Pattern Recognition 108 (2020) 107524

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Multi-view subspace learning via bidirectional sparsity

Ruidong Fan, Tingjin Luo, Wenzhang Zhuge, Sheng Qiang, Chenping Hou*

College of Liberal Arts and Sciences, National University of Defense Technology, Changsha, China, 410073

ARTICLE INFO

Article history:
Received 26 November 2019
Revised 17 April 2020
Accepted 30 June 2020
Available online 10 July 2020

Keywords:
Multi-view clustering
Subspace learning
Bidirectional sparsity
Non-convex optimization

ABSTRACT

With the improvement of multi-view data collection technology, multi-view learning has become a hot research area. How to deal with diverse and complex data is one of the challenging problems in multi-view learning. However, it is hard for traditional multi-view subspace learning methods to find an effective subspace dimension and deal with outliers simultaneously. In this paper, we propose a novel method, named as Multi-view Subspace Learning via Bidirectional Sparsity(SLBS), which is effective to overcome the above difficulties and learn a better representation. Specifically, we divide the shared subspace into two parts. One is a row sparse matrix to do a secondary extraction of features and the other is a column sparse matrix to reduce the influence of outliers. The proposed model is a non-convex problem which is difficult to be solved. To address this problem, we develop an efficient algorithm and analyze its convergence and computational complexity. Finally, compared with other multi-view subspace learning methods, the extensive experimental results on real-world datasets present the effectiveness of our SLBS.

1. Introduction

Driven by diverse access to information technology, many samples can be presented in two or more different ways. For example, we can describe images by different visual descriptors such as HOG [1], LBP [2], SIFT [3] and GIST [4] etc. Different types of visual descriptors can represent different features of an image. This kind of data which is obtained by the different approaches or different levels for the same object is called multi-view data.

Real-world data is always characterized by high dimensionality, complexity, and redundancy. Therefore, it is difficult to analyze the intrinsic structure of multi-view data. To solve this problem, lots of multi-view learning methods [5,6] have been proposed. Multi-view learning aims to better find the relationships of different views. In the extant literature, the approaches of multi-view learning are mainly divided into two types: unsupervised learning [7,8] and supervised learning [9,10]. Because the labels of data are difficult to obtain in many cases, we choose unsupervised multi-view learning as our research aspect.

There are many traditional unsupervised multi-view learning methods. We focus on two categories as follows: (1) Multi-view learning based on graph [11–13]: This category of methods aims to learn a fusion graph by using all views and then uses effective algorithms such as special clustering [14] on the fusion graph to get the results. (2) Multi-view learning based on shared subspace [15–17]: Based on the assumption of all views share a common representation, this category of methods aims to learn a unified feature representation of all views.

Although traditional unsupervised subspace learning approaches perform noticeably well in many cases, their efficiency can be improved by addressing the following challenges.

Errors may occur in the process of collecting and storing data. Thus how to deal with outliers is one of the difficulties in multi-view learning. To reduce the influence of outliers in the data and improve the robustness of the model, lots of robust multi-view methods have been proposed. They can be divided into two groups as follows: (1) Robust multi-view learning based on sparse norm [18–20]. For example, Pu et al. [19] adopts $\ell_{2,1}$-norm instead of ℓ_2-norm to improve the robustness of the model. (2) Robust multi-view learning based on low-rank and sparse decomposition [21–23]. For example, Xia et al. [22] proposes a robust learning scheme to remove errors and noise and learn a reliable low-rank representation.

Traditional multi-view approaches are hard to choose a suitable dimension of the unified representation. Lots of methods make the dimension of the unified representation same as the number of clusters. However, if the multi-view data is high-dimensional and the number of clusters is small, then adopting the above mentioned schemes may lose too much information. But if we choose a large dimension of feature extraction, it may not remove redundant features completely. Thus how to select the appropriate fea-

* Corresponding author.
E-mail addresses: fanruidong1996@163.com (R. Fan), tingjinluo@gmail.com (T. Luo), zgwznudt@yeah.net (W. Zhuge), 13874998059@163.com (S. Qiang), houchenping@nudt.edu.cn (C. Hou).

https://doi.org/10.1016/j.patcog.2020.107524
0031-3203/

Table 1
Notions.

Notations	Descriptions
d_v	The dimensionality of the vth view
n	Data size
V	The numbers of views
r_v	The reduced dimensionality of the vth view data
r	The dimensionality of feature extraction
$\alpha^{(v)}$	The weight coefficient of the vth view
c	The weight redistribution parameter
$\mathbf{X}^{(v)} \in \mathbb{R}^{d_v \times n}$	Data matrix of the vth view
$\mathbf{U}^{(v)} \in \mathbb{R}^{d_v \times r}$	The projection matrix of the vth view
$\mathbf{P}, \mathbf{Q} \in \mathbb{R}^{r \times n}$	The common representation matrix
$\alpha \in \mathbb{R}^{V \times 1}$	The weight coefficient vector

ture dimension is also one of the difficulties in multi-view learning.

To make better feature selection and reduce the influence of outliers effectively, we propose an approach called Multi-view Subspace Learning via Bidirectional Sparsity(SLBS) which simultaneously captures a common set of features among relevant data and identifies outliers. Specifically, we decompose the low-dimensional representation learned by matrix decomposition into two matrices. We impose $\ell_{2,p}$ row sparse norm into the first matrix for capturing the well-shared features without redundant features among relevant data. It is equivalent to the secondary feature extraction of the data. To simultaneously identify the outliers and reduce its influence, we impose $\ell_{p,2}$ column sparse norm into the second matrix. We propose an effective algorithm to solve the optimization problem in SLBS and show that the proposed algorithm is also suitable for large-size problems. Besides, we provide a detailed theoretical convergence analysis of the proposed SLBS algorithm. Compared with traditional multi-view unsupervised feature extraction approaches, our method has been demonstrated to have better performances on some data sets.

The remainder of this paper is organized as follows. Section 2 introduces the notations and reviews some prevalent multi-view subspace clustering methods. Section 3 states the problem to be solved and provides an effective solution to this problem. The convergence behavior, computational cost and parameter determination are analyzed in Section 4. Some promising comparing results are provided on various kinds of data sets in Section 5, followed by the conclusions and future work in Section 6.

2. Related work

In this section, we introduce some notations used in this paper firstly. Next, we briefly introduce some previous work of multi-view unsupervised learning.

2.1. Notation

In this paper, we write the matrices and vectors as boldface uppercase letters and boldface lowercase letters. For a matrix $\mathbf{W} = (w_{ij})$, the ith row of $\mathbf{W}$ is denoted by $\mathbf{W}_{i.}$ and the jth column of $\mathbf{W}$ is denoted by $\mathbf{W}_{.j}$. The Frobenius norm is denoted by $||\cdot||_F$. The ℓ_p-norm of a vector $\mathbf{v} \in \mathbb{R}^{n\times 1}$ is defined as $||\mathbf{v}||_p = (\sum_{i=1}^{n} |v_i|^p)^{\frac{1}{p}}$ and the $\ell_{q,p}$-norm of a matrix $\mathbf{W} \in \mathbb{R}^{m\times n}$ is defined as

$$||\mathbf{W}||_{q,p} = \left(\sum_{i=1}^{n} \left(\sum_{j=1}^{m} |m_{ij}|^q \right)^{\frac{p}{q}} \right)^{\frac{1}{p}}, q > 0, p > 0.$$

We summarize the notations used in this paper in Table 1.

2.2. RMF

Regularized Matrix Factorization(RMF) [24] is a general framework and many typical feature extraction methods can be considered as the special case of RMF. More precisely, the RMF framework factorizes a $d \times n$-dimensional matrix $\mathbf{X}$ into the product of a $d \times r$-dimensional matrix $\mathbf{D}$ and a $r \times n$-dimensional matrix $\mathbf{A}$ so that the error is minimized. Furthermore, RMF exploits the regularizers to constrain the forms of $\mathbf{D}$ and $\mathbf{A}$. So RMF can be formulated as

$$\begin{cases} \min\limits_{\mathbf{D},\mathbf{A}} \dfrac{1}{n} ||\mathbf{X} - \mathbf{D}\mathbf{A}||_F^2 + \lambda\phi(\mathbf{D}) + \gamma\varphi(\mathbf{A}) \\ s.t. \quad \mathbf{D} \in \Lambda_D, \mathbf{A} \in \Lambda_A. \end{cases}$$

Where Λ_D and Λ_A are the domains of the dictionary $\mathbf{D}$ and of latent embedding $\mathbf{A}$, respectively. These domains allow to enforce additional constraints on those matrices. Several existing algorithms, such as PCA, sparse coding(SC), group SC, structured sparse PCA(SSPCA), group Lasso and group structured sparse matrix factorization(GSSMF), can be considered as special cases of this general framework. In this framework, GSSMF performs best on most data sets (Table 2).

2.3. MVNMF

Gao et al. [29] proposed a NMF-based multi-view clustering algorithm (MVNMF) which formulates a joint matrix factorization process for all views. The algorithm pushes the representations of each view towards a consensus representation and uses the common representation for clustering. The objective function of MVNMF is formulated as follows:

$$\begin{cases} \sum\limits_{v=1}^{V} ||\mathbf{X}^{(v)} - \mathbf{U}^{(v)}(\mathbf{V}^{(v)})^T||_F^2 + \sum\limits_{v=1}^{V} \lambda_v ||\mathbf{V}^{(v)} - \mathbf{V}^*||_F^2 \\ s.t. \ \forall 1 \leqslant k \leqslant K, \ ||\mathbf{U}_{.k}^{(v)}|| = 1, \ \mathbf{U}^{(v)}, \mathbf{V}^{(v)}, \mathbf{V}^* \geqslant 0. \end{cases}$$

2.4. AMGL

Auto-Weighted Multiple Graph Learning(AMGL) [30] is a parameter-free graph learning framework that can be used both for multi-view clustering and semi-supervised classification. Since we aim to solve an unsurprised problem, then we only care about the part of multi-view clustering in AMGL. The objective function of AMGL is as follows:

$$\min_{\mathbf{F}^T\mathbf{F}=I} \sqrt{Tr(\mathbf{F}^T\mathbf{L}^{(v)}\mathbf{F})}$$

Here $\mathbf{L}^{(v)}$ is the normalized Laplacian matrix of the vth view.

2.5. RGC

Robust Graph Construction(RGC) [21] is a robust graph learning scheme which aims to learn a reliable graph from real-world data with noise and error. Specifically, this paper divides the raw data into error part and clean part, and then the graph learning is completed on the clean part.

$$\begin{cases} \min\limits_{\mathbf{D},\mathbf{E},\mathbf{S}} ||\mathbf{D}||_* + \alpha||\mathbf{E}||_1 + \beta Tr(\mathbf{D}\mathbf{L}\mathbf{D}^T) + \gamma||\mathbf{S}||_F^2 \\ s.t. \quad \mathbf{X} = \mathbf{D} + \mathbf{E}, \ \mathbf{S}\mathbf{1} = \mathbf{1}, \ 0 \leqslant \mathbf{S} \leqslant 1 \end{cases}.$$

Where α, β and γ are all trade-off parameters.

3. Multi-view subspace learning via bidirectional sparsity

In this section, we first introduce our objective function step by step. Then we provide an effective algorithm which combines the gradient method with the direct solution to solve this problem.

Table 2
RMF.

Method	$\phi(\mathbf{D})$	$\varphi(\mathbf{A})$	Λ_D or Λ_A
PCA	none	none	$\{\mathbf{D}\|\mathbf{D}^T\mathbf{D}=I\}$
SC [25]	none	$\|\|\mathbf{A}^T\|\|_{1,1}$	$\{\mathbf{D}\|\|\|\mathbf{D}_{.i}\|\|_2 \le 1, \forall i \le n_d\}$
Group SC [26]	$\|\|\mathbf{D}^T\|\|_{1,2}$	$\sum_g \|\|\mathbf{A}_g\|\|_{1,2}$	none
SSPCA [27]	$\sum_g \|\|\mathbf{D}_g\|\|_{\xi,2}$	none	$\{\mathbf{A}\|\|\|\mathbf{A}_{i.}\|\|_2 \le 1, \forall i \le n_d\}$
Group Lasso [28]	none	$\sum_g \|\|(\mathbf{A}_g)^T\|\|_{1,2}$	$\{\mathbf{D}\|\mathbf{D}^T\mathbf{D}=I\}$
GSSMF [24]	$\sum_g \|\|(\mathbf{D}_g)^T\|\|_{1,\infty}$	$\|\|\mathbf{A}\|\|_{1,\infty}$	none

3.1. The proposed model

Assume that we now have V views. Let $\chi = \{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(V)}\}$ represents the data of all views. In the traditional matrix factorization, we can find two sets of matrix factors $\mathbf{U}^{(v)} \in \mathbb{R}^{d_v \times r_v}$ and $\mathbf{L}^{(v)} \in \mathbb{R}^{r_v \times n}$ to approximate every single view:

$$\mathbf{X}^{(v)} \approx \mathbf{U}^{(v)}\mathbf{L}^{(v)}$$

We can view $\mathbf{L}^{(v)}$ as a potentially low-dimensional representation of $\mathbf{X}^{(v)}$. This method is not suitable for multi-view problem due to the correlation between different views. To solve this problem, we assume that the different features of multi-view data arise from the same potential space(i.e. low-dimensional representation). Specifically, when we map the different data to a shared, low-dimensional and potential space, we can get a compact distribution of data and reveal statistical relationships and essential structures between different views. Thus for all views, we have

$$\mathbf{X}^{(v)} \approx \mathbf{U}^{(v)}\mathbf{L}$$

Here $\mathbf{L} \in \mathbb{R}^{r \times n}$. Then we can use the shared low-dimensional representation to solve multi-view problems. One of the common reconstruction processes can be formulated as a Frobenius norm optimization problem which is defined as follows:

$$\min \sum_{v=1}^{V} ||\mathbf{X}^{(v)} - \mathbf{U}^{(v)}\mathbf{L}||_F^2$$

Although the above method can get a good low-dimensional representation of multi-view data, it only considers that the multi-view data has a shared space and ignores the inherent structure of the shared subspace itself. We think that even though we have reduced multi-view data to a low dimension, the low-dimensional representation may have redundant features or relevant features. So the inherent representation should be a row sparse matrix for feature extraction. Further more, we think that the real-world data is not all clean and may have a few outliers. So the inherent representation should be a column sparse matrix for samples. But the inherent low-dimensional representation is not a matrix with sparse rows and sparse columns. Because if so, the low-dimensional representation also optimizes the outliers. Taking the above points into account, we divide the low-dimensional representation into two parts: the first part is a row sparse matrix for optimizing features and the other part is a column sparse matrix for optimizing samples. The objective function can be formulated as:

$$\min \sum_{v=1}^{V} ||\mathbf{X}^{(v)} - \mathbf{U}^{(v)}(\mathbf{P}+\mathbf{Q})||_F^2$$

Here $\mathbf{P}$ is a row sparse matrix and $\mathbf{Q}$ is a column sparse matrix. Because $\mathbf{P}$ is a row sparse matrix and then by applying regularization, we can formulate it as

$$R(\mathbf{P}) = \sum_{i=1}^{r} ||\mathbf{P}_{i.}||_2^0$$

But this formula is a NP-hard problem which is hard to be solved and then we relax it as

$$R(\mathbf{P}) = \sum_{i=1}^{r} ||\mathbf{P}_{i.}||_2^p$$

Here $p \in (0, 1)$ represents sparsity. Then the objective function of our method can formulate as

$$\min \sum_{v=1}^{V} ||\mathbf{X}^{(v)} - \mathbf{U}^{(v)}(\mathbf{P}+\mathbf{Q})||_F^2 + \beta_1||\mathbf{P}||_{2,p}^p + \beta_2||\mathbf{Q}^T||_{2,p}^p$$

Since $\mathbf{P}$ and $\mathbf{Q}$ have the similar status in the formulation and then we set $\lambda = \beta_1 = \beta_2$.

Consider that different views should play various roles in the problem, we use parameters to balance the effectiveness of different views. Mathematically, the objective function is

$$\min \sum_{v=1}^{V} \alpha^{(v)}||\mathbf{X}^{(v)} - \mathbf{U}^{(v)}(\mathbf{P}+\mathbf{Q})||_F^2 + \lambda(||\mathbf{P}||_{2,p}^p + ||\mathbf{Q}^T||_{2,p}^p) + \gamma \sum_{v=1}^{V} (\alpha^{(v)})^c$$

Here γ is a smoothing factor and we set it to 1. Finally, we add the Frobenius norm of $\mathbf{U}^{(v)}$ on the objective function to avoid ill-conditioned problem and reduce the freedom of the variables. So the finally objective function of our method is

$$\begin{aligned}\min_{\mathbf{U}^{(v)},\mathbf{P},\mathbf{Q},\{\alpha^{(v)}\geqslant 0\}} \quad & \sum_{v=1}^{V} \alpha^{(v)}||\mathbf{X}^{(v)} - \mathbf{U}^{(v)}(\mathbf{P}+\mathbf{Q})||_F^2 + \lambda_1 \sum_{v=1}^{V} ||\mathbf{U}^{(v)}||_F^2 \\ & +\lambda_2(||\mathbf{P}||_{2,p}^p + ||\mathbf{Q}^T||_{2,p}^p) + \sum_{v=1}^{V} (\alpha^{(v)})^c.\end{aligned} \tag{1}$$

3.2. Optimization algorithm

From Eq. (1) we can see that four groups of variables need to be solved. $\mathbf{U}^{(v)}$s are the projection matrices of all views $\mathbf{X}^{(v)}$s. $\mathbf{P}$ and $\mathbf{Q}$ are the unified representations of all views. $\boldsymbol{\alpha}$ contains the weight coefficients which balance the importance of different views. But it is difficult to solve Eq. (1) directly because all variables are coupled in the formula. So we offer an effective algorithm which updates variables alternatively. Specifically, we fix three groups of variables and only optimize the remaining one variable alternatively.

(1) **Fix P, Q and $\boldsymbol{\alpha}$, optimize $\mathbf{U}^{(v)}$.** When **P, Q** and $\boldsymbol{\alpha}$ are fixed, we need to optimize a set of projection matrices $\mathbf{U}^{(v)}$. It is obvious that the second and third terms in the objective function are not related to $\mathbf{U}^{(v)}$. Then the optimization subproblem becomes

$$\min_{\mathbf{U}^{(v)}} G = \min_{\mathbf{U}^{(v)}} \sum_{v=1}^{V} \alpha^{(v)}||\mathbf{X}^{(v)} - \mathbf{U}^{(v)}(\mathbf{P}+\mathbf{Q})||_F^2 + \lambda_1 \sum_{v=1}^{V} ||\mathbf{U}^{(v)}||_F^2 \tag{2}$$

This problem is convex and derivable, so we can get its solution with close form through its derivative.

$$\frac{\partial G}{\partial \mathbf{U}^{(v)}} = 2\alpha^{(v)}[-\mathbf{X}^{(v)} + \mathbf{U}^{(v)}(\mathbf{P}+\mathbf{Q})](\mathbf{P}+\mathbf{Q})^T + 2\lambda_1\mathbf{U}^{(v)} \tag{3}$$

Set the derivative to zero, we get

$$\mathbf{U}^{(v)} = \mathbf{X}^{(v)}(\mathbf{P}+\mathbf{Q})^T[(\mathbf{P}+\mathbf{Q})(\mathbf{P}+\mathbf{Q})^T + \frac{\lambda_1}{\alpha^{(v)}}I_k]^{-1} \tag{4}$$

(2) **Fix** $\{\mathbf{U}^{(v)}\}_{v=1}^{V}$, **Q and** $\boldsymbol{\alpha}$, **optimize P**. We delete the terms which are not related to **P**. Then the optimization subproblem becomes

$$\min_{\mathbf{P}} \sum_{v=1}^{V} \alpha^{(v)} ||\mathbf{X}^{(v)} - \mathbf{U}^{(v)}(\mathbf{P}+\mathbf{Q})||_F^2 + \lambda_2 ||\mathbf{P}||_{2,p}^p \tag{5}$$

It is hard to solve the problem directly because the second term is non-convex and non-differentiable. Inspired by the basic idea in solving the $L_{2,1}$ term, we take the derivative of $||\mathbf{P}||_{2,p}^p$ with respect to **P**. The derivative of $||\mathbf{P}||_{2,p}^p$ is

$$\frac{\partial ||\mathbf{P}||_{2,p}^p}{\partial \mathbf{P}} = 2\mathbf{D}\mathbf{P} \tag{6}$$

Where $\mathbf{D} \in \mathbb{R}^{r\times r}$ is a diagonal matrix with the ith diagonal element as

$$d_{ii} = \frac{p}{2}||\mathbf{P}_{i\cdot}||_2^{p-2} \tag{7}$$

Thus the problem (5) can be solved by solving the following problem iteratively.

$$\sum_{v=1}^{V} \alpha^{(v)} ||\mathbf{X}^{(v)} - \mathbf{U}^{(v)}(\mathbf{P}+\mathbf{Q})||_F^2 + \lambda_2 tr(\mathbf{P}^T\mathbf{D}\mathbf{P}) \tag{8}$$

The new problem is convex and we can get its solution with close form.

$$\mathbf{P} = [\sum_{v=1}^{V} \alpha^{(v)}(\mathbf{U}^{(v)})^T\mathbf{U}^{(v)} + \lambda_2\mathbf{D}]^{-1}[\sum_{v=1}^{V} \alpha^{(v)}(\mathbf{U}^{(v)})^T(\mathbf{X}^{(v)} - \mathbf{U}^{(v)}\mathbf{Q})] \tag{9}$$

(3) **Fix** $\{\mathbf{U}^{(v)}\}_{v=1}^{V}$, **P and** $\boldsymbol{\alpha}$, **optimize Q**. We can get the objective function of this subproblem as follows:

$$\min_{\mathbf{Q}} \sum_{v=1}^{V} \alpha^{(v)} ||\mathbf{X}^{(v)} - \mathbf{U}^{(v)}(\mathbf{P}+\mathbf{Q})||_F^2 + \lambda_2 ||\mathbf{Q}^T||_{2,p}^p \tag{10}$$

As similar to **P**, the problem (10) can be solved by solving the following problem iteratively.

$$\min_{\mathbf{Q}} \sum_{v=1}^{V} \alpha^{(v)} ||\mathbf{X}^{(v)} - \mathbf{U}^{(v)}(\mathbf{P}+\mathbf{Q})||_F^2 + \lambda_2 tr(\mathbf{Q}\mathbf{S}\mathbf{Q}^T) \tag{11}$$

Where **S** is similar to **D**, $\mathbf{S} \in \mathbb{R}^{n\times n}$ and

$$s_{jj} = \frac{p}{2}||\mathbf{Q}_{j\cdot}||_2^{p-2} \tag{12}$$

The problem (11) is a sylvester equation and difficult to be solved with close form. So we adopt a gradient descent method to solve it. Assume that we have finished the kth iteration. Then the $(k+1)$th iteration is

$$\mathbf{Q}_{k+1} = \mathbf{Q}_k - t_k \nabla M(\mathbf{Q}_k) \tag{13}$$

Where $\nabla M(\mathbf{Q}_k)$ is the derivative of problem (11).

$$\nabla M(\mathbf{Q}_k) = \sum_{v=1}^{V} \alpha^{(v)}(\mathbf{U}^{(v)})^T\mathbf{U}^{(v)}\mathbf{Q}_k + \lambda_2\mathbf{Q}_k S - \sum_{v=1}^{V} \alpha^{(v)}(\mathbf{U}^{(v)})^T(\mathbf{X}^{(v)} - \mathbf{U}^{(v)}\mathbf{P}) \tag{14}$$

And t_k is the iteration step size and fixed by exact linear search method. That is

$$t_k = \arg\min_t M(\mathbf{Q}_k - t\nabla M(\mathbf{Q}_k)) \tag{15}$$

(4) **Fix** $\{\mathbf{U}^{(v)}\}_{v=1}^{V}$, **P and Q, optimize** $\boldsymbol{\alpha}$. The objective function of this subproblem is as follows:

$$\min_{\alpha^{(v)}\geqslant 0} \sum_{v=1}^{V} \alpha^{(v)} ||\mathbf{X}^{(v)} - \mathbf{U}^{(v)}(\mathbf{P}+\mathbf{Q})||_F^2 + \sum_{v=1}^{V} (\alpha^{(v)})^c \tag{16}$$

This problem has an inequality constraint. We set $h^{(v)} = ||\mathbf{X}^{(v)} - \mathbf{U}^{(v)}(\mathbf{P}+\mathbf{Q})||_F^2$ and then we construct the function as follows:

$$H(\alpha) = \sum_{v=1}^{V} \alpha^{(v)}h^{(v)} + \sum_{v=1}^{V} (\alpha^{(v)})^c \tag{17}$$

We can take the derivative on $\alpha^{(v)}$ and set it to zero.

$$\frac{\partial H}{\partial \alpha^{(v)}} = h^{(v)} + c(\alpha^{(v)})^{c-1} = 0 \tag{18}$$

Then we can get $\alpha^{(v)}$ as follows:

$$\alpha^{(v)} = (-\frac{h^{(v)}}{c})^{\frac{1}{c-1}} \tag{19}$$

One point should be mentioned here. Because of $h^{(v)} \geq 0$ and $c < 0$, the derived results of $\alpha^{(v)}$ satisfy its non-negativity condition. The procedure of our method is listed in Algorithm 1.

Algorithm 1

Input: Data matrix $\mathbf{X}^{(v)}$, parameters λ_1, p, λ_2 and c.
Output: the latent subspaces **P** and **Q**.
1.Initialize $\alpha^{(v)} = \frac{1}{V}$, $\mathbf{P} = \mathbf{P}_0$ and $\mathbf{Q} = \mathbf{Q}_0$;
Repeat
2.Update projection matrices $\mathbf{U}^{(v)}$ using Eq. (4);
3.Update row sparsely shared matrix **P** using Eq. (9);
4.Update column sparsely shared matrix **Q** using Eq. (13);
5.Update weight coefficients $\alpha^{(v)}$ using Eq. (20);
Until converges

4. Discussion

In this section, we analyze SLBS in three aspects. First, we prove that the objective function value of SLBS is non-increasing in each iteration. The second part is the computational complexity which can prove the efficiency of Algorithm 1. Finally, we discuss the influence of model parameters and then identify the update strategies.

4.1. Convergence analysis

Lemma 1. *Suppose that a, b are the nonzero vectors, then for any* $p \in (0, 2]$,

$$||a||_2^p - \frac{p}{2}\frac{||a||_2^2}{||b||_2^{2-p}} \leqslant ||b||_2^p - \frac{p}{2}\frac{||b||_2^2}{||b||_2^{2-p}} \tag{20}$$

Eq. (20) hold if and only if $||a||_2^p = ||b||_2^p$.

Proof. Denote $\varphi(t) = t^p - \frac{p}{2}t^2 + \frac{p}{2} - 1$, then we have

$$\varphi'(t) = pt^{p-1} - pt = pt(t^{p-2} - 1) \tag{21}$$

We analyze the Eq. (21). For any $p \in (0, 2]$, if $t \in (0, 1]$, $\varphi'(t) \geqslant 0$ and if $t > 1$, $\varphi'(t) < 0$. So $t = 1$ is the maximum point. We find that when $t = 1$, $\varphi(t) = 0$. In other words, when $t > 0$, $\varphi(t) \leq 0$ always stands up.

We let $t = \frac{||a||_2}{||b||_2}$ and then can get

$$\varphi(t) = t^p - \frac{p}{2}t^2 + \frac{p}{2} - 1 = \frac{||a||_2^p}{||b||_2^p} - \frac{p}{2}\frac{||a||_2^2}{||b||_2^2} + \frac{p}{2} - 1 \leqslant 0 \tag{22}$$

The Eq. (22) is equivalent to the following inequality.

$$||a||_2^p - \frac{p}{2}\frac{||a||_2^2}{||b||_2^{2-p}} \leqslant ||b||_2^p - \frac{p}{2}\frac{||b||_2^2}{||b||_2^{2-p}} \tag{23}$$

□

Lemma 2. *By lemma 1, we can prove that the optimal solution obtained by Formula (8) is also a solution which makes Formula (5) non-increasing.*

Proof. We set $F(P) = \sum_{v=1}^{V} ||X^{(v)} - U^{(v)}(P+Q)||_F^2$ and assume the solution of the third step is P^*. Then we can get

$$F(P^*) + \lambda tr((P^*)^T D P^*) \leqslant F(P) + \lambda tr((P)^T D P) \tag{24}$$

Here $D \in R^{r \times r}$ is a diagonal matrix with the ith diagonal element as $d_{ii} = \frac{p}{2}||P_{i.}||_2^{p-2}$. So the function (24) is equivalent to the following inequality.

$$F(P^*) + \lambda\frac{p}{2}\frac{||P^*||_2^2}{||P||_2^{2-p}} \leqslant F(P) + \lambda\frac{p}{2}\frac{||P||_2^2}{||P||_2^{2-p}} \tag{25}$$

By Lemma 1, we can get

$$\sum_{i=1}^{r}(||P_{i.}^*||_2^p - \frac{p}{2}\frac{||P_{i.}^*||_2^2}{||P_{i.}||_2^{2-p}}) \leqslant \sum_{i=1}^{r}(||P_{i.}||_2^p - \frac{p}{2}\frac{||P_{i.}||_2^2}{||P_{i.}||_2^{2-p}}) \tag{26}$$

Substituting formula (26) into formula (25), the result is

$$F(P^*) + \lambda\sum_{i=1}^{r}||P_{i.}^*||_2^p \leqslant F(P) + \lambda\sum_{i=1}^{r}||P_{i.}||_2^p \tag{27}$$

It is equal to the following formula.

$$F(P^*) + \lambda||P^*||_{2,p}^p \leqslant F(P) + \lambda||P||_{2,p}^p \tag{28}$$

□

Theorem 1. *By employing the optimization procedure in Algorithm 1, the objective function value of SLBS in formula (1) is non-increasing.*

Proof. We set

$$\begin{aligned} L(U^{(v)}, P, Q, \alpha^{(v)}) &= \sum_{v=1}^{V}\alpha^{(v)}||X^{(v)} - U^{(v)}(P+Q)||_F^2 \\ &+ \lambda_1\sum_{v=1}^{V}||U^{(v)}||_F^2 \\ &+ \lambda_2(||P||_{2,p}^p + ||Q^T||_{2,p}^p) + \sum_{v=1}^{V}(\alpha^{(v)})^c \end{aligned} \tag{29}$$

We use $\widetilde{U^{(v)}}, \widetilde{P}, \widetilde{Q}, \widetilde{\alpha^{(v)}}$ to denote the updated $U^{(v)}$, P, Q, $\alpha^{(v)}$ in each iteration. We first prove that updating $U^{(v)}$ will not increase the value of the objective function. According to the Eq. (4), we have

$$L(\widetilde{U^{(v)}}, P, Q, \alpha^{(v)}) \leqslant L(U^{(v)}, P, Q, \alpha^{(v)})$$

Then we prove that updating P and Q will not increase the value of the objective function. We set $F(P) = \sum_{v=1}^{V}||X^{(v)} - U^{(v)}(P+Q)||_F^2$ and we can get $F(\widetilde{P}) + \lambda tr((\widetilde{P})^T D\widetilde{P}) \leqslant F(P) + \lambda tr((P)^T DP)$ by the Eq. (9). Through the Lemma 2, we obtain that $F(\widetilde{P}) + \lambda||\widetilde{P}||_{2,p}^p \leqslant F(P) + \lambda||P||_{2,p}^p$. The update of Q is the same as P and then we can get that when we update P and Q through Algorithm 1, the value of the objective function will not increase. That is,

$$L(\widetilde{U^{(v)}}, \widetilde{P}, \widetilde{Q}, \alpha^{(v)}) \leqslant L(\widetilde{U^{(v)}}, P, Q, \alpha^{(v)})$$

Then, since the updated α_v are the optimal solution to the problems in Eq. (17) respectively, the following inequalities hold.

$$L(\widetilde{U^{(v)}}, \widetilde{P}, \widetilde{Q}, \widetilde{\alpha^{(v)}}) \leqslant L(\widetilde{U^{(v)}}, \widetilde{P}, \widetilde{Q}, \alpha^{(v)})$$

Combine the above conclusions, we get the results. □

Through the convergence analysis of our model, we can safely come to the following conclusions. We think that our convergence analysis is the first analysis of the multi-view model with bidirectional sparsity. Thus, it has extended the use of sparsity into new scope with theoretical guarantee. We think that our algorithm lays a preliminary foundation for further theoretical analysis of other models with bidirectional sparsity. For example, it is natural to extend the application to supervised multi-view learning and use bidirectional sparsity for the relevant features and outliers. In the potential application of multi-task learning, we can use bidirectional sparsity for the relevant tasks and outlier tasks. We think that our convergence analysis can also be borrowed for theoretical guarantee.

4.2. Computational complexity

As SLBS is solved in an alternative way, we can sum the computational complexity of every sub-step to calculate their total computational complexity. The computational complexity of every sub-step is listed as follows:

(1) Eq. (4) aims to update the group of projection matrices $\mathbf{U}^{(v)}$. It has closed form solution and the computational complexity is $\mathcal{O}(\sum_{v=1}^{V}(d_v \times n \times r))$.
(2) The main computational complexity of Eq. (9) is when we update the unified matrices $\mathbf{P}$. It has closed form solution and the computational complexity is $\mathcal{O}(\sum_{v=1}^{V}(d_v \times n \times r))$.
(3) Eq. (13) aims to update the shared matrix $\mathbf{Q}$. But it is hard to get the close form solution and we use the iterative method to solve it. Then the computational complexity is $\mathcal{O}(\sum_{v=1}^{V}(d_v \times n \times r \times k))$. Where k is the number of iterations of this subproblem.
(4) The problem in Eq. (20) aims to update the optimal weight for each view. The computational complexity is $\mathcal{O}(\sum_{v=1}^{V}(d_v \times n \times r))$.

The parameter k is also less than twenty and then the total computational complexity of SLBS is $\mathcal{O}(T \times \sum_{v=1}^{V}(d_v \times n \times r \times k))$, where T is the number of iterations. Accordingly, our proposed SLBS method costs time only linear in n.

4.3. Parameter determination

Parameter is an important part of the model and parameter determination is closely related to the performance of the model. There are many parameters which need to be determined in Eq. (1). But we can divide them into two parts. One part is optimization variables such as $\mathbf{P}$, $\mathbf{Q}$, $\boldsymbol{\alpha}$ and $\mathbf{U}^{(v)}$ which need to optimize in each iteration. The other part is hyper-parameters such as λ_1, λ_2, p and c which need to be determined before the iteration of SLBS.

Since every hyper-parameter has its unique impact, we can determine some hyper-parameters empirically by the previous researches. For our model, we will list that how each hyper-parameter determines. As for p, it is designed to guarantee the sparsity of $\mathbf{P}$ and $\mathbf{Q}$. The best sparse approximation is $p = 0$, but it is a NP-hard problem and hard to be solved. Most papers set $p = 1$ because it is the optimal convex approximation. We set $p = 1/2$ rather than $p = 1$ to get a more accurate sparse approximation to $p = 0$. As for c, it is designed to measure the importance of views. Different c can change the weight of views.

As for the parameter λ_1 and λ_2, it is very important for the final performance since they are employed to balance the importance of loss function, sparse representation and regularization factor. Since there is no prior information about these parameters, we change them in $[10^{-5}, 10]$. We will provide some experimental results in the next section.

5. Experiment

In this section, we design six groups of experiments. The first group contains the clustering results on six multi-view datasets. The second group is to show some results about convergence behavior. The third group compares some results with different parameters. The fourth group shows the time cost of compared methods. The fifth group shows the visualization of row sparse matrix **P** and column sparse matrix **Q** and the last group shows the robust analysis.

5.1. Data set description and evaluation metric

We select six standard datasets with different features to evaluate the effectiveness of our method. In many real applications, six multi-view datasets are commonly used.

MSRC_v1 data set consists of 240 images and is divided into 8 classes. According to Lee and Grauman [31], we select 7 classes composed of face, tree, airplane, building, bicycle, cow, car and each class has 30 images. To distinguish all scenes, we extract 256 Local Binary Pattern(LBP), 48 Color Moment(CMT), 1302 Centrist, 100 Histogram of Oriented Gradient(HOG), 512 GIST and 200 SIFT features.

Handwritten numerals(HW) data set is composed of 2000 data points for 0 to 9 ten digit classes and each class has 200 data points. Six published features can be used for clustering: 47 Zernike moment (ZER), 240 pixel averages in 2 $\times$ 3 windows (PIX), 64 Karhunen-love coefficients (KAR), 216 profile correlations (FAC), 76 Fourier coefficients of the character shapes (FOU) and 6 morphological (MOR) features.

Caltech_7 contains 8677 images which belong to 101 categories. According to [32], we choose the widely used 7 classes, i.e., Faces, Dolla-Bill, Snoopy, Windsor-Chair, Motorbikes, Garfield and Stop-Sign. We sample the data and we choose 441 images as Caltech-7 totally. We extract the same visual features: LBP, HOG, GIST, CMT, CENTRIST and SIFT.

Scene15 is composed of 4485 images belonging to 15 categories: highway, inside of cities, tall buildings, streets, suburb residence, forest, coast, mountain, open country, bedroom, kitchen, livingroom, office, industrial and store. Six visual features are extracted: SIFT, SURF, PHOG, LBP, GIST and wavelet texture (WT).

MvYale contains 165 images of 15 volunteers and everyone has fifteen images. This data set is a gray-scale data set and mainly includes changes in lighting, facial expressions and posture. To analyze data effectively, five visual features are extracted: SIFT, HOG, LBP, WT and GIST.

KSA includes four subjects performing five actions. Here, each action is regarded as a class. We select 2000 video frames from each action, forming a subset of 10,000 examples. We use four pose features, *i.e.*, $f_{\text{JJ\_d}}$, $f_{\text{JL\_d}}$, $f_{\text{LL\_a}}$ and $f_{\text{PP\_a}}$ [33] extracted in [34].

5.2. Experiment setup

Since we focus on unsupervised learning, we compare SLBS with some unsurprised methods and use K-means clustering method to evaluate the effectiveness of our method. We list the compared methods as follows.

- **Best Single View(BSV)**: We use the proposed approach SLBS on each single view and employ K-means on low-dimensional representations to obtain clustering results. The best results on these views will be reported.
- **Group Structured Sparsity Matrix Factorization(GSSMF)**: An approach which learns a latent space factorized into dimensions by structure sparse proposed in [24].
- **Multi-View Spectral Embedding (MSE)**: One multi-view graph learning method proposed in [35].
- **Multi-View Nonnegative Matrix Factorization (MVNMF)**: The NMF-based multi-view clustering method proposed in [29].
- **Auto-Weighted Multiple Graph Learning(AMGL)**: Another multi-view graph learning method proposed in [30].
- **Large-scale Multi-view Subspace Clustering(LMVSC)**: A novel multi-view subspace learning method proposed in [36] with linear time.

For the experimental results, two different metrics clustering accuracy(ACC) and normalized mutual information(NMI) are employed to evaluate the performances of our proposed methods SLBS for clustering.

5.3. Comparison between SLBS and other algorithms

Tables 3 and 4 show the NMI and ACC results of all the comparative methods with different dimensions of feature extraction respectively. In terms of the clustering results, we have the following observations:

(1) Compared with other feature extraction methods of the metric ACC, SLBS performs best in most cases on most datasets. Specifically, SLBS exceeds the best ACC of other methods by nearly 10% on the Scene15 dataset and exceeds average ACC of other methods by more than 5% in most datasets. And we can get similar conclusions for the other metric NMI. This experimental results can prove the validity of our method.
(2) As for the comparison between our best single view method(BSV) and previous multi-view approaches, the latter does not always perform better. This may be caused by the fact that previous methods characterize the structures of each view data separately and combine them by simple addition operations. But compared with the weighted multi-view methods, the performance of our best single view method is also better sometimes. This is may be caused by the bidirectional sparsity of the unified subspace which plays an essential role.
(3) As is well-known, graph-based multi-view feature selection approaches perform very well when the dimension of selected features is same as the number of data categories. We also compared our method with graph-based approaches and find that our method is better than the best result of them in most cases. This effectively proves the validity of our algorithm.
(4) By analyzing the ACC and NMI of methods based on matrix factorization, we can obtain that the ACC and NMI increase first and decrease later with the increase of dimension. It is consistent with intuition. Since as dimension increasing, the learned subspace is closer to the intrinsic dimension first and redundant features appear when the dimension of the subspace is more than the intrinsic dimension. But our method also performs well with high dimensions in most datasets. This proves the validity of bidirectional sparsity in SLBS.

5.4. Convergence behavior

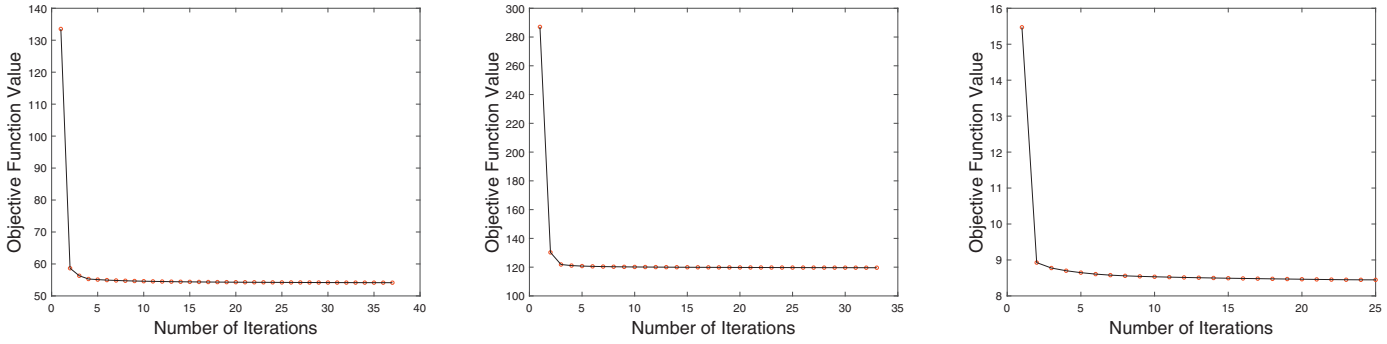
We draw the iterative convergence curves of MSRC_v1, Caltech_7 and HW datasets in Fig. 1 for verifying the convergence of our algorithm SLBS. By analyzing the iterative convergence curves, our algorithm is non-increasing during the iterations and converges to a definite value gradually. In addition, the algorithm con-

Table 3
ACC of different methods on six data sets with different dimensions of subspaces.

Dataset	*r*	BSV	GSSMF	MVNMF	MSE	AMGL	LMVSC	SLBS
Scene15	15	0.3698	0.4104	0.4142	0.2441	0.2526	0.4049	**0.4947**
	19	0.4125	0.4390	0.4470	0.2474	0.2637	0.4127	**0.5010**
	23	0.4341	0.4178	0.4093	0.2363	0.2972	0.4118	**0.5001**
	27	0.4357	0.4265	0.3991	0.2566	0.2956	0.4285	**0.4934**
	31	0.4140	0.4359	0.4341	0.2557	0.3152	0.4123	**0.5346**
MvYale	15	0.6303	0.6424	0.5535	0.6969	0.5575	0.3879	**0.7333**
	19	0.5818	0.6787	0.5757	0.6484	0.6060	0.3636	**0.7212**
	23	0.5515	0.6727	0.6141	0.6060	0.5515	0.3758	**0.6787**
	27	0.5333	0.6000	0.6020	0.5212	0.4424	0.3576	**0.6424**
	31	0.5393	**0.6303**	0.6020	0.5151	0.4303	0.3576	**0.6303**
MSRC_v1	7	0.6904	0.8143	0.7690	0.8761	0.9095	0.7714	**0.9190**
	10	0.7238	0.8000	0.7290	0.6476	0.6428	0.8048	**0.9381**
	15	0.6380	0.9095	0.6566	0.5476	0.5142	0.8143	**0.9619**
	20	0.6571	0.9333	0.6222	0.4619	0.4952	0.8143	**0.9810**
	25	0.7523	0.9333	0.5190	0.3476	0.4952	0.8190	**0.9857**
HW	10	0.7685	0.7765	0.7928	0.8367	0.8177	0.8560	**0.9350**
	15	0.8240	0.9020	0.8995	0.6085	0.6540	0.8425	**0.9295**
	20	0.7770	0.8920	0.8708	0.5615	0.5520	0.8445	**0.9410**
	25	0.8195	0.8905	0.8785	0.5400	0.6035	0.8315	**0.9360**
	30	0.8250	0.9020	0.9196	0.5250	0.5105	0.8900	**0.9475**
KSA	5	0.5790	0.6395	0.5287	0.4846	0.5060	0.5754	**0.6916**
	8	0.5660	0.6534	0.6590	0.4384	0.4599	0.5829	**0.7231**
	11	0.6005	0.6521	0.6284	0.4384	0.4301	0.5841	**0.7295**
	14	0.5964	0.6479	0.6740	0.4230	0.4065	0.6832	**0.7354**
	17	0.6344	0.6487	0.6632	0.3769	0.3932	0.6866	**0.7307**
Caltech_7	7	0.6825	0.7188	0.6386	0.5154	0.5736	0.7256	**0.7710**
	10	0.6916	0.7074	0.5993	0.5474	0.5895	**0.7324**	0.7234
	15	0.7256	0.7029	0.6213	0.5743	0.5555	**0.7347**	0.7324
	20	0.6780	0.6825	0.5661	0.5092	0.4807	0.7438	**0.7687**
	25	0.6439	0.7074	0.5684	0.4671	0.4965	0.7120	**0.7256**

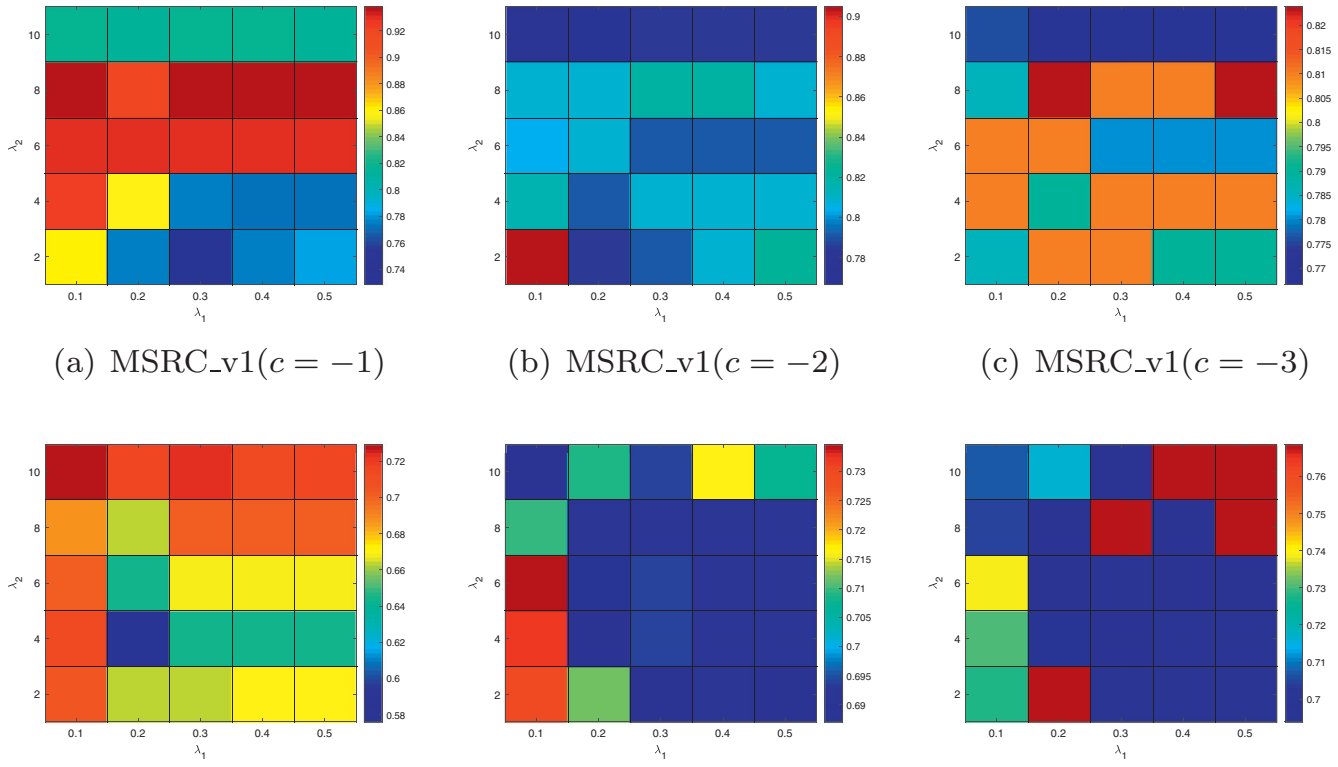
Table 4
NMI of different methods on six data sets with different dimensions of subspaces.

Dataset	*r*	BSV	GSSMF	MVNMF	MSE	AMGL	LMVSC	SLBS
Scene15	15	0.4015	0.4004	0.4334	0.3388	0.3494	0.3607	**0.5058**
	19	0.4255	0.4274	0.4360	0.3184	0.3595	0.3621	**0.5068**
	23	0.4266	0.4191	0.4438	0.3093	0.4159	0.3708	**0.5081**
	27	0.4255	0.4182	0.4391	0.3332	0.4460	0.3812	**0.5103**
	31	0.4214	0.4378	0.4336	0.3290	0.4397	0.3773	**0.5254**
MvYale	15	0.6623	0.7032	0.6049	**0.7476**	0.6626	0.3993	0.7267
	19	0.6621	0.7017	0.6366	0.7092	0.6587	0.3869	**0.7331**
	23	0.6312	0.7132	0.6591	0.6817	0.6565	0.3788	**0.7169**
	27	0.5953	0.6571	0.6594	0.6313	0.5700	0.4003	**0.7111**
	31	0.5994	0.6545	0.6683	0.6111	0.5619	0.3637	**0.6894**
MSRC_v1	7	0.6071	0.7648	0.7223	0.8097	0.8499	0.6970	**0.8533**
	10	0.5845	0.7557	0.7195	0.6039	0.6450	0.7305	**0.8839**
	15	0.5471	0.8554	0.6341	0.5224	0.5300	0.7198	**0.9146**
	20	0.5763	0.8777	0.6088	0.3848	0.4414	0.7018	**0.9570**
	25	0.6757	0.8764	0.5494	0.2978	0.4414	0.7229	**0.9677**
HW	10	0.7162	0.7440	0.7523	0.7370	0.7455	0.7512	**0.8778**
	15	0.7419	0.8204	0.8114	0.7629	0.7459	0.7415	**0.8670**
	20	0.7357	0.8150	0.8077	0.6745	0.6616	0.7537	**0.8823**
	25	0.7556	0.8118	0.8228	0.6256	0.6761	0.7157	**0.8775**
	30	0.7523	0.8204	0.8446	0.5865	0.5826	0.8077	**0.8947**
KSA	5	0.4050	0.4375	0.3353	0.4489	0.4539	0.3558	**0.4936**
	8	0.3840	0.4634	0.4029	0.4622	0.4753	0.3567	**0.5403**
	11	0.3898	0.4646	0.4274	0.4390	0.4356	0.3954	**0.5311**
	14	0.3920	0.4615	0.4491	0.3712	0.4008	0.4388	**0.5462**
	17	0.3976	0.4633	0.4530	0.3798	0.3905	0.4396	**0.5459**
Caltech_7	7	0.6412	0.6839	0.5368	0.4933	0.6336	0.6766	**0.7237**
	10	0.6553	0.6968	0.5647	0.5846	0.5752	0.7018	**0.7444**
	15	0.6386	0.6716	0.5935	0.4868	0.5048	0.6494	**0.7558**
	20	0.6271	0.6742	0.5379	0.4700	0.4736	0.6976	**0.7396**
	25	0.6417	0.6784	0.5366	0.3901	0.3785	0.6495	**0.7518**



(a) MSRC_v1 (b) Caltech_7 (c) Handwritten

Fig. 1. Objective values of SLBS with different numbers of iterations.



(a) MSRC_v1($c = -1$) (b) MSRC_v1($c = -2$) (c) MSRC_v1($c = -3$)

(d) Caltech_7($c = -1$) (e) Caltech_7($c = -2$) (f) Caltech_7($c = -3$)

Fig. 2. Sensitivity analysis on λ_1 and λ_2 with different parameter c.

verges well with about 10 iterations and thus fast convergence speed is an advantage of our method.

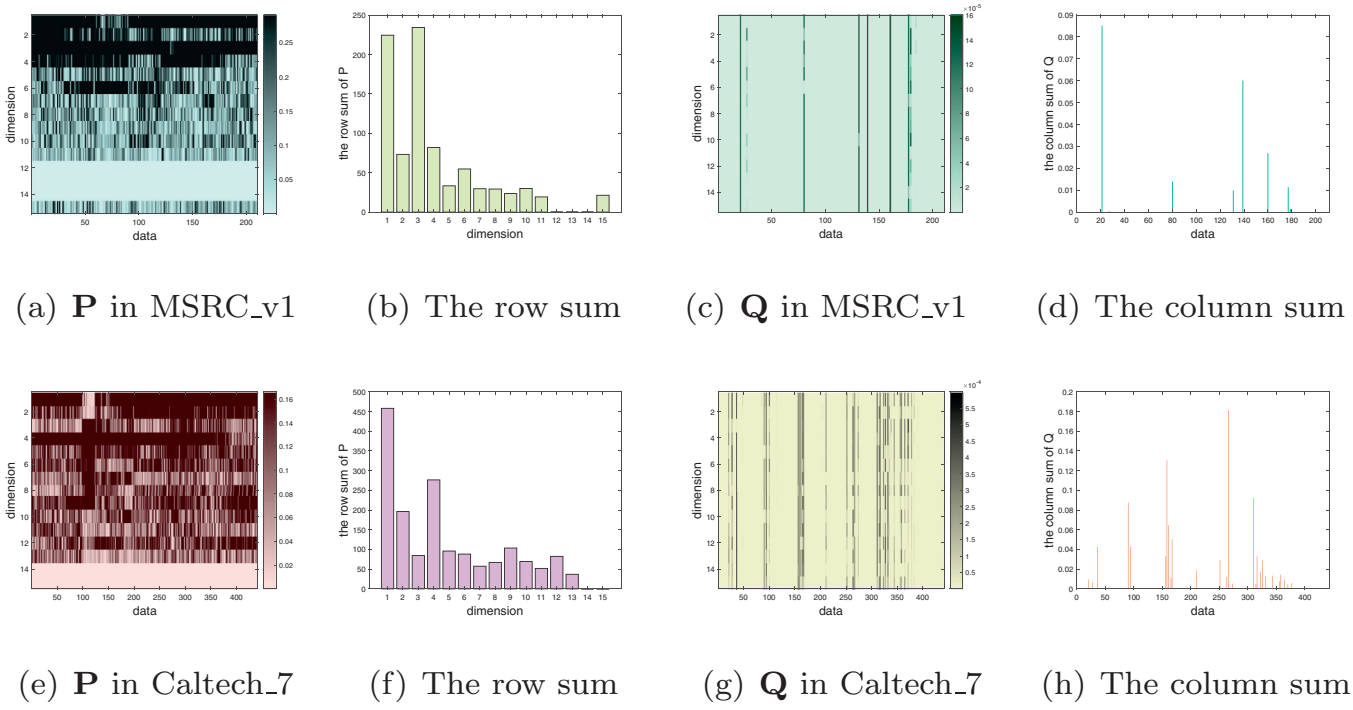
5.5. Computational time and parameter determination

To demonstrate the efficiency of our method for computational time, we compare the runtime on six data sets which have different data and feature scales in Table 6. Since all methods need a K-means algorithm for clustering and the focus of attention is the process of learning subspace, we do not report the computational time of the clustering algorithm.

We compare the time complexity of SLBS with other methods about the data size n in Table 5. From Table 6, we can find that our method costs the least time on six datasets. We discuss the essential reasons for the experimental results. (1) The computational time of graph-based methods is impacted by the number of data n most because it needs to decompose the n-dimensional data matrix and the computational complexity is $O(n^3)$. Our method only has linear complexity for the same case. Thus as the size of

Table 5
Computational complexity with respect to the data size n of all methods.

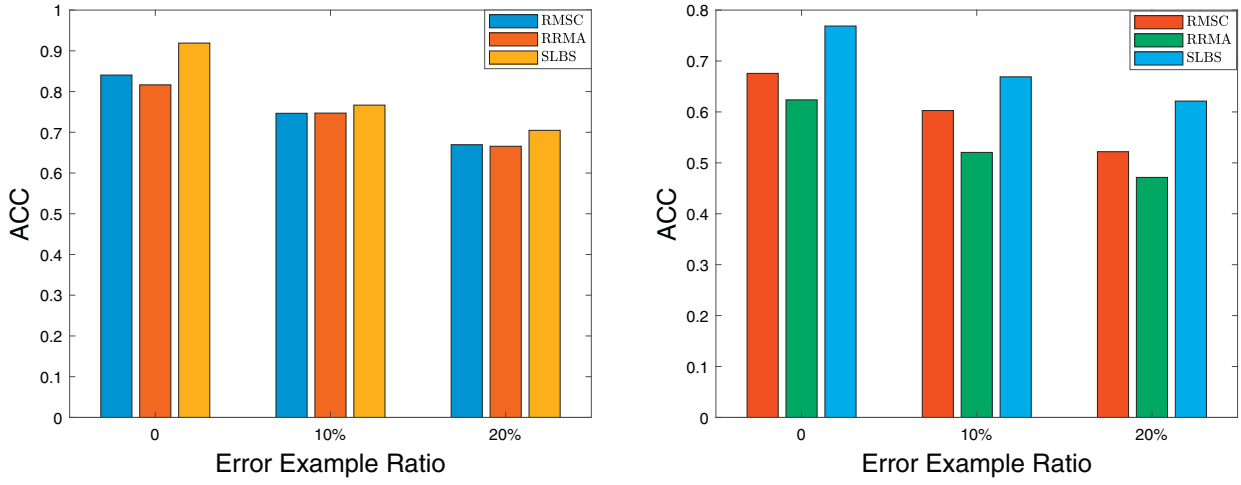
Method	Computational complexity
GSSMF	$\mathcal{O}(n)$
MVNMF	$\mathcal{O}(n)$
MSE	$\mathcal{O}(n^3)$
AMGL	$\mathcal{O}(n^3)$
LMVSC	$\mathcal{O}(n)$
SLBS	$\mathcal{O}(n)$



(a) **P** in MSRC_v1 (b) The row sum (c) **Q** in MSRC_v1 (d) The column sum

(e) **P** in Caltech_7 (f) The row sum (g) **Q** in Caltech_7 (h) The column sum

Fig. 3. The visualization of row sparse matrix **P** and column sparse matrix **Q** in two datasets.



(a) MSRC_v1 (b) Caltech_7

Fig. 4. Robust analysis with different error example ratios in two datasets.

Table 6
Average runtime comparison (seconds) on 6 datasets.

Dataset	GSSMF	MVNMF	MSE	AMGL	LMVSC	SLBS
MSRC_v1	0.2256	0.5308	0.3713	0.1721	0.2187	0.1237
Caltech_7	0.4360	0.9858	6.0967	0.7399	0.3628	0.2367
HW	0.7686	2.2252	118.4874	47.8417	1.0367	0.3460
MvYale	0.1935	0.4990	0.1788	0.1135	0.1502	0.0735
Scene15	7.2818	11.6786	552.3268	437.6027	3.3847	2.7510
KSA	1.7907	7.1253	4424.311	10145.02	5.5009	1.0249

the data increasing, our method is much faster than graph-based methods. (2) Even though MVNMF, GSSMF and LMVSC have the same computational complexity as our method, our method is still faster than them. By analyzing we find that our method has a faster convergence rate. For example, our method needs 35 iterations while MVNMF and GSSMF need more than 50 iterations on MSRC_v1.

As for the parameter determination problem, we design experiments on two data sets, i.e. MSRC_v1 and Caltech_7. Since we still need to determine three parameters, three experiments are performed on one data set and the difference of them is the value of c. We vary parameter c from {-1, -2, -3}. When the parameter c is fixed, we vary λ_1 from {0.1, 0.2, 0.3, 0.4, 0.5} and λ_2 from {2, 4, 6, 8, 10}. The dimension of the studied subspace is set to 10 and ACC was selected as the evaluation index. The results are shown in Fig. 2.

As seen from Fig. 2, the result shows that different parameters cause different results. This proves the importance of parameter selection. We find the results of two sets perform well when the

parameter c is equal to -1. It may be caused by the fact that the large c indicates the more attention to the diversity of different views. Besides, the optimal solutions of two data sets have different λ_1 and λ_2 since these two data sets have different data characteristics.

5.6. The visualization of common representation

We give the visualization of row sparse matrix **P** and column sparse matrix **Q** in Fig. 3. From Fig. 3, We can see that the row sum of **P** has several small values and the column sum of **Q** has several larger values. This is reasonable because most of the features after the first dimension reduction are useful and there are not lots of outliers in the datasets. The experimental results show the effectiveness of our algorithm.

5.7. Robust analysis

To analyze the robustness of our model, we compare our method SLBS with two robust multi-view subspace learning methods called RMSC [22] and RRMA [19] on two datasets. We have designed three error data ratios, which are 0, 10% and 20%. As shown in Fig. 4, with the increase of the number of error data, the performance of three methods tends to be decreased, but SLBS always performs best when compared with RMSC and RRMA. This fully demonstrates the robustness of our algorithm.

6. Conclusion

In this paper, we have proposed an unsupervised subspace learning method named as SLBS for multi-view data to learn a unified subspace representation. Specially, we analyze the structure of the common representation by the sparsity of both features and data. We further discuss the convergence and computational complexity of our algorithm, both theory and experiment results prove the effectiveness of SLBS. Finally, we verify the sparsity of the common representation and the robustness of our model in the experiment.

The convergence analysis can guarantee that the value of our objective function is non-increasing. Thus the first further work is how to design a new algorithm to obtain the location solution or even the optimal solution under the guarantee of theory. The second future work is to apply this bidirectional sparsity to other applications, such as surprised multi-view learning and multi-task learning.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the NSF of China under Grant 61922087 and Grant 61906201, and the NSF for Distinguished Young Scholars of Hunan Province under Grant 2019JJ20020. Chenping Hou is the corresponding author.

References

[1] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[2] C.R. Wang, J.J. Lien, Adaboost learning for human detection based on histograms of oriented gradients, in: ACCV, 2007, pp. 885–895.
[3] T. Ojala, M. Pietikainen, T. Maenpaa, Gray scale and rotation invariant texture classification with local binary patterns, in: ECCV, 2000, pp. 404–420.
[4] A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope, Int. J. Comput. Vis. 42 (3) (2001) 145–175.
[5] L. Ling, H.Y. Chao, C.D. Wang, Multi-view intact space clustering, in: ACPR, 2017, pp. 500–505.
[6] S. Liu, W. Cai, S. Liu, S. Pujol, R. Kikinis, D. Feng, Subject-centered multi-view feature fusion for neuroimaging retrieval and classification, in: ICIP, 2015, pp. 2505–2509.
[7] Y. Feng, J. Xiao, Y. Zhuang, X. Liu, Adaptive unsupervised multi-view feature selection for visual concept recognition, in: ACCV, 2012, pp. 343–357.
[8] X. Zhao, N.W.D. Evans, J.L. Dugelay, A subspace co-training framework for multi-view clustering, Pattern Recognit. Lett. 41 (1) (2014) 73–82.
[9] Y. Zhu, X. Jing, Q. Wang, W. Fei, F. Hui, S. Wu, Multi-view sparse embedding analysis based image feature extraction and classification, in: CCCV, 2015, pp. 51–60.
[10] J. Yu, D. Tao, Y. Rui, J. Cheng, Pairwise constraints based multiview features fusion for scene classification, Pattern Recognit. 46 (2) (2013) 483–496.
[11] W. Zhuge, F. Nie, C. Hou, D. Yi, Unsupervised single and multiple views feature extraction with structured graph, IEEE Trans. Knowl. Data Eng. 29 (10) (2017) 2347–2359.
[12] C. Hou, F. Nie, T. Hong, D. Yi, Multi-view unsupervised feature selection with adaptive similarity and view weight, IEEE Trans. Knowl. Data Eng. 29 (9) (2017) 1998–2011.
[13] Z. Kang, G. Shi, S. Huang, W. Chen, X. Pu, J.T. Zhou, Z. Xu, Multi-graph fusion for multi-view spectral clustering, Knowl. Based Syst. 189 (2020) 105102.
[14] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, in: NIPS, 2001, pp. 849–856.
[15] C. Zhang, Q. Hu, H. Fu, P. Zhu, X. Cao, Latent multi-view subspace clustering, in: CVPR, 2017, pp. 4333–4341.
[16] C. Zhang, H. Fu, Q. Hu, X. Cao, Y. Xie, D. Tao, D. Xu, Generalized latent multi-view subspace clustering, IEEE Trans. Pattern Anal. Mach. Intell. 42 (1) (2020) 86–99.
[17] Z. Kang, X. Zhao, C. Peng, H. Zhu, J.T. Zhou, X. Peng, W. Chen, Z. Xu, Partition level multiview subspace clustering, Neural Netw. 122 (2020) 279–288.
[18] Z. Kang, C. Peng, Q. Cheng, Robust subspace clustering via smoothed rank approximation, IEEE Signal Process. Lett. 22 (11) (2015) 2088–2092.
[19] J. Pu, Q. Zhang, L. Zhang, B. Du, J. You, Multiview clustering based on robust and regularized matrix approximation, in: ICPR, 2016, pp. 2550–2555.
[20] P. Ren, Y. Xiao, P. Xu, J. Guo, X. Chen, X. Wang, D. Fang, Robust auto-weighted multi-view clustering, in: IJCAI, 2018, pp. 2644–2650.
[21] Z. Kang, H. Pan, S.C.H. Hoi, Z. Xu, Robust graph learning from noisy data, IEEE Trans. Syst. Man Cybern. (2019) 1–11.
[22] R. Xia, Y. Pan, L. Du, J. Yin, Robust multi-view spectral clustering via low-rank and sparse decomposition, in: AAAI, 2014, pp. 2149–2155.
[23] M. Najafi, L. He, P.S. Yu, Error-robust multi-view clustering, in: Big Data, 2017, pp. 736–745.
[24] Y. Jia, M. Salzmann, T. Darrell, Factorized latent spaces with structured sparsity, in: NIPS, 2010, pp. 982–990.
[25] H. Lee, A. Battle, R. Raina, A.Y. Ng, Efficient sparse coding algorithms, in: NIPS, 2006, pp. 801–808.
[26] S. Bengio, F. Pereira, Y. Singer, D. Strelow, Group sparse coding, in: NIPS, 2009, pp. 82–89.
[27] R. Jenatton, G. Obozinski, F. Bach, Structured sparse principal component analysis, J. Mach. Learn. Res. 9 (2) (2009) 131–160.
[28] Y. Ming, L. Yi, Model selection and estimation in regression with grouped variables, J. R. Stat. Soc. 68 (1) (2006) 49–67.
[29] J. Gao, J. Han, J. Liu, C. Wang, Multi-view clustering via joint nonnegative matrix factorization, in: SDM, 2013, pp. 252–260.
[30] F. Nie, J. Li, X. Li, Parameter-free auto-weighted multiple graph learning: a framework for multiview clustering and semi-supervised classification, in: IJCAI, 2016, pp. 1881–1887.
[31] Y.J. Lee, K. Grauman, Foreground focus: unsupervised learning from partially matching images, Int. J. Comput. Vis. 85 (2) (2009) 143–166.
[32] D. Dueck, B.J. Frey, Non-metric affinity propagation for unsupervised image categorization, in: ICCV, 2007, pp. 1–8.
[33] C. Chen, Y. Zhuang, F. Nie, Y. Yang, F. Wu, J. Xiao, Learning a 3d human pose distance metric from geometric pose descriptor, IEEE Trans. Vis. Comput. Graph. 17 (11) (2011) 1676–1689.
[34] Z. Ma, Y. Yang, F. Nie, N. Sebe, S. Yan, A.G. Hauptmann, Harnessing lab knowledge for real-world action recognition, Int. J. Comput. Vis. 109 (1) (2014) 60–73.
[35] T. Xia, D. Tao, T. Mei, Y. Zhang, Multiview spectral embedding, IEEE Trans. Syst. Man Cybern. 40 (6) (2010) 1438–1446.
[36] Z. Kang, W. Zhou, Z. Zhao, J. Shao, M. Han, Z. Xu, Large-scale multi-view subspace clustering in linear time, AAAI, 2020.

Ruidong Fan is a master degree candidate at the National University of Defense Technology, Changsha, China. He received the B.S. degree from the Lanzhou University in 2018. His research interests include data mining, optimization and machine learning.

Tingjin Luo received the Ph.D. degree in College of Science from National University of Defense Technology, Changsha, China, in 2018. He has received the B.S. and M.S. from the College of Information System and Management, National University of Defense Technology, Changsha, China, in 2011 and 2013, respectively. He was a visiting Ph.D. student with the University of Michigan, Ann Arbor, MI, USA, from

2015 to 2017. He is currently an Assistant Professor with the College of Science, National University of Defense Technology. He has authored several papers in journals and conferences, such as IEEE TKDE, IEEE TCYB, IEEE TIP, Scientific Reports and KDD. His current research interests include machine learning, multimedia analysis, optimization, and computer vision.

Wenzhang Zhuge is a Ph.D. condidate at the National University of Denfense Technology, Changsha, China. He received the B.S. degree from Shandong University, Jinan, China, in 2015 and the M.S. degree from the National University of Defense Technology, Changsha, China in 2017. His research interests include machine learning, system science and data mining.

Sheng Qiang received the Ph.D. degrees from the National University of Defense Technology, Changsha, China in 2011. He has received the B.S. and M.S degrees with the same university. He has authored 20+ peer-reviewed papers in journals and conferences. His current research interests include machine learning, data mining, and computer vision.

Chenping Hou received the Ph.D. degrees from the National University of Defense Technology, Changsha, China in 2009. He is currently a full Professor with the Department of Systems Science of the same university. He has authored 80+ peer-reviewed papers in journals and conferences, such as the IEEE TPAMI, TNNLS/TNN, IEEE TSMCB/TCB, IEEE TIP, the IJCAI and AAAI. His current research interests include machine learning, data mining, and computer vision.