensibility of $\epsilon$ during our multi-view clustering procedure. By default, Fig. 12 shows the performance of SWNMF-RcS when varying $\epsilon$ while holding other optimal parameters. Fig. 12(a) - (f) illustrates the clustering performance changes with increasing value of $\epsilon$ from 0.0001 to 0.003 with an interval 0.0005. We can observe that the best
clustering performance of SWNMF-RI5 is obtained when the value of $\epsilon$ in the 0.001-0.0015. According to the above analysis, our results indicate that SWNMF framework is stable across a wide range of parameters.

G. CONVERGENCE ANALYSIS

We take D-MOV and D-BOOK as our example data. In Fig. 13 and Fig. 14 we show the convergence of our proposed algorithm. The stop criteria for our framework is defined as following:

$$\frac{\|r^{t+1} - r^t\|}{r^t} < 10^{-6}$$

where $r^t$ is the objective value in the $t$-th iteration. To show the difference among these four SWNMF-based methods clear, we only perform the first 200 iterations. We can find that SWNMF needs the least iterations to be converged.

Combining with similarity pair-wise constraint, our proposed SWNMF-RI5 consume more time due to the constraint strategy, especially on the larger dataset D-MOV.

VI. CONCLUSION

In this paper, we have proposed a novel semantic weighted multi-view clustering framework to cluster the multi-view web data, which also addressed the sparseness problem in real-world datasets. For this purpose, we collected three large comment-based web datasets in this work. We have developed an overall iterative optimization algorithm. Extensive experiments have demonstrated that the proposed method is effective. We especially showed that our algorithm could handle sparse data properly. The comparison between our approach and other unsupervised clustering algorithms indicates that SWNMF can cluster items effectively. We also take extra care to choose parameters wisely.
In the future, we will study how to model any other features generated by users, such as user preferences and user reactions. We will also investigate how to improve the algorithm efficiency when dealing with the huge real-world datasets.

REFERENCES


