



Research Article

Adaptive-Weighted Multiview Deep Basis Matrix Factorization for Multimedia Data Analysis

Shicheng Li¹, Qinghua Liu¹, Jiangyan Dai², Wenle Wang¹, Xiaolin Gui¹,¹ and Yugen Yi¹

¹School of Software, Jiangxi Normal University, Nanchang 330022, China

²School of Computer Engineering, Weifang University, Weifang 261061, China

Correspondence should be addressed to Qinghua Liu; 004799@jxnu.edu.cn and Yugen Yi; yiyg510@jxnu.edu.cn

Received 23 February 2021; Accepted 22 May 2021; Published 7 June 2021

Academic Editor: Wei Quan

Copyright © 2021 Shicheng Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Feature representation learning is a key issue in artificial intelligence research. Multiview multimedia data can provide rich information, which makes feature representation become one of the current research hotspots in data analysis. Recently, a large number of multiview data feature representation methods have been proposed, among which matrix factorization shows the excellent performance. Therefore, we propose an adaptive-weighted multiview deep basis matrix factorization (AMDBMF) method that integrates matrix factorization, deep learning, and view fusion together. Specifically, we first perform deep basis matrix factorization on data of each view. Then, all views are integrated to complete the procedure of multiview feature learning. Finally, we propose an adaptive weighting strategy to fuse the low-dimensional features of each view so that a unified feature representation can be obtained for multiview multimedia data. We also design an iterative update algorithm to optimize the objective function and justify the convergence of the optimization algorithm through numerical experiments. We conducted clustering experiments on five multiview multimedia datasets and compare the proposed method with several excellent current methods. The experimental results demonstrate that the clustering performance of the proposed method is better than those of the other comparison methods.

1. Introduction

With the rapid development of computer technology, the collected multimedia data from many research fields, such as computer vision, image processing, and natural language processing, always have features with high dimension and complex structures. These high-dimensional data can not only provide abundant information but also bring some problems such as the “curse of dimensionality” [1, 2]. Therefore, how to effectively deal with high-dimensional data has become a widespread concern [3]. Dimensionality reduction is an efficient way to solve this issue, which can map the original data to a low-dimensional space and obtain a low-dimensional representation derived from the hidden information in the original data [4].

In recent years, many dimensionality reduction methods have been proposed for multimedia data [5]. The matrix factorization method has become one of the research hotspots

owing to its simple theoretical basis and easy implementation. Principal component analysis (PCA) [6], independent components analysis (ICA) [7], vector quantization (VQ) [8], etc. are well-known matrix factorization methods that can obtain a low-rank approximation matrix by decomposing a high-dimensional data matrix, and they can effectively extract a low-dimensional representation from high-dimensional data. However, these methods do not utilize any constraints on the matrix elements during the process of matrix decomposition. It means that the results allow negative elements, which give rise to the loss of physical meaning in low-dimensional representations. To solve this problem, Lee et al. added nonnegative constraints into matrix decomposition and proposed a nonnegative matrix factorization (NMF) [9] method. The low-dimensional feature representations obtained by NMF method are part-based so that they have strong interpretability. Consequently, NMF has attracted the wide attention of researchers. There are a large

number of improved algorithms based on NMF have been emerged, which have achieved great success in computer vision, natural language processing, speech recognition, DNA sequence analysis, and other areas [10–13].

NMF decomposes the original nonnegative data matrix into the product of a nonnegative basis matrix and a nonnegative coefficient matrix (also called low-dimensional feature matrix). The original data can be expressed as a linear combination of basis matrices, and the combination coefficients can form the coefficient matrix. Since NMF uses nonnegative constraints, it reflects the intuitive notion of combining parts to form a whole and has better interpretability than other methods. The obtained experimental results indicate that NMF has achieved good performance on image and document clustering tasks. Nevertheless, the traditional NMF method only considers the nonnegativity constraints of the elements, which may result in the obtained basis matrix having poor sparseness and independence. To solve the above problems, researchers have imposed additional constraints on the basis matrix or the coefficient matrix and proposed a series of improved methods. For instance, Hoyer [14] designed a sparsity measurement criterion and proposed an NMF variant with sparsity constraints (NMF-SC). Moreover, to enhance the independence of the obtained basis matrices and low-dimensional representation, Choi [15] proposed orthogonal nonnegative matrix factorization (ONMF), which imposed orthogonal constraints on the basis matrix and the coefficient matrix. However, the above methods have nonnegative limitations on the original data, thereby limiting the applicability of these NMF-based algorithms. Therefore, Ding et al. [16] proposed a semi-nonnegative matrix factorization (SNMF). Different from traditional NMF, SNMF relaxed the limitations on the original data and coefficient matrix and only imposed a nonnegative constraint on the basis matrix. The methods mentioned above have better capabilities than their predecessors for feature extraction and achieved better results in real-world tasks, but they only extracted shallow features [17].

In recent years, deep learning has exhibited outstanding performance in feature representation tasks [18–20]. Therefore, many researchers have introduced deep learning into matrix factorization and proposed a large number of deep feature representation methods [21–27]. Ahn et al. [21] proposed multilayer nonnegative matrix factorization (MNMF). Different from traditional NMF-based approaches, MNMF decomposed the coefficient matrix several times to obtain an underlying part-based representation that can extract deep hierarchical features from the original data. In addition, to expand the application scope, Trigeorgis et al. [22] integrated deep factorization and semi-NMF to propose a deep semi-nonnegative matrix factorization (deep semi-NMF) method. However, both MNMF and deep semi-NMF only considered the deep decomposition of the coefficient matrix for the training data. For the new test data, the basis matrix was used to obtain the deep low-dimensional representation. Therefore, the basis matrix directly affected the results of the deep low-dimensional representation. To obtain a more accurate deep low-dimensional representation of the original

data matrix, Zhao et al. [23] applied deep factorization to the basis matrix and proposed a deep NMF method based on basis image learning.

With the rapid development of the Internet and data collection technology, a large amount of multiview multimedia data can be easily acquired [28–30]. For example, an object can be shot from different views. An image can be described with different types of features such as color, texture, and shape. These multiview multimedia data can provide different information for each view, but they also contain potential correlations among these different views. Furthermore, they contain more information than single-view data. It is possible to simply integrate multiview data into single-view data, which ignores the differences and potential correlations between the various views of the data [28–30].

Consequently, extensive multiview data dimensionality reduction methods have been proposed [31–33]. Liu et al. [34] proposed a multiview NMF (multi-NMF) method which established the relationship between different perspectives by learning the common coefficient matrix among different views. Subsequently, Chang et al. [35] introduced a new regularization term into the multi-NMF and used it for clothing image clustering. Inspired by ONMF, Liang et al. [36] proposed NMF with coorthogonal constraints (NMFCC) for multiview multimedia data clustering. Additionally, to consider the correlations between multiple views, Zhan et al. [37] jointly optimized the graph matrix and concept factorization process and proposed an adaptive structure concept factorization (ASCF) method for multiview clustering. Although the above methods can handle multiview multimedia data well, they still belong to the class of feature representation method based on shallow factorization [38, 39]. The underlying deep features in the multiview data are still not available. Therefore, Zhao et al. [40] maximized the mutual information between various views, which forced the non-negative representation of the last layer in each view to be as similar as possible. Then, the deep semi-NMF method was applied to multiview multimedia data clustering. Different from the existing studies, to adaptively provide feature weights for different perspectives in the multiperspective deep feature representation procedure, Huang et al. introduced an adaptive-weighted framework into the multiview deep semi-NMF and proposed an adaptive-weighted multiview clustering method based on deep matrix factorization [41]. Unlike the literature [40], it can adaptively assign weights to different views in a multiview deep feature representation. However, these methods still consider only the deep decomposition of the coefficient matrix. Therefore, an adaptive-weighted multiview deep basis matrix factorization (AMDBMF) is proposed for multimedia data clustering in this paper. Different from the above methods, AMDBMF first decomposes the basis matrix using a deep way on the data of each view simultaneously and then integrates the low-dimensional features of all view through the adaptive weighting mechanism to extract more accurate multiperspective deep low-dimensional representations. The flowchart of the proposed AMDBMF approach is shown in Figure 1. At last, we perform extensive experiments on five publicly

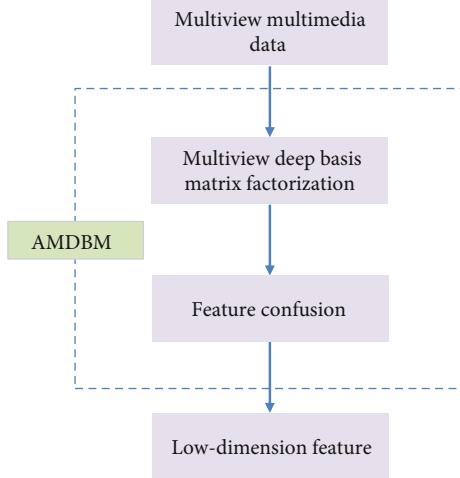


FIGURE 1: The flowchart of the proposed AMDBMF approach.

available multiview multimedia datasets. These experimental results show that the proposed AMDBMF approach outperforms the existing related approaches.

The remainder of this paper is organized as follows. “Related Works” describes the related algorithms including NMF and deep semi-NMF briefly. “Adaptive-Weighted Multiview Deep Basis Matrix Factorization” introduces the adaptive-weighted multiview deep basis matrix factorization (AMDBMF) algorithm in detail. The experimental results and analysis are discussed in “Experiments and Analysis.” Finally, the conclusions are given in “Conclusions and Future Work.”

2. Related Works

2.1. Nonnegative Matrix Factorization. Suppose that the given multimedia data can be represented as $X = [x_1, \dots, x_N] \in R_+^{D \times N}$, where D is the dimensionality of the data and N is the number of samples. Each sample can be represented as a D -dimensional feature vector $x_j (1 \leq j \leq N)$. NMF is aimed at finding two low-ranking nonnegative matrices $W = [w_1, \dots, w_d] \in R_+^{D \times d}$ and $H = [h_1, \dots, h_N] \in R_+^{d \times N} (k < N \text{ and } d < D)$ that fulfill $X \approx WH$. After obtaining W and H , the original data can be expressed as $x_j \approx \sum_{i=1}^d w_i h_{ij}$, that is, each sample can be expressed as a linear combination of the basis matrix $W = [w_1, \dots, w_d]$, and the coefficient vector is h_j . Therefore, the matrices W and H are called the basis matrix and coefficient matrix, respectively. The objective function of NMF is defined as follows:

$$\begin{aligned} \min_{W,H} & \|X - WH\|_F^2 \\ \text{s.t.} & W \geq 0, H \geq 0, \end{aligned} \quad (1)$$

where $\|\cdot\|_F$ is the Frobenius norm operation.

According to the Karush-Kuhn-Tucker (KKT) condition, the update formulas for variables W and H are as follows:

$$\begin{aligned} W_{il} &= W_{il} \frac{(XH^T)_{il}}{(WHH^T)_{il}}, \\ H_{lj} &= H_{lj} \frac{(W^TX)_{lj}}{(W^TWH)_{lj}}. \end{aligned} \quad (2)$$

2.2. Deep Nonnegative Matrix Factorization. The traditional NMF method can remove redundant information and reveal the hidden semantic features of multimedia data, but it cannot learn an effective feature representation for the data. For example, a facial image contains various changes such as posture, lighting, and expression changes. Therefore, Trigeorgis et al. [22] pointed out that the coefficient matrix, as a low-dimensional representation of high-dimensional data, should be able to continue to be decomposed so that more abstract low-dimensional features can be obtained. Thus, these processes of deep factorization are defined as

$$\begin{aligned} X &\approx W_1 H_1, \\ H_1 &\approx W_2 H_2, \\ &\dots \\ H_{l-1} &\approx W_l H_l, \end{aligned} \quad (3)$$

where W_i and H_i represent the factorization results of the i -th layer. It can be seen from Eq. (3) that deep NMF performs a procedure of matrix factorization at each layer and uses the decomposed coefficient matrix as the input data of the next layer to continue decomposing. Consequently, the process of deep matrix factorization performed on the data is expressed as

$$X \approx W_1 W_2 \cdots W_l H_l. \quad (4)$$

The objective function of deep NMF is defined as follows:

$$\min \|X - W_1 W_2 \cdots W_l H_l\|_F^2. \quad (5)$$

Similar to that of NMF, the update formula can be defined as follows:

$$\begin{aligned} W_i &= (\Psi^T \Psi)^{-1} \Psi^T X \tilde{H}_i^T \left(\tilde{H}_i \tilde{H}_i^T \right)^{-1}, \\ H_i &= H_i \odot \sqrt{\frac{[\Psi^T X]^{\text{pos}} + [\Psi^T \Psi]^{\text{neg}} H_i}{[\Psi^T X]^{\text{neg}} + [\Psi^T \Psi]^{\text{pos}} H_i}}, \end{aligned} \quad (6)$$

where $\Psi = W_1 \cdots W_{i-1}, \tilde{H}_i$ denotes the reconstruction of the i -th layer’s feature matrix, and the symbol \odot represents the dot product of matrices. $[A]^{\text{pos}} = |A| + A/2$ represents a matrix operation that restrains all the negative elements to zeros and keeps the positive elements unchanged. On the contrary, $[A]^{\text{neg}} = |A| - A/2$ turns the positive elements to be zeros while the negative elements are to be nonnegative.

3. Adaptive-Weighted Multiview Deep Basis Matrix Factorization

First, an adaptive-weighted multiview deep basis matrix factorization (AMDBMF) method is proposed, which incorporates the nonnegative matrix factorization and deep learning into a unified framework. Next, an optimization algorithm with an iterative updating rule is designed to solve the objective function of AMDBMF. Then, an adaptive-weighted fusion mechanism is provided. Finally, we provide the complexity analysis of the proposed algorithm.

Suppose that $X = [X_1, \dots, X_N]$ denotes a multimedia data set which contains N samples. Each sample x_i ($i = 1, \dots, N$) is described by M views. Thus, the m -th view's features for this sample can be represented as x_i^m ($m = 1, \dots, M$). The features of all samples in this view can be represented as $X^m = [x_1^m, \dots, x_N^m]$.

3.1. Objective Function. First, matrix factorization is performed on the features in each view of the multimedia data, and the objective function can be defined as

$$\begin{aligned} & \min_{W^m, H^m} \|X^m - W^m H^m\|_F^2 \\ & \text{s.t. } W^m \geq 0, H^m \geq 0, \end{aligned} \quad (7)$$

where W^m and H^m denote the basis matrix and the coefficient matrix of the m -th view's features, respectively.

Then, the deep factorization is performed on W^m . The process is defined as follows:

$$\begin{aligned} X^m &\approx W_1^m H_1^m, \\ W_1^m &\approx W_2^m H_2^m, \\ &\dots \\ W_{l-2}^m &\approx W_{l-1}^m H_{l-1}^m, \\ W_{l-1}^m &\approx W_l^m H_l^m, \end{aligned} \quad (8)$$

where $W_1^m, W_2^m, \dots, W_{l-1}^m, W_l^m$ and $H_1^m, H_2^m, \dots, H_{l-1}^m, H_l^m$ denote the basis matrices and coefficient matrices for each layer in the m -th view, respectively.

By combining it with Eq. (8), Eq. (7) is further rewritten as follows:

$$\begin{aligned} & \min_{W_l^m, H_l^m} \|X^m - W_l^m H_l^m H_{l-1}^m \cdots H_2^m H_1^m\|_F^2 \\ & \text{s.t. } W_l^m \geq 0, H_l^m \geq 0. \end{aligned} \quad (9)$$

Finally, to fuse the data from multiple perspectives, the final objective function is defined as

$$\begin{aligned} & \min_{W_l^m, H_l^m} \sum_{m=1}^M \|X^m - W_l^m H_l^m H_{l-1}^m \cdots H_2^m H_1^m\|_F^2 \\ & \text{s.t. } W_l^m \geq 0, H_l^m \geq 0. \end{aligned} \quad (10)$$

3.2. Optimization. From Eq. (10), we can find that the objective function is nonconvex for all variables, but it is convex for each of them on their own. Therefore, we design an iterative update algorithm to find the local optimal solution of the objective function. To solve this problem, one variable is updated while the other variables are fixed. The detailed updating rules are described as follows.

The optimal objective function for variables W_l^m and H_l^m can be defined as

$$\begin{aligned} & \min_{W_l^m, H_l^m} \|X^m - W_l^m H_l^m H_{l-1}^m \cdots H_2^m H_1^m\|_F^2 \\ & \text{s.t. } W_l^m \geq 0, H_l^m \geq 0. \end{aligned} \quad (11)$$

Let $\Lambda_l^m = H_{l-1}^m \cdots H_2^m H_1^m$, and then Eq. (11) can be simplified as

$$\begin{aligned} & \min_{W_l^m, H_l^m} \|X^m - W_l^m H_l^m \Lambda_{l-1}^m\|_F^2 \\ & = \text{tr}\left((X^m - W_l^m H_l^m \Lambda_{l-1}^m)^T (X^m - W_l^m H_l^m \Lambda_{l-1}^m)\right) \\ & = \text{tr}\left(-2X^{mT} W_l^m H_l^m \Lambda_{l-1}^m + \Lambda_{l-1}^m {}^T H_l^m {}^T W_l^m {}^T W_l^m H_l^m \Lambda_{l-1}^m\right) \\ & \text{s.t. } W_l^m \geq 0, H_l^m \geq 0. \end{aligned} \quad (12)$$

The Lagrangian function of Eq. (12) is expressed as

$$\varphi(W_l^m, H_l^m) = \|X^m - W_l^m H_l^m \Lambda_{l-1}^m\|_F^2 + \text{tr}(\gamma_l^m H_l^m) + \text{tr}(\eta_l^m W_l^m), \quad (13)$$

where γ_l^m and η_l^m are Lagrange multipliers.

Taking the partial derivatives of Eq. (13) with respect to W_l^m and H_l^m , and setting these derivatives to zero, we have

$$\begin{aligned} \frac{\partial \varphi(W_l^m, H_l^m)}{\partial W_l^m} &= \frac{\partial \|X^m - W_l^m H_l^m \Lambda_{l-1}^m\|_F^2}{\partial W_l^m} + \frac{\partial \text{tr}(\eta_l^m W_l^m)}{\partial W_l^m} = 0, \\ \frac{\partial \varphi(W_l^m, H_l^m)}{\partial H_l^m} &= \frac{\partial \|X^m - W_l^m H_l^m \Lambda_{l-1}^m\|_F^2}{\partial H_l^m} + \frac{\partial \text{tr}(\gamma_l^m H_l^m)}{\partial H_l^m} = 0. \end{aligned} \quad (14)$$

According to the KKT condition $(\gamma_l^m H_l^m)_{ij} = 0$ and $(\eta_l^m W_l^m)_{ij} = 0$ [41], the update rules of variables W_l^m and H_l^m are as follows:

$$\begin{aligned} W_l^m &= W_l^m \odot \frac{X \Lambda_{l-1}^m {}^T H_l^m {}^T}{W_l^m H_l^m \Lambda_{l-1}^m \Lambda_{l-1}^m {}^T H_l^m {}^T}, \\ H_l^m &= H_l^m \odot \frac{W_l^m {}^T X^m \Lambda_{l-1}^m {}^T}{W_l^m {}^T W_l^m H_l^m \Lambda_{l-1}^m \Lambda_{l-1}^m {}^T}, \end{aligned} \quad (15)$$

where the symbol \odot represents the dot product of matrices.

Finally, the algorithmic steps of the proposed method are given in Algorithm 1. To make it easier to understand,

```

Input:
Multiview nonnegative matrix  $X = [X^1, X^2, \dots, X^m, \dots, X^M]$ 
The hidden features of each layer  $\{d_1, d_2, \dots, d_l, \dots, d_L\}$ 
Initialize:
The basis matrix for each view  $\{W_l^m\}, l = 1, 2, \dots, L, m = 1, 2, \dots, M$ 
The coefficient matrix for each view  $\{H_l^m\}, l = 1, 2, \dots, L, m = 1, 2, \dots, M$ 
Pretrain:
for  $m = 1 : M$  do
  for  $l = 1 : L$  do
     $W_l^m, H_l^m = NMF(W_{l-1}^m, k_l)$ 
  end for
end for
Update:
Repeat
  for  $m = 1 : M$  do
    for  $l = 1 : L$  do
      if  $l == 1$ 
         $A_l^m = I$ 
      else
         $A_l^m = H_{l-1}^m \cdots H_2^m H_1^m$ 
      end
      Update  $W_l^m = W_l^m \odot X A_{l-1}^{m-1} H_l^m / W_l^m H_l^m A_{l-1}^m A_{l-1}^{m-1} H_l^m$ 
      Update  $H_l^m = H_l^m \odot W_l^m X^m A_{l-1}^{m-1} / W_l^m H_l^m A_{l-1}^m A_{l-1}^{m-1}$ 
    end for
  end for
Until Reach the convergence condition or the maximum number of iterations
Output:
The basis matrix for each view  $\{W_l^m\}, l = 1, 2, \dots, L, m = 1, 2, \dots, M$ 
The coefficient matrix for each view  $\{H_l^m\}, l = 1, 2, \dots, L, m = 1, 2, \dots, M$ 

```

ALGORITHM 1: The optimization algorithm.

Figure 2 depicts the block diagram of the proposed optimization algorithm.

3.3. Feature Confusion. After obtaining the basis matrix and coefficient matrix of each layer for each view through the optimization algorithm, an adaptive-weighted fusion mechanism is adopted to obtain a low-dimensional representation of the multiview data, and the weight calculation is

$$\alpha^m = \frac{1}{2\sqrt{\|X^m - W_l^m H_l^m H_{l-1}^m \cdots H_2^m H_1^m\|_F^2 + \varepsilon}}, \quad (16)$$

where ε is a small constant.

Then, α^m is normalized by Eq. (17)

$$\alpha^m = \frac{\alpha^m}{\sum_{m=1}^M \alpha^m}. \quad (17)$$

Finally, since the low-dimensional representation of each view is expressed as $H^m = H_l^m A_{l-1}^m$, the fusion of the low-dimensional features derived from the multiview data can be expressed as

$$H^* = \sum_{m=1}^M \alpha^m H^m. \quad (18)$$

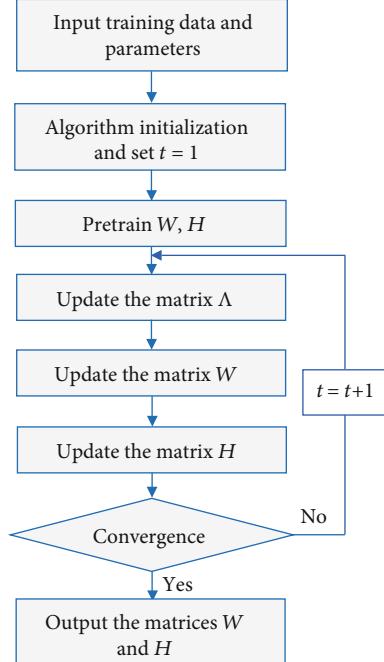
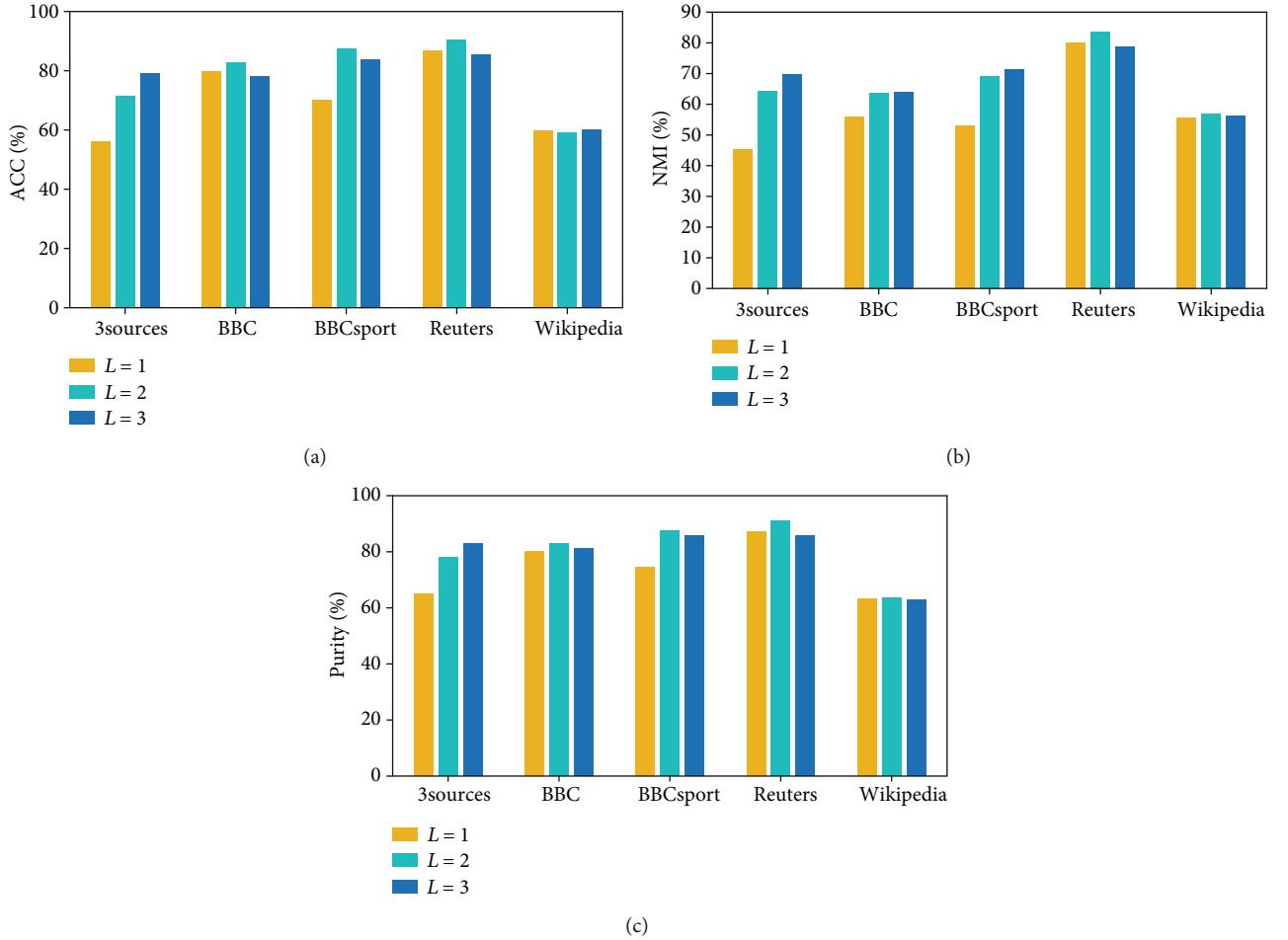


FIGURE 2: The block diagram of the proposed optimization algorithm.

TABLE 1: Statistical information about the different datasets.

Datasets	Size (N)	Classes (C)	Views (V)	Feature dimension of each view				
				d_1	d_2	d_3	d_4	d_5
3sources	169	6	3	3560	3068	3631	—	—
BBC	685	5	4	4659	4633	4465	4684	—
BBC Sport	544	5	2	3183	3203	—	—	—
Reuters	1200	6	5	12000	12000	12000	12000	12000
Wikipedia	693	10	2	128	10	—	—	—

FIGURE 3: The influence of factorization layer L on the clustering results, (a) ACC, (b) NMI, and (c) purity.

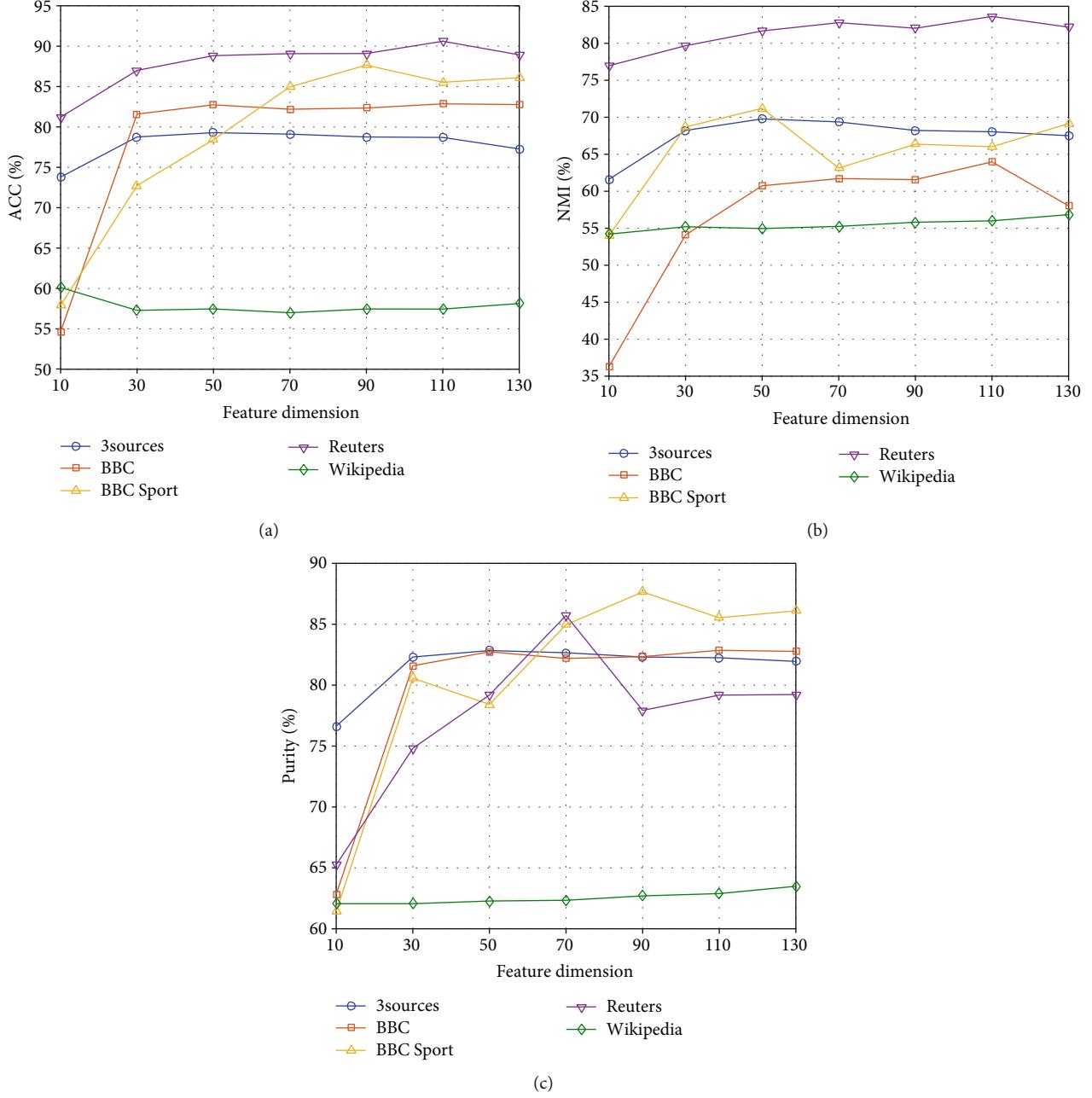
3.4. Complexity Analysis. Clearly, the proposed algorithm can be divided into two stages: pretraining and fine-tuning. For convenience, suppose that the number of iterations is T , M is the number of data views, and L is the number of layers. The number of features for all views is D , and the number of low-dimensional representations for each layer is K . In the pretraining process, the complexity of a single view is $O(TDNK)$. Therefore, the complexity of the whole pretraining process is $O(TMLDNK)$. For the fine-tuning part, the main computational complexity is derived from updating A_{l-1}^m , W_l^m , and H_l^m , which requires $O(TM(L-1)NK^2)$, $O(TMLDNK)$, and $O(TMLDNK)$ complexity, respectively.

Since $D > K$, the total computational complexity of the proposed algorithm is $O(TMLDNK)$.

4. Experiments and Analysis

4.1. Datasets. Five commonly used multiview multimedia datasets from the Internet are used in the experiments to verify the effectiveness of the proposed method.

4.1.1. 3sources. This dataset includes a collection of 416 news events and 948 related news reports from February to April 2009 from three well-known news media outlets, including

FIGURE 4: The influences of feature dimension D on clustering results, (a) ACC, (b) NMI, and (c) purity.

BBC, Reuters, and Guardian. In the experiments, 169 news items reported by all three news media outlets are used. These news events include six categories: business, entertainment, health, politics, sports, and technology (<http://mlg.ucd.ie/datasets/3sources.html>).

4.1.2. BBC [42]. This dataset contains 685 news articles collected from the BBC News Network between 2004 and 2005. Each article is divided into four parts, and the data consist of five kinds of news: business, entertainment, politics, sport, and technology (<http://mlg.ucd.ie/datasets/segment.html>).

TABLE 2: The optimal parameter groups in our proposed algorithm.

Datasets	ACC	NMI	Purity
3sources	{3, 50}	{3, 50}	{3, 50}
BBC	{2, 110}	{3, 110}	{2, 110}
BBC Sport	{2, 90}	{3, 50}	{2, 90}
Reuters	{3, 110}	{2, 110}	{2, 70}
Wikipedia	{3, 10}	{2, 130}	{2, 130}

TABLE 3: The ACCs (%) of the AMDBMF and single-view methods on different datasets.

Methods	3sources	BBC	BBC Sport	Reuters	Wikipedia
NMF	55.03	46.39	61.75	73.57	56.71
DBMF	76.98	63.84	80.24	67.67	57.11
AMDBMF	79.29	82.86	87.65	90.61	60.12

TABLE 4: The NMIs (%) of the AMDBMF and single-view methods on different datasets.

Methods	3sources	BBC	BBC Sport	Reuters	Wikipedia
NMF	51.90	18.74	40.45	66.84	51.23
DBMF	67.16	56.71	66.80	59.47	54.89
AMDBMF	69.78	63.97	71.21	86.63	56.84

TABLE 5: The purities (%) of the AMDBMF and single-view methods on different datasets.

Methods	3sources	BBC	BBC Sport	Reuters	Wikipedia
NMF	72.01	48.66	64.67	75.34	60.75
DBMF	81.72	73.74	83.62	69.98	59.60
AMDBMF	82.84	82.86	87.65	91.13	63.49

TABLE 6: The ACCs (%) of different multiview methods on different datasets.

Methods	3sources	BBC	BBC Sport	Reuters	Wikipedia
MVCF	69.94	63.18	66.82	63.42	54.04
DeepMVC	65.68	72.12	72.12	76.25	60.46
GMC	69.23	69.34	80.70	66.58	44.88
NMFCC	69.82	73.40	83.09	76.04	61.46
AMDBMF	79.29	82.86	87.65	90.61	60.12

4.1.3. *BBC Sport* [42]. This dataset includes 737 news articles from the BBC Sport network from 2004 to 2005. These news articles cover six fields, such as track, field, cricket, football, rugby, and tennis (<http://mlg.ucd.ie/datasets/segment.html>).

4.1.4. *Reuters* [43]. This is a dataset that includes 1200 English articles from six types of samples, and each article has been translated into French, German, Italian, and Spanish (<http://lig-membres.imag.fr/grimal/data.html>).

4.1.5. *Wikipedia* [43]. This dataset consists of specific Wikipedia material with 2669 articles in 29 categories (<http://www.svcl.ucsd.edu/projects/crossmodal/>). In the experiments, we select a subset of the 10 most popular categories containing a total of 693 samples. The detailed statistical information about the different datasets is given in Table 1.

4.2. *Metrics*. In the experiments, we select three commonly used clustering evaluation indicators [44]: accuracy (ACC), normalized mutual information (NMI), and purity to evaluate the performance of the proposed method.

TABLE 7: The NMIs (%) of different multiview methods on different datasets.

Methods	3sources	BBC	BBC Sport	Reuters	Wikipedia
MVCF	60.64	45.33	46.01	65.92	49.24
DeepMVC	56.50	52.58	52.58	73.76	55.45
GMC	54.80	48.52	72.26	70.43	36.14
NMFCC	61.91	53.95	67.38	73.92	56.00
AMDBMF	69.78	63.97	71.21	83.63	56.84

TABLE 8: The purities (%) of different multiview methods on different datasets.

Methods	3sources	BBC	BBC Sport	Reuters	Wikipedia
MVCF	78.52	64.64	68.36	68.84	57.17
DeepMVC	75.15	72.12	72.12	78.00	63.64
GMC	74.56	69.34	84.38	67.25	48.20
NMFCC	78.70	73.45	83.09	78.96	64.86
AMDBMF	82.84	82.86	87.65	91.13	63.49

Assuming that the clustering result of x_i is l_i and that the corresponding true label is t_i , then the clustering accuracy (ACC) [45] is defined as

$$\text{ACC} = \frac{\sum_{i=1}^N \delta(t_i, \text{map}(l_i))}{N}, \quad (19)$$

where the function $\delta(\cdot)$ is defined as follows:

$$\begin{cases} \delta(x, y) = 1 & x = y \\ \delta(x, y) = 0 & x \neq y. \end{cases} \quad (20)$$

The function $\text{map}(\cdot)$ maps the clustering result to the corresponding true label. The Kuhn-Munkres algorithm [46] is employed to find the best mapping result.

Assume that L and T are the clustering result and the true label set, respectively. The mutual information (MI) between them is defined as

$$\text{MI}(L, T) = \sum_{l_i \in L, t_i \in T} p(l_i, t_i) \cdot \log_2 \frac{p(l_i, t_i)}{p(l_i) \cdot p(t_i)}, \quad (21)$$

where $p(l_i)$ and $p(t_i)$ represent the probabilities that a sample is randomly selected from the dataset belonging to l_i and t_i , respectively. $p(l_i, t_i)$ represents the joint probability of a sample randomly being selected from the dataset belonging to l_i and t_i . Let $H(L)$ and $H(T)$ represent the entropies of L and T , respectively. Since the value range of mutual information is between 0 and $\max(H(L), H(T))$, the normalized mutual information (NMI) is defined as

$$\text{NMI} = \frac{\text{MI}(L, T)}{\max(H(L), H(T))}. \quad (22)$$

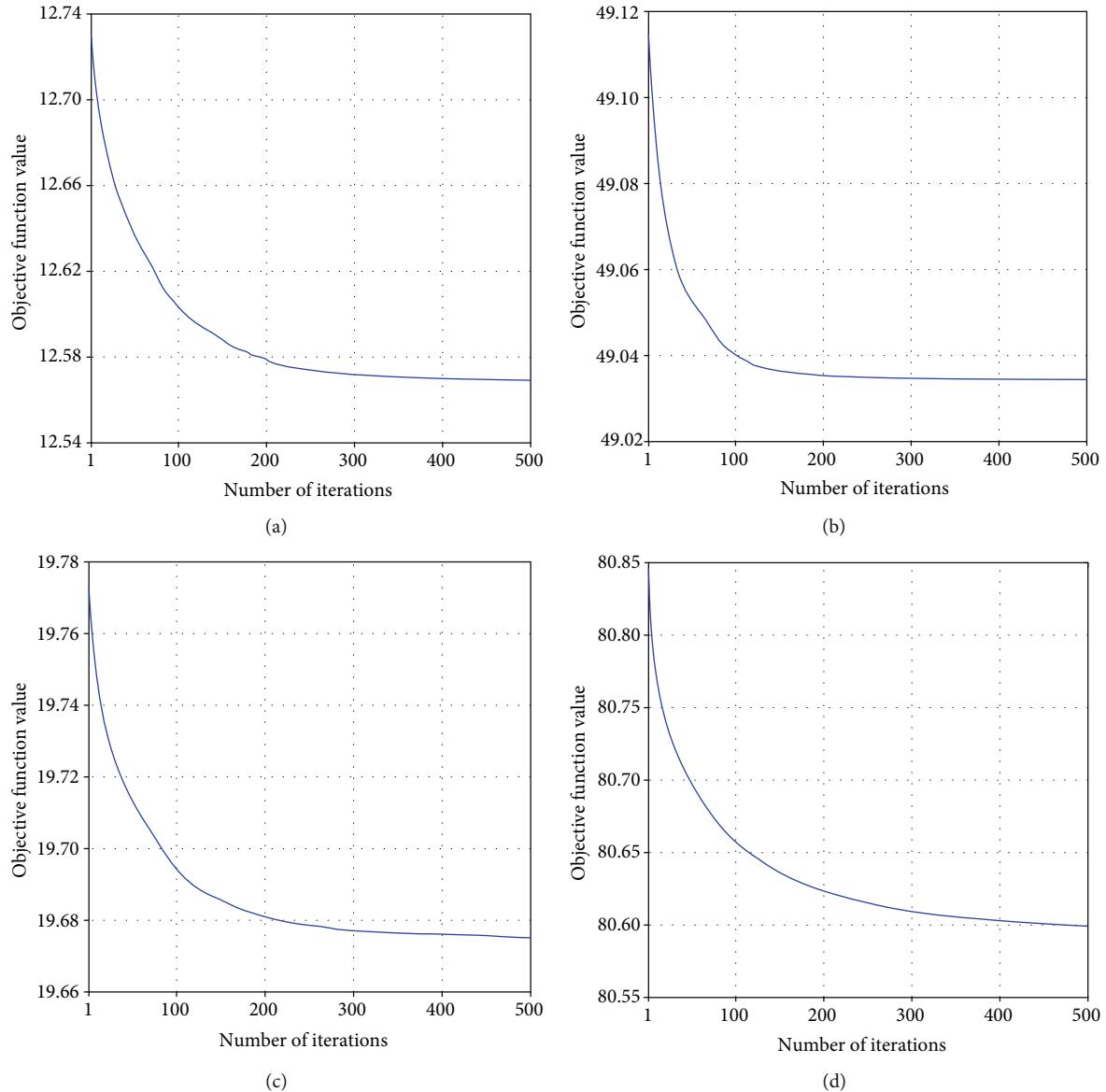


FIGURE 5: Continued.

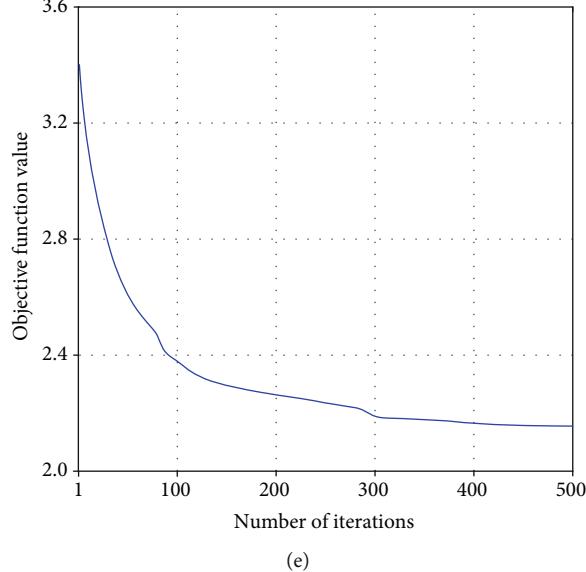


FIGURE 5: The convergence curves of the proposed method on five datasets: (a) 3sources, (b) BBC, (c) BBC Sport, (d) Reuters, and (e) Wikipedia.

Purity is a straightforward and transparent evaluation method that is defined as follows:

$$\text{purity} = \frac{1}{k} \sum_{i=1}^k \frac{|C_i^d|}{|C_i|}, \quad (23)$$

where k represents the number of clusters, $|C_i^d|$ is the number of elements in the most numerous category in cluster C_i , and $|C_i|$ is the number of elements in cluster C_i .

4.3. Experimental Results and Analysis. In the first experiment, to test the influences of the parameters on the proposed method, we set the number of factorization layers L and the feature dimension D of each layer to $\{1, 2, 3\}$ and $\{10, 30, 50, 70, 90, 110, 130\}$, respectively. Furthermore, we adopt a grid search to find the optimal parameter value. In the experiment, the low-dimensional features obtained by the proposed algorithm are clustered by the K -means algorithm. Since the initialization of the K -means algorithm has an impact on the clustering results, we repeat the random initialization process with 10 times and report the mean value. First, the optimal feature dimension of each layer is fixed, and the numbers of layers are changed. As shown in Figure 3, in most cases, when the number of layers is set to 1, the result of each measure is poorer than the rest. However, as the number of layers increases, the performance also increases. It shows that the deep factorization helps to improve the performance of the proposed method.

Then, the numbers of layers are fixed, and the dimension of the feature is changed. The result is shown in Figure 4. It can be seen that as the dimensionality increases, the clustering performance also improves in most cases. However, this trend is not always maintained, and the clustering perfor-

mance decreases or remains stable as the dimensionality increases once the performance reaches the optimal level. The details of the optimal parameter groups in our proposed algorithm are listed in Table 2.

The second experiment is conducted to verify that the fusion of multiview information is beneficial for improving the clustering performance of the proposed method. First, we perform traditional NMF and deep basis matrix factorization (DBMF) for the data of each view. Then, we obtain the low-dimensional features of the multiview data by fusing the features of different views with equal weight. Finally, the proposed AMDBMF method is compared with the above two methods. The comparison results are listed in Tables 3–5. According to the tables, the performance of the DBMF method is better than that of the traditional NMF method, which indicates that more abstract features can be obtained through the deep factorization. The performance of the proposed AMDBMF method is better than that of the DNBMF method, which verifies that the adaptive fusion of different views is beneficial for extracting more robust low-dimensional features from multiview data.

The third experiment compares the performance of the proposed AMDBMF method with those of some currently popular multiview algorithms, including MVCF [37], DeepMVC [41], GMC [47], and NMFCC [36]. MVCF utilized the correlation information between the views obtained by jointly optimizing the graph matrix of the data of each view. DeepMVC used a nonparameterized adaptive learning method to obtain the weights between views. NMFCC introduces orthogonal constraints into the basis matrix and coefficient matrix. The best results yielded by the different multiview learning methods on different datasets are shown in Tables 6–8. It can be seen that the performances of the proposed method are significantly better than those of the other comparison methods in most cases. Since these

methods use different mechanisms to fuse multiview data information, all the methods present different performances on different databases. Therefore, how to effectively integrate fusion mechanisms is still an open problem.

The final experiment verifies the convergence of the proposed optimization algorithm. The convergence curves of the proposed method on different datasets are given in Figure 5. As seen from the figures, the iterative update rules in Algorithm 1 decrease the objective function value obtained by our proposed method. Moreover, we can also see that our proposed method converges very quickly on these datasets.

5. Conclusions and Future Work

To efficiently learn the feature representations of multiview multimedia data, this paper proposes a new deep nonnegative matrix factorization method with multiview learning. Unlike traditional methods, the proposed method deeply decomposes the basis matrix, so it not only can learn the component representation of the original data but also can learn more abstract deep features. Furthermore, to effectively fuse the available multiview data information, this paper introduces an adaptive feature fusion mechanism.

To solve the shortcoming of information fusion for multiview data, a large number of fusion mechanisms have been proposed, and they achieve different performances on different datasets. Therefore, how to effectively integrate different mechanisms to improve the feature representation ability of a given approach is one of the key research tasks to be addressed in the future. Moreover, we will apply our method to other fields such as medical image procession and medical text analysis [48].

Data Availability

The data are derived from public domain resources.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is supported by the National Natural Science Foundation of China under grant nos. 62062040, 61962026, 62006174, and 71762018, the Chinese Postdoctoral Science Foundation (grant no. 2019M661117), the Provincial Key Research and Development Program of Jiangxi under grant nos. 20192ACBL21031 and 20202BABL202016, the Science and Technology Research Project of Jiangxi Provincial Department of Education (grant nos. GJJ191709 and GJJ191689), Fundamental Research Funds for the Central Universities under grant no. 2412019FZ049, the Graduate Innovation Foundation Project of Jiangxi Normal University under grant no. YJS2020045, and the Young Talent Cultivation Program of Jiangxi Normal University.

References

- [1] Y. Yi, Y. Chen, J. Wang, G. Lei, J. Dai, and H. Zhang, “Joint feature representation and classification via adaptive graph semi-supervised nonnegative matrix factorization,” *Signal Processing: Image Communication*, vol. 89, article 115984, 2020.
- [2] Y. Yi, J. Wang, W. Zhou, C. Zheng, J. Kong, and S. Qiao, “Non-negative matrix factorization with locality constrained adaptive graph,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 2, pp. 427–441, 2020.
- [3] G. T. Reddy, M. P. K. Reddy, K. Lakshmann et al., “Analysis of dimensionality reduction techniques on big data,” *IEEE Access*, vol. 8, pp. 54776–54788, 2020.
- [4] Y. Yi, J. Wang, W. Zhou, Y. Fang, J. Kong, and Y. Lu, “Joint graph optimization and projection learning for dimensionality reduction,” *Pattern Recognition*, vol. 92, pp. 258–273, 2019.
- [5] S. Ayesha, M. K. Hanif, and R. Talib, “Overview and comparative study of dimensionality reduction techniques for high dimensional data,” *Information Fusion*, vol. 59, pp. 44–58, 2020.
- [6] H. Abdi and L. J. Williams, “Principal component analysis,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.
- [7] A. Hyvärinen and E. Oja, “Independent component analysis: algorithms and applications,” *Neural Networks*, vol. 13, no. 4–5, pp. 411–430, 2000.
- [8] Y. Linde, A. Buzo, and R. Gray, “An algorithm for vector quantizer design,” *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, 1980.
- [9] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
- [10] Y. X. Wang and Y. J. Zhang, “Nonnegative matrix factorization: a comprehensive review,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, pp. 1336–1353, 2012.
- [11] A. A. Jamali, A. Kusalik, and F. X. Wu, “MDIPA: a micro-RNA–drug interaction prediction approach based on non-negative matrix factorization,” *Bioinformatics*, vol. 36, no. 20, pp. 5061–5067, 2020.
- [12] P. Chalise, Y. Ni, and B. L. Fridley, “Network-based integrative clustering of multiple types of genomic data using non-negative matrix factorization,” *Computers in Biology and Medicine*, vol. 118, p. 103625, 2020.
- [13] M. Hou, J. Li, and G. Lu, “A supervised non-negative matrix factorization model for speech emotion recognition,” *Speech Communication*, vol. 124, pp. 13–20, 2020.
- [14] P. O. Hoyer, “Non-negative matrix factorization with sparseness constraints,” *Journal of Machine Learning Research*, vol. 5, no. 9, pp. 1457–1469, 2004.
- [15] S. Choi, “Algorithms for orthogonal nonnegative matrix factorization,” in *2008 IEEE International Joint Conference on Neural Networks*, pp. 1828–1832, Hong Kong, China, June 2008.
- [16] C. H. Q. Ding, T. Li, and M. I. Jordan, “Convex and semi-nonnegative matrix factorizations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 45–55, 2008.
- [17] J. Fan and J. Cheng, “Matrix completion by deep matrix factorization,” *Neural Networks*, vol. 98, pp. 34–41, 2018.
- [18] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: a review and new perspectives,” *IEEE Transactions on*

- Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [19] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [20] G. Zhong, L. N. Wang, X. Ling, and J. Dong, “An overview on data representation learning: from traditional feature learning to recent deep learning,” *The Journal of Finance and Data Science*, vol. 2, no. 4, pp. 265–278, 2016.
- [21] J. H. Ahn, S. Kim, J. H. Oh, and S. Choi, “Multiple nonnegative-matrix factorization of dynamic PET images,” in *Proceedings of Asian Conference on Computer Vision*, pp. 1009–1013, Jeju, Korea, 2004.
- [22] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. W. Schuller, “A deep matrix factorization method for learning attribute representations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 3, pp. 417–429, 2016.
- [23] Y. Zhao, H. Wang, and J. Pei, “Deep non-negative matrix factorization architecture based on underlying basis images learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 6, pp. 1897–1913, 2021.
- [24] Y. Meng, R. Shang, F. Shang, L. Jiao, S. Yang, and R. Stolk, “Semi-supervised graph regularized deep NMF with bi-orthogonal constraints for data representation,” *IEEE transactions on neural networks and learning systems*, vol. 31, no. 9, pp. 3245–3258, 2020.
- [25] M. Tong, Y. Chen, L. Ma, H. Bai, and X. Yue, “NMF with local constraint and deep NMF with temporal dependencies constraint for action recognition,” *Neural Computing and Applications*, vol. 32, no. 9, pp. 4481–4505, 2020.
- [26] J. Li, G. Zhou, Y. Qiu, Y. Wang, Y. Zhang, and S. Xie, “Deep graph regularized non-negative matrix factorization for multi-view clustering,” *Neurocomputing*, vol. 390, pp. 108–116, 2020.
- [27] Z. Shu, X. Wu, C. Hu, C. Z. You, and H. H. Fan, “Deep semi-nonnegative matrix factorization with elastic preserving for data representation,” *Multimedia Tools and Applications*, vol. 80, no. 2, pp. 1707–1724, 2021.
- [28] J. Zhao, X. Xie, X. Xu, and S. Sun, “Multi-view learning overview: recent progress and new challenges,” *Information Fusion*, vol. 38, pp. 43–54, 2017.
- [29] Y. Li, M. Yang, and Z. Zhang, “A survey of multi-view representation learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 10, pp. 1863–1883, 2018.
- [30] T. Hussain, K. Muhammad, W. Ding, J. Lloret, S. W. Baik, and V. H. C. de Albuquerque, “A comprehensive survey of multi-view video summarization,” *Pattern Recognition*, vol. 109, p. 107567, 2021.
- [31] X. Xu, Y. Yang, C. Deng, and F. Nie, “Adaptive graph weighting for multi-view dimensionality reduction,” *Signal Processing*, vol. 165, pp. 186–196, 2019.
- [32] R. Zhang, F. Nie, X. Li, and X. Wei, “Feature selection with multi-view data: a survey,” *Information Fusion*, vol. 50, pp. 158–167, 2019.
- [33] P. Luo, J. Peng, Z. Guan, and J. Fan, “Dual regularized multi-view non-negative matrix factorization for clustering,” *Neurocomputing*, vol. 294, pp. 1–11, 2018.
- [34] J. Liu, C. Wang, J. Gao, and J. Han, “Multi-view clustering via joint nonnegative matrix factorization,” in *Proceedings of the 2013 SIAM International Conference on Data Mining*, pp. 252–260, Austin, TX, USA, 2013.
- [35] W. Y. Chang, C. P. Wei, and Y. C. F. Wang, “Multi-view non-negative matrix factorization for clothing image characterization,” in *2014 22nd International Conference on Pattern Recognition*, pp. 1272–1277, Stockholm, Sweden, August 2014.
- [36] N. Liang, Z. Yang, Z. Li, W. Sun, and S. Xie, “Multi-view clustering by non-negative matrix factorization with co-orthogonal constraints,” *Knowledge-Based Systems*, vol. 194, p. 105582, 2020.
- [37] K. Zhan, J. Shi, J. Wang, H. Wang, and Y. Xie, “Adaptive structure concept factorization for multiview clustering,” *Neural Computation*, vol. 30, no. 4, pp. 1080–1103, 2018.
- [38] S. Wei, J. Wang, G. Yu, C. Domeniconi, and X. Zhang, “Multi-view multiple clusterings using deep matrix factorization,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 4, pp. 6348–6355, 2020.
- [39] W. Zhao, C. Xu, Z. Guan, and Y. Liu, “Multiview concept learning via deep matrix factorization,” *IEEE transactions on neural networks and learning systems*, vol. 32, no. 2, pp. 814–825, 2021.
- [40] H. Zhao, Z. Ding, and Y. Fu, “Multi-view clustering via deep matrix factorization,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.
- [41] S. Huang, Z. Kang, and Z. Xu, “Auto-weighted multi-view clustering via deep matrix decomposition,” *Pattern Recognition*, vol. 97, article 107015, 2020.
- [42] D. Greene and P. Cunningham, “Practical solutions to the problem of diagonal dominance in kernel document clustering,” in *Proceedings of the 23rd international conference on Machine learning*, pp. 377–384, New York, NY, USA, 2006.
- [43] G. Bisson and C. Grimal, “An architecture to efficiently learn co-similarities from multi-view datasets,” in *Neural Information Processing. ICONIP 2012. Lecture Notes in Computer Science*, vol. 7663, T. Huang, Z. Zeng, C. Li, and C. S. Leung, Eds., pp. 184–193, Springer, Berlin, Heidelberg, 2012.
- [44] L. Fu, P. Lin, A. V. Vasilakos, and S. Wang, “An overview of recent multi-view clustering,” *Neurocomputing*, vol. 402, pp. 148–161, 2020.
- [45] Y. Yi, Y. Shi, H. Zhang, J. Wang, and J. Kong, “Label propagation based semi-supervised non-negative matrix factorization for feature extraction,” *Neurocomputing*, vol. 149, pp. 1021–1037, 2015.
- [46] H. Zhu and M. C. Zhou, “Efficient role transfer based on Kuhn–Munkres algorithm,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 42, no. 2, pp. 491–496, 2011.
- [47] H. Wang, Y. Yang, and B. Liu, “GMC: graph-based multi-view clustering,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 6, pp. 1116–1129, 2019.
- [48] X. Huang, B. Zhong, Y. Cao, Y. Yi, and M. Gu, “Chest X-ray lung Chinese description generation based on semantic labels and hierarchical LSTM,” in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1020–1023, Seoul, Korea (South), December 2020.