Information Processing and Management 59 (2022) 102931

Contents lists available at ScienceDirect

Information Processing and Management

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

Adaptive dictionary and structure learning for unsupervised feature selection

Yanrong Guo [a,b,*], Huihui Sun [a,b], Shijie Hao [a,b]

[a] *Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology, Ministry of Education, China*
[b] *School of Computer Science and Information Engineering, Hefei University of Technology, China*

ARTICLE INFO

Keywords:
Unsupervised feature selection
Adaptive sparse dictionary learning
Adaptive manifold learning
Matrix completion

ABSTRACT

Unsupervised feature selection is very attractive in many practical applications, as it needs no semantic labels during the learning process. However, the absence of semantic labels makes the unsupervised feature selection more challenging, as the method can be affected by the noise, redundancy, or missing in the originally extracted features. Currently, most methods either consider the influence of noise for sparse learning or think over the internal structure information of the data, leading to suboptimal results. To relieve these limitations and improve the effectiveness of unsupervised feature selection, we propose a novel method named Adaptive Dictionary and Structure Learning (ADSL) that conducts spectral learning and sparse dictionary learning in a unified framework. Specifically, we adaptively update the dictionary based on sparse dictionary learning. And, we also introduce the spectral learning method of adaptive updating affinity matrix. While removing redundant features, the intrinsic structure of the original data can be retained. In addition, we adopt matrix completion in our framework to make it competent for fixing the missing data problem. We validate the effectiveness of our method on several public datasets. Experimental results show that our model not only outperforms some state-of-the-art methods on complete datasets but also achieves satisfying results on incomplete datasets.

1. Introduction

Nowadays, big data is increasingly showing its advantages in various fields, such as information retrieval, pattern recognition, recommendation system (Zheng et al., 2020; Zhu et al., 2019; Wang et al., 2015; Mitra, Murthy & Pal, 2002). The expanding big data promotes the rapid development of artificial intelligence and deep learning. However, traditional machine learning can still be effective for the study of medical data with limited acquired subjects. For example, for the study of Alzheimer's disease, depression, and other mental disorders, it's hard to provide an extensive database for deep learning. Moreover, (Zhu, Ma, Yuan & Zhu, 2022; Gan et al., 2021) point out that data structure information and semantic features, which are difficult to be considered by most deep learning methods, can effectively improve classification performance. In addition, big data brings in the issue of the curse of dimensionality (Zhang, Wang, Jin & Wang, 2016; Zhu et al., 2017), posing challenges to computational effectiveness and efficiency. For example, a dataset may contain noisy or redundant information in high-dimensional features. Moreover, some feature items can be absent in some situations. These issues tend to have significant negative impacts on many pattern analysis tasks. Therefore, it is essential to learn a

[*] Corresponding author at: Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology, Ministry of Education, China.

E-mail address: yrguo@hfut.edu.cn (Y. Guo).

https://doi.org/10.1016/j.ipm.2022.102931
Received 5 September 2021; Received in revised form 17 February 2022; Accepted 9 March 2022
Available online 24 March 2022

more compact feature representation for the high-dimensional data, aiming to ensure the accuracy and computational efficiency of the pattern analysis tasks.

To this end, many feature selection methods have been proposed. Feature selection methods typically aim to remove noise and irrelevant information by reducing the feature dimension of data. Generally, they can be divided into three categories: unsupervised, semi-supervised, and supervised according to whether sample labels are available. Supervised feature selection methods use category labels of the sample to select discriminative features. Differently, without sample labels, unsupervised methods rely more on the graph structure of data and try to mine the intrinsic correlation between features. Semi-supervised methods usually take an in-between roadmap that first extracts the most discriminant features with unlabeled data and then further improves the learning model based on the rest labeled part (Zhu et al., 2019). Although the supervised methods can be more reliable with guidance from labels, the label acquisition process can be very expensive or labor-intensive in many real-world applications. In this context, it is valuable to develop an effective unsupervised feature selection method (Wang et al., 2015; Mitra et al., 2002).

Among the existing unsupervised feature selection algorithms, sparse dictionary learning (Chen, Guo & Hao, 2020; Mairal, Bach, Ponce & Sapiro, 2010) is one of the mainstream. The goal of dictionary learning is to extract the essential features of samples to find a suitable sparse representation of the dense and noisy original data. Its advantage is removing the influence of noise and redundant features and reducing the complexity of data. However, the classic sparse dictionary learning methods usually perform feature selection based on a fixed dictionary, which is learned from the original dataset. And its performance depends heavily on the quality of the dictionary initially constructed. In addition, sparse dictionary learning cannot guarantee the integrity of manifold structure within the original data. Spectral learning (Zhu, Gan, Lu, Li & Zhang, 2020; Wang et al., 2016) is also widely used in unsupervised feature selection methods, such as Spectral Feature selection (SPEC) (Zhao and Liu, 2007), Muti-cluster Feature Selection (MCFS) (Cai, Zhang & He, 2010), and Minimum Redundancy Spectral Feature Selection (MRSF) (Zhao, Wang & Liu, 2010). It can learn the manifold structure information of data well. Nevertheless, the above methods perform spectral learning and sparse feature representation independently, leading to the suboptimal solution (Zhu, Wu, Ding & Zhang, 2013). To resolve this, (Zhu et al., 2013; Zhu, Li, Zhang, Ju & Wu, 2017) simultaneously conduct manifold learning and sparse regression. But they are still limited in relying on fixed dictionaries and affinity matrices, which may also be suboptimal.

It is also noted that the dataset can suffer from the incompleteness issue. In many applications, the problem of missing data is often encountered, as data collection can be a complex and long-term process. For example, in the ADNI dataset, subjects are required to take different examinations to help doctors better judge their conditions but some subjects are unwilling or not suitable for one of the examinations. And, in a movie recommendation system, all the interviewees can't have seen every movie (Zhang et al., 2019). One possibility is to exclude these incomplete samples manually. However, this process increases the workload of researchers and causes the waste of samples.

To overcome the limitations mentioned above, we proposed a robust and effective feature selection method. It integrates sparse dictionary learning and spectral learning into a united framework. In general, the proposed method can maintain the intrinsic structure information of the original data while removing the noise. More specifically, instead of fixing the input dictionary, we adaptively update the dictionary to construct the best basis space for the original data, where each data point can be accurately expressed as a sparse linear combination of these basis vectors. We learn the affinity matrix dynamically according to spectral learning, which can not only remove the noise data but also maintain the internal relationship between the original data. In addition, we introduce matrix completion into our framework to solve the problem of missing data.

To sum up, the contributions of our research are:

- We propose a novel method, Adaptive Dictionary and Structure Learning (ADSL), which unifies spectral learning and sparse dictionary learning in a framework.
- Our method adaptively updates the dictionary and affinity matrix simultaneously, preserving the structure of the original data while removing the redundancy and noise. Therefore, the feature selection process becomes more robust.
- By equipping matrix completion, our method can also process incomplete datasets. It not only avoids the waste of data but also verifies the generalization ability of our method.

We organize the remainder of the paper as follows. In Section 2, we briefly introduce some closely related directions. In Section 3, some preliminaries are introduced. We describe the modeling process of ADSL in Section 4 and its optimization process in Section 5. In Section 6, the experimental results that validate the effectiveness of our method are reported. Section 7 concludes the paper.

2. Related work

In this section, we describe the related methods of unsupervised feature selection based on sparse learning and spectral learning. As we introduce matrix completion into our whole framework, we also briefly the related works for the matrix completion task.

2.1. Feature selection

The goal of feature selection is to reduce the feature dimension for retaining helpful information and removing irrelevant and redundant data. In this paper, we mainly focus on the unsupervised feature selection scenario, in which the semantic or category labels of the samples in a dataset are assumed unavailable.

The sparse-regularization-based methods are attractive because of their clear modeling process and good performance (Chen, Zhao

& Guo, 2020; Zhang, Kyaw, Chang & Chua, 2017; Zhang, , Zha, Yang, Yan & Chua, 2014). They treat feature selection as a sparse regression problem (Chen, Guo & Hao, 2020; Hou, Nie, Li, Yi & Wu, 2014), in which the objective function can be regarded as composed of the reconstruction term and regularization term (Peng & Fan, 2017). The former one is usually designed as a reconstruction or regression error loss term. The latter is typically designed as a sparse term that avoids overfitting and produces representation sparsity. For example, the methods proposed by Tibshirani (1996) and Wang, Zhu and Zou (2008) are both based on l_1 $-norm$ regularization that achieves good sparsity and has been widely adopted. However, the objective function composed of l_1 $-norm$ is non-convex and challenging to optimize. To solve the optimization problem, many researchers introduced $l_{2,1} - norm$ (Xiang, Nie, Meng, Pan & Zhang, 2012; Nie, Huang, Cai, & Ding, 2010; Liu, Ji, & Ye, 2009) for feature selection. In this way, the objective function is relatively easy to optimize. As $l_1 - norm$ and $l_{2,1} - norm$ have limitations in handling outliers (Liu, Ji, & Ye, 2009; Wu, Wang, Gao & Li, 2018), Chen, Guo and Hao (2020) adopt $l_{2,r} - norm(0 < r \leq 2)$ as the reconstruction loss term and $l_{2,p} - norm(0 < p \leq 1)$ as the sparsity regularization term.

Unsupervised spectral feature selection (USFS) is another effective method to deal with high-dimensional data, which takes manifold learning into account during feature selection. Spectral feature selection is a model that integrates the feature selection method and subspace learning method to take the advantage of the two methods (Yuan, Zhong, Lei, Zhu & Hu, 2021). As summarized in (Zhu, Zhang, Hu, Zhu and Song, 2018), USFS contains two key components, i.e., graph-based subspace learning and sparsity regularization. For example, Cai, Zhang and He (2010) and Zhao, Wang, Liu and Ye (2013) obtain the graph representation by performing eigenvalue decomposition of the original data, and then perform $l_1 - norm$ and $l_{2,1} - norm$ regularization on the graph representation to select significant features, respectively. The above two methods construct graph matrix and feature selection separately, possibly leading to suboptimal results. As a step further, several methods are proposed to simultaneously conduct manifold learning and sparse regression, such as the Joint Graph Sparse Coding (JGSC) method (Zhu et al., 2013) and the Robust Joint Graph Sparse Coding (RJGSC) method (Zhu et al., 2017). JGSC considers manifold learning and spectral clustering in a unified framework, in which $F-$norm is employed. RJGSC replaces the least square loss function in JGSC with a more robust one to avoid the outlier influence. The above methods are very dependent on the graph matrix initially learned. To address this limitation, the researchers propose automatically updating the graph matrix during learning. For example, the unsupervised Feature Selection with Adaptive Structure Learning (FSASL) method (Du and Shen, 2015) obtains the adaptive graph matrix through iteration until convergence.

Generally, the above methods directly construct the affinity matrix based on the original data and perform feature selection on a fixed dictionary. As the data points often lie in high-dimension and contain noise and redundancy, the learned affinity matrix and dictionary can be of low quality, therefore having a negative impact on the final feature selection. Differently, our method combines spectral learning and sparse dictionary learning in a unified framework via updating the dictionary and graph matrix simultaneously, making the whole model more robust to noise and redundancy.

2.2. Matrix completion method

Dealing with incomplete datasets for feature representation has received much attention in recent years. Missing data values can be broadly classified as missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR) (Wu & Lange, 2015). Researchers have proposed a series of methods to deal with the missing dataset problem, including deleting samples with missing data, mean substitution, multiple imputations. Removing the sample will cause the waste of samples so that the information of missing samples cannot be fully utilized. Then, Candès and Recht (2009) prove that a low-rank matrix can be almost perfectly recovered when the number of observed entries exceeds a certain level. Therefore, the task of matrix completion, which recovers missing data items from observed data subsets to obtain a complete matrix (Kwon & Choi, 2020), has attracted much attention. According to the algorithm of solving the objective function, low-rank matrix completion algorithms can be divided into the following categories: (1) Matrix completion based on kernel norm relaxation. Ma, Goldfarb, & Chen, 2011 and Toh and Yun (2010) relax the standard matrix completion problem into a matrix lasso model and solve it by the nearest neighbor gradient algorithm and the accelerated nearest neighbor gradient algorithm, respectively. Cai, Candès, & Shen, 2010 introduce the regularization term of the elastic net (elastic-net) to increase the stability of the matrix completion problem. (2) Matrix completion based on matrix decomposition/ factorization. To avoid the complicated singular value decomposition, the target matrix is decomposed into the product of two low-rank matrices, therefore improving the efficiency. Following this roadmap, (Jain, Netrapalli & Sanghavi, 2013; Kim, Lee, Choi, Kwak and Oh, 2015; Tanner and Wei, 2016; Gu, Wang & Liu, 2016) adopt the alternating minimization methods to solve the matrix decomposition model. Liu, Jiao and Shang (2013) proposed a double decomposition model. (3) Matrix completion based on non-convex relaxation. For example, Nie, Wang, Huang and Ding (2015) introduce matrix Schatten p-norm into the model to replace rank function. Ghasemi, Malek-Mohammadi, Babaie-Zadeh and Jutten (2011) use the Gaussian function instead of the rank function.

The above methods usually need to know the matrix rank in advance. However, the rank value is generally difficult to estimate in practical applications. A new roadmap that firstly estimates the matrix rank has been adopted. Wen, Yin and Zhang (2012) propose a low-rank matrix fitting algorithm (LMaFit) to estimate the rank. Shi, Lu and Cheung (2018) propose a novel low-rank matrix completion method (L1MC) that can automatically determine the rank of an incomplete matrix based on $l_1 - norm$ regularization on the weight vector. L1MC is more suitable for our research, as we do not know the matrix rank in advance. In our method, L1MC acts as the pre-processing module to complete the dataset if needed.

Table 1
Notations.

Notation	Description
$\boldsymbol{X}$	The original and complete feature matrix/The complete feature matrix after completion
$\boldsymbol{x}_i$	The $i-th$ row of $\boldsymbol{X}$
$\boldsymbol{x}^j$	The $j-th$ column of $\boldsymbol{X}$
x_{ij}	The element on $i-th$ row and $j-th$ column of $\boldsymbol{X}$
$\|\boldsymbol{X}\|_F$	The Frobenius norm of $\boldsymbol{X}$
$\|\boldsymbol{X}\|_{2,1}$	The $l_{2,1}-norm$ of $\boldsymbol{X}$
$tr(\boldsymbol{X})$	The trace of $\boldsymbol{X}$
$rank(\boldsymbol{X})$	The rank of $\boldsymbol{X}$
$\boldsymbol{X}^T$	The transpose of $\boldsymbol{X}$
$\boldsymbol{X}^{-1}$	The inverse of $\boldsymbol{X}$
$\boldsymbol{B}$	The base/dictionary matrix
$\boldsymbol{S}$	The learned representation of data
$\boldsymbol{D}$	The degree matrix
$\boldsymbol{W}$	The similarity matrix
$\boldsymbol{L}$	The Laplacian matrix
$\boldsymbol{M}$	The feature matrix of the incomplete original dataset
$\boldsymbol{G}$	The initial graph structure of dataset

3. Preliminary

In this section, we first introduce the notations used in this paper. Then we briefly introduce the problem formulation of sparse dictionary learning and spectral learning, as well as low-rank matrix completion. These are the fundamental elements of our ADSL method.

3.1. Notations

The notations in this paper are summarized in Table 1, where the matrices/vectors are denoted by italic boldface uppercase/lowercase letters, respectively. Scalars are represented by normal italic letters. For a matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$, its $l_{2,1}-$norm can be defined as follows:

$$\|\boldsymbol{X}\|_{2,1} = \sum_{i=1}^{m}\left(\sum_{j=1}^{n}|x_{ij}|^2\right)^{\frac{1}{2}} \tag{1}$$

3.2. Sparse dictionary learning

Given a dataset $\boldsymbol{X} \in \mathbb{R}^{m\times n}$, its columns indicate n data samples, and each sample is expressed as an m-dimensional vector. We can extract a set of basis vectors $\boldsymbol{b}^i$ from $\boldsymbol{X}$ as the dictionary matrix $\boldsymbol{B} = [\boldsymbol{b}^1,\boldsymbol{b}^2,\cdots,\boldsymbol{b}^k] \in \mathbb{R}^{m\times k}$, where $\boldsymbol{b}^i$ is a basis vector of m dimensions, and k is the vocabulary number of the dictionary. Then, each sample can be expressed as a sparse linear combination of these basis vectors. The objective function of the sparse dictionary learning method is defined as below:

$$\min_{\boldsymbol{S}} \|\boldsymbol{X}-\boldsymbol{BS}\|_F^2 + \alpha\sum_{i=1}^{n}\|\boldsymbol{s}^i\|_1 \ s.t. \sum_{i=1}^{m}\sum_{j=1}^{k} b_{ij}^2 \leq 1 \tag{2}$$

where $\boldsymbol{S} = [\boldsymbol{s}^1, \boldsymbol{s}^2, \cdots, \boldsymbol{s}^n] \in \mathbb{R}^{k\times n}$ is the sparse representation of the samples $\boldsymbol{X}$. It can be treated as the projection of the original data to the base space $\boldsymbol{B}$. The first term $\|\boldsymbol{X}-\boldsymbol{BS}\|_F^2$ is the reconstruction error term. The second term is the regularization term, aiming to control the sparsity of each vector $\boldsymbol{s}^i$. The constraint $b_{ij}^2 \leq 1$ is to prevent the dictionary from having too large a value. α is the balancing parameter.

3.3. Spectral learning

Spectral clustering based on the graph Laplacian well preserves the manifold structure of original data. Given the data points $\{\boldsymbol{x}^1,\cdots,\boldsymbol{x}^n\} \in \mathbb{R}^{m\times n}$, the manifold structure can be represented based on the graph Laplacian:

$$\sum_{i\neq j}\|\boldsymbol{x}^i-\boldsymbol{x}^j\|_2^2\omega_{ij} = tr\left(\boldsymbol{XLX}^T\right) \tag{3}$$

In this equation, ω_{ij} is the data similarity between the $i-$th and $j-$th sample:

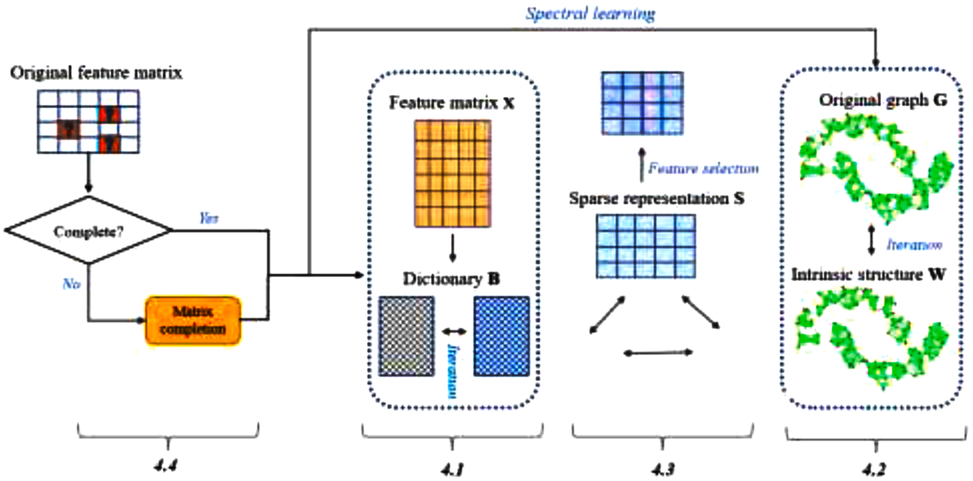

Fig. 1. The framework of the proposed method. It is mainly composed of four parts. If the dataset is incomplete, matrix completion (Section 4.4) should be performed for the missing dataset. Then, an initial dictionary is learned according to the feature matrix, and an adaptive dictionary is learned according to Section 4.1 to obtain the sparse representation of the original data. Meanwhile, an initial graph structure is learned from the original data, and then the graph is adaptively updated, according to Section 4.2. Finally, the dictionary $\boldsymbol{B}$ and the graph $\boldsymbol{W}$ are updated under a unified framework (Section 4.3) to obtain sparse feature representation $\boldsymbol{S}$, in which feature selection is carried out.

$$\omega_{ij}= \begin{cases} e^{-\frac{\|x^i-x^j\|^2}{\sigma}}, & \boldsymbol{x}^i \in N\left(\boldsymbol{x}^j\right) \ or \ \boldsymbol{x}^j \in N\left(\boldsymbol{x}^i\right) \\ 0, & else \end{cases} \tag{4}$$

$N(\boldsymbol{x}^i)$ is the k nearest neighbors of sample $\boldsymbol{x}^i$.$\boldsymbol{L}$ is the Laplacian matrix defined as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$, where $\boldsymbol{W} = [\omega_{ij}]|_{i,j=1,\cdots,n} \in \mathbb{R}^{n\times n}$ is the similarity matrix, and $\boldsymbol{D}$ is a diagonal matrix with its dialogue elements as the column sums of $\boldsymbol{W}$. σ is the scaling parameter. As $\boldsymbol{W}$ measures the similarity between all sample pairs within the data, their intrinsic structure properties can be reflected by the Laplacian matrix $\boldsymbol{L}$ (Ng, Jordan & Weiss, 2001).

3.4. Low rank matrix completion

The rank of a matrix measures the correlation between the rows and columns of a matrix. If the rows or columns of a matrix are linearly correlated, it is considered the matrix to be a low-rank matrix or to be sparse. According to the compressed sensing theory, we can make use of the existing observed data to fill in the missing data. The low-rank matrix completion problem can be modeled as the following rank minimization constrained optimization model:

$$\min_{\boldsymbol{X}} rank(\boldsymbol{X}) \ s.t.\mathscr{P}_{\boldsymbol{\Omega}}(\boldsymbol{X}) = \mathscr{P}_{\boldsymbol{\Omega}}(\boldsymbol{M}) \tag{5}$$

where $rank(\cdot)$ is the matrix rank, $\boldsymbol{M} \in \mathbb{R}^{I_1\times I_2}$ is a target matrix with missing data, and $\boldsymbol{\Omega} \in \mathbb{R}^{I_1\times I_2}$ is the binary index matrix. If $\boldsymbol{M}_{ij}$ is observed, Ω_{ij} is 1, otherwise Ω_{ij} is 0. $\mathscr{P}_{\boldsymbol{\Omega}}$ is the relevant sampling operator, which retrieves only the items indexed by $\boldsymbol{\Omega}$. $\boldsymbol{X} \in \mathbb{R}^{I_1\times I_2}$ is the complete matrix that approximates the original matrix $\boldsymbol{M}$.

4. Proposed methods

In this section, we first present the whole framework of our method, which is shown in Fig. 1. In the first part, an initial dictionary is first learned according to the feature matrix, and then an adaptive dictionary is learned through iteration. Meanwhile, in the second part, an initial graph structure is learned from the original data, and then the graph is adaptively updated. In the third part, the dictionary and the graph are updated under a unified framework to obtain the sparse feature representation. As a complementary component, the fourth part of our framework is the matrix completion module prepared for the dataset with missing items. The details of these four parts are introduced in the following subsections, respectively.

4.1. Adaptive dictionary learning

Previous dictionary learning methods used a fixed dictionary $\boldsymbol{B}$. However, if this dictionary is affected by noise, the obtained representation $\boldsymbol{S}$ is possibly in low quality. To solve this, we propose to adaptively update the dictionary, of which the objective function used for feature selection is shown as:

$$\min_{\boldsymbol{B},\boldsymbol{S}} \boldsymbol{J}(\boldsymbol{B},\boldsymbol{S}) = \| \boldsymbol{X} - \boldsymbol{BS} \|_F^2 + \alpha \| \boldsymbol{S} \|_{2,1} + \beta \| \boldsymbol{B} \|_{2,1} \tag{6}$$

In Eq. (6), the first term is the fidelity term that measures the reconstruction errors based on the learned dictionary $\boldsymbol{B}$ and data representation $\boldsymbol{S}$. The second term $\| \boldsymbol{S}\|_{2,1}$ avoids the overfitting issue, in which the $l_{2,1} - norm$ is adopted to control sparsity. It controls the sparseness of $\boldsymbol{S}$ and makes the row sum of $\boldsymbol{S}$ corresponding to irrelevant features equal to 0, thus preserving important features. The third term $\| \boldsymbol{B}\|_{2,1}$ controls the sparseness of the learned dictionary $\boldsymbol{B}$, which makes the dictionary itself update during the learning process. In this way, the quality of $\boldsymbol{B}$ can be improved.

4.2. Adaptive laplace graph matrix learning

The manifold structure represented by a graph Laplacian can be inaccurate, as the similarity between samples can be seriously affected by noise. As a consequence, the inaccurate graph Laplacian possibly further impacts the feature selection process based on spectral clustering.

To solve this issue, instead of the original features, we construct the similarity matrix on the new representation $\boldsymbol{S}$, which contains less redundancy and noise. The spectral learning based on the new representation $\boldsymbol{S}$ is formulated as below:

$$\sum_{i\neq j} \| \boldsymbol{s}^i - \boldsymbol{s}^j \|_2^2 \omega_{ij} = tr\left(\boldsymbol{SLS}^T\right) \tag{7}$$

From Eq. (7), we can see that the graph Laplacian can also be adaptively updated based on the newly obtained representation $\boldsymbol{S}$. Then the objective function used for feature selection is shown in Eq. (8):

$$\min_{\boldsymbol{W}} J(\boldsymbol{W}) = \sum_{i\neq j} \| \boldsymbol{s}^i - \boldsymbol{s}^j \|_2^2 \omega_{ij} + \lambda \sum\nolimits_{i=1}^{n} \| \boldsymbol{\omega}_i \|_2^2 \tag{8}$$

4.3. Adaptive dictionary and structure learning (ADSL)

Based on the above adaptive dictionary learning and adaptive Laplacian graph matrix learning, we can construct our unsupervised feature selection model, that is, Adaptive Dictionary and Structure Learning (ADSL). The overall objective function is shown as:

$$\underset{\boldsymbol{B},\boldsymbol{S},\boldsymbol{W}}{\operatorname{argmin}} J(\boldsymbol{B},\boldsymbol{S},\boldsymbol{W}) = \| \boldsymbol{X} - \boldsymbol{BS} \|_F^2 + \alpha \| \boldsymbol{S} \|_{2,1} + \beta \| \boldsymbol{B} \|_{2,1} + \gamma \sum_{i,j=1}^{n} \| \boldsymbol{s}^i - \boldsymbol{s}^j \|_2^2 \omega_{ij} + \lambda \sum_{i=1}^{n} \| \boldsymbol{\omega}_i \|_2^2 \tag{9}$$

$$s.t., \forall \boldsymbol{\omega}_i 1 = 1, \omega_{ii} = 0, \omega_{ij} \geq 0 if\ j \in N(i), otherwise\ 0$$

where α, β, γ, and λ are hyperparameters. $\mathbf{1}$ is a column vector with all 1, and the term$\forall \boldsymbol{\omega}_i 1 = 1$ constrain the sum of each row element of $\boldsymbol{W}$ as 1.

We can see that Eq. (9) is the combination of Eqs. (6) and (8). The ADSL model is a unified framework that jointly optimizes the dictionary $\boldsymbol{B}$, the affinity matrix $\boldsymbol{W}$, and the desired feature representation $\boldsymbol{S}$ simultaneously. In other words, the obtained representation $\boldsymbol{S}$ relieves the noise impact on the dictionary $\boldsymbol{B}$ and the affinity matrix $\boldsymbol{W}$, while the updated $\boldsymbol{B}$ and $\boldsymbol{W}$ enable an improved $\boldsymbol{S}$ of higher quality on the other hand. This process can be realized by solving Eq. (9) in an iterative framework, as described in Section 5.

4.4. Matrix completion as a complementary module

Here we assume that the matrix rank is unknown in advance, which is often encountered in practical conditions. We adopt the powerful matrix completion method (Shi, Lu & Cheung, 2018) as the complementary module of ADSL.

The matrix completion model based on low-rank matrix decomposition is expressed as follows:

$$\min_{\boldsymbol{Z},\boldsymbol{U},\boldsymbol{V}} \| \boldsymbol{Z} - \boldsymbol{UV}^T \|_F^2 \ s.t. \mathscr{P}_{\boldsymbol{\Omega}}(\boldsymbol{Z}) = \mathscr{P}_{\boldsymbol{\Omega}}(\boldsymbol{M}) \tag{10}$$

where $\boldsymbol{M} \in \mathbb{R}^{I_1 \times I_2}$ is a target matrix with missing terms. $\boldsymbol{\Omega} \in \mathbb{R}^{I_1 \times I_2}$ is the binary index matrix. If $\boldsymbol{M}_{ij}$ is observed, Ω_{ij} is 1, otherwise Ω_{ij} is 0. $\mathscr{P}_{\boldsymbol{\Omega}}$ is the relevant sampling operator, which retrieves only the items indexed by $\boldsymbol{\Omega}$. $\boldsymbol{Z} \in \mathbb{R}^{I_1 \times I_2}$ is a complete matrix that approximates the original matrix $\boldsymbol{M}$. According to the matrix factorization theory, $\boldsymbol{Z}$ can be factorized as $\boldsymbol{UV}^T$, where $\boldsymbol{U} \in \mathbb{R}^{I_1 \times R}$, $\boldsymbol{V} \in \mathbb{R}^{I_2 \times R}$, and R ($R < \min(I_1, I_2)$) is the rank of $\boldsymbol{M}$.

The rank-one approximation is widely used in matrix completion (Hu, Zhao, Cai, He & Li, 2016; Wang et al., 2015; Wang et al., 2014). For any matrix $\boldsymbol{Z}$, it can be expressed as the weighted sum of R rank-one matrices:

$$\boldsymbol{Z} = \sum_{r=1}^{R} q_r \boldsymbol{u}_r \boldsymbol{v}_r^T = \boldsymbol{U} diag(\boldsymbol{q}) \boldsymbol{V}^T \tag{11}$$

$$s.t.\ \ \| \boldsymbol{u}_r \|_2 = \| \boldsymbol{v}_r \|_2 = 1,\ \text{for } r = 1, \cdots, R$$

where $\boldsymbol{q} = [q_1, \cdots, q_r, \cdots, q_R]^T$ is the weight vector, $\boldsymbol{U} \in \mathbb{R}^{I_1 \times R} = \{\boldsymbol{u}_r\}_{r=1}^R$, $\boldsymbol{V} \in \mathbb{R}^{I_2 \times R} = \{\boldsymbol{v}_r\}_{r=1}^R$, and R is the rank of $\boldsymbol{Z}$.

According to the above low-rank matrix decomposition and rank-one approximation, the matrix completion problem can be expressed in the following form:

$$\min_{\boldsymbol{X},\boldsymbol{Z}} \frac{1}{2} \| \boldsymbol{X} - \boldsymbol{Z} \|_F^2 \tag{12}$$

$$s.t.\ \boldsymbol{Z} = \sum_{r=1}^{R} q_r \boldsymbol{u}_r \boldsymbol{v}_r^T,\ \mathscr{P}_{\boldsymbol{\Omega}}(\boldsymbol{X}) = \mathscr{P}_{\boldsymbol{\Omega}}(\boldsymbol{M}),\ \| \boldsymbol{u}_r \|_2 = \| \boldsymbol{v}_r \|_2 = 1,\ \text{for } r = 1, \cdots, R$$

Note that the matrix rank is unknown in advance, $l_1 - norm$ regularization is imposed on the weight matrix $\boldsymbol{q}$, and the matrix completion problem is finally formulated as follows:

$$\min_{\boldsymbol{X},\boldsymbol{q},\{\boldsymbol{u}_r,\boldsymbol{v}_r\}_{r=1}^R,R} \mu \| \boldsymbol{q} \|_1 + \frac{1}{2} \left\| \boldsymbol{X} - \sum_{r=1}^{R} q_r \boldsymbol{u}_r \boldsymbol{v}_r^T \right\|_F^2 \tag{13}$$

$$s.t.\ \mathscr{P}_{\boldsymbol{\Omega}}(\boldsymbol{X}) = \mathscr{P}_{\boldsymbol{\Omega}}(\boldsymbol{M}),\ \| \boldsymbol{u}_r \|_2 = \| \boldsymbol{v}_r \|_2 = 1,\ \text{for } r = 1, \cdots, R$$

where R is the rank for estimation. By minimizing Eq. (13), we can automatically determine the rank of the incomplete matrix and predict the missing terms (Shi, Lu & Cheung, 2018).

5. Optimization

As we need to find three optimal variables for solving the objective function in Eq. (9), the Iteratively Reweighted Least Squares (IRLS) framework is adopted. For example, in a typical iteration, we can firstly update $\boldsymbol{S}$ by fixing $\boldsymbol{W}$and $\boldsymbol{B}$, then update $\boldsymbol{B}$ by fixing $\boldsymbol{S}$ and $\boldsymbol{W}$, and finally update $\boldsymbol{W}$ by fixing $\boldsymbol{S}$ and $\boldsymbol{B}$. In the following, we present the details of the optimization process.

5.1. Update S by fixing B and W

Given fixed $\boldsymbol{B}$ and $\boldsymbol{W}$, Eq. (9) can be reformulated as minimizing the following object function $J(\boldsymbol{S})$:

$$J(\boldsymbol{S}) = \| \boldsymbol{X} - \boldsymbol{BS} \|_F^2 + \alpha \| \boldsymbol{S} \|_{2,1} + \gamma \sum_{i,j=1}^{n} \| \boldsymbol{s}^i - \boldsymbol{s}^j \|_2^2 \omega_{ij} \tag{14}$$

By taking the derivative of $\boldsymbol{S}$ in the above equation, we can obtain:

$$\frac{\partial J(\boldsymbol{S})}{\partial \boldsymbol{S}} = -2\boldsymbol{B}^T(\boldsymbol{X} - \boldsymbol{BS}) + 2\alpha \boldsymbol{D}_1 \boldsymbol{S} + 2\gamma \boldsymbol{SL} \tag{15}$$

where $\boldsymbol{D}_1$ is a diagonal matrix, with $\boldsymbol{D}_1(i,i) = \frac{1}{2\|s_i\|_2}$. By setting Eq. (15) to zero:

$$\left(\boldsymbol{B}^T\boldsymbol{B} + \alpha \boldsymbol{D}_1\right)\boldsymbol{S} + \gamma \boldsymbol{SL} = \boldsymbol{B}^T\boldsymbol{X} \tag{16}$$

As both $\boldsymbol{B}^T\boldsymbol{B} + \alpha\boldsymbol{D}_1$ and $\gamma\boldsymbol{L}$ are semidefinite positive matrices, the singular value decomposition is carried on them:

$$\boldsymbol{B}^T\boldsymbol{B} + \alpha\boldsymbol{D}_1 = \boldsymbol{U}_1\boldsymbol{C}_1\boldsymbol{U}_1^T \tag{17}$$

$$\gamma\boldsymbol{L} = \boldsymbol{V}_1\boldsymbol{C}_2\boldsymbol{V}_1^T \tag{18}$$

where $\boldsymbol{U}_1$ and $\boldsymbol{V}_1$ are unitary matrices. Then Eq. (16) can be characterized as:

$$\boldsymbol{U}_1\boldsymbol{C}_1\boldsymbol{U}_1^T\boldsymbol{S} + \boldsymbol{S}\boldsymbol{V}_1\boldsymbol{C}_2\boldsymbol{V}_1^T = \boldsymbol{B}^T\boldsymbol{X} \tag{19}$$

By multiplying $\boldsymbol{U}_1^T$ and $\boldsymbol{V}_1$ from left and right side of above equation respectively, we obtain:

$$\boldsymbol{C}_1\boldsymbol{U}_1^T\boldsymbol{S}\boldsymbol{V}_1 + \boldsymbol{U}_1^T\boldsymbol{S}\boldsymbol{V}_1\boldsymbol{C}_2 = \boldsymbol{U}_1^T\boldsymbol{B}^T\boldsymbol{X}\boldsymbol{V}_1 \tag{20}$$

To simplify the representation, the above formulation can be denoted with $\boldsymbol{\varPsi} = \boldsymbol{U}_1^T\boldsymbol{S}\boldsymbol{V}_1$ and $\mathrm{T} = \boldsymbol{U}_1^T\boldsymbol{B}^T\boldsymbol{X}\boldsymbol{V}_1$. Thereby, we have:

$$\boldsymbol{C}_1\boldsymbol{\varPsi} + \boldsymbol{\varPsi}\boldsymbol{C}_2 = \mathrm{T} \tag{21}$$

where $\boldsymbol{C}_1 = diag(\sigma_1^{(1)}, \cdots, \sigma_1^{(m)})$, $\boldsymbol{C}_2 = diag(\sigma_2^{(1)}, \cdots, \sigma_2^{(k)})$ and $\Psi_{ij} = \frac{\mathrm{T}_{ij}}{\sigma_1^i + \sigma_2^j}$, respectively.

Finally, the learned representation can be denoted as:

$$\boldsymbol{S} = \boldsymbol{U}_1\boldsymbol{\varPsi}\boldsymbol{V}_1^T \tag{22}$$

Table 2
Algorithm 1 The pseudo-code of ADSL.

```
Algorithm 1 The pseudo-code of ADSL

Input: The original dataset
    Output: The learned representation $\boldsymbol{S}$
    if the original dataset is complete
    Initialize: The Laplacian matrix $\boldsymbol{L}$ and the base $\boldsymbol{B}$
    end
    else if the original dataset is incomplete, i.e., $\boldsymbol{M}$
    Matrix completion is performed for $\boldsymbol{M}$ according to Section 4.4
    Initialize: The Laplacian matrix $\boldsymbol{L}$ and the base $\boldsymbol{B}$
    end
    Repeat
    Calculating $\boldsymbol{B}^T\boldsymbol{B} + \alpha\boldsymbol{D}_1$ and $\gamma\boldsymbol{L}$, $\beta\boldsymbol{D}_2$ and $\boldsymbol{S}\boldsymbol{S}^T$
    Update $\boldsymbol{S}$ according to Eq. (22)
    Update $\boldsymbol{B}$ according to Eq. (30)
    Update $\boldsymbol{W}$ according to Eq. (34)
    Calculate Laplacian matrix $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$
    Until convergence
```

5.2. *Update* $\boldsymbol{B}$ *by fixing* $\boldsymbol{S}$ *and* $\boldsymbol{W}$

By fixing $\boldsymbol{S}$ and $\boldsymbol{W}$, the second, the fourth, and the fifth term in Eq. (9) can be regarded as constants, and Eq. (9) can be transformed into:

$$J(\boldsymbol{B}) = \| \boldsymbol{X} - \boldsymbol{B}\boldsymbol{S}\|_F^2 + \beta \| \boldsymbol{B}\|_{2,1} \tag{23}$$

Similar to the above process for solving the variable $\boldsymbol{S}$ above, we take the derivative of the variable $\boldsymbol{B}$ and set it to zero. Then we have the following equation:

$$\beta\boldsymbol{D}_2\boldsymbol{B} + \boldsymbol{B}\boldsymbol{S}\boldsymbol{S}^T = \boldsymbol{X}\boldsymbol{S}^T \tag{24}$$

where $\boldsymbol{D}_2$ is a diagonal matrix, with $\boldsymbol{D}_2(i,i) = \frac{1}{2\|\boldsymbol{b}_i\|_2}$. While both $\boldsymbol{S}\boldsymbol{S}^T$and $\beta\boldsymbol{D}_2$ are semidefinite positive matrices, the singular value decomposition is then carried on them:

$$\boldsymbol{S}\boldsymbol{S}^T = \boldsymbol{U}_2\boldsymbol{C}_3\boldsymbol{U}_2^T \tag{25}$$

$$\beta\boldsymbol{D}_2 = \boldsymbol{V}_2\boldsymbol{C}_4\boldsymbol{V}_2^T \tag{26}$$

where $\boldsymbol{U}_2$ and $\boldsymbol{V}_2$ are unitary matrices. Then Eq. (24) can be characterized as:

$$\boldsymbol{V}_2\boldsymbol{C}_4\boldsymbol{V}_2^T\boldsymbol{B} + \boldsymbol{B}\boldsymbol{U}_2\boldsymbol{C}_3\boldsymbol{U}_2^T = \boldsymbol{X}\boldsymbol{S}^T \tag{27}$$

By multiplying $\boldsymbol{V}_2^T$ and $\boldsymbol{U}_2$ from the left and right side of the above equation respectively, we obtain:

$$\boldsymbol{C}_4\boldsymbol{V}_2^T\boldsymbol{B}\boldsymbol{U}_2 + \boldsymbol{V}_2^T\boldsymbol{B}\boldsymbol{U}_2\boldsymbol{C}_3 = \boldsymbol{V}_2^T\boldsymbol{X}\boldsymbol{S}^T\boldsymbol{U}_2 \tag{28}$$

To simplify the representation, we let $\boldsymbol{\Phi} = \boldsymbol{V}_2^T\boldsymbol{B}\boldsymbol{U}_2$ and $\boldsymbol{\Gamma} = \boldsymbol{V}_2^T\boldsymbol{X}\boldsymbol{S}^T\boldsymbol{U}_2$. Then we have:

$$\boldsymbol{C}_4\boldsymbol{\Phi} + \boldsymbol{\Phi}\boldsymbol{C}_3 = \boldsymbol{\Gamma} \tag{29}$$

where $\boldsymbol{C}_3 = diag(\sigma_3^{(1)}, \cdots, \sigma_3^{(m)})$, $\boldsymbol{C}_4 = diag(\sigma_4^{(1)}, \cdots, \sigma_4^{(k)})$ and $\Phi_{ij} = \frac{\Gamma_{ij}}{\sigma_3^i + \sigma_4^j}$, respectively.

Finally, the learned dictionary can be denoted as:

$$\boldsymbol{B} = \boldsymbol{V}_2\boldsymbol{\Phi}\boldsymbol{U}_2^T \tag{30}$$

5.3. *Update* $\boldsymbol{W}$ *by fixing* $\boldsymbol{S}$ *and* $\boldsymbol{B}$

By fixing $\boldsymbol{S}$ and $\boldsymbol{B}$, Eq. (9) is presented as:

$$\underset{\boldsymbol{W}}{argmin}\gamma\sum_{i,j=1}^{n} \| \boldsymbol{s}^i - \boldsymbol{s}^j\|_2^2\omega_{ij} + \lambda\sum_{i=1}^{n} \| \boldsymbol{\omega}_i\|_2^2 \tag{31}$$

$$s.t., \forall\boldsymbol{\omega}_i 1 = 1, \omega_{ii} = 0, \omega_{ij} \geq 0 if\ j \in N(i), otherwise\ 0$$

Table 3
The details of the complete/incomplete ADNI dataset used in our experiment.

	Label	Instances	Dimension
Complete	*NC*	190	2161
	MCI	303	
	AD	131	
Incomplete	*NC*	522	999
	MCI	866	
	AD	341	

By denoting $h_{ij} = \| \boldsymbol{s}^i - \boldsymbol{s}^j \|_2^2$, Eq. (31) can be converted into:

$$\min_{\boldsymbol{\omega}_i} \left\| \boldsymbol{\omega}_i + \frac{1}{2\frac{\lambda}{\gamma}} \boldsymbol{h}_i \right\|_2^2 \tag{32}$$

$$s.t., \forall \boldsymbol{\omega}_i 1 = 1, \omega_{ii} = 0, \omega_{ij} \geq 0 if\ j \in N(i), otherwise\ 0$$

Furthermore, the Lagrange function of Eq. (32) is:

$$\min_{\boldsymbol{\omega}_i, \tau, \boldsymbol{\eta}} \left\| \boldsymbol{\omega}_i + \frac{1}{2\frac{\lambda}{\gamma}} \boldsymbol{h}_i \right\|_2^2 - \tau(\boldsymbol{\omega}_i 1 - 1) - \boldsymbol{\eta}^T \boldsymbol{\omega}_i^T \tag{33}$$

where τ and $\boldsymbol{\eta}$ are the Lagrange multipliers. Based on the Karush-Kuhn-Tucker (KKT) conditions, the closed-form solution can be achieved as follows:

$$\omega_{ij} = \left(-\frac{1}{2\frac{\lambda}{\gamma}} h_{ij} + \tau \right)_+ \tag{34}$$

The whole optimization process of our proposed ADSL algorithm is summarized in Table 2.

6. Experiment

We validate the effectiveness of our ADSL model on several public datasets, including the Alzheimer's Disease dataset[1], the RELATHE dataset and the Colon dataset[2]. In this section, the datasets, evaluation metrics, methods for comparison, and implementation details of our model are first introduced. Then, we report and analyze the quantitative experimental results.

6.1. Datasets

The Alzheimer's Disease Neuroimaging Initiative (ADNI) is a longitudinal multicenter study designed to develop clinical, imaging, genetic, and biochemical biomarkers for the early detection and tracking of Alzheimer's disease (AD). In our study, we only use a part of ADNI data to validate our model. Specifically, we choose 624 participants, containing 190 Normal Control (NC), 303 Mild Cognitive Impairment (MCI), and 131 Alzheimer's Disease (AD). The multi-modal biomarkers used for classification include genetic information (APOE4), demographic information (such as age, education, etc.), cognitive tests data, and MRI ROIs measures.

Moreover, an ADNI dataset with missing data from the TADPOLE challenge[3] is also adopted to evaluate the robustness of our model. Of note, the unlabeled samples and the PET feature are removed because of the severe absence of data. More details about the sample number are shown in Table 3. The RELATHE dataset is a text dataset derived from the 20 Newsgroups original dataset. It has two categories and contains 1427 samples with 4322-dimensional features. And the Colon dataset is a biomedical dataset about chemotherapy for B/C colon cancer. It includes 62 instances with 2000 features.

6.2. Evaluation metrics

To validate the performance of feature selection, we follow the experiment settings commonly adopted in the related works (Zhao et al., 2010; Nie et al., 2010; Du & Shen, 2015), including SVM-based classification and K-means-based clustering. For the above two methods, we adopt two evaluation metrics that are commonly used in unsupervised feature selection research, including ACC (accuracy) and F1 (F1_measure):

[1] http://adni.loni.usc.edu/, https://www.synapse.org/#!Synapse:syn2290704/wiki/64634

[2] https://jundongl.github.io/scikit-feature/datasets.html

[3] https://tadpole.grand-challenge.org/Data/

Table 4
Average classification/clustering results (ACC%, F1-measure%) of different feature selection algorithms on the complete datasets. The best two results are highlighted in bold.

Algorithm		NCvs.AD		NCvs.MCI		MCIvs.AD		RELATHE		Colon	
		ACC	F1	ACC	F1	ACC	F1	ACC	F1	Acc	F1
SVM	**baseline**	67.69	60.45	65.66	77.27	61.82	20.67	73.43	79.68	76.92	57.14
	JGSC	73.85	62.22	64.65	73.68	69.32	27.03	86.01	86.84	84.62	75.00
	RJGSC	76.92	65.12	70.71	**78.52**	71.59	35.90	88.46	91.15	**92.31**	**90.91**
	GSSR	77.85	67.16	65.45	74.18	69.32	32.69	88.39	89.18	89.23	82.07
	w/o updating B	**78.15**	**78.57**	**72.12**	77.23	**77.27**	**62.96**	**90.28**	**91.59**	87.69	80.56
	ADSL (our)	**86.15**	**86.27**	**73.74**	**80.00**	**74.32**	**53.33**	**92.52**	**92.95**	**96.92**	**93.89**
K-means	**baseline**	52.31	53.21	51.35	51.13	52.39	**54.16**	54.45	60.98	54.84	37.04
	JGSC	60.44	**56.95**	52.13	52.32	57.37	44.04	54.62	62.39	63.55	41.38
	RJGSC	60.44	**56.95**	**58.92**	57.76	62.21	43.75	**54.66**	**62.49**	**63.87**	**68.29**
	GSSR	**60.75**	37.32	57.00	**71.88**	**65.21**	45.47	**54.94**	**70.16**	60.32	44.44
	w/o updating B	57.38	**60.08**	57.61	57.08	**65.67**	51.47	54.62	62.39	63.23	42.10
	ADSL (our)	**69.47**	53.33	**60.85**	**75.35**	61.75	**69.49**	**54.94**	**70.16**	**64.52**	**70.27**

- **ACC**: (1) For the classification method, accuracy represents the percentage of the correctly classified samples in the total number of samples:

$$Acc = \frac{N_c}{N}$$

where N denotes the number of samples and N_c is correctly classified samples.

(2) For the clustering method, cluster accuracy is used to compare the predicted labels with the true labels:

$$Acc = \frac{\sum_{i=1}^{N} \delta(r_i, map(g_i))}{N}$$

where r_i, g_i and N are the true labels, the predicted labels and the number of samples, respectively. $map(x)$ is used to predict clustering labels to best match the true labels. δ is the indicator function:

$$\delta(x, y) = \begin{cases} 1, & if x = y \\ 0, & otherwise \end{cases}$$

- **F1**: The metric of $F1\_measure$ jointly considers precision and recall.

$$F1\_measure = \frac{2 \ \times Precision \times Recall}{Precision + Recall}$$

Here $Precision = \frac{TP}{TP+FP}$, $Recall = \frac{TP}{TP+FN}$. For classification, *TP, FP*, and *FN* mean true positive, false positive, and false negative, respectively. For clustering, *TP* assigns two similar samples to the same cluster. *TN* assigns two dissimilar samples to different clusters. *FP* assigns two dissimilar samples to the same cluster. *FN* assigns two similar samples to different clusters.

6.3. Methods for comparison

We briefly introduce the methods for comparison, including a baseline model and some state-of-the-art algorithms. Then we introduce the implementation details of our ADSL model.

- **Baseline**: The baseline means that all the sample features are directly used for classification without feature selection. That is to demonstrate the effectiveness of feature selection in general.
- **JGSC** (Joint Graph Sparse Coding)(Zhu, Wu, Ding and Zhang, 2013): **JGSC** considers manifold learning and regression in a unified framework. It employs $F-$norm to perform feature selection.
- **RJGSC** (Robust Joint Graph Sparse Coding)(Zhu et al., 2017): Based on **JGSC, RJGSC** further adopts $l_{2,1}-$norm to boost the robustness of the classification model.
- **GSSR** (General Spectral Sparse Regression)(Chen, Guo and Hao, 2020): **GSSR** handles the outlier features by learning the joint sparsity and handles the noisy features by preserving the local structure of the data.

Table 5
Average classification/clustering results (ACC%, F1-measure%) of different feature selection algorithms on the incomplete datasets completed by L1MC. The best two results are highlighted in bold.

Algorithm		**NCvs.AD**		**NCvs.MCI**		**MCIvs.AD**		**RELATHE**	
		ACC	**F1**	**ACC**	**F1**	**ACC**	**F1**	**ACC**	**F1**
SVM	**baseline**	78.74	64.08	65.95	**77.86**	72.02	17.07	63.99	74.94
	JGSC	82.18	71.56	69.18	**78.39**	73.66	30.43	82.52	84.47
	RJGSC	82.76	72.73	68.46	75.56	**74.90**	37.11	**85.66**	**87.83**
	GSSR	85.06	**81.16**	**69.89**	77.42	74.07	**43.24**	84.62	85.90
	w/o updating B	**86.26**	80.00	68.32	76.52	72.67	18.62	81.82	83.85
	ADSL (our)	**89.31**	**85.76**	**70.25**	76.58	**75.06**	**40.21**	**86.76**	**87.77**
K-means	**baseline**	50.87	41.60	54.03	54.36	50.70	38.97	54.45	60.98
	JGSC	55.97	37.29	51.15	55.45	58.24	40.78	54.52	61.91
	RJGSC	55.97	**48.56**	54.03	**56.45**	**71.52**	**43.75**	**54.66**	**62.49**
	GSSR	56.20	**48.46**	**56.74**	34.24	62.97	21.41	**54.66**	**62.49**
	w/o updating B	**56.43**	48.14	**62.10**	31.58	**71.67**	**44.09**	54.52	61.91
	ADSL (our)	**57.71**	46.60	**62.10**	**76.56**	**71.52**	**43.75**	**54.94**	**70.16**

Table 6
The Colon dataset was sampled randomly at 100% to 60% sampling rates. Then, the missing dataset was completed by L1MC. The best two results are highlighted in bold.

Algorithm		**SR=1.0** **ACC**	**SR=0.9** **ACC**	**SR=0.8** **ACC**	**SR=0.7** **ACC**	**SR=0.6** **ACC**
SVM	**baseline**	76.92	69.23	69.23	69.23	69.23
	JGSC	84.62	84.62	76.92	76.92	73.84
	RJGSC	**92.31**	90.77	**84.62**	**83.08**	76.92
	GSSR	89.23	86.16	**84.62**	**84.62**	**80.00**
	w/o updating B	87.69	**92.31**	83.08	80.00	76.92
	ADSL (our)	**96.92**	**96.92**	**86.16**	**84.62**	**83.08**
K-means	**baseline**	54.84	54.84	54.84	54.84	54.84
	JGSC	63.55	63.23	**62.75**	60.83	60.00
	RJGSC	**63.87**	**63.55**	**63.87**	**62.45**	**60.65**
	GSSR	60.32	58.06	60.00	57.10	56.13
	w/o updating B	63.23	61.29	61.29	58.06	56.45
	ADSL (our)	**64.52**	**63.87**	**63.87**	**63.23**	**61.29**

Table 7
The Colon dataset was sampled randomly at 100% to 60% sampling rates. Then, the missing dataset was filled with zero. The best two results are highlighted in bold.

Algorithm		**SR=1.0** **ACC**	**SR=0.9** **ACC**	**SR=0.8** **ACC**	**SR=0.7** **ACC**	**SR=0.6** **ACC**
SVM	**baseline**	76.92	69.23	61.54	61.54	53.85
	JGSC	84.62	76.92	69.23	61.54	61.54
	RJGSC	**92.31**	**80.00**	**76.92**	69.23	64.62
	GSSR	89.23	**84.62**	73.84	**70.77**	**66.15**
	w/o updating B	87.69	76.92	66.15	66.15	61.54
	ADSL (our)	**96.92**	**84.62**	**75.38**	**75.38**	**69.23**
K-means	**baseline**	54.84	54.84	54.84	50.00	53.23
	JGSC	63.55	56.13	**55.16**	54.19	54.19
	RJGSC	**63.87**	**57.42**	**55.81**	**54.84**	**54.52**
	GSSR	60.32	55.81	54.84	54.52	54.19
	w/o updating B	63.23	55.81	54.84	53.23	53.23
	ADSL (our)	**64.52**	**56.45**	**55.81**	**55.16**	**54.84**

- **ADSL (ours):ADSL** learns and updates the dictionary, intrinsic manifold structure, and the data representation in a unified framework.

Some implementation details of ADSL-based classification are introduced. The hyperparameters α, β, γ, and λ of ADSL are determined through 5-folds cross-validation. As for the feature selection, it is realized based on the new representation $\boldsymbol{S}$. By applying the $l_{2,1}-$norm to $\boldsymbol{S}$, the discriminative features are indicated by non-zero rows, while the non-discriminative features are indicated by all-zero rows. According to the descending order value of $l_{2,1}-$ norm, we select corresponding k top-level row features.

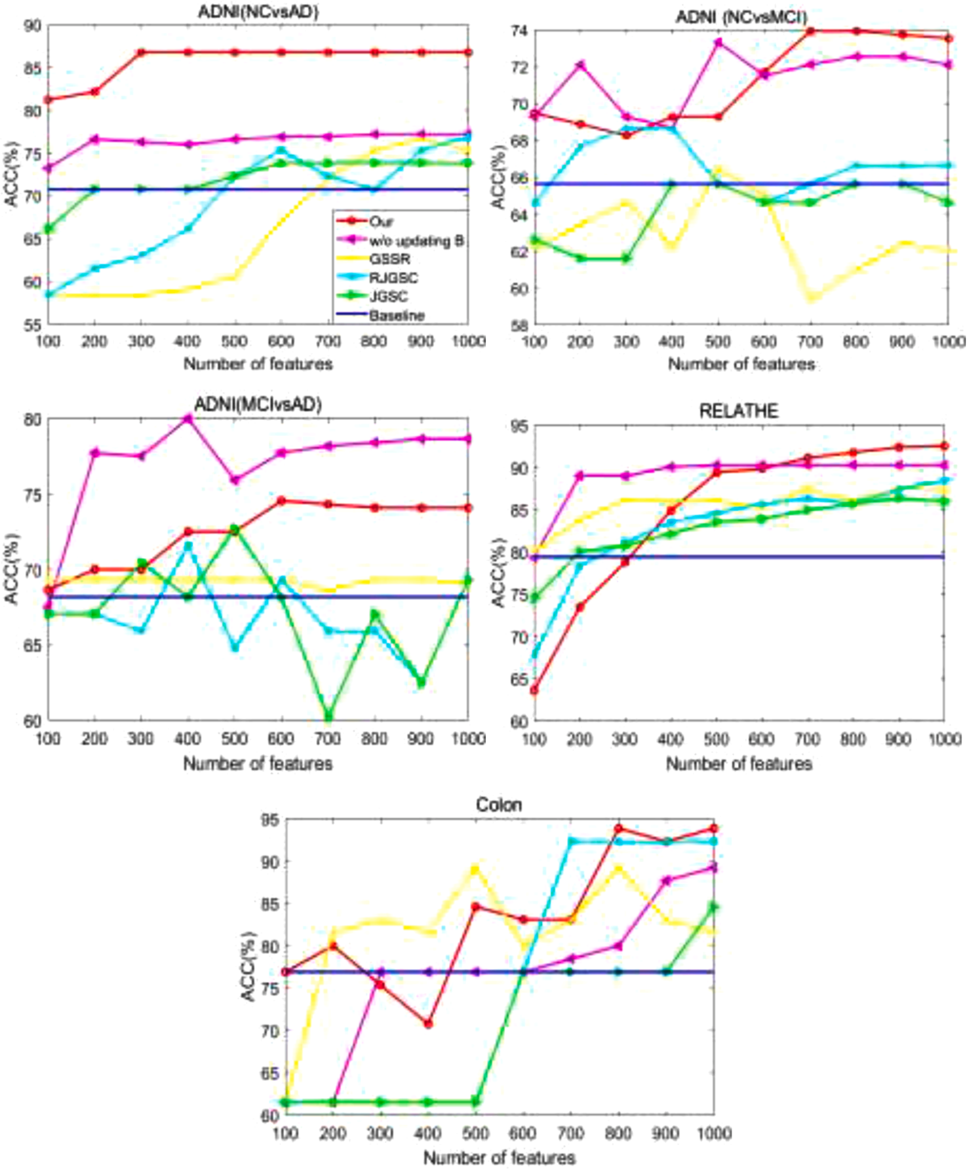

Fig. 2. The average classification accuracy varies with the number of feature selections. The figure only shows the accuracy of SVM classification for each comparison method in Table 4.

6.4. Results and analysis

Table 4 shows the test results of different feature selection methods on complete datasets. We have the following general observations. First, we can see that our algorithm (ADSL) generally achieves the best performance on the completed dataset, both in the SVM classification track and in the K-means clustering track. In the SVM classification track, our ADSL algorithm achieves the best results in the three tasks, which are 8.0% (NC vs. AD), 1.62% (NC vs. MCI), 2.24% (RELATHE), and 4.61% (Colon) higher than the suboptimal method, respectively. As for the K-means clustering track, ADSL achieves the best results in almost all the tasks. Second, by comparing the two tracks, we observe that the performance of the SVM classification is consistently better than the K-means clustering. Similar trends can be found in all the following experiments. It is understandable as the K-means clustering does not utilize the label information. On the other hand, the results from the K-means clustering demonstrate that the unsupervised feature selection method can still obtain satisfactory results. Third, according to the experimental results, the classification performance of our method on the ADNI dataset is better than that of other methods. Especially, the detection of the early MCI stage can assist the patients with intervention treatment as soon as possible.

We also have some detailed observations from Table 4. Here ‘w/o updating ***B***’ refers to the intermediate version of our method (same for the rest of experiments), where the dictionary ***B*** in our model is not updated. On the one hand, we can see that this

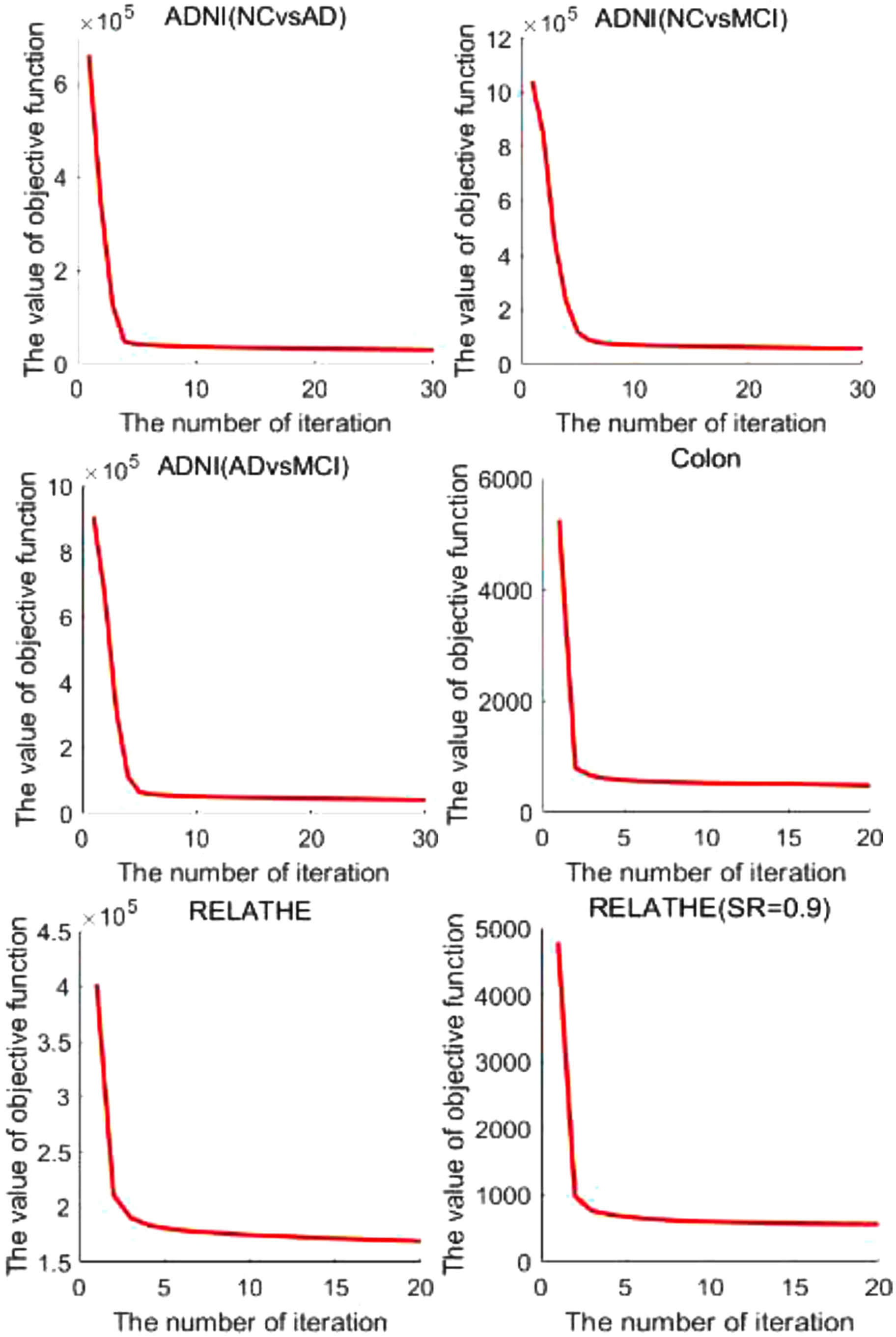

Fig. 3. The convergence of the objective function for different datasets. The convergence graph is obtained using the SVM classification method of the objective function on different complete datasets.

intermediate version still has competitive performance over other methods in general, which empirically demonstrates the effectiveness of updating the spectral structure in our model. On the other hand, by comparing it with our full version ADSL, the gap between the two methods further empirically shows the effectiveness of gradually updating the dictionary $\boldsymbol{B}$.

We also test the performance under more challenging situations with missing data. The miss rate of the incomplete ADNI dataset is 37.46%. As for RELATHE, 10% of the dataset is randomly deleted. For fairness, matrix completion is conducted for all the methods for comparison. Table 5 shows the results of all the methods tested on the two incomplete datasets. The ADSL achieves the best accuracies on all the tasks in the SVM classification track. As for the K-means clustering track, our ADSL still reaches the best accuracy except on the MCI vs. AD classification task. These results empirically validate the effectiveness of introducing the matrix completion module. In addition, by comparing the RELATHE dataset in Tables 4 and 5, or comparing the Colon dataset in Tables 6 and 7, we can see that the performance based on incomplete data is relatively lower than that based on complete data in general, showing the influence of missing data. On the other hand, however, we can see that most of the results in Table 5 are still in an acceptable range, showing the usefulness of introducing the matrix completion module.

We further investigate the robustness of our ADSL module equipped with the matrix completion module. Table 6 shows the experimental results on the Colon dataset under different sampling rates (SR), i.e., from 100% to 60%. Similar to the experiments in Table 5, all the methods adopt matrix completion. We have the following observations in Table 6. First, as a general trend, the performance drops along with the decreased sampling rate. Second, our method keeps the best performance under all the sampling rates, and the classification accuracy remains at an acceptable level. Unlike the setting in Table 6, we fill the missing data with 0 and the experimental results are shown in Table 7. Compared with Tables 6, Table 7 is generally lower for all the methods. Moreover, as shown in Table 7, as the missing rate increases, the accuracy decreases rapidly especially for SVM classification results, while the accuracy in Table 6 declines more slowly. These results prove the usefulness of the matrix completion module for dealing with the missing data issue in feature selection applications. Of note, as shown in Table 7, we can see that our ADSL still achieves the best performances among all the methods for comparison. As the filled zeros can be seen as noises, these results demonstrate the robustness of our method from another perspective.

In the following, we analyze the influence of the dimension of selected features, as shown in Fig. 2. As for all the five classification tasks, we can see that a higher feature dimension generally brings in better classification accuracy, except for several conditions, e.g., GSSR under the NC vs. MCI task and the Colon task, JGSC and RJGSC under the MCI vs. AD task. On the contrary, our method has an obvious correlation between feature dimension and accuracy, showing better consistency and robustness.

At last, we investigate the converging ability of our ADSL. Fig. 3 shows the convergence curve of the objective function under different datasets and experimental settings. The last row of Fig. 3 shows that the objective function converges on both the full datasets (RELATHE) and the missing datasets with 90% sampling rate (RELATHE(SR=0.9)). All the subfigures show that the optimization process of ADSL performs well in terms of its convergence. For example, in the three Alzheimer's disease classification tasks, the objective function value quickly converges within ten iterations.

7. Conclusion

In this paper, we propose a novel unsupervised feature selection method, ADSL, jointly conducting adaptive sparse dictionary learning and adaptive spectral learning. On the one hand, the new dictionary is updated to obtain better bases. On the other hand, the intrinsic manifold structure is kept on explored through the adaptively updating of the Laplacian graph in the spectral learning process. The above two aspects are encoded into a unified learning model, solved under an alternative optimization framework. During the iterative optimization process, the newly learned representation enables the dictionary and manifold structure to be updated adaptively, promoting a better-learned data representation. In addition, we introduce the matrix completion module into our framework, aiming to make it suitable to handle datasets with missing items. Experimental results on both complete and incomplete datasets demonstrate the effectiveness of the proposed ADSL method.

Our proposed ADSL method still has two shortcomings. The first one is the long computational times, especially for high feature dimensions. The second one is that a two-step method is adopted to process incomplete datasets, which may introduce additional noises. To deal with these problems, in our future work, feature selection and matrix completion can be optimized under a unified framework.

CRediT authorship contribution statement

Yanrong Guo: Conceptualization, Methodology, Supervision, Funding acquisition. **Huihui Sun:** Software, Writing – original draft, Data curation, Validation. **Shijie Hao:** Writing – review & editing, Project administration, Resources, Funding acquisition.

Acknowledgments

This work was supported by National Key Research and Development Program (Grant No. 2019YFA0706200), the National Nature Science Foundation of China under Grant No. 62072152, 62172137, and the Fundamental Research Funds for the Central Universities under Grant No. PA2020GDKC0023 and PA2019GDZC0095. Data collection and sharing for this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: AbbVie, Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Araclon Biotech; BioClinica, Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir, Inc.; Cogstate; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Lumosity; Lundbeck; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's

Therapeutic Research Institute at the University of Southern California. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of Southern California

References

Cai, D., Zhang, C., & He, X. (2010). Unsupervised feature selection for multi-cluster data. In *Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining - KDD '10* (p. 333). ACM Press. https://doi.org/10.1145/1835804.1835848.

Cai, J.-F., Candès, E. J., & Shen, Z. (2010). A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization, 20*, 1956–1982. https://doi.org/10.1137/080738970

Candès, E. J., & Recht, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational Mathematics, 9*, 717. https://doi.org/10.1007/s10208-009-9045-5

Chen, T., Guo, Y., & Hao, S. (2020). Unsupervised feature selection based on joint spectral learning and general sparse regression. *Neural Computing & Applications, 32*, 6581–6589. https://doi.org/10.1007/s00521-019-04117-9

Chen, T., Zhao, Y., & Guo, Y. (2020). Sparsity-regularized feature selection for multi-class remote sensing image classification. *Neural Computing & Applications, 32*, 6513–6521. https://doi.org/10.1007/s00521-019-04046-7

Du, L., & Shen, Y. D. (2015). Unsupervised feature selection with adaptive structure learning. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, association for computing machinery* (pp. 209–218). https://doi.org/10.1145/2783258.2783345

Gan, J., Peng, Z., Zhu, X., Hu, R., Ma, J., & Wu, G. (2021). Brain functional connectivity analysis based on multi-graph fusion. *Medical Image Analysis, 71*, Article 102057. https://doi.org/10.1016/j.media.2021.102057

Ghasemi, H., Malek-Mohammadi, M., Babaie-Zadeh, M., & Jutten, C. (2011). SRF: Matrix completion based on smoothed rank function. In *Proceedings of the IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 3672–3675). https://doi.org/10.1109/ICASSP.2011.5947147

Gu, Q., Wang, Z. W., & Liu, H. (2016). Low-rank and sparse structure pursuit via alternating minimization. In *Proceedings of the 19th international conference onartificial intelligence and statistics, PMLR* (pp. 600–609). http://proceedings.mlr.press/v51/gu16.html accessed July 26, 2021.

Hou, C., Nie, F., Li, X., Yi, D., & Wu, Y. (2014). Joint embedding learning and sparse regression: A framework for unsupervised feature selection. *IEEE Transactions on Cybernetics, 44*, 793–804. https://doi.org/10.1109/TCYB.2013.2272642

Hu, Y., Zhao, C., Cai, D., He, X., & Li, X. (2016). Atom decomposition with adaptive basis selection strategy for matrix completion. *ACM Transactions on Multimedia Computing, Communications, and Applications, 12*, 43:1–43:25. https://doi.org/10.1145/2903716, 1-43:25.

Jain, P., Netrapalli, P., & Sanghavi, S. (2013). Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on theory of computing* (pp. 665–674). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2488608.2488693.

Kim, E., Lee, M., Choi, C. H., Kwak, N., & Oh, S. (2015). Efficient L1-norm-based low-rank matrix approximations for large-scale problems using alternating rectified gradient method. *IEEE Transactions on Neural Networks and Learning Systems, 26*, 237–251. https://doi.org/10.1109/TNNLS.2014.2312535

Kwon, M., & Choi, H. (2020). Learning low-rank representation for matrix completion. In *Proceedings of the IEEE international conference on big data and smart computing (BigComp)* (pp. 161–164). https://doi.org/10.1109/BigComp48618.2020.00-82

Liu, Y., Jiao, L. C., & Shang, F. (2013). A fast tri-factorization method for low-rank matrix recovery and completion. *Pattern Recognition, 46*, 163–173. https://doi.org/10.1016/j.patcog.2012.07.003

Liu, J., Ji, S., & Ye, J. (2009). Multi-task feature learning via efficient ℓ2, 1-norm minimization. In Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, UAI 2009 (pp. 339–348).

Ma, S., Goldfarb, D., & Chen, L. (2011). Fixed point and Bregman iterative methods for matrix rank minimization. *Mathematical Programming, 128*, 321–353. https://doi.org/10.1007/s10107-009-0306-5

Mairal, J., Bach, F., Ponce, J., & Sapiro, G. (2010). Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research, 11*, 19–60.

Mitra, P., Murthy, C. A., & Pal, S. K. (2002). Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*, 301–312. https://doi.org/10.1109/34.990133

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *Proceedings of the 14th international conference on neural information processing systems: Natural and synthetic* (pp. 849–856). Cambridge, MA, USA: MIT Press.

Nie, F., Huang, H., Cai, X., & Ding, C. (2010). Efficient and robust feature selection via joint ℓ2,1-norms minimization. *Proceedings of the 23rd International Conference on Neural Information Processing Systems, 2*, 1813–1821. https://doi.org/10.5555/2997046.2997098

Nie, F., Wang, H., Huang, H., & Ding, C. (2015). Joint Schatten p-norm and Lp-norm robust matrix completion for missing value recovery. *Knowledge and Information Systems, 42*, 525–544. https://doi.org/10.1007/s10115-013-0713-z

Peng, H., & Fan, Y. (2017). A general framework for sparsity regularized feature selection via iteratively reweighted least square minimization. In *Proceedings of the thirty-first AAAI conference on artificial intelligence* (pp. 2471–2477). AAAI Press. https://doi.org/10.5555/3298483.3298595.

Shi, Q., Lu, H., & Cheung, Y. (2018). Rank-one matrix completion with automatic rank estimation via L1-norm regularization. *IEEE Transactions on Neural Networks and Learning Systems, 29*, 4744–4757. https://doi.org/10.1109/TNNLS.2017.2766160

Tanner, J., & Wei, K. (2016). Low rank matrix completion by alternating steepest descent methods. *Applied and Computational Harmonic Analysis, 40*, 417–429. https://doi.org/10.1016/j.acha.2015.08.003

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological), 58*, 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Toh, K. C., & Yun, S. (2010). An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Optimization, 6*, 615–640. https://scholarbank.nus.edu.sg/handle/10635/102811 accessed July 16, 2021.

Wang, L., Zhu, J., & Zou, H. (2008). Hybrid huberized support vector machines for microarray classification and gene selection. *Bioinformatics, 24*, 412–419. https://doi.org/10.1093/bioinformatics/btm579

Wang, Y., Lin, X., Wu, L., Zhang, W., Zhang, Q., & Huang, X. (2015). Robust subspace clustering for multi-view data by exploiting correlation consensus. *IEEE Transactions on Image Processing, 24*, 3939–3949. https://doi.org/10.1109/TIP.2015.2457339

Wang, Y., Zhang, W., Wu, L., Lin, X., Fang, M., & Pan, S. (2016). Iterative views agreement: An iterative low-rank based structured optimization method to multi-view spectral clustering. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2153–2159, 1608.05560 [Cs, Stat] http://arxiv.org/abs/1608.05560 accessed April 18, 2021.

Wang, Z., Lai, M. J., Lu, Z., Fan, W., Davulcu, H., & Ye, J. (2014). Rank-one matrix pursuit for matrix completion. In *, 2. Proceedings of the 31st international conference on machine learning, ICML* (pp. 1260–1268).

Wang, Z., Lai, M. J., Lu, Z., Fan, W., Davulcu, H., & Ye, J. (2015). Orthogonal rank-one matrix pursuit for low rank matrix completion. *Siam Journal of Scientific Computing, 37*, A488–A514. https://doi.org/10.1137/130934271

Wen, Z., Yin, W., & Zhang, Y. (2012). Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation, 4*, 333–361. https://doi.org/10.1007/s12532-012-0044-1

Wu, L., Wang, Y., Gao, J., & Li, X. (2018). Deep adaptive feature embedding with local sample distributions for person re-identification. *Pattern Recognition, 73*, 275–288. https://doi.org/10.1016/j.patcog.2017.08.029

Wu, T. T., & Lange, K. (2015). Matrix completion discriminant analysis. *Computational Statistics & Data Analysis, 92*, 115–125. https://doi.org/10.1016/j.csda.2015.06.006

Xiang, S., Nie, F., Meng, G., Pan, C., & Zhang, C. (2012). Discriminative least squares regression for multiclass classification and feature selection. *IEEE Transactions on Neural Networks and Learning Systems, 23*, 1738–1754. https://doi.org/10.1109/TNNLS.2012.2212721

Yuan, C., Zhong, Z., Lei, C., Zhu, X., & Hu, R. (2021). Adaptive reverse graph learning for robust subspace learning. *Information Processing & Management, 58*, Article 102733. https://doi.org/10.1016/j.ipm.2021.102733

Zhang, C., Han, Z., Cui, Y., Fu, H., Zhou, J. T., Hu, Q., & H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett. (2019). CPM-Nets: Cross partial multi-view networks. *Advances in neural information processing systems*. Curran Associates, Inc.. https://proceedings.neurips.cc/paper/2019/file/11b9842e0a271ff252c1903e7132cd68-Paper.pdf

Zhang, H., Kyaw, Z., Chang, S. F., & Chua, T. S. (2017). Visual translation embedding network for visual relation detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3107–3115. https://doi.org/10.1109/CVPR.2017.331, 1702.08319 [Cs] http://arxiv.org/abs/1702.08319 accessed April 18, 2021.

Zhang, H., Zha, Z., Yang, Y., Yan, S., & Chua, T. (2014). Robust (Semi) nonnegative graph embedding. *IEEE Transactions on Image Processing, 23*, 2996–3012. https://doi.org/10.1109/TIP.2014.2325784

Zhang, Y., Wang, Y., Jin, J., & Wang, X. (2016). Sparse Bayesian learning for obtaining sparsity of EEG frequency bands based feature vectors in motor imagery classification. *International Journal of Neural Systems, 27*, Article 1650032. https://doi.org/10.1142/S0129065716500325

Zhao, Z., & Liu, H. (2007). Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the 24th international conference on machine learning - ICML '07* (pp. 1151–1157). ACM Press. https://doi.org/10.1145/1273496.1273641.

Zhao, Z., Wang, L., & Liu, H. (2010). Efficient spectral feature selection with minimum redundancy. In *, 1. In AAAI-10 / IAAI-10 - Proceedings of the 24th AAAI Conference on Artificial Intelligence and the 22nd Innovative Applications of Artificial Intelligence Conference* (pp. 673–678). AI Access Foundation. https://asu.pure.elsevier.com/en/publications/efficient-spectral-feature-selection-with-minimum-redundancy accessed August 22, 2021.

Zhao, Z., Wang, L., Liu, H., & Ye, J. (2013). On similarity preserving feature selection. *IEEE Transactions on Knowledge and Data Engineering, 25*, 619–632. https://doi.org/10.1109/TKDE.2011.222

Zheng, W., Zhu, X., Wen, G., Zhu, Y., Yu, H., & Gan, J. (2020). Unsupervised feature selection by self-paced learning regularization. *Pattern Recognition Letters, 132*, 4–11. https://doi.org/10.1016/j.patrec.2018.06.029

Zhu, X., Gan, J., Lu, G., Li, J., & Zhang, S. (2020). Spectral clustering via half-quadratic optimization. *World Wide Web, 23*, 1969–1988. https://doi.org/10.1007/s11280-019-00731-8

Zhu, X., Hu, R., Lei, C., Thung, K. H., Zheng, W., & Wang, C. (2019). Low-rank hypergraph feature selection for multi-output regression. *World Wide Web, 22*, 517–531. https://doi.org/10.1007/s11280-017-0514-5

Zhu, X., Li, X., Zhang, S., Ju, C., & Wu, X. (2017). Robust joint graph sparse coding for unsupervised spectral feature selection. *IEEE Transactions on Neural Networks and Learning Systems, 28*, 1263–1275. https://doi.org/10.1109/TNNLS.2016.2521602

Zhu, X., Li, X., Zhang, S., Xu, Z., Yu, L., & Wang, C. (2017). Graph PCA hashing for similarity search. *IEEE Transactions on Multimedia, 19*, 2033–2044. https://doi.org/10.1109/TMM.2017.2703636

Zhu, X., Wu, X., Ding, W., & Zhang, S. (2013). Feature selection by joint graph sparse coding. In *Proceedings of the 2013 SIAM international conference on data mining (SDM), society for industrial and applied mathematics* (pp. 803–811). https://doi.org/10.1137/1.9781611972832.89

Zhu, X., Zhang, S., Hu, R., Zhu, Y., & Song, J. (2018). Local and global structure preservation for robust unsupervised spectral feature selection. *IEEE Transactions on Knowledge and Data Engineering, 30*, 517–529. https://doi.org/10.1109/TKDE.2017.2763618

Zhu, Y., Ma, J., Yuan, C., & Zhu, X. (2022). Interpretable learning based dynamic graph convolutional networks for Alzheimer's disease analysis. *Information Fusion, 77*, 53–61. https://doi.org/10.1016/j.inffus.2021.07.013