Self-Weighted Multi-View Clustering with Deep Matrix Factorization

Beilei Cui DLUT_CBL@163.COM
Hong Yu* HONGYU@DLUT.EDU.CN
Tiantian Zhang ZTT_DLUT@163.COM
Siwen Li LISIWEN_DUT@FOXMAIL.COM
School of Software, Dalian University of Technology, Dalian, China

Editors: Wee Sun Lee and Taiji Suzuki

Abstract

Due to the efficiency of exploring multiple views of the real-word data, Multi-View Clustering (MVC) has attracted extensive attention from the scholars and researches based on it have made significant progress. However, multi-view data with numerous complementary information is vulnerable to various factors (such as noise). So it is an important and challenging task to discover the intrinsic characteristics hidden deeply in the data. In this paper, we present a novel MVC algorithm based on deep matrix factorization, named Self-Weighted Multi-view Clustering with Deep Matrix Factorization (SMDMF). By performing the deep decomposition structure, SMDMF can eliminate interference and reveal semantic information of the multi-view data. To properly integrate the complementary information among views, it assigns an automatic weight for each view without introducing supernumerary parameters. We also analyze the convergence of the algorithm and discuss the hierarchical parameters. The experimental results on four datasets show our algorithm is superior to other comparisons in all aspects.

Keywords: Multi-view Clustering, Deep Matrix Factorization, Self-Weighted.

1. Introduction

Real-world datasets are always obtained from multiple sources with abundant feature representations, which means these source data describe various information of the same dataset (Zhao et al., 2017; Huang et al.). For example, a webpage has both text and image information, which is depicted from different perspectives (i.e. views or modalities). And each picture can be specified by different features, e.g. LBP (Ojala et al., 2000), SIFT (Lowe, 2004) and HOG (Dalal and Triggs, 2005). These different views composed of multiple features constitute a huge data network, which provide more comprehensive and useful information. MVC is a typical unsupervised classification method dealing with the heterogeneous data information. It divides the data into a set of disjoint subsets with high intra-cluster similarity and low inter-cluster similarity (Liu et al., 2013; Gao et al., 2016; Xu et al., 2017).

Recently, lots of researches on MVC has emerged, which are devoted to the development of effective MVC algorithms. To fully use the complementary information, most of them

* Corresponding author.

are based on single-view clustering algorithms with carefully fusion of views (Bruno and Marchand-Maillet, 2009). By using the common regularizing constraints, co-regularized multi-view spectral clustering makes the clustering results of different views agree with each other (Kumar et al., 2011b). Another multi-view spectral clustering method based on the co-training idea learns clustering of one view, and then alternately modifies the graphical structure of others with the help of marks (Kumar and Iii, 2011a). Cai et al. extended the traditional K-means algorithm into the multi-view case with a structured sparse induction norm (Cai et al., 2013). Yu et al. normalized the consistent subspace representation by adding proper regular terms (Yu et al., 2018). Guo et al. proved the correlation between the spectral clustering algorithm and the kernel matrix learning process theoretically, so that the method can determine kernel weights and clustering simultaneously (Guo et al., 2014) .

As one of the most popular high-dimensional data processing tools, non-negative matrix factorization (NMF) has received more and more attention. And it has been used to improve the correctness and effectiveness of clustering (Du and Swamy, 2010; Liu et al., 2013; Leng et al., 2018). Although these algorithms have improved the clustering performance to a large extent, they can only explore the shallow information limited by structure, which also reduces their application scopes. The Deep Semi-NMF structure can not only retain the interpretability of single-layer NMF, but also eliminate the interference in multi-view data, which is helpful to extract the best common clustering features. H. Zhao et al. used this deep structure to capture the hidden information and generate a valid consensus at the last level (Zhao et al., 2017). Cai Xu et al. also utilized this hierarchical model to learn the semantic structure of the multi-view data, and they particularly took the consistent and complementary information among different views into account (Xu et al., 2018).

While operating multiple views of data, most existing algorithms choose a manual parameter to control the distribution of their weights, which is easy to implement. However, the hyper-parameters will also inevitably reduce the convenience of execution and the simplicity of program (Nie et al., 2017b).Yang Liu et al. designed a proper updating rule for parameters. Their parameter-free model can automatically learn and update the weight of each view with clustering performance improved (Liu et al., 2018). This idea can also apply to multiple kernel learning frameworks. It enables algorithms to automatically assign appropriate weights to kernels without adding additional parameters (Kang et al., 2018).

Based on this, we extend the semi-NMF with deep hierarchical framework, and initiatively apply the self-weighted strategy to this structure. In addition, there exists some similarities between the deep semi-NMF structure and the multi-layer neural network of deep learning (Hinton and Salakhutdinov, 2006; Bengio et al., 2009; Trigeorgis et al., 2014). Fig.1 shows a schematic diagram of SMDMF. The deep model decomposes and filters the multi-view data matrices ($X^{(v)}$) layer by layer. Cooperating with weight parameters (α), our model extracts the underlying complementary information and finally automatically obtains the common representation matrix $\boldsymbol{G}_m$ shared by all the views in the highest abstraction layer.

In short, our major contributions are organized in three sections:

- A new deep semi-NMF method is designed for a consensus representation of the multi-view data. The hierarchical structure captures the implicit and valid information of data without interference, such as outliers and noise, and finally gains a unified

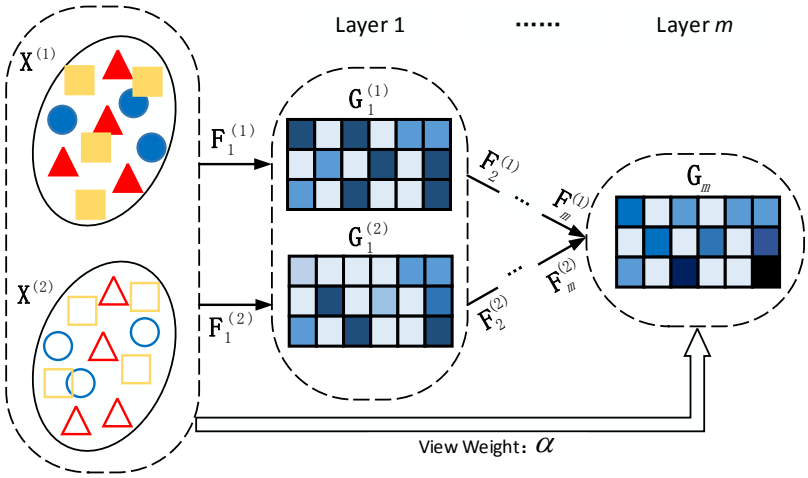

Figure 1: Framework of SMDMF. For the sake of brevity, we take two views as an example. While the data in each view is decomposed by the deep factorization structure, an automatic weight is updated correspondingly. SMDMF combines the representation matrices of the multi-view data in the highest abstraction layer. And finally we obtain a consensus for future operation.

representation matrix of all the views. The consensus matrix can be considered as a proper approximation of the original data to improve the performance of MVC.

- Except the necessary hierarchical parameters (layer size and single-layer dimensions), our model does not use hyper-parameters to control the combination of complementary information like these previous. It can automatically allocate an appropriate weight to each view for information fusion, which greatly simplifies the whole structure and reduces the complexity of operation.

- Furthermore, We also evaluate SMDMF on four real-world datasets. The proposed algorithm achieves superior results and high performance in comparison with some state-of-art ones.

Notations. Throughout the paper, matrices are written as uppercase. Given a matrix X, its ij-th element is denoted by X_{ij}. The Frobenius norm of matrix X is denoted by $||X||_F$. The transpose of matrix X is denoted by X^T. $X^\dagger$ refers to the Moore-Penrose pseudo inverse of matrix X.

2. Related work

In this section, we will introduce a brief review of semi-NMF, and then extend it to form a deep structure. Finally we elaborate the structure in the setting of multi-view data with weight coefficient and trade-off parameter.

2.1. A Review of semi-NMF

Semi-NMF extends NMF in which the elements of partial matrices are relaxed to real values in the formulation, so that it can retain the characteristics of NMF with enough interpretability of semantic representation, and gain more general range of applications in the real world (Du and Swamy, 2010). Due to the consistency of the objective function form, we can intuitively understand Semi-NMF from the perspective of the K-means algorithm. In other words, Semi-NMF is also known as the relaxed version of the K-means clustering (Chris et al., 2010; Liu et al., 2013; Zhao et al., 2017). Its objective function can be written as follows:

$$\min_{F,G\geq 0} \quad ||X - FG^T||_F^2 \tag{1}$$

where the input data set $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n] \in \mathcal{R}^{d\times n}$ has n samples with d-dimensional features, $F \in \mathcal{R}^{d\times K}/G \in \mathcal{R}^{n\times K}$ is called the basis/encoding matrix of K clusters (Xu et al., 2018). For clustering, F can be approximated to the cluster centroid matrix. G is the "soft" cluster indicator matrix in the hidden space (Zhao et al., 2017), which carries more effective information than the sparse indicator matrix.

2.2. The Deep Semi-NMF structure

However, the simple single-layer semi-NMF ignores the deep intrinsic information existing in the multi-view data, which adversely affects the clustering effect (Georghiades et al., 2002). Inspired by the work in Xu et al. (2018), we utilize the pattern of the Deep Semi-NMF model (Trigeorgis et al., 2014). It looks for a low-dimensional hidden representation embedded in all the views. The deep model decomposes dataset X hierarchically, and the process can be formulated as:

$$X \approx F_1 F_2 \cdots F_m G_m^T \tag{2}$$

where $F_1 \in \mathcal{R}^{d\times p_1}, F_2 \in \mathcal{R}^{p_1\times p_2}, \cdots, F_m \in \mathcal{R}^{p_{m-1}\times p_m}$ denote a series of basic mapping matrices, p_i denotes the dimension of the i-th layer and m represents the total number of layers, $G_m \in \mathcal{R}^{n\times p_m}$ denotes the final common latent representation (Trigeorgis et al., 2014). Actually, the whole step of the deep structure can be translated as G_i, the representation matrix of layer i, is decomposed into a new basis matrix F_{i+1} and a more general consensus matrix G_{i+1} of the next layer.

While preserving the interpretability of the original single-layer NMF, the deep model can also eliminate the negative interference in the multi-source data through layer-wise decomposition. It effectively extracts the clustering-friendly feature of each attribute in the single layer, and then outputs the consensus representation at the abstraction level (Zhao et al., 2017; Xu et al., 2018).

2.3. Multi-view Combination

Some related work focus on the fusion of views in the multi-view scenario (Cai et al., 2013; Guo et al., 2014). A multi-view dataset X is represented by different modalities $X^{(1)}, \cdots, X^{(v)}, \cdots, X^{(V)}$, where V denotes the number of views. After introducing a proper

weight parameter $\omega^{(v)}(v = 1, 2, \cdots, V)$ to integrate different views, we can obtain the Deep Semi-NMF model in the multi-view setting (Nie et al., 2017a). The formulation can be written as

$$\begin{aligned} \min_{\substack{F_i^{(v)}, G_i^{(v)}, \\ \omega^{(v)}, G_m}} \quad & \sum_{v=1}^{V} (\omega^{(v)})^{\gamma} ||X^{(v)} - F_1^{(v)} \cdots F_m^{(v)} G_m^T||_F^2 \\ \text{s.t.} \quad & G_i^{(v)} \geq 0, G_m \geq 0, \sum_{v=1}^{V} \omega^{(v)} = 1, \omega^{(v)} \geq 0 \end{aligned} \tag{3}$$

where X is the given multi-view dataset, and for each view $X^{(v)} \in \mathcal{R}^{d^{(v)} \times n}$, $d^{(v)}$ represents the dimension of data in view v. The adding parameter γ is necessary to smooth the distribution of $\omega^{(v)}$ (Zhao et al., 2017). $F_i^{(v)}$and $G_i^{(v)}$ denote basis and encoding matrices, respectively. And in the highest layer, all views with homology share a common representation matrix G_m.

The introduction of weight hyper-parameter is to consider the significance of different views, and make full use of the complementary information of the multi-view data. However, in order to have satisfactory performance of the algorithm , it is usually essential to adjust the parameter γ within a large range, which increases the runtime challenge drastically. In consideration of this, we propose a parameter-free framework to tackle such problem (Nie et al., 2017a,b).

3. Our Proposal: Self-Weighted Multi-View Clustering with Deep Matrix Factorization

In this section, we present our novel parameter-free framework. In order to omit the weight hyper-parameter, we used an ingenious strategy to generate the objective function of our SMDMF framework, and it can be expressed as

$$\begin{aligned} \min_{F_i^{(v)}, G_i^{(v)}, G_m} \quad & \sum_{v=1}^{V} ||X^{(v)} - F_1^{(v)} \cdots F_m^{(v)} G_m^T||_F \\ \text{s.t.} \quad & G_i^{(v)} \geq 0, G_m \geq 0 \end{aligned} \tag{4}$$

and the Lagrange function of Eq.(4) is

$$\sum_{v=1}^{V} ||X^{(v)} - F_1^{(v)} \cdots F_m^{(v)} G_m^T||_F + \mathcal{G}(\Lambda, A) \tag{5}$$

where $\mathcal{G}(\Lambda, A)$ denotes the linear combination of the Lagrange multipliers Λ (a vector) and the constraints in Eq.(4). For simplicity, Matrix A represents all relevant basis and encoding matrices $(F_i^{(v)}, G_i^{(v)}, G_m)$. By setting to be 0, the derivative of Eq.(5) w.r.t. G_m is equal to

$$\sum_{v=1}^{V} \alpha^{(v)} \frac{\partial ||X^{(v)} - F_1^{(v)} \cdots F_m^{(v)} G_m^T||_F^2}{\partial G_m} + \frac{\partial \mathcal{G}(\Lambda, A)}{\partial G_m} = 0. \tag{6}$$

where

$$\alpha^{(v)} = \frac{1}{2||X^{(v)} - F_1^{(v)} F_2^{(v)} \cdots F_m^{(v)} G_m^T||_F} \tag{7}$$

Since all the relevant matrices in Eq.(7) are known during a complete optimization, $\alpha^{(v)}$ can be considered fixed to some extent. After substituting Eq.(7), Eq.(4) is equivalent to the final objective function

$$\begin{aligned} \min_{F_i^{(v)}, G_i^{(v)}, G_m} \quad & \sum_{v=1}^{V} \alpha^{(v)} ||X^{(v)} - F_1^{(v)} \cdots F_m^{(v)} G_m^T||_F^2 \\ \text{s.t.} \quad & G_i^{(v)} \geq 0, G_m \geq 0 \end{aligned} \tag{8}$$

An intuitive comprehension of $\boldsymbol{\alpha^{(v)}}$, our new parameter-free "weight", is that if the original data in a view is closer to the decomposed matrices, the obtained representation from the view will extract better data information, so the view should gain a larger weight (Nie et al., 2017a; Kang et al., 2018), which is also consistent with the meaning conveyed by Eq.(7). Besides, by comparison to Eq.(3), $\alpha^{(v)}$ plays the same "role" with $\omega^{(v)}$ according to the position. However, ours can be adjusted without human intervention. Specifically, when getting the value of matrices by solving Eq.(8), we can obtain a new weight by Eq.(7), which can be used to calculate a series of new matrices in the next iteration. By repeating this process until convergence, we can acquire the local optimal solution of all parameters corresponding to each other.

Because there is only an automatic weight parameter for each view, SMDMF is more concise and efficient without the process of adjusting hyper-parameters. It can succinctly grasp the hierarchical semantic and complementary information of the multi-view data, and discover a more universal feature representation for MVC algorithm.

4. Optimization

To promote the efficiency of our algorithm, we borrow the idea of layer-wise pre-training in deep leaning of Hinton and Salakhutdinov (2006) to initialize the model variables. We set $\alpha^{(v)} = \frac{1}{V}$ to balance the importance of different views before clustering. According to Trigeorgis et al. (2014), we use semi-NMF to decompose the data matrices of every view as $X^{(v)} = F_1^{(v)} G_1^{(v)T}$, and then $G_1^{(v)}$ is treated as the new data matrix to perform decomposition in the layer 2 until all the layers have been pre-trained. We minimize our final objective by an alternative and iterative algorithm, which optimizes the function value w.r.t. one part of parameters while fixing the remaining ones.

4.1. Updating $F_i^{(v)} (i = 1 \cdots m)$

The cost function is denoted as $\mathcal{C} = \sum_{v=1}^{V} \alpha^{(v)} ||X^{(v)} - F_1^{(v)} \cdots F_m^{(v)} G_m^T||_F^2$. By setting $\partial \mathcal{C} / \partial F_i^{(v)} = 0$, we have

$$F_i^{(v)} = (\xi_i{}^T \xi_i)^{-1} \xi_i{}^T X^{(v)} \widetilde{G}_i^{(v)} (\widetilde{G}_i^{(v)T} \widetilde{G}_i^{(v)})^{-1}. \tag{9}$$

where $\widetilde{G} = \prod_{j=i+1}^{m} F_j^{(v)} G_m$ denotes the reconstruction of the encoding matrix in this layer, and $\xi_i = \prod_{j=1}^{i-1} F_j^{(v)}$. Utilizing the pseudo-inverse notation, the final update rule of $F_i^{(v)}$ is

$$F_i^{(v)} = {\xi_i}^{\dagger} X^{(v)} \widetilde{G}_i^{(v)T\dagger}. \tag{10}$$

4.2. Updating $G_i^{(v)}(i = 1 \cdots m-1)$

Similar to Chris et al. (2010), we firstly derive the solution of $G_i^{(v)}(i = 1 \cdots m-1)$, followed by the convergence proof. That is

$$G_i^{(v)} \leftarrow G_i^{(v)} \sqrt{\frac{[X^{(v)T}\xi_{i+1}]^+ + [G_i^{(v)}{\xi_{i+1}}^T\xi_{i+1}]^-}{[X^{(v)T}\xi_{i+1}]^- + [G_i^{(v)}{\xi_{i+1}}^T\xi_{i+1}]^+}} \tag{11}$$

where $[A]^+$ and $[A]^-$ denote the positive and negative parts of matrix A, separately. Their elements are

$$[A]_{ij}^+ = (|A_{ij}| + A_{ij})/2, \quad [A]_{ij}^- = (|A_{ij}| - A_{ij})/2 \tag{12}$$

Owing to the space limitation, we only give a concise proof process, please refer to Chris et al. (2010) for specific definition of $Z(G, \widetilde{G})$ and more details.

Theorem 1. The solution with restrictions of the update rule in Eq.(11) satisfies the KKT condition.

*Proof.*The Lagrange function is

$$\mathcal{L}(G_i^{(v)}) = \sum_{v=1}^{V} \alpha^{(v)} Tr(-2X^{(v)T}\xi_{i+1}G_i^{(v)T} + G_i^{(v)}\xi_{i+1}^T\xi_{i+1}G_i^{(v)T} - \lambda G_i^{(v)T}) \tag{13}$$

where the Lagrangian multiplier λ imposes the constraints, $G_i^{(v)} \geq 0$. The zero gradient condition offers $\frac{\partial \mathcal{L}}{\partial G_i^{(v)}} = -2X^{(v)T}\xi_{i+1} + 2G_i^{(v)}\xi_{i+1}^T\xi_{i+1} - \lambda = 0$. By the complementary slackness condition, the solution must satisfy a fixed point equation at convergence as

$$(-2X^{(v)T}\xi_{i+1} + 2G_i^{(v)}\xi_{i+1}^T\xi_{i+1})_{kj}(G_i^{(v)})_{kj} = \lambda(G_i^{(v)})_{kj} = 0 \tag{14}$$

When reaching convergence, the limited solution of Eq.(11) meets $G_i^{(v)t} = G_i^{(v)t+1} = G_i^{(v)\infty} = G_i^{(v)}$, i.e.,

$$(G_i^{(v)})_{kj} = (G_i^{(v)})_{kj} \sqrt{\frac{[X^{(v)T}\xi_{i+1}]_{kj}^+ + [G_i^{(v)}{\xi_{i+1}}^T\xi_{i+1}]_{kj}^-}{[X^{(v)T}\xi_{i+1}]_{kj}^- + [G_i^{(v)}{\xi_{i+1}}^T\xi_{i+1}]_{kj}^+}} \tag{15}$$

Since $A = [A]^+ - [A]^-$, Eq.(15) reduces to

$$\left(-2X^{(v)T}\xi_{i+1} + 2G_i^{(v)}\xi_{i+1}^T\xi_{i+1}\right)_{kj}(G_i^{(v)})_{kj}^2 = 0 \tag{16}$$

Obviously, Eq.(14) is equivalent to Eq.(16) (Chris et al., 2010; Zhao et al., 2017). Theorem 1. proves the correctness of the update strategy.

Theorem 2. The residual of $\mathcal{C}$ is monotonically decreasing (non-increasing) under the update rule given in Eq.(11) by fixing others.

Proof. We construct an auxiliary function $Z(G, \widetilde{G})$ to convert the problem,which satisfies

$$Z(G, \widetilde{G}) \geq \mathcal{L}(G), \quad Z(G, G) = \mathcal{L}(G) \tag{17}$$

For any $G, \widetilde{G}$, define

$$G^{(t+1)} = \arg\min_{G} Z(G, G^{(t)}) \tag{18}$$

We have $\mathcal{L}(G^{(t)}) = Z(G^{(t)}, G^{(t)}) \geq Z(G^{(t+1)}, G^{(t)}) \geq \mathcal{L}(G^{(t+1)})$, Thus, $\mathcal{L}(G)$ is monotonically decreasing, i.e. the update rule of $G_i^{(v)}(i = 1 \cdots m-1)$ can converge normally.

4.3. Updating G_m

The Lagrange function is

$$\mathcal{L}(G_m) = \sum_{v=1}^{V} \alpha^{(v)} Tr(-2X^{(v)T}\xi_{m+1}G_m^T + G_m\xi_{m+1}^T\xi_{m+1}G_m^T - \varphi G_m^T) \tag{19}$$

where φ is the Lagrangian multiplier. Taking derivative of $\mathcal{L}(G_m)$ w.r.t. to G_m, we can get

$$\frac{\partial \mathcal{L}}{\partial G_m} = \sum_{v=1}^{V} 2\alpha^{(v)}(-X^{(v)T}\xi_{m+1} + G_m\xi_{m+1}^T\xi_{m+1}) - \varphi\sum_{v=1}^{V}\alpha^{(v)} = 0 \tag{20}$$

According to the updating strategy of $G_i^{(v)}(i = 1 \cdots m-1)$, G_m can be updated as follows

$$G_m \leftarrow G_m\sqrt{\frac{\sum_{v=1}^{V}\alpha^{(v)}([X^{(v)T}\xi_{m+1}]^+ + [G_m\xi_{m+1}{}^T\xi_{m+1}]^-)}{\sum_{v=1}^{V}\alpha^{(v)}([X^{(v)T}\xi_{m+1}]^- + [G_m\xi_{m+1}{}^T\xi_{m+1}]^+)}} \tag{21}$$

The complete procedure is depicted in Algorithm 1. We iteratively repeat all the update rules orderly until convergence. At the highest level, we can obtain the consensus representation of all the views used for clustering (Zhao et al., 2017). For visualization, we exploit spectral clustering (Ng et al., 2001) to process the graph established on G_m by the k-NN algorithm.

Algorithm 1 The algorithm of SMDMF

Input: Multi-view data $\{X^{(v)}\}_{v=1}^{V}$; layer parameter $[p^1, \cdots, p^m]$.

Output: Basis matrices $\{F_i^{(v)}\}, \forall v, i$ and the consensus matrix G_m.

```
Initialization:
  for i = 1 to m, v = 1 to V do
      (F_i^(v), G_i^(v)) ← Semi-NMF (G_{i-1}^(v), p^i). α^(v) ← 1/V.
  end
  while not converge do
      Update the weight of each view α^(v) by Eq. (7).
      Update the basis matrices F_i^(v)(i = 1 ··· m) by Eq. (10).
      Update the encoding matrices G_i^(v)(i = 1 ··· m − 1) by Eq. (11).
      Update the common representation matrix G_m by Eq. (21).
  end
```

5. Experiments

5.1. Datasets and Experiment Setup

We select four datasets for evaluation. The important statistics are summarized in Table 1 and a brief introduction of these datasets is as follows.

Table 1: Description of the datasets

	# instances	# views	# classes
Handwritten	2000	3	10
Yale	165	3	15
Sources	169	3	6
C101-7	1474	3	7

- *Handwritten*: The handwritten number (within 0 to 9 digits) dataset comes from the UCI. Three selected components for the test datasets are: Fourier coefficients of the shape, pixel averages in windows, and profile correlations.
- *Yale*: It consists of 165 original pixel images for 15 subjects, each of which has 11 images with different conditions, e.g. lighting conditions, with/without glasses, etc. Three of these views are chosen for experiments, they are Intensity, LBP, and Gabor.
- *Sources*: The multi-view text dataset was collected from three well-known online news sources: Guardian, BBC and Reuters. We selected 169 articles that were reported by all of them.
- *C101-7*: Caltech101 is a digital image dataset created by California Institute of Technology. We commonly used 1474 images with 7 categories and three views: HOG, GIST, and LBP.

Before operation, we preprocess all datasets in accordance with the method mentioned in Cao et al. (2015). To show the strong adaptability of the deep structure algorithms, we set the number of layers with 1, 2, 3 and 4 respectively when applied to these four datasets.

For comparison , we select 7 classical baselines and state-of-art algorithms. The parameters involved in all algorithms are adjusted to the optimal to ensure comparability. The details are summarized below.

- **BestSV** (Kumar et al., 2011b) is the result of selecting the optimal performance after performing standard spectral clustering on the attributes in each view.
- **ConcatFea** (Kumar et al., 2011b) firstly connects all the features of all views, and then performs standard spectral clustering on the Laplacian operator derived from the joint view representation.
- **Co-reg(P)** (Kumar et al., 2011b) adjusts the clustering hypothesis (Pairwise) among multiple views, aiming at implicitly combining the graphs of multi-view data for better clustering performance.

- **Co-reg(C)** (Kumar et al., 2011b) is similar to Co-reg(P), which is unique in converging the feature vectors of each view into a common centroid to achieve mutual recognition among views.
- **Co-trained** (Kumar and Iii, 2011a) learns clustering result of one view, and then alternately modifies the graph structure of others with the help of a "tag" information.
- **Multi-NMF** (Liu et al., 2013) applies NMF to project each view data into a common potential subspace. This is equivalent to the single-layer version of our structure.
- **DMSNMF** (Zhao et al., 2017) seeks a valid consensus representation at the last level by deep matrix factorization structure. In order to guarantee comparability, the same number of layer is selected and other parameters are adjusted to the optimal when performing DMSNMF and ours.

We use six different evaluation criteria:**ACC, NMI, AR, Precision, Recall, and F-score**. Their ranges are between 0 and 1, with larger values mean better clustering performance. For algorithms involving graph similarity, we follow the strategy in Kumar and Iii (2011a). We repeat every algorithms for 10 times and record the results in the combination of mean and standard deviation.

Table 2: Clustering performance on Handwritten dataset (Mean and Sta Dev,%).

Method	Values	ACC	NMI	AR	F-core	Precision	Recall
BestSV	Mean	0.6871	0.6432	0.5446	0.5907	0.5810	0.6008
	Sta Dev	0.0154	0.0069	0.0128	0.0115	0.0119	0.0114
ConcatFea	Mean	0.6146	0.6238	0.4940	0.5467	0.5227	0.5731
	Sta Dev	0.0079	0.0042	0.0072	0.0065	0.0066	0.0065
Co-reg(P)	Mean	0.8064	0.7582	0.6909	0.7224	0.7088	0.7372
	Sta Dev	0.0188	0.0077	0.0138	0.0122	0.0163	0.0080
Co-reg(C)	Mean	0.7974	0.7636	0.6965	0.7277	0.7090	0.7481
	Sta Dev	0.0199	0.0084	0.0159	0.0141	0.0175	0.0109
Co-trained	Mean	0.8044	0.7722	0.7081	0.7377	0.7276	0.7501
	Sta Dev	0.0068	0.0038	0.0065	0.0058	0.0061	0.0049
Multi-NMF	Mean	**0.8771**	0.7984	0.7534	0.7781	**0.7828**	0.7734
	Sta Dev	0.0132	0.0138	0.0224	0.0202	0.0202	0.0203
DMSNMF	Mean	0.8090	0.8604	0.7791	0.8028	0.7450	0.8702
	Sta Dev	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
SMDMF	Mean	0.8130	**0.8614**	**0.7794**	**0.8032**	0.7437	**0.8730**
	Sta Dev	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 3: Clustering performance on Yale dataset (Mean and Sta Dev,%).

Method	Values	ACC	NMI	AR	F-core	Precision	Recall
BestSV	Mean	0.6059	0.6444	0.4221	0.4590	0.4384	0.4822
	Sta Dev	0.0144	0.0086	0.0122	0.0113	0.0122	0.0106
ConcatFea	Mean	0.5475	0.5986	0.3611	0.4021	0.3826	0.4241
	Sta Dev	0.0058	0.0052	0.0057	0.0054	0.0052	0.0058
Co-reg(P)	Mean	0.5752	0.6306	0.4063	0.4444	0.4233	0.4682
	Sta Dev	0.0132	0.0082	0.0123	0.0115	0.0121	0.0112
Co-reg(C)	Mean	0.6081	0.6606	0.4482	0.4840	0.4567	0.5155
	Sta Dev	0.0170	0.0115	0.0139	0.0131	0.0127	0.0151
Co-trained	Mean	0.6340	0.6690	0.4614	0.4958	0.4735	0.5225
	Sta Dev	0.0123	0.0094	0.0118	0.0111	0.0108	0.0112
Multi-NMF	Mean	0.6006	0.6225	0.3944	0.4335	0.4091	0.4615
	Sta Dev	0.0318	0.0249	0.0246	0.0229	0.0240	0.0257
DMSNMF	Mean	0.7625	0.7496	0.5722	0.5989	0.5838	0.6149
	Sta Dev	0.0021	0.0013	0.0143	0.0133	0.0155	0.0109
SMDMF	Mean	**0.7896**	**0.7552**	**0.5852**	**0.6111**	**0.5967**	**0.6263**
	Sta Dev	0.0023	0.0024	0.0037	0.0035	0.0034	0.0037

5.2. Results

Table 2, Table 3, Table 4 and Table 5 list the experimental results on four datasets. As shown, the methods based on the deep NMF structure (DMSNMF and our SMDMF) outperform almost all the competitors. It shows that these baseline methods based on original data are difficult to process these experimental datasets containing complex information. Nonetheless, the deep structure is forceful to obtain an abstract consensus through deep data mining, which enormously improves the final performance of algorithms.

For Yale dataset in Table 3 and C101-7 dataset in Table 5, our SMDMF model occupies an absolute advantage over all the other algorithms. Table 2 and Table 4 show some slight outliers. However, our parameter-free model simplifies the complexity of operation as well as ensuring effectiveness. Especially, it achieves convergence in about 5 iterations when applied to the Handwritten dataset. We can also conservatively conclude that SMDMF attains better performance than the others, considering the stability (small standard deviation) and efficiency on this two datasets. In brief, results show SMDMF outperforms all the baseline algorithms.

5.3. Analysis

In this part, we will evaluate our SMDMF from the perspective of robustness and sensitivity, i.e. parameter analysis and convergence property. Firstly, we introduce the parameter in our model with optimality analysis. The convergence is discussed later with the help of the

Table 4: Clustering performance on Sources dataset (Mean and Sta Dev,%).

Method	Values	ACC	NMI	AR	F-core	Precision	Recall
BestSV	Mean	0.5698	0.4679	0.3583	0.5015	0.5228	0.4848
	Sta Dev	0.0086	0.0083	0.0129	0.0087	0.0151	0.0117
ConcatFea	Mean	0.5572	0.5191	0.3478	0.4901	0.5253	0.4612
	Sta Dev	0.0058	0.0056	0.0103	0.0075	0.0104	0.0062
Co-reg(P)	Mean	0.5401	0.4656	0.2985	0.4491	0.4881	0.4177
	Sta Dev	0.0093	0.0064	0.0052	0.0033	0.0069	0.0031
Co-reg(C)	Mean	0.5588	0.5029	0.3431	0.4842	0.5257	0.4513
	Sta Dev	0.0110	0.0072	0.0079	0.0065	0.0107	0.0113
Co-trained	Mean	0.6034	0.5707	0.4360	0.5564	0.6070	0.5175
	Sta Dev	0.0084	0.0126	0.0121	0.0102	0.0103	0.0121
Multi-NMF	Mean	0.4858	0.4854	0.2355	0.4200	0.4048	0.4366
	Sta Dev	0.0139	0.0174	0.0193	0.0124	0.0170	0.0083
DMSNMF	Mean	0.7101	**0.6271**	0.5388	0.6452	0.6500	0.6405
	Sta Dev	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
SMDMF	Mean	**0.7515**	0.6245	**0.5734**	**0.6711**	**0.6810**	**0.6614**
	Sta Dev	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

objective value and NMI. To greatly reflects the dimensional reduction effect of our model, we select Yale dataset with high dimensional feature as experimental one.

Parameter analysis. The parameter k of the k-NN algorithm is fixed as 5. As a result, parameters in SMDMF only include the layer settings, i.e. the layer number m and the dimensions of single layer p_i, introduced by the deep NMF process. Lots of pervious work has discovered the significance of the last layer's size, p_m, which directly determines the quality of the consensus matrix (Zhao et al., 2017; Xu et al., 2018). Therefore, we change p_m in settings with different layer numbers, while fixing the other dimensions by trial and error. Fig.2 explores the ACC results of SMDMF on Yale dataset varying p_m in four different layer settings, i.e.{[100 p_m], [200 100 p_m], [300 200 100 p_m]}. As we can see, [100 p_m] is the most competitive layer setting among them. Although [200 100 p_m] and [300 200 100 p_m] are deeper than the best one, they still do not obtain a better clustering performance, which maybe results from the over-fitting problem or the dimensions of the remaining layers. In experiments on Yale dataset, we choose [100 50] as the default layer parameters.

Convergence analysis. We also analyze the convergence property of SMDMF by means of the objective and ACC values. Consistent with the foregoing, the relevant layer parameters are set as [100 50]. Fig.3 shows the objective value in Eq.(8) and ACC result against iteration times for SMDMF on Yale dataset. We observe the ACC value rises rapidly in the initial iterations, and then increases slowly until achieving the stable value, finally oscillates nearby it. At the same time, the objective value of our model decreases drastically

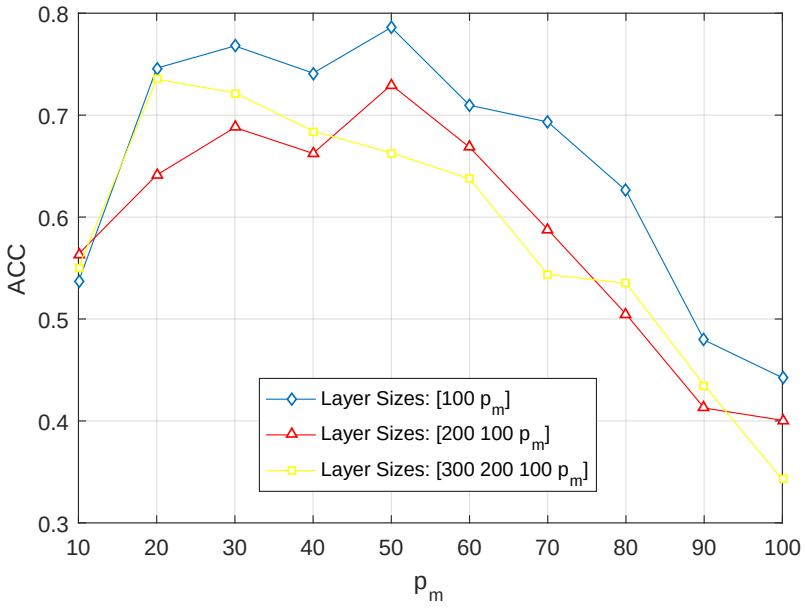

Figure 2: The ACC results curves of SMDMF with four different layer settings on Yale dataset.

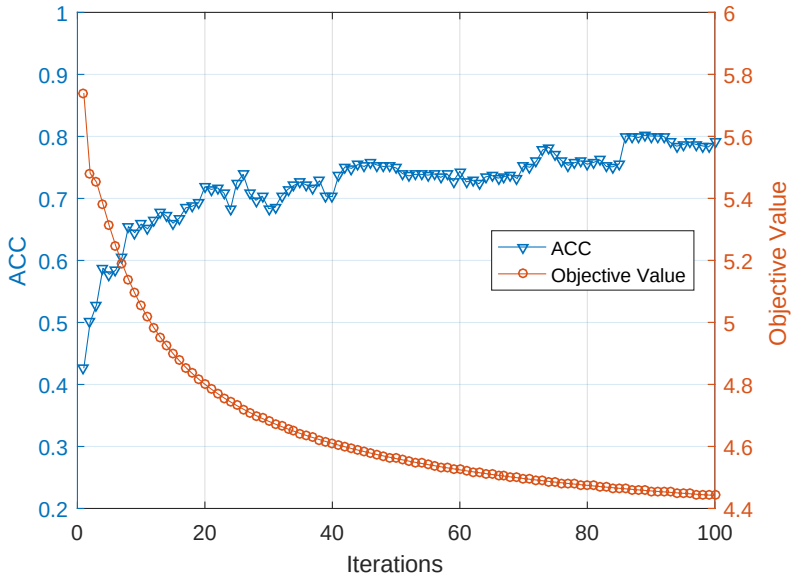

Figure 3: The Objective Value and ACC performance of SMDMF with respect to the iteration times on Yale dataset.

Table 5: Clustering performance on C101-7 dataset (Mean and Sta Dev,%).

Method	Values	ACC	NMI	AR	F-core	Precision	Recall
BestSV	Mean	0.4280	0.4222	0.2735	0.4353	0.7524	0.3063
	Sta Dev	0.0092	0.0099	0.0078	0.0069	0.0085	0.0055
ConcatFea	Mean	0.4120	0.4874	0.2973	0.4506	0.7961	0.3143
	Sta Dev	0.0057	0.0041	0.0042	0.0034	0.0066	0.0026
Co-reg(P)	Mean	0.4282	0.4906	0.3092	0.4610	0.8076	0.3227
	Sta Dev	0.0185	0.0046	0.0081	0.0080	0.0056	0.0070
Co-reg(C)	Mean	0.4402	0.4861	0.3153	0.4683	0.8062	0.3303
	Sta Dev	0.0069	0.0046	0.0071	0.0059	0.0087	0.0045
Co-trained	Mean	0.4487	0.5295	0.3376	0.4838	0.8427	0.3401
	Sta Dev	0.0110	0.0056	0.0051	0.0048	0.0049	0.0039
Multi-NMF	Mean	0.4864	0.4988	0.3418	0.4976	0.8099	0.3592
	Sta Dev	0.0206	0.0179	0.0173	0.0153	0.0147	0.0130
DMSNMF	Mean	0.5611	0.5436	0.4102	0.5756	0.7913	0.4523
	Sta Dev	0.0000	0.0022	0.0022	0.0018	0.0015	0.0017
SMDMF	Mean	**0.6592**	**0.5876**	**0.5037**	**0.6419**	**0.8886**	**0.5025**
	Sta Dev	0.0018	0.0006	0.0018	0.0016	0.0004	0.0018

and reaches the convergence value generally in 60 iterations, which is consistent with the conclusions of our proof.

6. Conclusion

In this paper, we propose a Self-Weighted Multi-view Clustering with Deep Matrix Factorization (SMDMF) method. Benefitting from the hierarchical framework, it can exclude adverse interference from other factors and capture the semantic structure of the multi-view data, which is instrumental in obtaining the consensus representation. For balancing the parameter quantities and performance, we utilize a self-weighted strategy to construct our parameter-free formulation, which can automatically assign a proper weight to each view and fully consider the complementarity among views. An iterative optimization algorithm is developed to deal with the SMDMF objective. Experiments on four datasets and performance analysis revealed the effectiveness of SMDMF compared with other seven algorithms.

7. Acknowledgments

Research reported in this publication was supported by the National Natural Science Foundation of China(61602081) and Natural Foundation of Liaoning Province(201602180).

References

Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

Eric Bruno and Stéphane Marchand-Maillet. Multiview clustering: a late fusion approach using latent models. In *International Acm Sigir Conference on Research & Development in Information Retrieval*, 2009.

Xiao Cai, Feiping Nie, and Heng Huang. Multi-view k-means clustering on big data. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2598–2604. AAAI Press, 2013.

Xiaochun Cao, Changqing Zhang, Huazhu Fu, Liu Si, and Zhang Hua. Diversity-induced multi-view subspace clustering. In *Computer Vision & Pattern Recognition*, 2015.

Ding Chris, Li Tao, and Michael I Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 32(1):45–55, 2010.

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR (1)*, pages 886–893. IEEE Computer Society, 2005.

Ke Lin Du and M. N. S. Swamy. Nonnegative matrix factorization. In *Twenty-fourth Aaai Conference on Artificial Intelligence*, 2010.

Hongchang Gao, Feiping Nie, Xuelong Li, and Heng Huang. Multi-view subspace clustering. In *IEEE International Conference on Computer Vision*, 2016.

Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 23(6):643–660, 2002.

Dongyan Guo, Jian Zhang, Xinwang Liu, Ying Cui, and Chunxia Zhao. Multiple kernel learning based multi-view spectral clustering. In *International Conference on Pattern Recognition*, 2014.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

Hsin-Chien Huang, Yung-Yu Chuang, and Chu-Song Chen. Affinity aggregation for spectral clustering. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEEConference on*, pages 773–780.

Zhao Kang, Xiao Lu, Jinfeng Yi, and Zenglin Xu. Self-weighted multiple kernel learning for graph-based clustering and semi-supervised classification. In *IJCAI*, pages 2312–2318. ijcai.org, 2018.

Abhishek Kumar and Hal Daumé Iii. A co-training approach for multi-view spectral clustering. In *International Conference on International Conference on Machine Learning*, 2011a.

Abhishek Kumar, Piyush Rai, and Hal Daumé. Co-regularized multi-view spectral clustering. In *International Conference on Neural Information Processing Systems*, 2011b.

Chengcai Leng, Hai Zhang, and Guorong Cai. A novel data clustering method based on smooth non-negative matrix factorization. In *International Conference on Smart Multimedia*, 2018.

Jialu Liu, Chi Wang, Jing Gao, and Jiawei Han. Multi-view clustering via joint nonnegative matrix factorization. In *Proc. of 2013 SIAM Data Mining Conf.*, 2013.

Yang Liu, Quanxue Gao, Zhaohua Yang, and Shujian Wang. Learning with adaptive neighbors for image clustering. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 2483–2489. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, 2001.

Feiping Nie, Guohao Cai, and Xuelong Li. Multi-view clustering and semi-supervised classification with adaptive neighbours. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017a.

Feiping Nie, Jing Li, and Xuelong Li. Self-weighted multiview clustering with multiple graphs. In *IJCAI*, pages 2564–2570. ijcai.org, 2017b.

Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Gray scale and rotation invariant texture classification with local binary patterns. In *European Conference on Computer Vision*, 2000.

G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. W. Schuller. A deep semi-nmf model for learning hidden representations. In *International Conference on Machine Learning*, 2014.

Cai Xu, Ziyu Guan, Wei Zhao, Yunfei Niu, Quan Wang, and Zhiheng Wang. Deep multiview concept learning. In *IJCAI*, pages 2898–2904. ijcai.org, 2018.

J. Xu, J. Han, F. Nie, and X. Li. Re-weighted discriminatively embedded k-means for multi-view clustering. *IEEE Transactions on Image Processing*, 26(6):3016–3027, 2017.

Hong Yu, Tiantian Zhang, Yahong Lian, and Yu Cai. Co-regularized multi-view subspace clustering. In *Asian Conference on Machine Learning*, pages 17–32, 2018.

Handong Zhao, Zhengming Ding, and Yun Fu. Multi-view clustering via deep matrix factorization. In *AAAI*, pages 2921–2927. AAAI Press, 2017.