Low-rank graph optimization for multi-view dimensionality reduction

Youcheng Qian1,2, Xueyan Yin3, Jun Kong4, Jianzhong Wang4*, Wei Gao1*

1 Key Laboratory for Applied Statistics of MOE, School of Mathematics and Statistics, Northeast Normal University, Changchun, Jilin, China, 2 School of Science, Jilin Institute of Chemical Technology, Jilin, China, 3 School of Computer Science and Technology, Dalian University of Technology, Dalian, Liaoning, China, 4 School of Information Science and Technology, Northeast Normal University, Changchun, Jilin, China

* wangjz2019@nenu.edu.cn (JW); gaow@nenu.edu.cn (WG)

Abstract

Graph-based dimensionality reduction methods have attracted substantial attention due to their successful applications in many tasks, including classification and clustering. However, most classical graph-based dimensionality reduction approaches are only applied to data from one view. Hence, combining information from different data views has attracted considerable attention in the literature. Although various multi-view graph-based dimensionality reduction algorithms have been proposed, the graph construction strategies utilized in them do not adequately take noise and different importance of multiple views into account, which may degrade their performance. In this paper, we propose a novel algorithm, namely, Low-Rank Graph Optimization for Multi-View Dimensionality Reduction (LRGO-MVDR), that overcomes these limitations. First, we construct a low-rank shared matrix and a sparse error matrix from the graph that corresponds to each view for capturing potential noise. Second, an adaptive nonnegative weight vector is learned to explore complementarity among views. Moreover, an effective optimization procedure based on the Alternating Direction Method of Multipliers scheme is utilized. Extensive experiments are carried out to evaluate the effectiveness of the proposed algorithm. The experimental results demonstrate that the proposed LRGO-MVDR algorithm outperforms related methods.

Introduction

In real-world applications, data can be represented by heterogeneous features. For example, images are often described by various types of features such as color, texture and shape. Web pages can also be represented in different styles, in which the text content and hyperlink information are two types of features. In these cases, each type of feature can be regarded as a particular view of data. Data represented by multiple types of features is referred to as multi-view data. In multi-view data, each individual view has its specific meaning to describe a particular aspect of data that cannot be reflected by other views. However, different views also hold potential connection. Therefore, multiple views can be used to provide a complementary description of the data [1, 2]. Although the multi-view features are beneficial for distinguishing the samples of various classes, the feature vector of each view usually lies in a high-dimensional feature...
space and the combination of multi-view features typically results in the “curse of dimensionality” [3–5]. Therefore, it is essential to utilize dimensionality reduction technologies to reduce the feature redundancy among views while preserving most useful low-dimensional features.

In the past few decades, numerous dimensionality reduction methods have been developed for various tasks [2, 6–10]. These methods can be roughly categorized into two groups: linear and nonlinear [11, 12]. Linear dimensionality reduction methods map high-dimensional data to a suitable low-dimensional subspace via a linear projection. The classical linear dimensionality reduction methods are Principal Component Analysis (PCA) [13] and Linear Discriminant Analysis (LDA) [14]. The traditional linear dimensionality reduction algorithms are easy to implement because they only need to find an appropriate linear projection. However, the nonlinear structure of data cannot be exploited by them. Various nonlinear manifold-learning-based dimensionality reduction algorithms have been proposed to address this limitation, such as Locally Linear Embedding (LLE) [15], Isometric Feature Mapping (ISOMAP) [16], Laplacian Eigenmaps (LE) [17], Hessian Eigenmaps (HE) [18] and Local Tangent Space Alignment (LTSA) [19]. All the aforementioned dimensionality reduction algorithms are graph-based since they can be unified into a graph embedding framework [11]. However, these algorithms mainly focus on the features from a single view rather than multiple views.

A naive approach for multi-view feature embedding is simply to concatenate the features from various views as an input vector and apply dimensionality reduction methods on it directly. However, the concatenation strategy has several disadvantages: Firstly, the dimension of the concatenated vector is frequently high, which may cause the “curse of dimensionality” and lead to a high computational cost. Moreover, since each view describes a specific kind of information about the data, simply concatenating features of different views into a vector may not be physically meaningful. Furthermore, some research has shown that the performance of feature concatenation is worse than single view [20]. Recently, many algorithms have been proposed for constructing graphs by integrating information from multiple views. In [21], Xia et al. developed a new spectral embedding algorithm that learns a low-dimensional and sufficiently smooth embedding over all views simultaneously. In [20], Kumar and Daum applied co-training for multi-view data such that the graph in one view can be influenced the graph in the other view. In [22], Shu et al. constructed graphs separately from each view of the data and integrated multiple graphs by exploring higher-order information. In [23], Bisson and Grimal used multiple instances of an existing co-similarity approach to represent all the information in multi-view datasets simultaneously. By this way, their approach can transform the global data matrix into a graph. In [24], Tzortzis and Likas learned a weighted graph by minimizing the intra-cluster variance in the space induced by combining the individual graph of each view. In [25], De Sa et al. constructed a multipartite graph that was based on the minimizing-disagreement criterion [26]. In [27], Li et al. approximated the similarity graphs by using bipartite graphs. In [28], Zong et al. constructed a unified graph by learning local geometrical information in the original data space from multiple views simultaneously. However, none of the aforementioned algorithms has an explicit mechanism for handling potential noise and outliers in the data, which may degrade their performance.

Recently, some research showed that low-rank approximation is an effective method to remedy the influence of noise and outliers on graph [29]. As a result, many low-rank based algorithms have been developed to enhance the robustness of graphs to noise and outliers. In [30], Ye et al. proposed a robust late fusion method that decomposes each original graph from an individual model into a common rank-2 graph matrix and sparse deviation errors. In [31], Pan et al. proposed a robust rank aggregation algorithm that recovers a latent rank list from the possibly incomplete and noisy input rank lists. Similarly, Xia et al. [32] proposed a robust multi-view spectral clustering (RMSC) method that seeks a shared transition probability graph matrix with a low-rank constraint and a sparse error matrix. In [33], Hong et al. constructed a
hypergraph Laplacian matrix by integrating various low-rank affinity graphs. Although these approaches have achieved satisfactory performance in dealing with noise, they neglected the different importance of multiple views and use the same weight for all input graphs. In fact, different views often contribute unequally in practice. Therefore, one challenge is how to aggregate the strengths of various heterogeneous graphs by exploring the rich information among them, which certainly can lead to more accurate and robust performance than by treating each individual type of graph equally [34].

We propose a novel algorithm, Low-Rank Graph Optimization for Multi-View Dimensionality Reduction (LRGO-MVDR) to address the aforementioned shortcomings. Compared with the existing methods, the proposed algorithm possesses two advantages. First, unlike the methods which ignore the noise and outliers in the datasets, our approach possesses a mechanism for dealing with noise and outliers based on a low-rank constraint. Thus, the features that obtained in our algorithm are more stable and robust. Second, different from most of the approaches which treat all views equally, a nonnegative weight vector is introduced into our algorithm to combine graphs from various views and an adaptive strategy is provided for optimizing the elements in the vector. Furthermore, we propose an optimization procedure based on the Alternating Direction Method of Multipliers (ADMM) scheme [35] to solve our model. We employ a simulated dataset and six real-world datasets to verify the performance of the proposed algorithm on classification and clustering tasks. Experimental results indicate that the proposed LRGO-MVDR algorithm outperforms other related algorithms in terms of classification accuracy rate [36, 37], Clustering Accuracy and Normalized Mutual Information [37, 38].

The remaining of the paper is organized as follows: Section ‘Related Work’ briefly reviews three methods for multi-view graph construction. Section ‘Proposed Algorithm’ describes the details of our algorithm. Section ‘Optimizing Algorithm’ describes the optimization process. Section ‘Experiments’ presents the experimental results and comparisons. Finally, Section ‘Conclusion and Future Work’ presents the conclusions and future work of this paper.

Related work
An integral part of our method is the construction of an accurate and robust shared graph matrix by combining multiple graphs. In this section, three methods for multi-view graph construction will be reviewed.

Robust late fusion with rank minimization
In [30], Ye et al. proposed a rank-based fusion method that utilizes rank minimization and sparse error to fuse the predicted confidence scores of multiple models, each of which is obtained based on a specified view of data. Given m confidence score vectors obtained from m views, which are denoted as $s^{(1)}, s^{(2)}, \ldots, s^{(m)}$, where $s^{(i)} = [s^{(i)}_1, s^{(i)}_2, \ldots, s^{(i)}_n]$, in which n is the number of samples, the method first converts each confidence score vector into a pairwise relationship matrix $T^{(i)}$. Specifically, $T^{(i)}_{jk} = \text{sign}(s^{(j)}_k - s^{(k)}_j)$, i.e., $T^{(i)}_{jk} = 1$ if $s^{(j)}_k > s^{(k)}_j$, $T^{(i)}_{jk} = -1$ if $s^{(j)}_k < s^{(k)}_j$, and $T^{(i)}_{jk} = 0$ if $s^{(j)}_k = s^{(k)}_j$, where $s^{(j)}_k$ and $s^{(k)}_j$ denote the scores of the j-th and k-th samples, respectively. The matrix $T^{(i)}$ can encode the pairwise comparative relation of scores of every two samples under the i-th view. Each pairwise relationship matrix is combined with a shared graph matrix and an independent sparse residue matrix. The constrained optimization problem is formulated as follows:

$$\begin{align*}
\min_{T \in \mathbb{R}^{n \times n}} \text{rank}(\hat{T}) + \lambda \sum_{i=1}^{m} \|E^{(i)}\|_0, \\
\text{s.t. } T^{(i)} = \hat{T} + E^{(i)}, \hat{T} = -\hat{T}^T, i = 1, \ldots, m,
\end{align*}$$

(1)
where \( \text{rank}(\hat{T}) \) is the rank of \( \hat{T} \), the \( l_0 \)-norm \( \|E^0\|_0 \) represents the number of non-zero elements in \( E^0 \), and \( \lambda \) is a non-negative tradeoff parameter. In fact, \( l_0 \)-norm is not actually a norm, and the term ‘norm’ is used in this study for convenience [39].

Since the problem in Eq (1) is NP-hard, it is difficult to be solved. Recently, [40] proved that the nuclear norm function is the convex envelope of the rank function on the matrix unit sphere, so the nuclear norm is the best convex approximation of the rank function. More recently, it has been shown in [41–43] that the solution of the minimize rank can be obtained by solving the minimize nuclear problem. We also learned that the nuclear norm heuristic could produce very low-rank solutions in practice [40, 44] and corresponding theoretical characterization in [43]. In addition, \( l_1 \)-norm has been observed to be a good convex approximation to \( l_0 \)-norm [45]. Therefore, we could get a tractable optimization problem by minimizing the following convex optimization problem in Eq (2) instead of minimizing Eq (1). It is not difficult to see the minimization problems coincide each other with high probability,

\[
\min_{T, E^0} \|\hat{T}\|_s + \lambda \sum_{i=1}^m \|E^0\|_1, \\
\text{s.t. } T^0 = T + E^0, \hat{T} = -\hat{T}^T, i = 1, \ldots, m. 
\]

where the nuclear norm \( \|\hat{T}\|_s \) denotes the sum of the singular values of \( \hat{T} \), the \( l_1 \)-norm \( \|E^0\|_1 \) represents the error matrix that corresponds to the \( i \)-th view, and \( \hat{T} \) defines a shared graph matrix. The skew-symmetric constraint \( \hat{T} = -\hat{T}^T \) is employed to make the decomposed \( \hat{T} \) remain a pairwise comparative matrix. By minimizing the constrained optimization problem in Eq (2), a consistent shared graph among views can be discovered while overcoming the noise issues.

**Rank aggregation via low-rank and structured-sparse decomposition**

In [31], Pan et al. stated that the method in [30] requires the input confidence score vectors to be complete (with no missing values), which is rare in practice. To overcome this shortcoming, a Robust Rank Aggregation (RAA) algorithm, which can simultaneously deal with the possible noise and missing values in the individual confidence score vector, was proposed. Given \( m \) confidence score vectors \( s^{(1)}, s^{(2)}, \ldots, s^{(m)} \) obtained from \( m \) views and \( s^{(0)} = [s^{(0)}_1, s^{(0)}_2, \ldots, s^{(0)}_n] \) in which \( n \) is the number of samples, RAA first converts each confidence score into a comparison matrix, which is denoted as \( T^{(0)} \). Specifically, \( T^{(0)}_{jk} = \text{sign}(s^{(0)}_j - s^{(0)}_k) \) if \( s^{(0)}_j \) and \( s^{(0)}_k \) are observed, and \( T^{(0)}_{jk} = \text{unknown} \) if \( s^{(0)}_j \) or \( s^{(0)}_k \) is missing, where \( s^{(0)}_j \) and \( s^{(0)}_k \) denote the scores of the \( j \)-th and \( k \)-th samples in the \( i \)-th view. The constrained optimization problem of a relaxation of RAA is defined as:

\[
\min_{Z, E^0} \|Z\|_* + \lambda \sum_{i=1}^m \|E^0\|_{2,1}, \\
\text{s.t. } W^{(i)} \odot T^{(i)} = W^{(i)} \odot (Z + E^{(0)} - (E^{(0)})^T), i = 1, 2, \ldots, m, 
\]

where the \( l_{2,1} \)-norm regularization term \( \|E^0\|_{2,1} = \sum_{i=1}^n \sqrt{\sum_{j=1}^m (E^{(0)}_{jk})^2} \) encourages the column-sparsity in \( E^{(0)} \), \( \lambda \) is a non-negative tradeoff parameter, \( \odot \) denotes elementwise (Hadamard) multiplication, \( T^{(0)} \) is a comparison matrix for the \( i \)-th view, \( W^{(i)} \) is an indicator matrix that corresponds to the \( i \)-th view such that \( W^{(i)}_{jk} = 0 \) if \( T^{(0)}_{jk} \) is unknown or missing and \( W^{(i)}_{jk} = 1 \) otherwise, \( Z \) is the target graph matrix that encodes the true order relations among items, and \( E^{(0)} \) is used to encode the noise for the \( i \)-th view. From the definitions of \( W^{(i)} \) in Eq (3), it can be
found that the objective function of RRA will not affected by the unknown or missing values in \( T^0 \) since their corresponding elements in \( W^0 \) is zero. Thus, as analyzed in [31], the main advantage of RRA over the method in [30] is that it can handle missing values.

**Robust multi-view spectral clustering via low-rank and sparse decomposition**

Xia et al. [32] applied low-rank and sparse decomposition in a Markov chain and proposed an algorithm named Robust Multi-view Spectral Clustering via Low-rank and Sparse Decomposition (RMSC) for clustering. RMSC initially constructs a graph on each view and subsequently extracts a shared graph matrix using rank minimization. The constrained optimization problem of a relaxation of RMSC can be written as:

\[
\min_{\hat{P}, E^0} \|\hat{P}\|_1 + \lambda \sum_{i=1}^{m} \|E^0\|_1,
\]

\[
s.t. \quad P^0 = \hat{P} + E^0, \quad \hat{P} \geq 0, \quad \hat{P} 1 = 1, \quad i = 1, 2, \ldots, m,
\]

where \( \hat{P} \) denotes a column vector with all elements as 1, the \( l_1 \)-norm regularization term encourages the sparsity in \( E^0 \), \( \lambda \) is a non-negative tradeoff parameter, \( P^0 \) is a graph for the \( i \)-th view, \( \hat{P} \) is a low-rank shared graph matrix and \( E^0 \) is the error matrix for the \( i \)-th view. The constraints \( \hat{P} \geq 0, \hat{P} 1 = 1 \) are employed to make \( \hat{P} \) has a desired probability property, i.e., the optimal \( \hat{P} \) can be considered as a probability that the \( i \)-th and \( j \)-th samples are connected as a neighboring nodes in the graph [46, 47].

**Proposed algorithm**

The methods in [30–32] are all graph-based multi-view methods and can effectively deal with noise. However, all views are treated equally and the diversity among multi-view data is not explicitly considered in these methods [30–32]. Therefore, their performance may suffer from the graph of less informative views [34]. To overcome this limitation, Low-Rank Graph Optimization for Multi-View Dimensionality Reduction (LRGO-MVDR) is proposed in this section, which consists of the following two steps.

**Shared graph matrix construction**

First, the graph of each view is constructed. Here, we also suppose \( X = \{X^{(1)}, X^{(2)}, \ldots, X^{(m)}\} \) is a multi-view dataset, where \( m \) is the number of views. \( X^{(i)} = \{x^{(i)}_1, \ldots, x^{(i)}_n\} \in \mathbb{R}^{d \times n} \) is a set of \( n \) data points in the \( i \)-th view, which contains \( c \) classes. We define a similarity matrix \( A^{(i)} \in \mathbb{R}^{n \times n} \) to represent the similarity between instances of the \( i \)-th view. Let \( G^{(i)} = (V^{(i)}, E^{(i)}, A^{(i)}) \) be the weighted graph with vertex set \( V^{(i)} \), edge set \( E^{(i)} \), and similarity matrix \( A^{(i)} \). In general, the Gaussian kernel is used to define the similarity matrix:

\[
A^{(i)}_{jk} = \exp \left( -\frac{\|x^{(i)}_j - x^{(i)}_k\|_2^2}{\sigma^2} \right),
\]

where \( \|x^{(i)}_j - x^{(i)}_k\|_2^2 = (x^{(i)}_j - x^{(i)}_k)^T (x^{(i)}_j - x^{(i)}_k) \) denotes the \( l_2 \)-norm and \( \sigma^2 \) denotes average Euclidean distance over all pairs of data points. Let \( S^{(i)} \) be the normalization matrix of \( G^{(i)} \). We can define \( S^{(i)} \) as \( S^{(i)} = (D^{(i)})^{-1} A^{(i)} \), where \( D^{(i)} \) is a diagonal matrix with \( D^{(i)}_{ii} = d^{(i)}_i = \sum_{k=1}^{n} A^{(i)}_{jk} \) and \( d^{(i)}_i \) represents the degree of a vertex \( v^{(i)}_j \) for the \( i \)-th view. Fig 1 illustrates the framework of our proposed method for the shared graph construction. There are two primary assumptions.
of LRGO-MVDR: (1) The features in each view are adequate for exploring most useful information. (2) The features in each view may be corrupted by noise. Based on these assumptions, each normalization matrix $S^{(i)}$ when associated with an individual view can be naturally decomposed into two parts: a shared graph matrix $S$ that reflects the underlying true structural information and an error matrix $E^{(i)}$ that encodes the noise in each view.

$$\forall i, \ S^{(i)} = S + E^{(i)}.$$  \hspace{1cm} (6)

**Problem formulation**

To learn the final shared graph matrix $S$ from each $S^{(i)}$, we minimize the average disagreement between $S$ and the cleaned matrices, i.e., $S^{(i)} - E^{(i)}$. Moreover, the normalization matrix from each view must be similar to the shared graph matrix. This requirement demands that the error matrices be small and sparse. It is formulated as follows:

$$\min_{S, E^{(i)}} \sum_{i=1}^{m} \|S - (S^{(i)} - E^{(i)})\|_F^2 + \lambda \|E^{(i)}\|_1,$$

s.t. $S \geq 0, S1 = 1,$  \hspace{1cm} (7)

where $\lambda$ is a nonnegative tradeoff parameter. The $l_1$-norm is used to ensure the sparsity of $E^{(i)}$. The constraints $S \geq 0, S1 = 1$ enforce $S$ to be nonnegative and the sum of each row to be 1.
Ideally, the similarity between two samples from the same class is high, whereas the similarity between the samples from different classes is low or even nearly zero. Therefore, as can be seen in Fig 2A, the similarity matrix in the ideal situation is diagonal block and its rank is low. However, the noise in the real-world data may affect the similarity of samples, e.g. make the similarity between samples from different classes be high, which will corrupt the diagonal block of similarity matrix and increase its rank (Fig 2B). Thus, the nuclear norm is introduced in our model to make the shared graph matrix \( S \) to be low rank, which would remedy the possible noise in the similarity matrices associated with different views and increase the robustness of our proposed approach. In summary, we can obtain the following formulation:

\[
\min \gamma \| S \|_*, + \sum_{i=1}^{m} \| S - (S^{(i)} - E^{(i)}) \|_F^2 + \lambda \| E^{(i)} \|_1, \\
\text{s.t. } S \geq 0, S1 = 1,
\]

where \( \| \cdot \|_* \) denotes the nuclear norm of a matrix and \( \lambda \) is a nonnegative tradeoff parameter. Intuitively, no weight factor is explicitly defined in Eq (8) and all views are treated equally. Thus, the different importance of views is ignored. To further explore the complementarity and different importance of various views, a nonnegative weight vector denoted as \( \omega = [\omega_1, \omega_2, \ldots, \omega_m]^T \in \mathbb{R}^m \) is introduced in our approach, where \( \omega(i = 1, 2, \ldots, m) \) is the weight for the \( i \)-th view. Thus, the Tikhonov regularization \([48]\) term can be included and the final constrained optimization problem of our proposed LRGO-MVDR algorithm is:

\[
\min \gamma \| S \|_*, + \sum_{i=1}^{m} \omega_i (\| S - (S^{(i)} - E^{(i)}) \|_F^2 + \lambda \| E^{(i)} \|_1) + \beta \| \omega \|_2^2, \\
\text{s.t. } \omega_i \geq 0, \sum \omega_i = 1, S \geq 0, S1 = 1, i = 1, \ldots, m,
\]
where $\beta$ is a nonnegative tradeoff parameter. The $l_2$-norm regularization term $\|\omega\|^2_2$ is utilized to avoid overfitting of the weight vector $\omega$ to a single normalization matrix [49].

We can execute dimensionality reduction and obtain the low-dimensional representation by the shared graph matrix $S$, which will be discussed at Section 'Dimensionality reduction with the shared graph'.

**Optimization algorithm**

Since Eq (9) is not convex in all variables jointly, it is hardly to expect an algorithm to find its global optimal solution. However, due to Eq (9) is convex to in $S, E^{(i)}$ and $\omega_i$ individually, the subproblem of solving each variable in this equation is tractable. We solve the problem via the Alternating Direction Method of Multipliers (ADMM) [35] scheme, which has demonstrated a satisfactory balance between efficiency and accuracy in many matrix learning problems [30–32].

By introducing an auxiliary variable $Q$, we convert Eq (9) into the following equivalent form:

$$
\min_{S, E^{(i)}, \omega, Q} \gamma \|Q\|_2 + \sum_{i=1}^m \omega_i \left( \|S - (S^{(i)} - E^{(i)})\|^2_F + \lambda \|E^{(i)}\|_1 \right) + \beta \|\omega\|^2_2,
$$

s.t. $\omega_i \geq 0, \sum \omega_i = 1, S \geq 0, S1 = 1, S = Q, i = 1, \ldots, m.$

(10)

The corresponding augmented Lagrangian function of Eq (10) is:

$$
L(S, Q, E^{(i)}, \omega, \gamma) = \gamma \|Q\|_2 + \sum_{i=1}^m \omega_i \left( \|S - (S^{(i)} - E^{(i)})\|^2_F + \lambda \|E^{(i)}\|_1 \right) + \beta \|\omega\|^2_2 + \langle Y, S - Q \rangle + \frac{\mu}{2} \|S - Q\|^2_F,
$$

s.t. $\omega_i \geq 0, \sum \omega_i = 1, S \geq 0, S1 = 1, i = 1, \ldots, m,$

where $\langle P, S - Q \rangle = \text{tr}(P^T(S - Q))$ denotes the inner product of two matrices, $Y$ represents the Lagrange multiplier, and $\mu > 0$ is an adaptive penalty parameter.

Next we will present the update rules for each of $S, Q, E^{(i)}$ and $\omega_i$, which are obtained by minimizing $L$ in Eq (11) with the other variables fixed.

**Optimize $Q$ by fixing $S, E^{(i)}$ and $\omega_i$:** Through fixing $S, E^{(i)}$ and $\omega_i$, and removing the irrelevant terms, the optimization problem of $Q$ can be obtained as:

$$
\min_Q \gamma \|Q\|_2 + \frac{\mu}{2} \left\| S - Q + \frac{Y}{\mu} \right\|_F^2,
$$

which can be solved via the Singular Value Threshold method [50]. More specifically, let $U\Sigma V^T$ be the SVD form of $S+Y/\mu$, the solution of Eq (12) is as follows:

$$
Q = US_{\gamma/\mu}(\Sigma)V^T,
$$

(13)

where $S_{\gamma/\mu}(\Sigma) = \max(\Sigma - \gamma/\mu, 0) + \min(\Sigma + \gamma/\mu, 0)$ is the shrinkage operator [51].

**Optimize $E^{(i)}$ by fixing $Q, S$ and $\omega_i$:** Through fixing $Q, S$ and $\omega_i$, and removing the irrelevant terms, the optimization of $E^{(i)}$ becomes:

$$
\min_{E^{(i)}} \lambda \|E^{(i)}\|_1 + \|S - (S^{(i)} - E^{(i)})\|^2_F,
$$

which has a closed-form solution of $E^{(i)} = S_{\lambda}(S^{(i)} - S)$ according to [50] and $S_{\lambda}(S^{(i)} - S) = \max(S^{(i)} - S - \lambda, 0) + \min(S^{(i)} - S + \lambda, 0)$ is the shrinkage operator [51].
Optimize $S$ by fixing $Q$, $E^0$, and $\omega$: When $Q$, $E^0$, and $\omega$ are fixed, the optimization problem with respect to $S$ becomes:

$$
S = \arg\min_S \sum_{i=1}^n \omega_i \left(\|S - (S^0 - E^0)\|_1^2 + \frac{\mu}{2} \|S - Q + \frac{Y}{t}\|_2^2\right)
$$

$$
s.t.S \geq 0, SI_1 = 1.
$$

(15)

By simple algebraic formulation, we define:

$$
C = \sum_{i=1}^n \omega_i \left(\sum_{j=1}^m \omega_j (S^0 - E^0)\right) + \frac{\mu}{2} Q - \frac{Y}{2}.
$$

(16)

Then, Eq (15) is equivalent to

$$
S = \arg\min_S \left(\sum_{i=1}^n \omega_i \left(\|S - C\|_2^2 + \frac{\mu}{2} \|S - C\|_2^2\right)\right)
$$

$$
= \arg\min_S \left(\sum_{i=1}^n \omega_i \left(\sum_{j=1}^m |S_i - C_{ij}|\right)\right),
$$

$$
s.t. \sum_{j=1}^m S_{ij} = 1, S_{ij} \geq 0,
$$

(17)

where $S_i$ and $C_{ij}$ denote the $i$-th rows of the matrices $S$ and $C$, respectively. The problem in Eq (17) can be decomposed into $n$ independent subproblems:

$$
\min_{S_i} \left(\sum_{i=1}^n \omega_i \left(\|S_i - C_i\|_2^2\right)\right), s.t. \sum_{j=1}^m S_{ij} = 1, S_{ij} \geq 0.
$$

(18)

Each subproblem is a projection of $C_i$ onto the probability simplex, which can be efficiently solved via the projection algorithm in [52]. Here, we detail the algorithm in Algorithm 1.

**Algorithm 1** The projection of a vector onto the probability simplex.

Input: A vector $C_i \in \mathbb{R}^n$

Sort $C_i$ into $u$: $u_1 \geq u_2 \geq \cdots \geq u_n$

Find $j = \max\left\{ j : 1 - \sum_{k=1}^{j-1} (u_k - u_{j}) \geq 0 \right\}$

Let $\sigma = \left(\sum_{j=1}^n u_j - 1\right)/j$

Output: $S_{ij}$, where $S_{ij} = \max(C_{ij} - \sigma, 0), j = 1, 2, \ldots, n$

Optimize $\omega$ by fixing $S$, $Q$, $E^0$: When $S$, $Q$ and $E^0$ are fixed, we can rewrite Eq (11) with respect to $\omega$ as:

$$
\min_{\omega} q^T q + \beta \|\omega\|_1^2,
$$

$$
s.t. \omega_i \geq 0, \sum_i \omega_i = 1, i = 1, \ldots, m,
$$

(19)

where $q = [q_1, \ldots, q_m]^T$ and $q_i = \|S - (S^0 - E^0)\|_2^2 + \frac{\mu}{2} \|E^0\|_2^2$. In Eq (19), if $\beta = 0$, the trivial solution will be

$$
\omega_i = \begin{cases} 1, \text{ if } q_i = \min_{i=1,2,\ldots,m} q_i, \\ 0, \text{ otherwise.} \end{cases}
$$

(20)

Since this situation only considers the information of a single view, the latent complementary information of multi-view datasets is ignored, which results in excessively sparse and unsatisfactory results. Contrarily, if $\beta \to \infty$, a uniform weight will be assigned to all multi-view datasets and the diversity of views will be neglected. Therefore, the best aggregated graph corresponds to $\beta$ in the middle range value, i.e., a balance between averaged weighting and a single view.
The minimization in Eq (19) is a convex quadratic programming problem, which can be solved by the generic Lagrange multiplier and stochastic gradient decent methods [53]. However, for large-scale problems, these methods are usually time consuming and converge slowly. Thus, the Coordinate Descent Algorithm (CDA), which has very inexpensive iterations and can be easily parallelized, is employed in our algorithm to solve Eq (19) as suggested in [54–58]. The pseudocode of CDA is presented as Algorithm 2.

**Algorithm 2 Coordinate Descent Algorithm (CDA)**

1: Input: the tradeoff parameter $\beta$, the vector $q = [q_1, \ldots, q_m]$

2: Output: the nonnegative weight vector $\omega$

3: Initialize $\omega_i = 1/m, i = 1, 2, \ldots, m$

4: for $i = 1$ to $m$

5: for $j = 1$ to $m$ ($j \neq i$)

6: repeat

7: if $2\beta(\omega_i + \omega_j) + (q_j - q_i) \leq 0$ then

8: $\omega_i' = 0$, $\omega_j' = \omega_i + \omega_j$

9: else

10: if $2\beta(\omega_i + \omega_j) + (q_i - q_j) \leq 0$ then

11: $\omega_i' = 0$, $\omega_j' = \omega_i + \omega_j$

12: else

13: $\omega_i' = (2\beta(\omega_i + \omega_j) + (q_i - q_j))/(4\beta)$

14: $\omega_j' = \omega_i + \omega_j - \omega_i''$

15: end if

16: end if

17: until convergence

18: end for

19: end for

**Complexity analysis for LRGO-MVDR**

Our method LRGO-MVDR consists of four subproblems. The complexity of updating $Q$ depends on the SVD of $(S+Y/\mu)$ and three matrix multiplications for $Q = US_{\rho/\mu}(\Sigma)V^T$, which is $O(n^3 + n^3)$ [50], where $n$ is the sample number of data. For updating $E^{(i)}$ in Eq (14) is $O(n^2)$, and for all $m$ views, the complexity of $E$ is $O(mn^2)$. For updating $S$, according to [52], we know that the complexity of $S$ is $O(n^2 \log n)$. For updating $\omega$, the main complexity is $O(tm^2)$ via Algorithm 2 [56]. Overall, the total complexity is $O(n^3 + mn^2 + n^2 \log n + tm^2)$ for each iteration. Under the condition $m \ll n$ and $t$ is empirically set to 30, the total complexity is basically $O(Tn^3)$, where $T$ is the number of iterations.

**Dimensionality reduction with the shared graph**

Once we have obtained the shared graph matrix $S$, we can execute dimensionality reduction on it and obtain the low-dimensional representation. The constrained optimization problem is as follows:

$$
\frac{1}{2} \sum_i \|b_i - b_j\|^2 S_{ij} = \frac{1}{2} \sum_i \sum_{t=1}^r (b_i - b_{jt})^2 S_{ij} = \sum_{t=1}^r B_t L B_t^T = tr(BLB^T),
$$

where $B = [b_1, b_2, \ldots, b_n] \in \mathbb{R}^{r \times n}$ is the low-dimensional representation of $X^{(i)}, i = 1, 2, \ldots, m$. 
and $L$ is the Laplacian matrix, which can be obtained as follows:

$$L = \| - S,$$

(22)

where $\|$ denotes a diagonal matrix of the shared graph matrix whose entries are the column or row sums of $S$. By introducing the constraint $B\Pi B^T = I$, the final constrained optimization problem is as follows:

$$\arg\min_B tr(BLB^T),$$

s.t. $B\Pi B^T = I.$

(23)

Eq (23) can be transformed into a generalized eigenvalue problem as:

$$LB^T = \alpha QB^T,$$

(24)

Suppose $\alpha_1, \alpha_2, \ldots, \alpha_r$ are the $r$ smallest eigenvalues of Eq (24) and $b_1, b_2, \ldots, b_r$ are the corresponding eigenvectors. The optimal low-dimensional features obtained by our algorithm is given by:

$$B = [b_1, b_2, \ldots, b_r]^T.$$  

(25)

Finally, the overall optimization process of the LRGO-MVDR algorithm can be summarized in Algorithm 3. In Algorithm 3, the stop condition can be defined as convergence, i.e., $\|S - Q\|_\infty \leq \varepsilon (\varepsilon = 10^{-6})$, or a prespecified maximum iteration number $T_{\text{max}} (T_{\text{max}} = 300)$ is reached.

**Algorithm 3 LRGO-MVDR algorithm**

**Input:** the data $X^{(i)} (i = 1, 2, \ldots, m)$ and the parameters $\gamma$, $\lambda$, and $\beta$

**Output:** the shared graph matrix $S$, noise matrices $E^{(i)}$, weight vector $\omega$, and the low-dimensional features $B$.

**Initialize:** $S = 0$, $Q = 0$, $Y = 0$, $\mu = 10^{-3}$, $\rho = 1.1$, $\max_u = 10^{10}$, $\omega_i = 1/m$, $E^{(i)}$ is a random matrix.

Compute similarity matrices $A^{(i)}$ according to Eq (4) and compute the corresponding normalization matrices $S^{(i)}$ via $S^{(i)} = (D^{(i)})^{-1} A^{(i)}$.

Repeat

1. Let $C = \sum_{i=1}^{m} c_i = \left( \sum_{i=1}^{m} \omega_i (S^0 - E^0) + \frac{L}{2} Q - \frac{Y}{2} \right)$
2. for $j = 1, 2, \ldots, n$
3. Run Algorithm 1 using $C_j$ as input to update $S_j$, where $C_j / S_j$ is the $j$-th row of $C / S$
4. end for
5. Update $Q$ via Eq (13)
6. for $i = 1, 2, \ldots, m$
7. Update $E^{(i)}$ via Eq (14)
8. end for
9. Update $\omega$ according to Algorithm 2
10. Update $Y = Y + \mu (S - Q)$
11. Update $\mu = \min (\rho \mu, \max_u)$

Until stop condition is met

Obtain the low-dimensional features $B$ via Eq (24).

**Convergence analysis**

In this subsection, we analyze the convergence of the proposed LRGO-MVDR algorithm. According to the section of optimization, the optimization process of our algorithm can be divided into four subproblems, which are formulated in Eqs (12), (14), (15) and (19). According to [50], we know that there is a unique minimum in Eq (12), and the value of $E^0$ decrease the function value of Eq (14). Following [52], we can solve for the minimum value of Eq (15). As for the subproblem
in Eq (19), the objective function is obviously convex and then the minimization problem has unique solution. Thus, solving the four subproblems in Eqs (12), (14), (15) and (19) will decrease the value of our constrained optimization problem in each iteration [59]. In addition, since each term in Eq (9) is greater than zero, the constrained optimization problem of our LRGO-MVDR algorithm has a lower bound. Therefore, according to Monotonic Sequence Theorem, the proposed LRGO-MVDR algorithm will converge to a local minimum of Eq (9).

Experiments
Since our LRGO-MVDR algorithm is a graph-based learning model, its performance is evaluated and compared with those of other related multi-view graph construction or optimization methods [20, 21, 23, 24, 32, 60, 61] on several benchmark datasets.

Dataset description
Simulated Dataset: Our simulated experiment is designed on a toy three-view data. In this dataset, 200 data points in each cluster of each view are sampled from a Gaussian distribution with three dimensions. The points in these clusters have diverse centers and covariances, which are overlapped and hence difficult to distinguish. Detailed information on this dataset is listed in Table 1, where $\mu_{ij}$ and $\Sigma_{ij}$ represent mean and covariance of the $i$-th cluster in the $j$-th view, respectively. Fig 3 is the illustration of the simulated dataset.

| View 1 | Centers | $\mu_{11} = (1,1,2)$ | $\mu_{21} = (3,4,3)$ | $\mu_{31} = (5,5,5)$ |
|---|---|---|---|---|
| Covariances | $\Sigma_{11} = \begin{pmatrix} 1 & 0.5 & 0.8 \\ 0.5 & 1.5 & 0.6 \\ 0.8 & 0.6 & 1.2 \end{pmatrix}$ | $\Sigma_{12} = \begin{pmatrix} 1.2 & 0.2 & 0.4 \\ 0.2 & 0.9 & 0.1 \\ 0.4 & 0.1 & 1.1 \end{pmatrix}$ | $\Sigma_{13} = \begin{pmatrix} 1 & -0.5 & 0.2 \\ -0.5 & 1 & 0.3 \\ 0.2 & 0.3 & 1 \end{pmatrix}$ |

| View 2 | Centers | $\mu_{12} = (1,2,1.4)$ | $\mu_{22} = (3,3,3)$ | $\mu_{32} = (4,3,4)$ |
|---|---|---|---|---|
| Covariances | $\Sigma_{21} = \begin{pmatrix} 1.2 & -0.2 & 0.3 \\ -0.2 & 1.3 & -0.4 \\ 0.3 & -0.4 & 1.5 \end{pmatrix}$ | $\Sigma_{22} = \begin{pmatrix} 0.6 & 0.1 & 0.2 \\ 0.1 & 0.9 & 0.3 \\ 0.2 & 0.3 & 1 \end{pmatrix}$ | $\Sigma_{23} = \begin{pmatrix} 0.8 & 0.1 & 0.2 \\ 0.1 & 0.9 & 0.3 \\ 0.2 & 0.3 & 1.5 \end{pmatrix}$ |

| View 3 | Centers | $\mu_{13} = (1,1,2)$ | $\mu_{23} = (3,1,2)$ | $\mu_{33} = (5,4,5)$ |
|---|---|---|---|---|
| Covariances | $\Sigma_{31} = \begin{pmatrix} 0.6 & 0.2 & 0.3 \\ 0.2 & 1 & 0.4 \\ 0.3 & 0.4 & 1 \end{pmatrix}$ | $\Sigma_{32} = \begin{pmatrix} 1 & 0.4 & 0.2 \\ 0.4 & 0.7 & 0.3 \\ 0.2 & 0.3 & 0.9 \end{pmatrix}$ | $\Sigma_{33} = \begin{pmatrix} 1 & 0.4 & -0.2 \\ 0.4 & 1 & 0.3 \\ -0.2 & 0.3 & 0.9 \end{pmatrix}$ |

Table 1. Detailed information on the simulated dataset.

https://doi.org/10.1371/journal.pone.0225987.t001

Fig 3. Simulated dataset example.

https://doi.org/10.1371/journal.pone.0225987.g003
Caltech101 Dataset [62]: This is an object recognition dataset that contains 101 categories of images. We select 11 widely used classes, namely, dollar-bill, anchor, ant, cougar-body, elephant, flamingo, panda, platypus, seahorse, snoopy, and wildcat, and obtain 512 images. Three types of features are extracted to represent each image: 512-dimensional GIST features, 254-dimensional CENTRIST features and 36-dimensional LBP features.

Wiki Dataset [63]: This is a widely used dataset for cross-modal retrieval, which consists of 693 image-text pairs that are divided into 10 categories. In each pair, the image is encoded by 128-dimensional SIFT descriptors and the text consists of 10-dimensional topics that are derived from a Latent Dirichlet Allocation model.

Yale Dataset [64]: The Yale face dataset contains 165 grayscale images of 15 individuals. There are 11 images per subject and each has a different facial expression or configuration. We extract three types of features: 9-dimensional color moment, 50-dimensional LBP and 512-dimension GIST features.

Cornell Dataset [65]: This dataset is composed of web pages that were collected from the computer science department of Cornell University. There are 195 pages over 5 labels (student, project, course, staff, faculty). Two heterogeneous feature sets, namely, cites and content, are utilized here for experiments. Specifically, the pages are described by 1703 words in the content view, which is a sparse matrix containing 0 and 1 values indicating absence and presence of a word in a page; and the number of citation links between other pages in the cites view.

Wisconsin Dataset [65]: This dataset is composed of web pages that were collected from the computer science department of the University of Wisconsin. The archive contains 265 documents over 5 labels. Two views are considered for each webpage: content and cites. The detailed description is similar to the Cornell dataset.

WebKB Dataset [66]: This dataset contains a subset of the web pages that were collected from computer science departments of various universities in January 1997 by the World Wide Knowledge Base project of the CMU text learning group. The 1051 pages were manually classified into two categories. Each webpage can be described by content view and cites view. The detailed description is similar to the Cornell dataset.

Data preprocessing is used for all datasets. Specifically, each feature variable in the simulated data set and the six real-world datasets have been transformed to have sample mean 0 and variance 1.

**Compared algorithms**

BSV [67]: This algorithm performs tasks independently on each view and chooses a view that achieves the best performance.

FeaConc [32]: This algorithm concatenates the features from each view and performs subsequent tasks directly on this concatenated feature representation.

KA [32]: This algorithm constructs a kernel matrix for the data from each view and sums these matrices over all views to obtain a single kernel matrix.

MvSim [23]: This algorithm simultaneously deals with all the information that is contained in multi-view datasets by using several instances of an existing co-similarity algorithm.

MvSpec [24]: This algorithm learns a weighted combination of the specified kernels in parallel.

MvSpecCE [61]: This algorithm extends clustering ensembles to multi-view clustering.

Cospectral [20]: This algorithm constructs a graph in one view by using the eigenvectors in another view and the tasks are subsequently performed based on the graph.

Corespectral [60]: This algorithm regularizes the eigenvectors of view-dependent graph Laplacians and a pairwise co-regularization scheme is used in our experiment.
MSE [21]: This algorithm combines features into a feature matrix and uses matrix decomposition methods to obtain a low-dimensional embedding matrix.

RMSC [32]: This algorithm recovers a shared graph matrix via low rank and sparse decomposition.

All experiments in this work are implemented in MATLAB R2018b and run on an Intel Core i7-8700K CPU at 3.70 GHz with 16 GB physical memory.

Clustering

In this section, the performance of the proposed algorithm on clustering task is evaluated on four datasets: Simulated, Wisconsin, WebKB and Cornell. Information on these datasets is listed in Table 2.

Experimental settings. We set the values of parameters in the comparison algorithms according to the experiments in their corresponding literature [20, 21, 23, 24, 32, 60, 61]. In our approach, we empirically tune the values of parameters $\gamma$ and $\lambda$ in the range $\{10^{-4}, 10^{-3}, \ldots, 10^5\}$ and we tune $\beta$ by searching the grid $\{0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ in an alternating manner [54].

For the clustering task, we use two types of evaluation metrics to measure the performance: Clustering Accuracy (ACC) and Normalized Mutual Information (NMI) [37, 38]. The Clustering Accuracy is defined as follows:

$$\text{ACC} = \frac{\sum_{i=1}^{n} \sigma(l_i, \text{map}(c_i))}{n},$$

where $\sigma(x,y) = 1$ if $x=y$ and $\sigma(x,y) = 0$ otherwise, $n$ is the total number of samples, $l_i$ is the true class label of the $i$-th sample, $c_i$ is the clustering result label of the $i$-th sample, and $\text{map}(\cdot)$ is a best mapping function that uses the Kuhn-Munkres algorithm to map clustering labels to equivalent ground-truth labels [68]. NMI is defined as follows:

$$\text{NMI} = \frac{I(L, C)}{\sqrt{H(L)H(C)}},$$

where $I(L,C)$ denotes the mutual information between true class label $L$ and obtained cluster label $C$, $H(L)$ and $H(C)$ are the entropies of $L$ and $C$. According to Eqs (26) and (27), a larger value of ACC or NMI indicates a better clustering performance. Moreover, the $k$-means algorithm is adopted to perform clustering on the low-dimensional features obtained by different approaches. Because the performance of $k$-means is heavily dependent on the initialization, we repeat the experiment 10 times with random initialization. The average accuracies and the corresponding standard deviations are recorded in the following section.

Experimental results and analysis. The highest clustering accuracies obtained by various algorithms are listed in Table 3. According to this table, FeaConc outperforms BSV in most cases, which demonstrates that multi-view data can provide more complementary

| Dataset    | Size | Views | Clusters |
|------------|------|-------|----------|
| Simulated  | 600  | 3     | 3        |
| Wisconsin  | 265  | 2     | 5        |
| WebKB      | 1051 | 2     | 2        |
| Cornell    | 195  | 2     | 5        |

https://doi.org/10.1371/journal.pone.0225987.t002
information. Although RMSC [32] has a mechanism for dealing with noise, our approach outperforms it. This is because RMSC treats all views equally, which is not appropriate since the differences among views are ignored. Moreover, different algorithms employ different ways to fuse graphs. Nevertheless, their results are inferior to those of our method. This is because our method adequately takes noise and different importance among multiple views into consideration. Last, the proposed LRGO-MVDR algorithm outperforms all the other algorithms; the best NMI results obtained by the algorithms are shown in Fig 4.

Next, the performance of our LRGO-MVDR algorithm under various parameter values ($\gamma$, $\lambda$ and $\beta$) is evaluated. According to the experimental results in Fig 5A, the performance of our method is stable across a wide range of $\gamma$ values (especially for the experiments on the

![Fig 4. NMI results of various algorithms.](https://doi.org/10.1371/journal.pone.0225987.g004)
Wisconsin, WebKB and Cornell datasets). According to the results in Fig 5B, when the value of $\lambda$ is very small, the results that are obtained by LRGO-MVDR are relatively poor. With the increase of the value of $\lambda$, the performance of our algorithm improves significantly. This indicates that considering the noise in the data plays an important role in the proposed algorithm. Moreover, when $\lambda$ exceeds a specified value, the influence of $\lambda$ on our algorithm becomes small; hence, a larger $\lambda$ is preferable for the proposed LRGO-MVDR algorithm. In Fig 5C, we find that $\beta$ influences the performance of our algorithm. As we have analyzed, our algorithm performs better when $\beta$ is set as a moderate value since a small $\beta$ will make our LRGO-MVDR overlook the latent complementary information of multi-view data while a large $\beta$ will lead the different importance of views to be ignored. In general, the best $\beta$ value for our algorithm is highly data-dependent, but we can roughly observe that the satisfactory results are produced in the range of $[0.05, 0.5]$.

Then, the convergence curves of our LRGO-MVDR algorithm on four datasets are shown in Fig 6. In this figure, the x-axis and logarithm y-axis represent the iteration number and
the value of Eq (9), respectively. Fig 6 demonstrates that the value of Eq (9) monotonically decreases at each iteration. Here, we should point out that although the curves in Fig 6B and Fig 6C seem to be flat after only a few (i.e., 1–2) iterations, it actually keeps decreasing. Since the value changes of Eq (9) in these two cases are relatively small after 1–2 iterations, it is hard to observed from the figure.

Finally, in order to justify the choice of coordinate descent algorithm in our approach, we compare the performance of LRGO-MVDR with the same model which utilizes Lagrange multiplier and stochastic gradient descent methods for solving Eq (19). Here, we use LRGO-MVDR-C, LRGO-MVDR-L and LRGO-MVDR-S to denote the proposed LRGO-MVDR which adopts coordinate descent, Lagrange multiplier and stochastic gradient descent methods to optimize Eq (19), respectively. From the experimental results in Table 4, it can be found that LRGO-MVDR-C and LRGO-MVDR-L achieve very similar clustering performances. This may because that they converge to the nearly identical solutions during the optimization. However, due to the efficiency of coordinate descent algorithm, the training time of LRGO-MVDR-C is less than LRGO-MVDR-L. Since stochastic gradient descent needs to compute the gradient of Eq (19) and solve it along the negative of gradient, the convergence speed of LRGO-MVDR-S is slower than LRGO-MVDR-C and LRGO-MVDR-L. Moreover, we can also see that the clustering performance of LRGO-MVDR-S is inferior to other two approaches, which may due to it cannot find the optimum solution when the maximum number of iterations is reached.

### Classification

In this section, we use four publicly available datasets, namely, Caltech101, Wiki, Yale and Cornell, to assess the performance of the proposed approach on classification tasks. Detailed information on the datasets is tabulated in Table 5.

**Experimental settings.** For classification task, we repeated 10 times of hold-out validation for each dataset to evaluate the performance [69]. For each hold-out validation, each data set is randomly split into two sets of sizes \( l \) and \( t \), the first is treated as the training data, and the other part is test data. The values of \( l \) and \( t \) for various datasets are listed in Table 5. Then we obtain an average accuracy rate of 10 repetitions as hold-out validated accuracy (HVA). In this task, the nearest-neighbor classifier is used to measure each method’s performance due to its

| Dataset    | Size | Views | Classes | \( l \) | \( t \) |
|------------|------|-------|---------|-------|-------|
| Caltech101 | 512  | 3     | 11      | 258   | 254   |
| Wiki       | 693  | 2     | 10      | 349   | 344   |
| Yale       | 165  | 3     | 15      | 90    | 75    |
| Cornell    | 195  | 2     | 5       | 99    | 96    |

The numbers in parentheses are the feature dimension corresponds to the best result and the training time of each algorithm, respectively.

https://doi.org/10.1371/journal.pone.0225987.t004

https://doi.org/10.1371/journal.pone.0225987.t005
simplicity. The HVA is defined as follows:

$$\text{HVA} = \frac{1}{h} \sum_{i=1}^{h} \frac{N^{(i)}_{\text{cor}}}{N_{\text{total}}}$$  \hspace{1cm} (28)$$

where $h$ is the times of hold-out validation, $N^{(i)}_{\text{cor}}$ is the number of test samples that are correctly classified in $i$-th time hold-out validation by the nearest-neighbor classifier and $N_{\text{total}}$ is the total number of test samples [36, 37]. We report the highest HVA for all of parameter sets of $(\gamma, \lambda, \beta)$.

**Experimental results and analysis.** Fig 7 shows the classification accuracy comparison among the algorithms. In this figure, the x-axis and y-axis represent the low-dimensional representation dimensions and HVA, respectively. From Fig 7, we observe the following points: First, the performance of our algorithm is basically consistent with those of BSV which reports the best results that are achieved using a single view. However, choosing the view that yields the optimal recognition performance in practical applications is time consuming and computationally costly. Thus, adaptively combining graphs that are obtained by multiple views is important. Second, our method becomes more stable than other algorithms with dimension varies. Specifically, the HVA of our algorithm improves steadily with the increase of dimension, and fluctuates less than some other algorithms. Hence, the proposed method is more robust. Third, our proposed LRGO-MVDR algorithm outperforms other algorithms that treat all graphs equally, such as FeaConc, KA, MVSim, Cospectral, Core respectral and RMSC, which demonstrates the effectiveness of introducing the weighted vector for the various graphs in our algorithm. Furthermore, the HVA obtained by different algorithms in Table 6 can also demonstrate the advantage of our algorithm. Last, the proposed LRGO-MVDR algorithm outperforms all the other algorithms. This is attributed to LRGO-MVDR explicitly taking noise and differences in importance among the data of multiple views into consideration.
Fig 8 shows the impacts of various parameter values on the performance of our algorithm. Similar observations are made to those in the section on clustering. For the parameter $\gamma$ which fuses the multi-view data, the performance of our method is stable across a wide range of values in most cases. For $\lambda$, our algorithm performs better as the value of $\lambda$ increases. This demonstrates that the noise term plays an important role. When the parameter $\lambda$ reaches a certain value, the performance of our algorithm becomes less affected on most datasets. For the parameter $\beta$, we observed that our proposed algorithm achieves its best performance when the value of $\beta$ is in a middle range such as $\beta \in [0.05, 0.5]$, which is consistent with the clustering experiments and our discussions in the section on solving for $\omega$.

The convergence curves of our LRGO-MVDR algorithm on four datasets for the classification tasks are plotted in Fig 9. Similar to those in Fig 6, it is also found that the objective function rapidly decreases at the first few iterations and becomes nearly stable within 10 iterations for most datasets and 50 iterations for Wiki dataset, which empirically proves the convergence of our algorithm.

At last, we also compare the optimal classification performance of three optimization techniques to solve Eq (19) in our LRGO-MVDR. From the experimental results in Table 7, we find that LRGO-MVDR-C outperforms other two methods in terms of HVA and training time.

### Table 6. Best classification results (HVA±std) of various algorithms.

| Methods       | Caltech101     | Wiki           | Yale            | Cornell         |
|---------------|----------------|----------------|-----------------|-----------------|
| BSV           | 0.5110±0.0337  | 0.6500±0.0181  | 0.9187±0.0284   | 0.6646±0.0367   |
| FeaConc       | 0.3028±0.0225  | 0.6195±0.0118  | 0.8773±0.0250   | 0.6917±0.0602   |
| KA            | 0.4339±0.0256  | 0.6224±0.0189  | 0.8027±0.0300   | 0.6312±0.0165   |
| MVSim         | 0.2807±0.0209  | 0.5625±0.0179  | 0.6053±0.0475   | 0.5844±0.0487   |
| MVSpec        | 0.2165±0.0201  | 0.6497±0.0253  | 0.4040±0.0361   | 0.5615±0.0349   |
| MvSpecCE      | 0.2795±0.0000  | 0.1988±0.1727  | 0.5333±0.0000   | 0.5521±0.0000   |
| Cospectral    | 0.2764±0.0096  | 0.6360±0.0160  | 0.7453±0.0341   | 0.6664±0.0264   |
| Corespectral  | 0.3028±0.0225  | 0.2032±0.0079  | 0.6253±0.0528   | 0.5094±0.0581   |
| MSE           | 0.4169±0.0303  | 0.6032±0.0173  | 0.7953±0.0385   | 0.6271±0.0319   |
| RMSC          | 0.4232±0.0246  | 0.6125±0.0219  | 0.8000±0.0377   | 0.6438±0.0274   |
| LRGO-MVDR     | 0.5500±0.0257  | 0.6637±0.0168  | 0.9453±0.0203   | 0.7177±0.0271   |

The numbers in parentheses are the feature dimension corresponding to the best result and the training time of each algorithm, respectively.
To further evaluate the performance of LRGO-MVDR, a statistical significance test is adopted to determine whether the advantage of our algorithm over other methods is significant. The one-tailed Wilcoxon rank sum test is utilized. In this test, the null hypothesis is that LRGO-MVDR makes the same performance with other methods and the alternative hypothesis is that LRGO-MVDR results in improved performance compared with other methods. For instance, to compare the performance of our algorithm with that of BSV (LRGO-MVDR vs. BSV), the null and alternative hypotheses can be defined as

\[ H_0: M_{LRGO-MVDR} = M_{BSV} \]
\[ H_1: M_{LRGO-MVDR} > M_{BSV} \]

where \( M_{LRGO-MVDR} \) and \( M_{BSV} \) are the medians of the results that are obtained by LRGO-MVDR and BSV. In this experiment, the significance level is set to 0.05. According to the test results in Tables 8–10, the \( p \)-values obtained by all pairwise one-tailed Wilcoxon rank sum tests are less than 0.05. Therefore, the alternative hypotheses are accepted, and the null hypotheses are rejected in all tests. Hence, the proposed algorithm significantly outperforms the other algorithms.

### Conclusion and future work

In this paper, a graph optimization approach is proposed for multi-view dimensionality reduction. With reasonable low-rank and sparse constraints, the algorithm can effectively deal with the noise in the multi-view input data. A robust and shared graph matrix can be learned by...
minimizing the disagreement over the cleaned matrices. Furthermore, a weighted scheme is introduced into our algorithm to take the latent complementary information among multi-view datasets into consideration. We provide an effective iterative scheme for optimizing the LRGO-MVDR algorithm and analyze the convergence of this scheme. The experimental results on several datasets have demonstrated that the learned graph boosts the classification and clustering performance.

| Table 8. p-values of the Wilcoxon rank sum tests on clustering tasks (ACC). |
|-------------------------------------------------|-----------------|
| p-values                                        |                 |
| LRGO-MVDR vs. BSV                               | 0.0038          |
| LRGO-MVDR vs. FeaConc                           | 0.0011          |
| LRGO-MVDR vs. KA                                | 5.7275e-06      |
| LRGO-MVDR vs. MVSim                             | 4.5902e-04      |
| LRGO-MVDR vs. MVSpec                            | 0.00019         |
| LRGO-MVDR vs. MvSpecCE                          | 5.7392e-05      |
| LRGO-MVDR vs. Cospectral                        | 1.2268e-04      |
| LRGO-MVDR vs. Corespectral                      | 8.2736e-05      |
| LRGO-MVDR vs. MSE                               | 0.0019          |
| LRGO-MVDR vs. RMSC                              | 0.0019          |

https://doi.org/10.1371/journal.pone.0225987.t008

| Table 9. p-values of the Wilcoxon rank sum tests on clustering tasks (NMI). |
|-------------------------------------------------|-----------------|
| p-values                                        |                 |
| LRGO-MVDR vs. BSV                               | 1.0348e-04      |
| LRGO-MVDR vs. FeaConc                           | 0.0031          |
| LRGO-MVDR vs. KA                                | 0.0019          |
| LRGO-MVDR vs. MVSim                             | 3.63e-09        |
| LRGO-MVDR vs. MVSpec                            | 0.0019          |
| LRGO-MVDR vs. MvSpecCE                          | 3.5697e-09      |
| LRGO-MVDR vs. Cospectral                        | 0.0019          |
| LRGO-MVDR vs. Corespectral                      | 0.0035          |
| LRGO-MVDR vs. MSE                               | 0.0019          |
| LRGO-MVDR vs. RMSC                              | 0.0019          |

https://doi.org/10.1371/journal.pone.0225987.t009

| Table 10. p-values of the Wilcoxon rank sum tests on classification tasks (HVA). |
|-------------------------------------------------|-----------------|
| p-values                                        |                 |
| LRGO-MVDR vs. BSV                               | 0.0479          |
| LRGO-MVDR vs. FeaConc                           | 0.0470          |
| LRGO-MVDR vs. KA                                | 0.0030          |
| LRGO-MVDR vs. MVSim                             | 2.8704e-07      |
| LRGO-MVDR vs. MVSpec                            | 2.2929e-09      |
| LRGO-MVDR vs. MvSpecCE                          | 3.1995e-11      |
| LRGO-MVDR vs. Cospectral                        | 0.0020          |
| LRGO-MVDR vs. Corespectral                      | 5.4697e-11      |
| LRGO-MVDR vs. MSE                               | 0.0039          |
| LRGO-MVDR vs. RMSC                              | 0.0064          |

https://doi.org/10.1371/journal.pone.0225987.t010
In future work, we will try to use multiple restarts of ADMM with random initial points, which can be regarded as a heuristic algorithm, to approximate the non-convex problem in Eq (19) [70]. Furthermore, the effectiveness of evolutionary algorithm for optimizing our approach will also be tested.

Supporting information

S1 Appendix. LRGO-MVDR_code. A file contains matlab codes of LRGO-MVDR and the datasets used in this paper.

(ZIP)

Acknowledgments

This work was supported by the National Nature Science Foundation of China (No. 61672150, 11871141), the Fund of the Jilin Provincial Science and Technology Department (20190201305(C, 20180201089GX, and 20180520027JH) and the fundamental research funds for the central universities (2412019FZ049).

Author Contributions

Conceptualization: Youcheng Qian, Xueyan Yin, Jun Kong, Jianzhong Wang, Wei Gao.
Data curation: Xueyan Yin, Jun Kong, Jianzhong Wang, Wei Gao.
Formal analysis: Jianzhong Wang.
Funding acquisition: Wei Gao.
Methodology: Xueyan Yin, Jun Kong, Jianzhong Wang, Wei Gao.
Resources: Wei Gao.
Software: Youcheng Qian, Xueyan Yin.
Supervision: Jun Kong, Jianzhong Wang.
Visualization: Xueyan Yin.
Writing – original draft: Youcheng Qian, Xueyan Yin.
Writing – review & editing: Xueyan Yin, Jun Kong, Jianzhong Wang.

References

1. Xu C, Tao D, Xu C. A survey on multi-view learning. arXiv:1304.5634. [Preprint]. 2013 [cited 2013 April 20]. Available from: https://arxiv.org/abs/1304.5634.
2. Xu C, Tao D, Xu C. Large-margin multi-view information bottleneck. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2014; 36(8):1559–1572. https://doi.org/10.1109/TPAMI.2013.2296528 PMID: 26353338
3. Gheyas IA, Smith LS. Feature subset selection in large dimensionality domains. Pattern recognition. 2010; 43(1):5–13.
4. De la Torre F. A least-squares framework for component analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2012; 34(6):1041–1055. https://doi.org/10.1109/TPAMI.2011.184 PMID: 21911913
5. Ni J, Qiu Q, Chellappa R. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2013. p.692-699.
6. Song D, Tao D. Biologically inspired feature manifold for scene classification. IEEE Transactions on Image Processing. 2010; 19(1):174–184. https://doi.org/10.1109/TIP.2009.2032938 PMID: 19783505
7. Xu C, Tao D, Xu C, Rui Y. Large-margin weakly supervised dimensionality reduction. In: International Conference on Machine Learning (ICML); 2014. p.865-873.
8. Sugiyama M. Local fisher discriminant analysis for supervised dimensionality reduction. In: International Conference on Machine Learning (ICML); 2006. p.905-912.
9. Xu D, Yan S, Tao D, Lin S, Zhang H-J. Marginal fisher analysis and its variants for human gait recognition and content-based image retrieval. IEEE Transactions on Image Processing. 2007; 16(11):2811–2821. https://doi.org/10.1109/TIP.2007.906769 PMID: 17990757
10. Zhang Y, Zhou Z-H. Multilabel dimensionality reduction via dependence maximization. ACM Transactions on Knowledge Discovery from Data. 2010; 4(3):14.
11. Yan S, Xu D, Zhang B, Zhang H-J, Yang Q, Lin S. Graph embedding and extensions: A general framework for dimensionality reduction. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2007; 29(1):40–51. https://doi.org/10.1109/TPAMI.2007.12 PMID: 17108382
12. Wang J, Zhang B, Qi M, Kong J. Linear discriminant projection embedding based on patches alignment. Image Vision Computing. 2010; 28(12):1624–1636.
13. Turk M, Pentland A. Eigenfaces for recognition. Journal of cognitive neuroscience. 1991; 3(1):71–86. https://doi.org/10.1162/jocn.1991.3.1.71 PMID: 23964806
14. Belhumeur PN, Hespanha JP, Kriegman DJ. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1997; 19(7):711–720.
15. Roweis ST, Saul LK. Nonlinear dimensionality reduction by locally linear embedding. Science. 2000; 290(5500):2323–2326. https://doi.org/10.1126/science.290.5500.2323 PMID: 11125150
16. Tenenbaum JB, De Silva V, Langford JC. A global geometric framework for nonlinear dimensionality reduction. Science. 2000; 290(5500):2319–2323. https://doi.org/10.1126/science.290.5500.2319 PMID: 11125149
17. Belkin M, Niyogi P. Laplacian eigenmaps for dimensionality reduction and data representation. Neural computation. 2003; 15(6):1373–1396.
18. Donoho DL, Grimes C. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. Proceedings of the National Academy of Sciences. 2003; 100(10):5591–5596.
19. Zhang Z, Zha H. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. SIAM journal on scientific computing. 2004; 26(1):313–338.
20. Kumar A, Daume H. A co-training approach for multi-view spectral clustering. In: International Conference on Machine Learning (ICML); 2011. p.393-400.
21. Xia T, Tao D, Mei T, Zhang Y. Multiview spectral embedding. IEEE Transactions on Systems, Man, Cybernetics, Part B. 2010; 40(6):1438–1446.
22. Shu L, Latecki LJ. Integration of single-view graphs with diffusion of tensor product graphs for multi-view spectral clustering. In: Asian Conference on Machine Learning; 2016. p.362-377.
23. Bisson G, Grimal C. Co-clustering of multi-view datasets: a parallelizable approach. In: International Conference on Data Mining (ICDM); 2012. p.828-833.
24. Tzortzis G, Likas A. Kernel-based weighted multi-view clustering. In: International Conference on Data Mining (ICDM); 2012. p.675-684.
25. De Sa VR, Gallagher PW, Lewis JM, Malave VL. Multi-view kernel construction. Machine learning. 2010; 79(1–2):47–71.
26. De Sa VR, Ballard DH. Category learning through multimodality sensing. Neural Computation. 1998; 10(5):1097–1117. https://doi.org/10.1162/089976698300017368 PMID: 9654768
27. Li Y, Nie F, Huang H, Huang J. Large-Scale Multi-View Spectral Clustering via Bipartite Graph. In: Association for the Advancement of Artificial Intelligence (AAAI); 2015. p.2750–2756.
28. Zong L, Zhang X, Yu H, Zhao Q, Ding F. Local linear neighbor reconstruction for multi-view data. Pattern Recognition Letters. 2016; 84:56-62.
29. Liu G, Lin Z, Yu Y. Robust subspace segmentation by low-rank representation. In: International Conference on Machine Learning (ICML); 2010. p.663-670.
30. Ye G, Liu D, Jhuo I-H, Chang S-F. Robust late fusion with rank minimization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2012. p.3021-3028.
31. Pan Y, Lai H, Liu C, Tang Y, Yan S. Rank Aggregation via Low-Rank and Structured-Sparse Decomposition. In: Association for the Advancement of Artificial Intelligence (AAAI); 2013.
32. Xia R, Pan Y, Du L, Yin J. Robust Multi-View Spectral Clustering via Low-Rank and Sparse Decomposition. In: Association for the Advancement of Artificial Intelligence (AAAI); 2014. p.2149–2155.
33. Hong C, Yu J, Wan J, Tao D, Wang M. Multimodal deep autoencoder for human pose recovery. IEEE Transactions on Image Processing. 2015; 24(12):5659–5670. https://doi.org/10.1109/TIP.2015.2487860 PMID: 26452284

34. Zhuge W, Hou C, Jiao Y, Yue J, Tao H, Yi D. Robust auto-weighted multi-view subspace clustering with common subspace representation matrix. PloS one. 2017; 12(5):e0176769. https://doi.org/10.1371/journal.pone.0176769 PMID: 28542234

35. Boyd S, Parikh N, Chu E, Peleato B, Eckstein J. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine learning. 2011; 3(1):1–122.

36. Ahmadian A, Mostafa A. An efficient texture classification algorithm using Gabor wavelet. In: IEEE engineering in medicine and biology society (IEEE Cat No 03CH37439); 2003. p.930–933.

37. Wang J, Zhao R, Wang Y, Zheng C, Kong J, Yi Y. Locality Constrained Graph Optimization for Dimensionality Reduction. Neurocomputing. 2017; 245:55–67.

38. Zhang H, Zhuang Y, Wu F. Cross-modal correlation learning for clustering on image-audio dataset. In: ACM international conference on Multimedia; 2007. p.273-276.

39. Nie F, Huang H, Cai X, Ding CH. Efficient and robust feature selection via joint l2,1-norms minimization. In: Conference on Neural Information Processing Systems (NIPS); 2010. p.1813-1821.

40. Fazel M. Matrix rank minimization with applications: PhD thesis, Stanford University; 2002.

41. Candès EJ, Recht B. Exact matrix completion via convex optimization. Foundations of Computational mathematics. 2009; 9(6):717.

42. Candès EJ, Tao T. The power of convex relaxation: Near-optimal matrix completion. IEEE Transactions on Information Theory. 2010; 56(5):2053–2080.

43. Recht B, Fazel M, Parrilo PA. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM review. 2010; 52(3):471–501.

44. Fazel M, Hindi H, Boyd SP. A rank minimization heuristic with application to minimum order system approximation. In: American control conference; 2001. p.4734–4739.

45. Ramírez C, Kreinovich V, Argaez M. Why l1 is a good approximation to l0: A geometric explanation. Journal of Uncertain Systems. 2013; 7(3):203–207.

46. Zhou D, Huang J, Schölkopf B. Learning from labeled and unlabeled data on a directed graph. In: International Conference on Machine Learning (ICML); 2005. p.1036-1043.

47. Wu J, Lin Z, Zha H. Essential tensor learning for multi-view spectral clustering. IEEE Transactions on Image Processing. 2019.

48. Groetsch C. The theory of tikhonov regularization for fredholm equations. 104p, Boston Pitman Publication. 1984.

49. Li P, Bu J, Chen C, He Z, Cai D. Relational multimanifold coclustering. IEEE Transactions on cybernetics. 2013; 43(6):1871–1881. https://doi.org/10.1109/TSMCB.2012.2234108 PMID: 23757578

50. Cai J-F, Candès EJ, Shen Z. A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization. 2010; 20(4):1956–1982.

51. Lin Z, Chen M, Ma Y. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv:10095055 [Preprint]. 2010 [cited 2010 Sept.26]. Available from: https://arxiv.org/abs/1009.5055

52. Duchi J, Shalev-Shwartz S, Singer Y, Chandra T. Efficient projections onto the l1-ball for learning in high dimensions. In: International Conference on Machine Learning (ICML); 2008. p.272-279.

53. Boyd S, Vandenberghe L. Convex optimization: Cambridge university press; 2004.

54. Geng B, Tao D, Xu C, Yang L, Hua X-S. Ensemble manifold regularization. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2012; 34(6):1227–1233. https://doi.org/10.1109/TPAMI.2012.57 PMID: 22371429

55. Shen J, Deng C, Gao X. Attraction recommendation: Towards personalized tourism via collective intelligence. Neurocomputing. 2016; 173:789–798.

56. Bai S, Zhou Z, Wang J, Bai X, Jan Latecki L, Tian Q. Ensemble diffusion for retrieval. In: IEEE International Conference on Computer Vision (ICCV); 2017. p.774-783.

57. Xiao Y, Shen W, Wang Z, Liu S, Wang J. Multiple graph regularized graph transduction via greedy gradient Max-Cut. Information Sciences. 2018; 423:187–199.

58. Hong C, Yu J, You J, Chen X, Tao D. Multi-view ensemble manifold regularization for 3D object recognition. Information sciences. 2015; 320:395–405.
59. Tao D, Jin L, Yuan Y, Xue Y. Ensemble manifold rank preserving for acceleration-based human activity recognition. IEEE transactions on neural networks and learning systems. 2016; 27(6):1392–1404. https://doi.org/10.1109/TNNLS.2014.2357794 PMID: 25265635

60. Kumar A, Rai P, Daume H. Co-regularized multi-view spectral clustering. In: Conference on Neural Information Processing Systems (NIPS); 2011. p.1413-1421.

61. Xie X, Sun S. Multi-view clustering ensembles. In: International Conference on Machine Learning and Computing (ICMLC); 2013. p.51-56.

62. Fei-Fei L, Fergus R, Perona PJ. Understanding I. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. Computer vision and Image understanding. 2007; 106(1):59–70.

63. Pereira JC, Coviello E, Doyle G, Rasiwasia N, Lanckriet GR, Levy R, et al. On the role of correlation and abstraction in cross-modal multimedia retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2014; 36(3):521–535.

64. Cai D, He X, Hu Y, Han J, Huang T. Learning a spatially smooth subspace for face recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2007. p.1-7.

65. Bisson G, Grimal C. An architecture to efficiently learn co-similarities from multi-view datasets. In: International Conference on Neural Information Processing (ICNIP); 2012. p.184-193.

66. Sindhwani V, Niyogi P, Belkin M. Beyond the point cloud: from transductive to semi-supervised learning. In: International Conference on Machine Learning (ICML); 2005. p.824-831.

67. Ng AY, Jordan MI, Weiss Y. On spectral clustering: Analysis and an algorithm. In: Conference on Neural Information Processing Systems (NIPS); 2002. p.849-856.

68. Lovász L, Plummer MD. Matching theory: American Mathematical Soc.; 2009.

69. Yadav S, Shukla S. Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. In: International Conference on Advanced Computing (IACC); 2016. p.78-83.

70. Kanno Y, Kitayama S. Alternating direction method of multipliers as a simple effective heuristic for mixed-integer nonlinear optimization. Structural and Multidisciplinary Optimization. 2018; 58(3):1291–1295.