Signal Processing: Image Communication 67 (2018) 118–131

Contents lists available at ScienceDirect

Signal Processing: Image Communication

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

Ordinal preserving matrix factorization for unsupervised feature selection

Yugen Yi [a], Wei Zhou [c,*], Qinghua Liu [a], Guoliang Luo [a], Jianzhong Wang [b,e,*], Yuming Fang [d], Caixia Zheng [b,e]

[a] School of Software, Jiangxi Normal University, Nanchang, China
[b] College of Information Science and Technology, Northeast Normal University, Changchun, China
[c] College of Information Science and Engineering, Northeastern University, Shenyang, China
[d] School of Information Technology, Jiangxi University of Finance and Economics, Nanchang, China
[e] Key Laboratory of Applied Statistics of MOE, Northeast Normal University, Changchun, China

ARTICLE INFO

Keywords:
Unsupervised feature selection
Matrix factorization
Ordinal locality structure preserving
Sparsity and low redundancy

ABSTRACT

Feature selection aims to remove the irrelevant and redundant features to reduce the dimensionality of data and increase the efficiency of learning algorithms. Specifically, unsupervised feature selection without any label information has become a challenging and significant task in machine learning applications. In this paper, a novel algorithm called Ordinal Preserving Matrix Factorization (OPMF), which incorporates matrix factorization, ordinal locality structure preserving and inner-product regularization into a unified framework, is proposed for feature selection. The advantages of our algorithm are three-fold. First, the ordinal locality property of original data is preserved by introducing a triplet-based loss function to the selected features, which is of great importance for distance-based classification and clustering tasks. Second, an inner product regularization term is incorporated into the proposed framework, so that the selected features obtained by our OPMF can be sparse and low redundant. Third, a simple and efficient iteratively updating algorithm is derived to solve the objective function of the proposed algorithm. Extensive experimental results on six datasets demonstrate that the proposed OPMF can obtain competitive performance compared to the existing state-of-the-art unsupervised feature selection approaches.

1. Introduction

With the rapid growth of the storage technologies, the amount of available data explodes in sample number and input space dimension [1–3]. However, the high-dimensional data always contain some irrelevant and redundant features, which leads to high computational and space complexity for data processing. Moreover, the irrelevant and redundant features also adversely affect the clustering or classification performance. Hence, as one of typical methods to reduce dimensionality of the data and address aforementioned issues, feature selection has attracted more and more attention in the research community [1–3].

In general, feature selection methods can be categorized into supervised and unsupervised ones in terms of the availability of class label information [4–7]. Supervised feature selection methods search the most discriminative feature subset with the guidance of class label information and have achieved good performance in classification and recognition tasks [8–11]. However, they are not feasible for some real-world applications in which the labels of training data are unavailable. Compared with supervised feature selection, unsupervised feature selection methods determine the subset of selected features by investigating only the intrinsic property or structure of the high-dimensional data [12–14], which makes them more flexible and practical. Therefore, it is much desired to design effective unsupervised feature selection methods for machine learning applications.

Recently, there have been many unsupervised feature selection approaches proposed in the literature [5]. Among them, Variance Score (VS) is a classical and simple unsupervised feature selection algorithm [15]. VS first calculates the variance of each feature and then selects the features with large variances as the optimal feature subset. He et al. took the locality preserving ability of features into account and proposed an unsupervised feature selection approach named as Laplacian Score (LS) [16]. Through LS algorithm, the feature subset that can best maintain the manifold structure of original high-dimensional data can be selected. Zhao et al. incorporated the spectral graph theory into feature selection and proposed a Spectral-Feature Selection (SPEC) approach [17]. SPEC first constructs a similarity graph based

* Corresponding authors.
E-mail addresses: zhouweineu@outlook.com (W. Zhou), wangjz019@nenu.edu.cn (J. Wang).

https://doi.org/10.1016/j.image.2018.06.005
Received 9 October 2017; Received in revised form 6 May 2018; Accepted 11 June 2018
Available online 18 June 2018

(a) Extended YaleB.

(b) CMU PIE.

(c) AR.

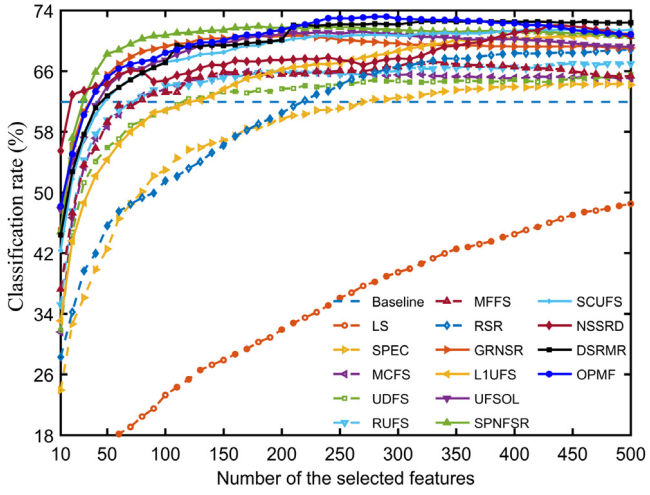
(d) JAFFE.

(e) ORL.

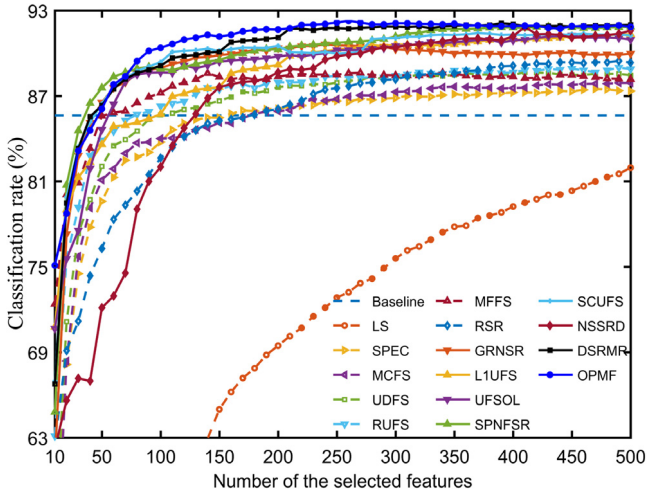
(f) COIL20.

Fig. 1. Some images from the different databases.

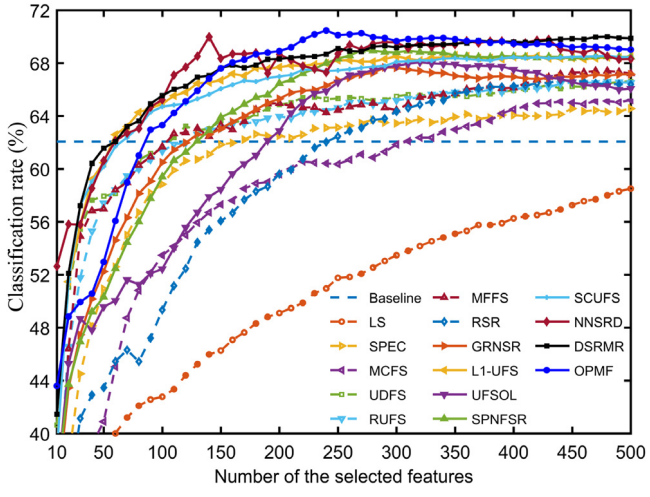
on high-dimensional data and then selects the feature subset which can preserve the structure information of the graph by spectral graph theory. However, the aforementioned algorithms estimate the quality of the features in one-by-one manner and the possible correlations among features are neglected, which may lead the selected feature subset to be sub-optimal [18,19]. To overcome this shortage, several sparsity regularization based approaches have been proposed and shown better performances [18–23]. Cai et al. took the underlying manifold structure and l_1-norm regularization into a unified framework to obtain the optimal feature subset [18]. In [19], an Unsupervised Discriminative Feature Selection (UDFS) algorithm which takes the discriminative information and the correlation between features into consideration was proposed by Yang et al. Moreover, a Nonnegative Discriminative Feature Selection (NDFS) algorithm was also proposed to select the feature subset with nonnegative spectral analysis [20]. In order to handle the outliers or noise in data, a series of unsupervised feature selection approaches based on $l_{2,1}$-norm have been proposed recently [21–27]. In [21], Nie et al. introduced $l_{2,1}$-norm into both the loss function and the feature selection matrix to reduce the effect of outliers and noise. Robust Unsupervised Feature Selection (RUFS) integrated $l_{2,1}$-regularized regression with $l_{2,1}$-norm-based nonnegative matrix factorization into a unified framework to select the most discriminative features while capturing the manifold structure of data [22]. Zhu et al. presented a Regularized Self-representation (RSR) approach for unsupervised feature selection [23]. In RSR, a linear regression model was firstly established so that each feature can be represented as a linear combination of its relevant features. Then, the most representative features were selected by introducing $l_{2,1}$-norm into the linear regression model. For the sake of taking the structure preserving ability of features into account, Yi et al. [24] proposed a Graph Regularized Nonnegative Self-Representation (GRNSR) algorithm which combined the local structure and non-negative $l_{2,1}$-norm sparse regularization for unsupervised feature selection. After that, Zhou et al. integrated the local and global structures, $l_{2,1}$-norm sparse regularization, and the non-negativity constraint into a unified framework and proposed Structure-preserving Nonnegative Feature Self-Representation (SPNFSR) for feature selection [25]. Similarly, Tang et al. [28] employed both $l_{2,1}$-norm and l_1-norm to regularize the feature self-representation and graph Laplacian regularizations respectively, which makes their feature selection model more robust. Different from the most existing methods that generated the local similarity graph by kernel functions, Zhu et al. [29] proposed a Subspace Clustering Guided Unsupervised Feature Selection (SCUFS) by learning a global similarity matrix to capture the multi-subspace structure of data. To better represent the local geometrical structure of data and make the selected features insensitive to the influence caused by parameters, a feature selection algorithm termed as Dual Self-representation and Manifold Regularization (DSRMR) was proposed by Tang et al. [30]. In DSRMR, the similarity graph of data samples is learned by a sample self-representation strategy so that the local geometrical structure of data can be adaptively captured and well preserved. Recently, Shang et al. [31] proposed a Non-negative Spectral Learning and Sparse Regression-based Dual-graph Regularized (NSSRD) approach for feature selection. Since NSSRD leveraged the geometry



(a) Extended YaleB.

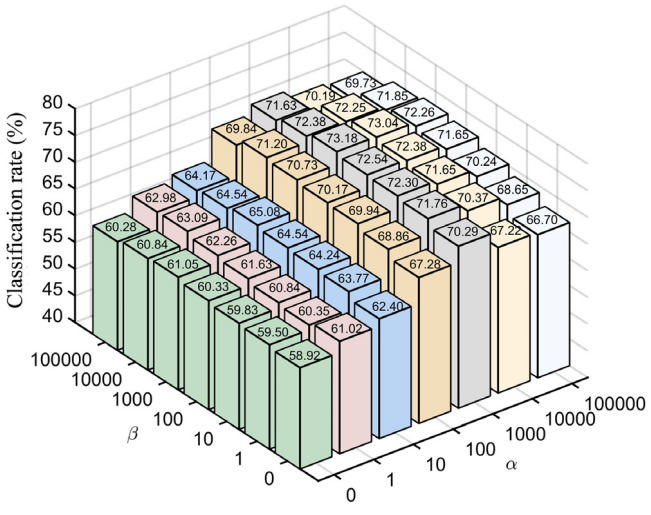


(b) CMU PIE.

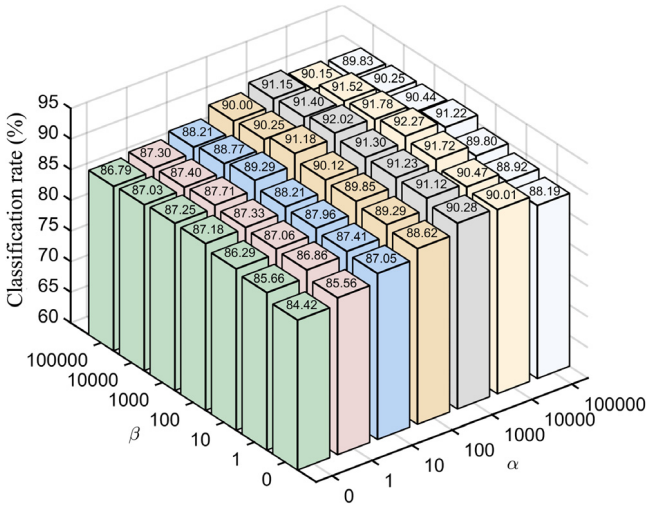


(c) AR.

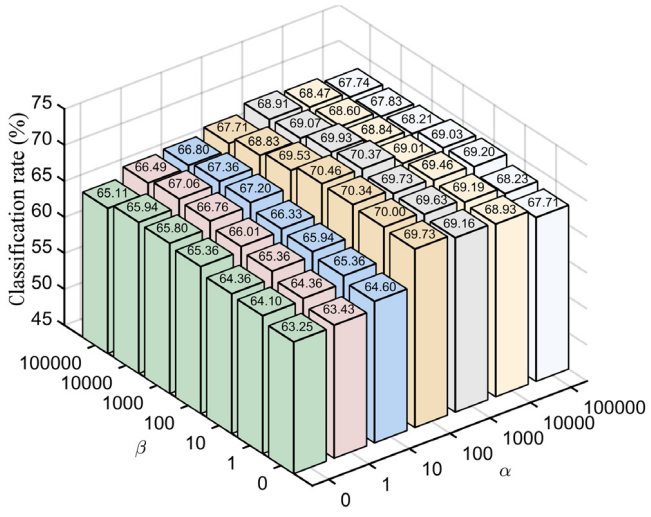
Fig. 2. The classification rates of different algorithms on three databases.



(a) Extended YaleB.



(b) CMU PIE.



(c) AR.

Fig. 3. The classification rate (%) of the proposed OPMF under various values of parameters α and β on three databases.

information in both data space and feature space to guide feature selection, it can select the feature subset with accurate discrimination information.

Nowadays, matrix factorization has attracted a lot of attention for dimensionality reduction problem and many related approaches have been proposed [32–34]. However, the main problem in these

methods is that the low dimensional features obtained by them may lack interpretability. To address this shortcoming, Wang et al. [35] proposed a method called Matrix Factorization based Feature Selection (MFFS), which is developed from the viewpoint of subspace learning. MFFS treats feature selection as a matrix factorization problem and the optimal feature subset can be formed by introducing an orthogonal constraint into its objective function. Although MFFS outperforms some state-of-the-art unsupervised feature selection approaches, there still exist several limitations in it. First, the orthogonal constraint in MFFS is too strict to be satisfied, which may hinder its application in some practical problems [36]. Second, the relative neighborhood proximities of original data are not preserved in MFFS, which will weaken its performance for the classification and clustering tasks. In addition, considering the fact that the correlations among features are neglected in MFFS, some redundancy features will be contained in the selected feature subset, which also makes the feature subset far from optimal [37].

To address these problems and improve the effectiveness of matrix factorization based feature selection, we propose a novel unsupervised feature selection algorithm called Ordinal Preserving Matrix Factorization (OPMF) in this paper. The proposed approach selects the features by incorporating matrix factorization, ordinal locality structure preserving and inner-product regularization into a unified framework. Compared with existing unsupervised feature selection approaches, there are three advantages in the proposed approach. First, we use the concept of ordinal locality structure to preserve the relative neighborhood proximities of original data. That is, the underlying local structures of data will be captured during the process of feature selection, which can improve the performance of feature selection. Moreover, an inner product regularization term that can be regarded as a combination of l_2-norm and l_1-norm on the feature weight matrix is introduced into our algorithm to achieve the characteristics of sparsity and low redundancy among the selected features simultaneously. At last, we design a simple and efficient iteratively updating algorithm to solve the objective function and provide the convergence analysis of our algorithm. Comprehensive experiments on six datasets show that the proposed approach is effective in terms of classification and clustering performances.

The remainder of this paper is organized as follows. In Section 2, we present the proposed approach in detail and provide an effective solution for our algorithm. Then, the related experimental results and analysis are given in Section 3. In the end, the conclusions are drawn in Section 4.

To facilitate the presentation, some notation frequently used throughout this paper is listed in Table 1.

2. The proposed method

In this section, we first present the proposed algorithm in detail. Next, a simple yet efficient iterative update algorithm is provided to solve our algorithm. Then the convergence analysis and the computational complexity of our algorithm are given. Finally, a guideline for parameters setting in our OPMF is also provided.

2.1. OPMF model

Given a high-dimensional original data matrix $X = [x_1; x_2; \dots; x_n] \in R^{n\times d}$, where n is the number of samples, and each sample is a d-dimensional feature vector. In our study, we regard the distance between the spaces spanned by the original high-dimensional data samples and the selected features as the evaluation criterion. Based on this criterion, a number of optimal features can be selected to approximately represent all features. Therefore, the problem of feature selection can be solved by the viewpoint of matrix factorization and formulized as follows:

$$\begin{aligned} &\arg\min_{P,A} \|X - XPA\|_F^2 \\ &s.t. P \geq 0, A \geq 0, P^T P = I_{m\times m}, \end{aligned} \tag{1}$$

where $A \in R^{m\times d}$ denotes the coefficient matrix of the original feature space in the selected feature space, $P \in R^{d\times m}$ represents the feature weight matrix and m indicates the number of selected features. The constraint $P^T P = I_{m\times m}$ aims to guarantee that each element in P is either one or zero, and any row or column of it has at most one non-zero element. Hence, we can regard the matrix P as an indicator matrix of the selected features.

Although Eq. (1) can accomplish the feature selection task, two shortcomings exist in it. For one thing, the relative neighborhood proximities of original data are neglected, which weakens the quality of feature selection. For the other, the strict orthogonality constraint in it is hard to be satisfied in some practical applications [36], so some redundant and correlation features will be selected.

To address the first shortcoming of Eq. (1), we introduce the triplet-based ordinal locality preserving loss function into our model to capture the relative neighborhood proximities of the original data during feature selection. The concept of ordinal locality means a kind of topology information in each sample's neighborhood and its important role for graph based representation has been proved [38,39].

Given an arbitrary original sample x_i, and $y_i = P^T x_i^T$ is the selected feature group, therefore $Y = P^T X^T$. Let a triplet (x_i, x_p, x_q) denotes x_i and its neighbors x_p and x_q. The corresponding selected feature group is represented as (y_i, y_p, y_q). In [38], the feature selection process can be regarded as ordinal locality preserving problem when the following condition holds: if $Dist(x_i, x_p) \leq Dist(x_i, x_q)$, then $Dist(y_i, y_p) \leq Dist(y_i, y_q)$, where $Dist(x, y)$ is a distance metric. Based on the aforementioned ordinal locality preserving property, the feature groups can be determined by optimizing the following ordinal locality preserving loss function:

$$\max_Y \sum_{i=1}^{n} \sum_{p\in N_i} \sum_{q\in N_i} S_{p,q}^i [Dist(y_i - y_p) - Dist(y_i - y_q)], \tag{2}$$

where N_i is the set of k nearest neighbors of x_i. S^i denotes an antisymmetric matrix, where element $S_{p,q}^i$ is equivalent to $Dist(x_i, x_p) - Dist(x_i, x_q)$. Similar to [38], we denote C as a weighting matrix defined as:

$$C_{ij} = \begin{cases} \sum_{p\in N_i} S_{pj}^i, & \forall j \in N_i \\ 0, & \forall j \notin N_i. \end{cases} \tag{3}$$

Therefore, the ordinal locality preserving loss function of Eq. (2) is equivalent to

$$\min_Y \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} Dist(y_i, y_j). \tag{4}$$

Since we denote the $Dist(x, y)$ as a squared Euclidean distance metric in this work, we can rewrite Eq. (4) as:

$$\min_Y \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} \|y_i - y_j\|_2^2. \tag{5}$$

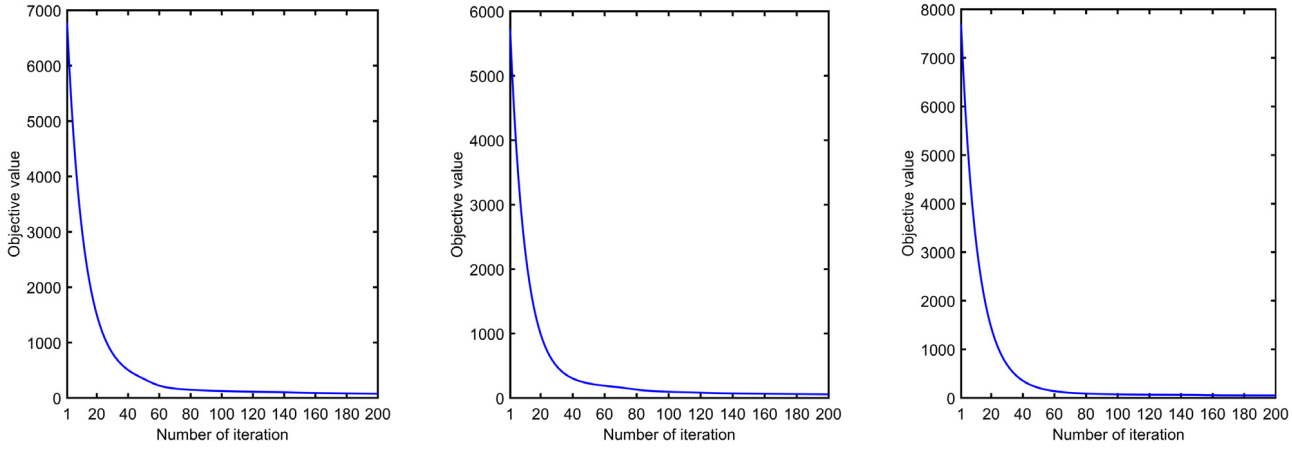
Thus, Eq. (5) has an equivalent compact matrix form as:

$$\min_P tr(YLY^T) = tr(P^T X^T L X P), \tag{6}$$

where $L = D - \frac{C+C^T}{2}$ is a Laplacian matrix, D is a diagonal matrix with $D_{ii} = \sum_{j=1}^{n} \frac{C_{ij}+C_{ji}}{2}$.

In fact, through minimizing Eq. (5), we tend to find a matrix P that preserves the ranking of original sample's neighbors as much as possible after feature selection.

The second shortcoming of Eq. (1) is the strict orthogonality constraint. Actually, this issue can be simply addressed through introducing l_1-norm or $l_{2,1}$-norm regularization with respect to P in Eq. (1). Nevertheless, both sparsity and low redundancy cannot be simultaneously achieved by this simple strategy [40]. Therefore, the optimal feature subset is hard to be obtained. Recently, Han et al. [40] presented a



(a) Extended YaleB. (b) CMU PIE. (c) AR.

Fig. 4. The convergence curves of the proposed approach on three different databases.

Table 1
Some notation used throughout the paper.

Notation	Description	Notation	Description
n	The number of instances	$P \in R^{d\times m}$	The feature weight matrix
d	The number of features	$p_i \in R^{1\times m}$	The ith row of P
m	The number of selected features	p_{ij}	The (i, j)th entry of matrix P
$I_{m\times m}$	The m-by-m identity matrix	$A \in R^{m\times d}$	The coefficient matrix
$1_{d\times d}$	The d-by-d all-ones matrix	A_{ji}	The (j, i)th entry of matrix A

novel regularization term which can be regarded as the combination of matrix l_1 and l_2 norms on the weights of features. And the regularization term is capable of characterizing the independence and saliency of variables. Therefore, we introduce the regularization term into OPMF to relax the strict orthogonality constraint, so that the sparsity and low redundancy can be achieved simultaneously by our algorithm. In the proposed algorithm, we define the regularization term as the absolute values of inner product between feature weight vectors. That is, $|\langle p_i, p_j\rangle|$, in which $p_i \in R^{1\times m}(i = 1, 2, \dots, d)$ is the ith row vector of P. Hence, taking all the weight vectors of P into consideration, the regularization term in our model can be represented as:

$$\begin{aligned}\Omega(P) &= \sum_{i=1}^{d}\sum_{j=1,j\neq i}^{d}|\langle p_i, p_j\rangle| = \sum_{i=1}^{d}\sum_{j=1}^{d}|\langle p_i, p_j\rangle| - \sum_{i=1}^{d}|\langle p_i, p_i\rangle| \\ &= \sum_{i=1}^{d}\sum_{j=1}^{d}|\langle p_i, p_j\rangle| - \sum_{i=1}^{d}\|p_i\|_2^2.\end{aligned} \tag{7}$$

Then, we can rewrite Eq. (6) as a more compact form as:

$$\Omega(P) = (\|PP^T\|_1 - tr(P^TP)) = (\|PP^T\|_1 - \|P\|_2^2). \tag{8}$$

To obtain a low redundant and informative feature subset, we expect the value of Eq. (8) to be as small as possible.

Now, through combining Eqs. (1), (6) and (8), we obtain the objective function of our OPMF as:

$$\begin{aligned}&\min_{P,A}\|X - XPA\|_F^2 + \alpha tr(P^TX^TLXP) + \beta\sum_{i=1}^{d}\sum_{j=1,j\neq i}^{d}|\langle p_i, p_j\rangle| \\ &= \min_{P,A}\|X - XPA\|_F^2 + \alpha tr(P^TQP) + \beta(\|PP^T\|_1 - \|P\|_2^2) \\ &s.t.\ P \geq 0, A \geq 0,\end{aligned} \tag{9}$$

where $Q = X^TLX$, α and β are two tradeoff parameters. In Eq. (9), the first term is to measure the representation ability of selected features; the second term is to ensure that the relative neighborhood proximities of the original data are preserved during feature selection and the third term is to make the feature weight matrix be sparse and low redundant.

Through optimizing Eq. (9), we can learn the feature weight matrix P. Then, we can rank all features according to the value of $\|p_i\|_2$ in descending order and select the top m features as the optimal feature subset.

2.2. Iterative updating algorithm

In our OPMF, there are two variables (i.e., P and A) that are required to be optimized. Nevertheless, the objective function in Eq. (9) is convex in P and A separately but not convex when combining them together. Therefore, we cannot obtain a closed-form solution. To address the problem, an efficient iterative updating algorithm is designed to optimize our model in this subsection.

Optimize P

First, suppose that A is fixed, the optimization problem for P in Eq. (9) can be reduced to

$$\begin{aligned}&\min_P\|X - XPA\|_F^2 + \alpha tr(P^TQP) + \beta(\|PP^T\|_1 - \|P\|_2^2) \\ &s.t.\ P \geq 0.\end{aligned} \tag{10}$$

By simple algebraic manipulations, some irrelevant terms can be removed from Eq. (10). Then, we can rewrite Eq. (10) as:

$$\begin{aligned}&\min_P tr(A^TP^TX^TXPA) - 2tr(A^TP^TX^TX) + \alpha tr(P^TQP) \\ &\quad + \beta(tr(1_{d\times d}PP^T) - tr(P^TP)) \\ &s.t.\ P \geq 0.\end{aligned} \tag{11}$$

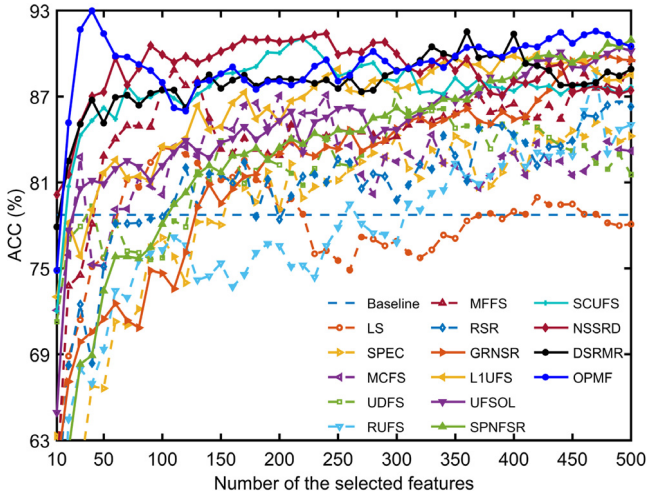
where $1_{d\times d}$ is an all-ones matrix.

To solve the optimization problem in Eq. (11), we introduce the Lagrange multiplier λ to our model and rewrite it as:

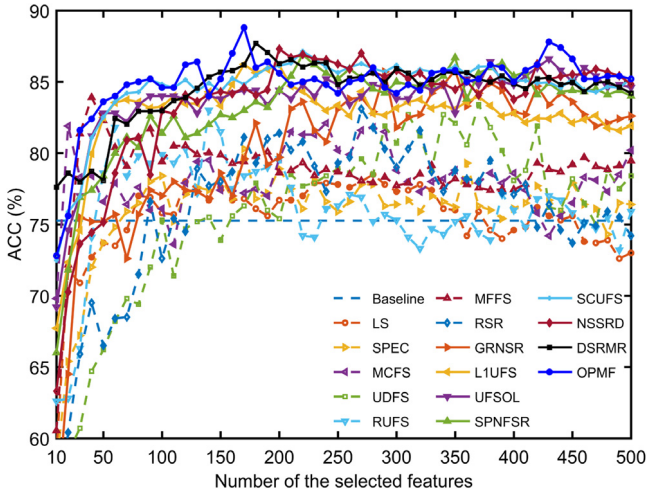
$$\varphi(P, \lambda) = \left\{\begin{matrix} tr(A^TP^TX^TXPA) - 2tr(A^TP^TX^TX) + \alpha tr(P^TQP) \\ +\beta(tr(1_{d\times d}PP^T) - tr(P^TP)) + tr(\lambda P)\end{matrix}\right\}. \tag{12}$$

By taking the derivative of Eq. (12) with respect to P, we get:

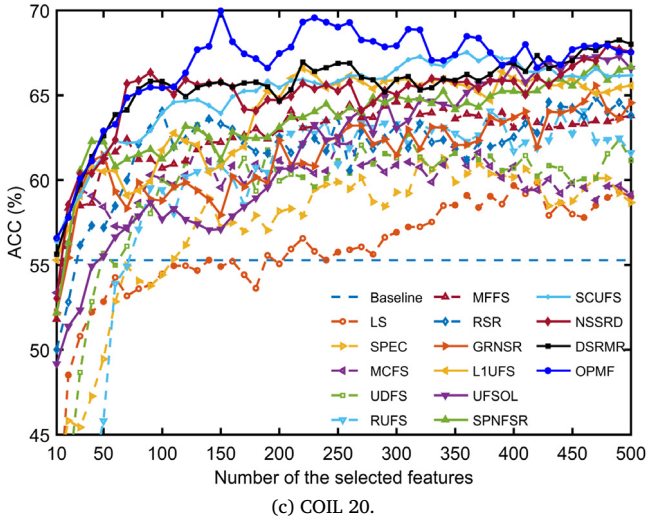
$$\begin{aligned}\frac{\partial\varphi(P,\lambda)}{\partial P} &= -2X^TXA^T + 2X^TXPAA^T + 2\alpha QP \\ &\quad + 2\beta(1_{d\times d}P - P) + \lambda.\end{aligned} \tag{13}$$



(a) JAFFE.

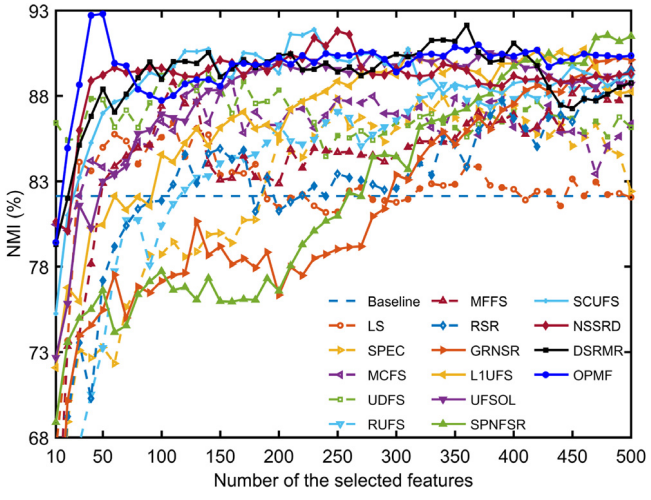


(b) ORL.

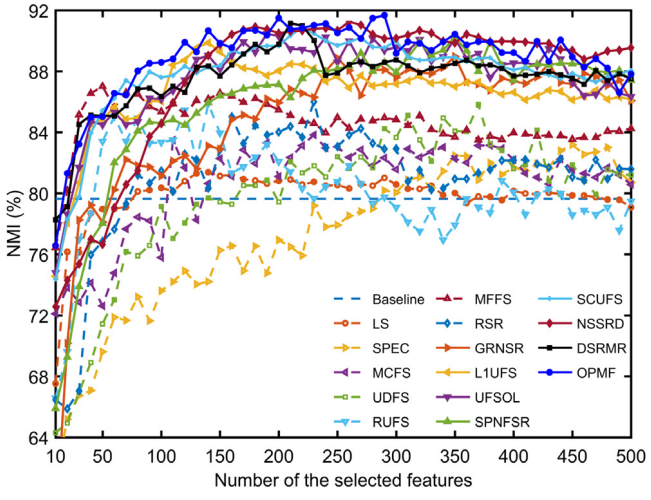


(c) COIL 20.

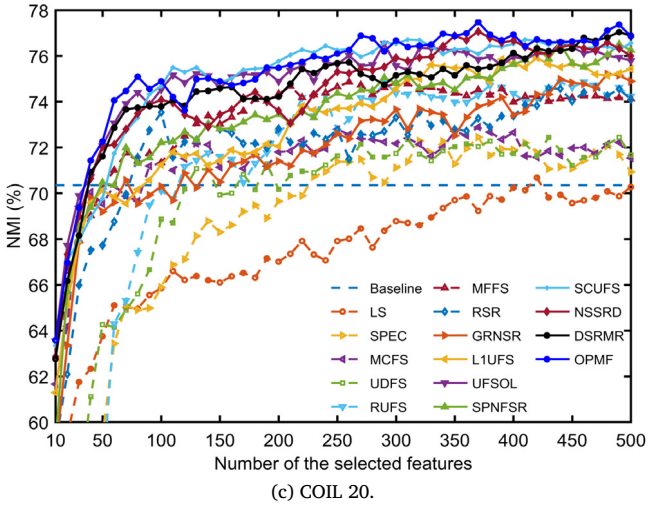
Fig. 5. The ACC (%) of different feature selection algorithms.



(a) JAFFE.



(b) ORL.



(c) COIL 20.

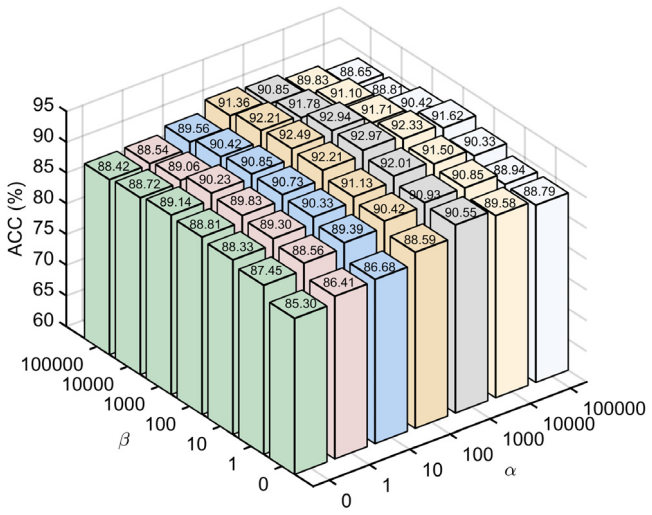
Fig. 6. The NMI (%) of different feature selection algorithms.

Using the Karush–Kuhn–Tucker (KKT) [41] condition $\lambda_{ij} P_{ij} = 0$, we obtain:

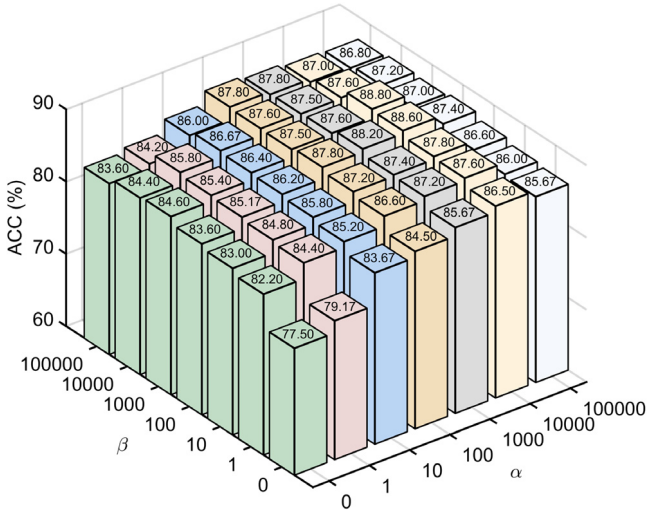
$$(-2X^T X A^T + 2X^T X P A A^T + 2\alpha Q P + 2\beta(1_{d\times d} P - P))_{ij} P_{ij} = 0. \quad (14)$$

Similar to [42], in order to guarantee the non-negativity of P, we define $Q = Q^+ - Q^-$, where

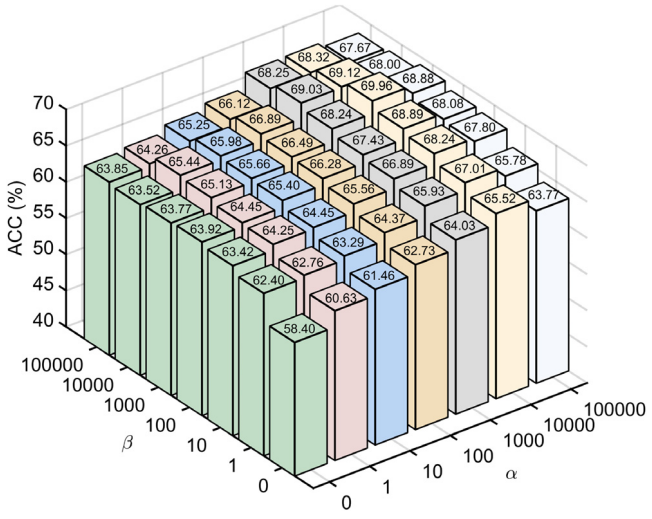
$$Q^+_{ij} = \frac{(|Q_{ij}| + Q_{ij})}{2}, \quad Q^-_{ij} = \frac{(|Q_{ij}| - Q_{ij})}{2}. \quad (15)$$



(a) JAFFE.

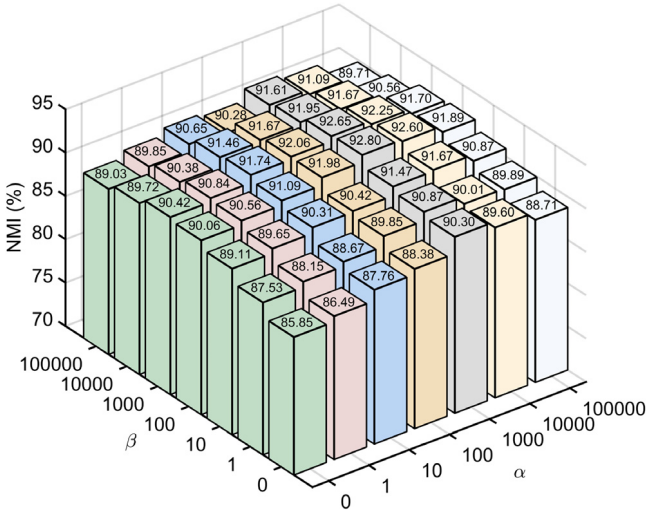


(b) ORL.

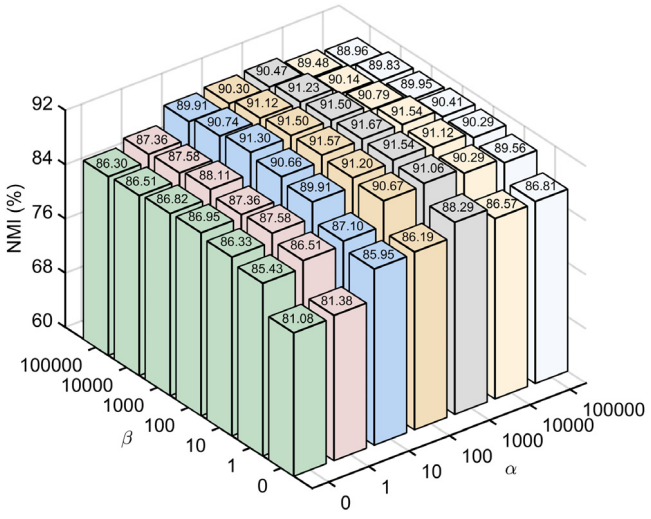


(c) COIL 20.

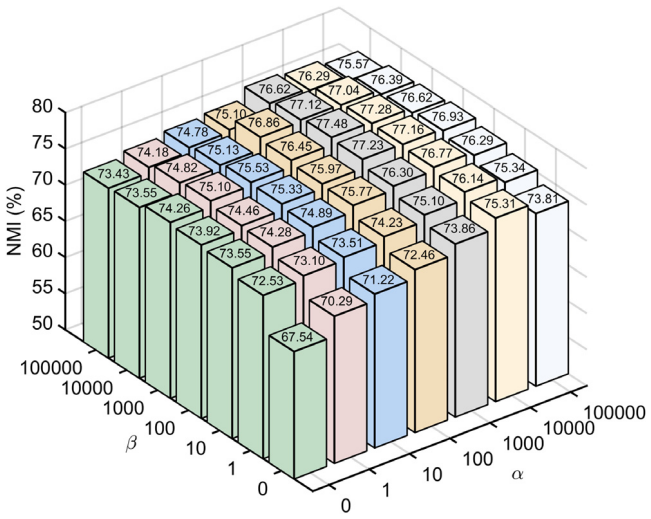
Fig. 7. The ACC (%) of the proposed OPMF vs. parameters α and β on three databases.



(a) JAFFE.



(b) ORL.



(c) COIL 20.

Fig. 8. The NMI (%) of the proposed OPMF vs. parameters α and β on three databases.

By substituting the decomposed positive and negative parts of Eq. (15) into Eq. (14), we obtain the updating rule of P as

$$P_{ij} \leftarrow P_{ij} \frac{(X^T X A^T + \beta P + \alpha Q^- P)_{ij}}{(X^T X P A A^T + \beta 1_{d\times d} P + \alpha Q^+ P)_{ij}}. \tag{16}$$

Optimize A

Then, we fix P to optimize the variable A, the optimization problem about A in Eq. (9) can be reduced to

$$\begin{aligned} &\min_A \|X - XPA\|_F^2 \\ &s.t.\ A \geq 0. \end{aligned} \tag{17}$$

After some algebraic manipulations, Eq. (17) is equivalent to

$$\begin{aligned} &\min_A -2tr(A^T P^T X^T X) + tr(A^T P^T X^T X P A) \\ &s.t.\ A \geq 0. \end{aligned} \tag{18}$$

Likewise, by introducing a Lagrange multiplier ϑ into the constraint $A \geq 0$, we can rewrite Eq. (18) as:

$$\varphi(A, \vartheta) = -2tr(A^T P^T X^T X) + tr(A^T P^T X^T X P A) + tr(\vartheta A). \tag{19}$$

The derivative of Eq. (19) with respect to A, we get:

$$\frac{\partial \varphi(A, \vartheta)}{\partial A} = -2P^T X^T X + 2P^T X^T X P A + \vartheta. \tag{20}$$

Using the Karush–Kuhn–Tucker (KKT) [41] condition $\vartheta_{ij} A_{ij} = 0$, we obtain:

$$(-2P^T X^T X + 2P^T X^T X P A)_{ij} A_{ij} = 0. \tag{21}$$

The updating rule of A is as follows:

$$A_{ij} \leftarrow A_{ij} \frac{(P^T X^T X)_{ij}}{(P^T X^T X P A)_{ij}}. \tag{22}$$

The whole procedure of our algorithm is summarized in Algorithm 1.

Algorithm 1. OPMF

Input: The data matrix $X \in R^{n\times d}$, balance parameters α and β, number of selected features m, each sample's nearest neighborhood size k.
Output: An index set $\{i_1, i_2, \ldots, i_m\}$ of the selected features.

Stage one: Graph construction
1. Construct the nearest neighborhood graph;
2. Compute the weighting matrix C using Eq. (3) and Laplacian matrix L;

Stage two: Alternative optimization
1. Initialize: $P \in R^{d\times m}$ and $A \in R^{m\times d}$;
2. Alternatively update P and A until convergence.
   a. Fix A, update P by Eq. (16);
   b. Fix P, update A by Eq. (22);

Stage three: Feature selection
1. Calculate all feature weight values based on $\|p_i\|_2$ $(i = 1, 2, \ldots, d)$;
2. Sort them in descending order and select the top m features to form the optimal feature subset;
3. The selected feature index set $\{i_1, i_2, \ldots, i_m\}$ is returned.

2.3. Convergence analysis

In this subsection, we analyze the convergence of the updating rules in Eqs. (16) and (22).

Theorem 1. *For $P \geq 0, A \geq 0$, the value of the objective function in Eq. (9) is non-increasing and has a lower boundary under the updating rules in Eqs. (16) and (22).*

To prove Theorem 1, we incorporate an auxiliary function similar to Ref. [43], which is defined as follows:

Definition 1. $\phi(v, v')$ is an auxiliary function of $\psi(v)$ if conditions $\phi(v, v') \geq \psi(v)$ and $\phi(v, v) = \psi(v)$ are satisfied.

The auxiliary function is useful because of the following lemma:

Lemma 1. *If ϕ is an auxiliary function of ψ; then, ψ is non-increasing under the following updating rule:*

$$v^{(c+1)} = \arg\min_v \phi(v, v^{(c)}) \tag{23}$$

where c indicates the cth iteration.

Proof. $\psi(v^{(c+1)}) \leq \phi(v^{(c+1)}, v^{(c)}) \leq \phi(v^{(c)}, v^{(c)}) = \psi(v^{(c)})$. □

First, we need to prove that the updating rule for variable P in Eq. (16) is consistent with Eq. (23) when an auxiliary function is properly designed. We define $\psi_{ij}(P_{ij})$ as the part of objective function Eq. (9) that is only related to P_{ij}. Therefore, we have:

$$\begin{aligned} \psi_{ij}(P_{ij}) = &(A^T P^T X^T X P A - A^T P^T X^T X + \alpha P^T Q P \\ &+ \beta(1_{d\times d} P P^T - P^T P))_{ij}, \end{aligned} \tag{24}$$

$$\begin{aligned} \nabla\psi_{ij}(P_{ij}) = &(-2X^T X A^T + 2X^T X P A A^T + 2\alpha Q P \\ &+ 2\beta 1_{d\times d} P - 2\beta P)_{ij}, \end{aligned} \tag{25}$$

$$\nabla^2\psi_{ij}(P_{ij}) = 2(X^T X)_{ii}(A^T A)_{jj} + 2Q_{ii} + 2\beta(1_{d\times d} - I)_{ii}, \tag{26}$$

where $\nabla\psi_{ij}(P_{ij})$ and $\nabla^2\psi_{ij}(P_{ij})$ represent the first-order and second-order derivatives of the objective function ψ_{ij} with respect to P_{ij}, respectively. I is the identity matrix.

Lemma 2. *The following function in Eq. (27) is a reasonable auxiliary function of $\psi_{ij}(P_{ij})$.*

$$\begin{aligned} \phi(P_{ij}, P_{ij}^{(c)}) = &\ \psi_{ij}(P_{ij}^{(c)}) + \nabla\psi_{ij}(P_{ij}^{(c)})(P_{ij} - P_{ij}^{(c)}) \\ &+ \frac{(X^T X P A A^T + \beta 1_{d\times d} P + \alpha Q^- P)_{ij}}{P_{ij}^{(c)}} (P_{ij} - P_{ij}^{(c)})^2. \end{aligned} \tag{27}$$

Proof. Through the Taylor series expansion of $\psi_{ij}(P_{ij})$, we can get:

$$\begin{aligned} \psi_{ij}(P_{ij}) &= \left\{ \begin{array}{c} \psi_{ij}(P_{ij}^{(c)}) + \nabla\psi_{ij}(P_{ij}^{(c)})(P_{ij} - P_{ij}^{(c)}) \\ + \frac{1}{2}\nabla^2\psi_{ij}(P_{ij}^{(c)})(P_{ij} - P_{ij}^{(c)})^2 \end{array} \right\} \\ &= \left\{ \begin{array}{c} \psi_{ij}(P_{ij}^{(c)}) + \nabla\psi_{ij}(P_{ij}^{(c)})(P_{ij} - P_{ij}^{(c)}) \\ + (X^T X)_{ii}(A^T A)_{jj} + \alpha Q_{ii} + \beta(1_{d\times d} - I)_{ii}(P_{ij} - P_{ij}^{(c)})^2 \end{array} \right\}. \end{aligned} \tag{28}$$

By comparing Eq. (27) with Eq. (28), we can find that $\phi(P_{ij}, P_{ij}^{(c)}) \geq \psi_{ij}(P_{ij})$ is equivalent to the following inequality:

$$\begin{aligned} &\frac{(X^T X P A A^T + \beta 1_{d\times d} P + \alpha Q^+ P)_{ij}}{P_{ij}^{(c)}} \\ &\geq (X^T X)_{ii}(A^T A)_{jj} + \alpha Q_{ii} + \beta(1_{d\times d} - I)_{ii}. \end{aligned} \tag{29}$$

According to linear algebra, we can obtain:

$$\begin{aligned} (X^T X P A A^T)_{ij} &= \sum_{l=1}^{m} (X^T X P^{(c)})_{il}(A^T A)_{lj} \geq (X^T X P^{(c)})_{ij}(A^T A)_{jj} \\ &\geq \sum_{l=1}^{d} (X^T X)_{il} P_{lj}^{(c)} (A A^T)_{jj} \\ &\geq (X^T X)_{ii} P_{ij}^{(c)} (A A^T)_{jj} = P_{ij}^{(c)} (X^T X)_{ii} (A A^T)_{jj}, \end{aligned} \tag{30}$$

$$\begin{aligned} \alpha(Q^+ P)_{ij} &= \alpha \sum_{l=1}^{d} (Q^+)_{il}(P^{(c)})_{lj} \geq \alpha(Q^+)_{ii}(P^{(c)})_{ij} \\ &\geq \alpha(Q^+ - Q^-)_{ii} P_{ij}^{(c)} = \alpha Q_{ii} P_{ij}^{(c)}, \end{aligned} \tag{31}$$

$$\begin{aligned} \beta(1_{d\times d} P)_{ij} &= \beta \sum_{l=1}^{d} (1_{d\times d})_{il} P_{lj}^{(c)} \geq \beta \sum_{l=1}^{d} (1_{d\times d} - I)_{ii} P_{ij}^{(c)} \\ &\geq \beta(1_{d\times d} - I)_{ii} P_{ij}^{(c)}. \end{aligned} \tag{32}$$

From Eqs. (30), (31) and (32), we observe that Eq. (29) holds and $\phi(P_{ij}, P_{ij}^{(c)}) \geq \psi_{ij}(P_{ij})$. Besides, $\phi(P_{ij}, P_{ij}) = \psi_{ij}(P_{ij})$ is obvious. Therefore, Lemma 2 is proved. □

Next, we analyze the variable A in the same way. Here, we use $\psi_{ij}(A_{ij})$ to denote the part of Eq. (9) that is only related to A_{ij}. Then, we get:

$$\psi_{ij}(A_{ij}) = (-2A^T P^T X^T X + A^T P^T X^T X P A)_{ij}, \tag{33}$$

$$\nabla\psi_{ij}(A_{ij}) = (-2P^T X^T X + 2P^T X^T X P A)_{ij}, \tag{34}$$

$$\nabla^2\psi_{ij}(A_{ij}) = 2(P^T X^T X P)_{ii}, \tag{35}$$

where $\nabla\psi_{ij}(A_{ij})$ and $\nabla^2\psi_{ij}(A_{ij})$ represent the first-order and second-order derivatives of ψ_{ij} with respect to variable A_{ij}, respectively.

Lemma 3. *The following function is a reasonable auxiliary function of* $\psi_{ij}(A_{ij})$.

$$\begin{aligned}\phi(A_{ij}, A_{ij}^{(c)}) &= \psi_{ij}(A_{ij}^{(c)}) + \nabla\psi_{ij}(A_{ij}^{(c)})(A_{ij} - A_{ij}^{(c)}) \\ &\quad + \frac{(P^T X^T X P A)_{ij}}{A_{ij}^{(c)}}(A_{ij} - A_{ij}^{(c)})^2.\end{aligned} \tag{36}$$

Proof. Through the Taylor series expansion of $\psi_{ij}(A_{ij})$, we get:

$$\begin{aligned}\psi(A_{ij}) &= \psi_{ij}(A_{ij}^{(c)}) + \nabla\psi_{ij}(A_{ij}^{(c)})(A_{ij} - A_{ij}^{(c)}) \\ &\quad + \frac{1}{2}\nabla^2\psi_{ij}(A_{ij}^{(c)})(A_{ij} - A_{ij}^{(c)})^2 \\ &= \psi_{ij}(A_{ij}^{(c)}) + \nabla\psi_{ij}(A_{ij}^{(c)})(A_{ij} - A_{ij}^{(c)}) \\ &\quad + (P^T X^T X P)_{ii}(A_{ij} - A_{ij}^{(c)})^2.\end{aligned} \tag{37}$$

By comparing Eq. (36) with Eq. (37), it is easy to find that $\phi(A_{ij}, A_{ij}^{(c)}) \geq \psi_{ij}(A_{ij})$ is equivalent to the following inequality:

$$\frac{(P^T X^T X P A)_{ij}}{A_{ij}^{(c)}} \geq (P^T X^T X P)_{ii}. \tag{38}$$

After the linear algebra, we have:

$$(P^T X^T X P A)_{ij} = \sum_{l=1}^{m}(P^T X^T X P)_{il} A_{lj}^{(c)} \geq (P^T X^T X P)_{ii} A_{ij}^{(c)}. \tag{39}$$

From Eq. (39), we know that Eq. (38) holds and $\phi(A_{ij}, A_{ij}^{(c)}) \geq \psi_{ij}(A_{ij})$. Considering that $\phi(A_{ij}, A_{ij}) = \psi_{ij}(A_{ij})$ is easily checked, Lemma 3 is proved. □

At last, we will give the proof of the convergence of Theorem 1.

Proof of Theorem 1. By using the auxiliary function in Eq. (27) to replace $\phi(v, v^{(c)})$ in Eq. (23), we get:

$$\begin{aligned}P_{ij}^{(c+1)} &= P_{ij}^{(c)} - P_{ij}^{(c)} \frac{\nabla\psi_{ij}(P_{ij}^{(c)})}{2(X^T X P A A^T + \alpha Q^+ P + \beta 1_{d\times d} P)_{ij}} \\ &= P_{ij}^{(c)} \frac{(X^T X A^T + \alpha Q^- P + \beta P)_{ij}}{(X^T X P A A^T + \alpha Q^+ P + \beta 1_{d\times d} P)_{ij}}.\end{aligned} \tag{40}$$

Similarly, by using the auxiliary function in Eq. (36) to replace $\phi(v, v^{(c)})$ in Eq. (23), we obtain:

$$A_{ij}^{(c+1)} = A_{ij}^{(c)} - A_{ij}^{(c)} \frac{\nabla\psi_{ij}(A_{ij}^{(c)})}{2(P^T X^T X P A)_{ij}} = A_{ij}^{(c)} \frac{(P^T X^T X)_{ij}}{(P^T X^T X P A)_{ij}}. \tag{41}$$

Since Eqs. (27) and (36) are the auxiliary functions of ψ_{ij}, ψ_{ij} is non-increasing under the updating rules in Eqs. (16) and (22). In the end, our objective function has a lower bound due to all terms in Eq. (9) are greater than zero. Hence, the proposed algorithm is convergent via Cauchy's convergence rule [44].

2.4. Computational complexity analysis

In this section, we analyze the computation complexity of the proposed algorithm. First, the computation complexity of constructing the weighting matrix C among samples is $O(d^2 n)$. Then, the cost of each iteration in Algorithm 1 is equal to $O(d^2 m + d^2 n)$, where m is the number of the selected features. Therefore, the total computation complexity of our algorithm is equal to $O(d^2 n + t(d^2 m + d^2 n))$, where t is the number of iterations. Moreover, the computation complexities of other related algorithms are also listed in Table 2. In this table, n is the number of samples, d is the number of features, p represents the dimension of the embedding space, m is the number of the selected features, k is the number of nearest neighbors and t_1 is the number of iterations for solving the LRR problem in SPNFSR. From this table, the computational complexity of OPMF is lower than those of L1UFS, UFSOL, SUCFS and DSRMR.

Table 2
The computation complexities of different algorithms.

Algorithms	Computational complexity (O)
LS [16]	$O(dn^2)$
SPEC [17]	$O(dn^2)$
MCFS [18]	$O(dn^2 + pm^3 + pnm^2)$
UDFS [19]	$O(n^2 + td^3)$
RUFS [22]	$O(t(n^2 + nd))$
MFFS [35]	$O(t(dm^2 + nd^2 + md^2))$
RSR [23]	$O(t(d^3 + dn^2))$
GRNSR [24]	$O(n^2 d + dnk^3 + t(\min(n, d)dn))$
L1UFS [28]	$O(n^3 + t\min(n, d)^3)$
UFSOL [38]	$O(td^3)$
SPNFSR [25]	$O(t_1 nd^2 + t_1 n^3 + t(\min(n, d)dn))$
SCUFS [29]	$O(t(dn^3 + \max(n, d)d^2))$
DSRMR [30]	$O(t(dn^3 + \min(n, d)^3))$
NSSRD [31]	$O(dn^2 + d^2 n + tpdn)$
OPMF	$O(d^2 n + t(d^2 n + d^2 m))$

2.5. Guideline for parameter setting

The tradeoff parameters α and β are used to adjust the importance of the ordinal locality structure preserving and inner product regularization terms in our model. Thus, their values should be set according to the characteristic of employed database. Specifically, if the samples from the same class are similar to each other and easily to be separated from samples of other classes in a dataset, it is suitable to set a large value for α so that the locality information of data can be well preserved. On the contrary, if the neighbors of current sample belong to different classes, a smaller α is more appropriate. For the parameter β, it is used to control the correlation and redundancy of the selected features obtained by our OPMF. Therefore, its value should be set as relatively small when the original features in a dataset contain small redundancy. Otherwise, we should set a relative large value for it.

3. Experimental results and analysis

In this section, we conduct classification and clustering experiments to evaluate the performance of the proposed approach.

3.1. Database

Six publicly available databases, including Extended YaleB [45], CMU PIE [46], AR [47], JAFFE [48], ORL [49], and COIL20 [50], are used in our experiments to compare the performance of our approach with those of other unsupervised feature selection approaches. Detailed descriptions of these databases are given in Table 3 and some examples of these databases are shown in Fig. 1.

(1) Extended YaleB face database [45]: it contains 2414 frontal cropped facial images belonging to 38 individuals, i.e., each subject has 64 images with the size of 32 × 32 pixels.
(2) CMU PIE face dataset [46]: it contains 41,368 images of 68 human subjects. The images were captured with different poses, illumination conditions, and expressions. We choose a subset (C29) of this database that contains 24 images of each person with only lighting change in our experiment.

Table 3
Database description.

Database	No. of instances	No. of features	No. of classes	Domain
Extended YaleB	2432	1024	38	Face
CMU PIE	1632	1024	24	Face
AR	1400	1024	14	Face
JAFFE	213	1024	10	Face
ORL	400	1024	40	Face
COIL20	1440	1024	20	Object

(3) AR face database [47]: it consists of 4000 facial images from 126 individuals (70 male and 56 female faces). Each subject has 26 facial images, which were captured with several expressions (anger, smiling, and screaming), varying illumination conditions, and some occlusions (sun glasses and scarf). In our work, we choose a subset that contains 14 images of each person with several expressions and varying illumination conditions.
(4) JAFFE face database [48]: it has 213 facial images that depict ten Japanese female models. Each image is depicted with seven kinds of facial expressions.
(5) ORL face database [49]: it is comprised of 400 images that depict 40 distinct subjects. They were taken at different times, with varying lighting and facial expressions as well as facial details.
(6) COIL20 database [50]: it contains 1440 images of 20 objects viewed from varying angles at intervals of 5^0. Each object has 72 images with the size of 32×32 pixels.

3.2. Experimental settings

In our experiments, we choose some classical and state-of-the-art unsupervised feature selection algorithms to compare and evaluate the performance of our proposed approach, these comparison algorithms include LS [16], SPEC [17], MCFS [18], UDFS [19], RUFS [22], RSR [23], GRNSR [24], SPNFSR [25], MFFS [35], UFSOL [38], L1UFS [28], SCUFS [29], NSSRD [30] and DSRMR [31]. We also utilize all the features to perform classification and clustering as Baseline algorithm. For LS, MCFS, SPEC, UDFS, GRNSR, UFSOL, L1UFS, NSSRD and OPMF, the number of neighborhoods is set as 5 on all the databases. The sparsity parameters are tuned by a grid-search strategy from $\{10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3\}$ for all compareds methods except MFFS. According to [35], the value of parameter in MFFS is fixed to be 10^8. For our approach, we tune the values of parameters α and β from $\{0, 10^0, 10^1, 10^2, 10^3, 10^4, 10^5\}$ on all databases and report the best results with standard deviations obtained by the optimal parameters. To evaluate the effectiveness of selected features, we use classification rate as the evaluation criterion for classification experiment and clustering accuracy (ACC) and normalized mutual information (NMI) for clustering experiment in this paper. All algorithms are implemented by Matlab 2012a and executed on a *desktop* computer with Inter (R) Core (TM) i7-4970 CPU@3.60 GHz and 8 GB RAM.

3.3. Classification results and analysis

In this subsection, three databases, including Extended YaleB, CMU PIE and AR, are used for classification experiment. For each database, we randomly select l images as the training samples (Extended YaleB ($l = 20$), CMU PIE ($l = 12$), and AR ($l = 7$)), and the remaining images are regarded as the testing samples. In this experiment, we repeat the process of sample selection 10 times and the average classification results and standard deviations of different algorithms are reported in Table 4. The nearest neighbor classifier (NNC) with Euclidean distance is used for classification due to its simplicity. Moreover, the running time of different algorithms are also listed in Table 4.

From Table 4, several interesting points can be observed as follows. (1) Most of the feature selection approaches except LS perform better than the baseline approach, which indicates feature selection plays an important role to improve the classification performance. (2) The performances of LS and SPEC are worse than other methods. The main reason is that LS and SPEC select the features in a one-by-one manner and ignore the correlations between the features. (3) GRNSR, L1UFS, SPNFSR, SCUFS, NSSRD and DSRMR consider the local structure of data during the procedure of feature selection, thus they achieve better performances than MCFS, UDFS, RUFS and MMFS. (4) Since UFSOL selects the features which can preserve the ordinal locality structure of data, its performance is superior to most of the compared methods. This phenomenon demonstrates that ordinal locality structure of data is very helpful for feature selection. (5) DSRMR integrates the processes of feature selection and similarity matrix learning into a unified framework. Therefore, its performance is better than GRNSR, L1UFS, SPNFSR and SCUFS. (6) The proposed OPMF outperforms other unsupervised feature selection approaches on all three databases. This is due to our algorithm incorporates the matrix factorization, ordinal locality structure preserving and inner-product regularization into a unified framework for feature selection. As a result, the feature subset selected by our algorithm not only preserves the ordinal locality structure of original data, but also contains low redundancy. (7) Since the iterative updating strategy is utilized to optimize the proposed OPMF, its running time is longer than some classical non-iterative approaches such as LS, SPEC and MCFS. However, we can also find that the running time of our algorithm is less than SCUFS and DSRMR, which is consistent with the computational complexity analysis in Section 2.4.

The classification rates under various numbers of selected features obtained by all feature selection approaches are shown in Fig. 2. First, it can be seen that the classification performances of all algorithms are improved with the increase of the number of selected features. Nevertheless, after achieving their best performances, the recognition rates of most algorithms begin to be stable. Second, we can find that the performances of matrix factorization based approaches including MFFS and our OPMF are inferior to some other methods when the number of selected features is relatively small. The main reason may lie in that the space spanned by only a small number of features cannot approximate the space spanned by original input samples. Thus, the information of high-dimensional data is not sufficiently maintained.

In order to test the influence of parameters α and β to the proposed approach, the recognition results of OPMF under different parameter values are evaluated in Fig. 3. Firstly, we can see that the proposed OPMF performs worse when the values of α and β are set to zero. This is due to the zero parameter values would lead our algorithm reduce to traditional matrix factorization. Thus, the ordinal locality property and redundancy of the selected features are both neglected. Secondly, it can be found that the proposed algorithm performs better when the values of parameters are neither too large nor too small. The reason to this phenomenon is that a large α will make the objective function of our model be dominated by the second term, thus matrix factorization and inner product regularization terms will be neglected. Similarity, a large parameter β will overemphasize the inner product regularization term and meanwhile overlook the other two terms. At last, we can find that our OPMF obtains its best performances under relatively larger α and β values on Extended YaleB and CMU PIE databases. The reasons may lie in two aspects. On the one hand, Extended YaleB and CMU PIE databases contain more samples with less variations than AR database. Thus, the samples from the same class are more likely to be adjacent in the feature space and a larger α value could benefit the ordinal locality preservation in our algorithm. On the other hand, since the uninformative and redundant face components such as cheek and chin take up more area in the images of Extended YaleB and CMU PIE databases than those in AR database, a larger β value is more preferred for our algorithm to reduce the redundancy. Moreover, the running times of our algorithm under various parameter values are also provided in Tables 5–7.

In the end, we give the convergence curves of the proposed approach on three different databases, as shown in Fig. 4. From this figure, we can learn that the proposed approach converges fast on all the databases.

Table 4
The best average classification rates (%) ± standard deviations (%) of different algorithms on three databases. The best results are highlighted in bold.

Methods	Extended YaleB	CMU PIE	AR
Baseline	61.93 ± 0.81(1024)	85.63 ± 0.72(1024)	62.06 ± 1.62(1024)
LS	48.50 ± 1.42(500,0.09)	81.96 ± 1.80(500,0.14)	58.51 ± 1.55(500,0.11)
SPEC	64.18 ± 0.96(500,5.11)	87.49 ± 0.82(470,7.71)	64.56 ± 1.54(500,8.45)
MCFS	65.89 ± 1.78(200,0.31)	87.91 ± 0.84(490,0.28)	65.21 ± 1.58(500,0.23)
UDFS	65.54 ± 2.33(500,41.3)	88.66 ± 0.93(410,43.1)	66.60 ± 1.50(500,42.9)
RUFS	66.97 ± 1.32(480,30.1)	88.99 ± 0.91(490,263)	66.61 ± 1.71(480,246)
MFFS	67.22 ± 0.93(330,62.6)	88.62 ± 0.88(270,68.1)	67.41 ± 1.47(470,64.6)
RSR	68.83 ± 1.06(500,30.3)	89.37 ± 0.85(440,34.7)	66.71 ± 1.47(440,32.4)
GRNSR	70.69 ± 0.79(210,49.3)	90.62 ± 1.13(240,51.2)	67.67 ± 1.38(290,48.3)
L1UFS	70.96 ± 1.16 (500,128)	91.33 ± 1.14 (500,138)	68.59 ± 1.65 (350,131)
UFSOL	71.20 ± 1.10(270,318)	91.25 ± 1.54(450,331)	68.05 ± 1.40(320,305)
SPNFSR	71.90 ± 0.77(180,79.8)	91.84 ± 1.03(430,102)	68.95 ± 1.17(270,92.6)
SCUFS	71.47 ± 1.07(410,1309)	91.40 ± 0.97(370,1383)	68.56 ± 0.88(480,1271)
NSSRD	72.02 ± 1.19(440,5.49)	91.85 ± 0.96(410,6.18)	69.79 ± 1.52(430,5.54)
DSRMR	72.58 ± 0.97(340,594)	92.10 ± 1.14(390,975)	70.00 ± 1.94(480,710)
OPMF	**73.18 ± 1.03(290,65.7)**	**92.27 ± 0.57(260,71.7)**	**70.46 ± 1.42(240,72.4)**

Note that the numbers in parentheses are the number of the selected features that correspond to the best classification result and the running time (s).

Table 5
The running time (s) of the proposed approach under different parameters α and β on the Extended YaleB database.

	$\alpha = 0$	$\alpha = 1$	$\alpha = 10$	$\alpha = 100$	$\alpha = 1000$	$\alpha = 10000$	$\alpha = 100000$
$\beta = 0$	68.20	87.24	87.61	87.64	91.48	95.49	102.06
$\beta = 1$	76.14	78.65	88.40	89.51	93.50	97.48	102.67
$\beta = 10$	72.69	70.20	78.60	87.75	92.49	94.48	101.60
$\beta = 100$	75.14	76.15	77.33	80.32	88.34	92.60	99.15
$\beta = 1000$	73.72	74.17	76.64	78.04	78.00	80.98	91.13
$\beta = 10000$	71.16	72.54	73.48	77.78	78.94	76.82	79.81
$\beta = 100000$	65.16	70.81	76.26	75.39	77.68	70.12	77.79

Table 6
The running time (s) of the proposed approach under different parameters α and β on the CMU PIE database.

	$\alpha = 0$	$\alpha = 1$	$\alpha = 10$	$\alpha = 100$	$\alpha = 1000$	$\alpha = 10000$	$\alpha = 100000$
$\beta = 0$	30.29	38.38	39.17	41.96	42.61	48.62	51.38
$\beta = 1$	38.63	40.36	45.34	45.83	46.61	47.64	51.83
$\beta = 10$	40.25	39.97	43.76	42.60	45.63	46.92	52.37
$\beta = 100$	46.26	48.84	51.91	54.03	55.74	55.96	52.60
$\beta = 1000$	43.81	46.27	50.33	55.58	50.55	45.80	55.87
$\beta = 10000$	47.24	48.58	48.26	50.78	57.20	48.94	45.09
$\beta = 100000$	45.23	50.75	49.95	50.05	55.86	60.28	47.32

Table 7
The running time (s) of the proposed approach under different parameters α and β on the AR database.

	$\alpha = 0$	$\alpha = 1$	$\alpha = 10$	$\alpha = 100$	$\alpha = 1000$	$\alpha = 10000$	$\alpha = 100000$
$\beta = 0$	41.18	53.88	63.34	62.79	62.86	68.62	71.80
$\beta = 1$	50.13	57.28	66.42	65.38	66.83	67.66	73.72
$\beta = 10$	49.85	56.19	64.00	66.24	67.01	67.96	72.60
$\beta = 100$	50.16	57.20	65.92	67.12	68.93	68.99	73.95
$\beta = 1000$	45.56	50.15	56.19	60.47	63.89	65.47	69.31
$\beta = 10000$	40.00	45.35	50.57	58.76	60.58	63.28	65.99
$\beta = 100000$	42.51	50.89	46.73	52.62	58.72	56.84	57.49

3.4. Clustering results and analysis

In this subsection, we carry out clustering experiment on three public databases including JAFFE, ORL and COIL20 to verify the effectiveness of the proposed approach. Two widely used evaluation metrics including Accuracy (ACC) and Normalized Mutual Information (NMI) [9] are adopted in our clustering experiment. The larger ACC and NMI are, the better the results are.

In this work, *k*-means clustering algorithm is utilized to cluster samples based on the selected features. Given that the performance of *k*-means heavily depends on the initialization, we repeat this clustering process 50 times with varying initializations and the average clustering results together with standard deviations are reported.

In our clustering experiment, we tune the number of selected features from 10 to 500 with the interval of 10 for all the databases. From the clustering results in Tables 8 and 9, we can learn that all feature selection algorithms perform better than the baseline, which demonstrates that feature selection is necessary in clustering. Besides, it can be found that the proposed approach outperforms other feature selection approaches, which is consistent with the experimental results in Section 3.3. Moreover, from the clustering results under various numbers of selected features obtained by different approaches in Figs. 5 and 6, we can see that none of the fifteen feature selection algorithms performs consistently better or worse than all others under all selected feature numbers. However, the proposed OPMF outperforms most of the feature selection approaches under most numbers of selected features and achieves the best clustering results on all three databases, which demonstrates the advantage of our algorithm.

Next, we discuss the sensitiveness of parameters in our proposed approach for clustering task. The results in terms of ACC and NMI over

Table 8
Clustering results (ACC% ± std%) of different algorithms on three databases.

Methods	JAFFE	ORL	COIL20
Baseline	78.73 ± 2.28(1024)	75.26 ± 4.39(1024)	55.27 ± 2.71(1024)
LS	83.43 ± 6.30(100,0.05)	78.50 ± 3.10(280,0.06)	59.84 ± 2.46(420,0.23)
SPEC	85.21 ± 7.08(470,2.09)	80.30 ± 7.56(170,4.07)	61.28 ± 4.76(340,15.5)
MCFS	87.09 ± 8.71(240,0.03)	82.10 ± 5.55(240,0.65)	62.14 ± 5.12(250,0.78)
UDFS	87.42 ± 8.23(300,11.9)	83.40 ± 4.47(370,39.1)	63.25 ± 2.89(130,64.3)
RUFS	88.64 ± 7.81(470,10.3)	83.00 ± 5.42(140,53.5)	64.08 ± 4.84(360,461)
MFFS	89.58 ± 2.98(500,26.7)	83.90 ± 5.23(40,53.2)	64.60 ± 2.86(300,90.5)
RSR	87.28 ± 5.18(500,5.91)	83.10 ± 3.78(270,24.4)	64.86 ± 2.72(470,47.8)
GRNSR	89.81 ± 3.75(470,10.6)	85.80 ± 5.78(350,42.7)	65.55 ± 1.92(470,65.68)
L1UFS	89.94 ± 6.09 (420,24.4)	86.22 ± 5.89 (170,103)	66.57 ± 3.54 (220,186)
UFSOL	90.42 ± 3.46(490,76.7)	86.60 ± 3.58(430,111)	67.30 ± 2.02(490,415)
SPNFSR	90.93 ± 2.53(500,13.2)	86.90 ± 4.28(220,515)	66.79 ± 1.47(470,259)
SCUFS	91.04 ± 6.31(220,256)	87.14 ± 5.48(220,1040)	67.53 ± 2.90(360,1752)
NSSRD	91.38 ± 5.43(250,1.53)	87.30 ± 4.59(200,3.42)	67.93 ± 2.80(480,7.88)
DSRMR	91.51 ± 6.35(360,180)	87.70 ± 5.29(180,1429)	68.25 ± 2.29(490,4773)
OPMF	**92.97 ± 3.47(40,24.9)**	**88.80 ± 2.17(170,58.1)**	**69.96 ± 2.76(150,87.7)**

Note that the numbers in parentheses are the number of the selected features that correspond to the best clustering result and the running time (s).

Table 9
Clustering results (NMI% ± std%) of different feature selection algorithms on different databases.

Methods	JAFFE	ORL	COIL20
Baseline	82.13 ± 1.43(1024)	79.64 ± 3.10(1024)	70.35 ± 1.31(1024)
LS	87.56 ± 3.01(110,0.05)	81.38 ± 4.55(140,0.06)	70.68 ± 1.22(420,0.23)
SPEC	88.28 ± 6.14(390,2.09)	83.18 ± 2.28(450,4.07)	72.45 ± 1.28(370,15.5)
MCFS	88.87 ± 5.00(160,0.03)	83.83 ± 3.90(230,0.65)	72.89 ± 2.47(370,0.78)
UDFS	89.27 ± 3.57(140,11.9)	85.83 ± 2.38(370,39.1)	72.44 ± 1.67(430,64.3)
RUFS	89.14 ± 4.48(500,10.3)	85.88 ± 3.16(140,53.5)	74.86 ± 0.96(290,461)
MFFS	89.60 ± 2.64(500,26.7)	87.01 ± 2.36(50,53.2)	74.86 ± 1.36(260,90.5)
RSR	89.52 ± 2.76(500,5.91)	86.00 ± 2.98(230,24.4)	74.63 ± 1.63(470,47.8)
GRNSR	90.21 ± 2.12(490,10.6)	88.96 ± 2.36(350,42.7)	75.21 ± 1.02(480,65.8)
L1UFS	90.74 ± 4.01 (460,24.4)	89.87 ± 2.54 (140,103)	75.92 ± 2.02 (420,186)
UFSOL	90.52 ± 2.09(380,76.7)	90.47 ± 3.25(190,111)	76.60 ± 1.35(440,415)
SPNFSR	91.54 ± 1.25(480,13.2)	90.22 ± 3.04(350,51.5)	76.52 ± 0.94(440,259)
SCUFS	91.88 ± 3.24(230,256)	90.86 ± 3.29(220,1040)	76.71 ± 1.72(310,1752)
NSSRD	91.79 ± 2.92(240,1.53)	91.25 ± 2.87(260,3.42)	76.96 ± 1.30(500,7.88)
DSRMR	92.14 ± 3.71(360,180)	91.15 ± 3.00(210,1429)	77.04 ± 1.52(500,4773)
OPMF	**92.80 ± 1.63(50,24.9)**	**91.67 ± 1.67 (290,58.1)**	**77.48 ± 0.99 (370,87.7)**

Note that the numbers in parentheses are the number of the selected features that correspond to the best clustering result and the running time (s).

Table 10
The running time (s) of the proposed approach under different parameters α and β on the JAEEF database.

	$\alpha = 0$	$\alpha = 1$	$\alpha = 10$	$\alpha = 100$	$\alpha = 1000$	$\alpha = 10000$	$\alpha = 100000$
$\beta = 0$	30.48	36.64	35.07	34.07	33.60	36.95	36.44
$\beta = 1$	32.55	35.61	35.71	33.37	33.93	36.14	36.86
$\beta = 10$	34.39	42.55	34.49	33.66	33.55	36.39	38.27
$\beta = 100$	32.67	43.64	42.48	34.69	34.34	39.10	36.43
$\beta = 1000$	30.40	41.40	43.22	42.51	36.23	35.43	36.48
$\beta = 10000$	32.48	40.38	41.84	43.18	38.52	36.19	35.45
$\beta = 100000$	31.39	39.93	42.68	42.66	42.50	38.78	35.69

Table 11
The running time (s) of the proposed approach under different parameters α and β on the ORL database.

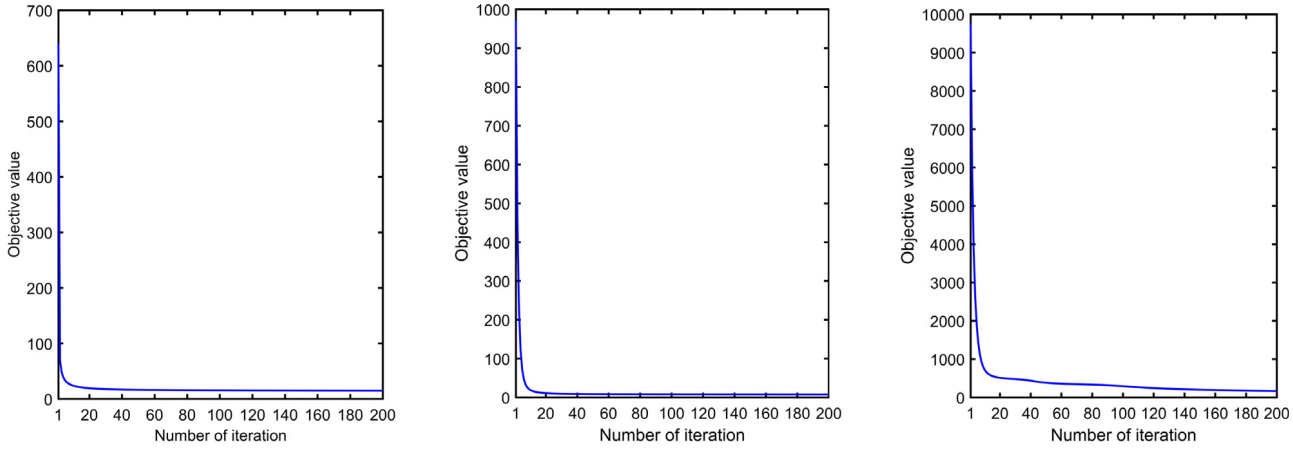
	$\alpha = 0$	$\alpha = 1$	$\alpha = 10$	$\alpha = 100$	$\alpha = 1000$	$\alpha = 10000$	$\alpha = 100000$
$\beta = 0$	21.09	25.42	23.52	21.03	22.00	20.07	22.18
$\beta = 1$	20.92	21.48	21.41	20.28	19.97	19.87	23.85
$\beta = 10$	21.93	22.85	19.37	19.69	21.42	19.89	23.00
$\beta = 100$	23.91	24.50	22.77	21.18	19.40	20.13	22.99
$\beta = 1000$	20.02	21.54	27.08	24.63	23.75	21.10	23.56
$\beta = 10000$	19.49	20.18	24.81	26.75	24.39	23.17	23.84
$\beta = 100000$	21.95	22.96	19.99	25.16	26.47	24.40	21.55

three databases are shown in Figs. 7 and 8. From the two figures, we can see that the proposed OPMF obtains its best performance under moderate parameter values, which is consistent with the classification experiments. Furthermore, since the view angle changes of each object is small in COIL20, the samples of this database are more likely to be separable and redundant. Hence, larger α and β values are more suitable for our algorithm to achieve its best performance. The running times of the proposed OPMF under different parameter values are also given in Tables 10–12.

Finally, from Fig. 9, we can find that the proposed approach converges within approximately 200 iterations in most of our clustering experiments.

Table 12
The running (s) of the proposed approach under different parameters α and β on the COIL20 database.

	$\alpha = 0$	$\alpha = 1$	$\alpha = 10$	$\alpha = 100$	$\alpha = 1000$	$\alpha = 10000$	$\alpha = 100000$
$\beta = 0$	101.91	121.55	124.11	115.41	120.22	118.65	115.06
$\beta = 1$	115.77	125.48	118.98	119.94	110.29	121.52	115.31
$\beta = 10$	108.54	117.73	110.32	112.23	115.37	110.38	115.48
$\beta = 100$	100.69	98.78	121.80	121.08	114.30	116.26	115.80
$\beta = 1000$	95.82	101.81	114.84	118.73	120.58	116.54	117.16
$\beta = 10000$	97.86	99.86	117.88	121.87	119.02	119.37	124.12
$\beta = 100000$	91.79	95.72	101.79	111.80	108.76	118.38	113.40



(a) JAEEF. (b) ORL. (c) COIL20.

Fig. 9. The convergence curves of the proposed approach on three different databases.

4. Conclusions

This paper has proposed an Ordinal Preserving Matrix Factorization Feature Selection approach, which combines the matrix factorization, ordinal locality structure preserving and inner-product regularization into a joint framework. In our algorithm, we introduce a triplet-based loss function to preserve the ordinal locality structure of the original data during feature selection. Moreover, to achieve sparsity and low redundancy among the features, an inner product regularization term is incorporated into our algorithm. In addition, an alternating optimization algorithm has been developed for efficient optimization. Extensive experiments are carried out in this paper, which show that our proposed approach outperforms several classical and state-of-the-art comparison algorithms.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Nos. 61602221, 61672150, 61702092, 61562044, 61762050, 61772091 and 61602222), the Science and Technology Research Project of Jiangxi Provincial Department of Education (No. GJJ160333), the doctoral fund of Jiangxi Normal University (No. 7525), the Natural Science Foundation of Jiangxi Province (No. 20171BAB212009), the Fundamental Research Funds for the Central Universities (No. 2412017QD). the China Postdoctoral Science Foundation (No. 2017M621193) and the Fund of the Jilin Provincial Science and Technology Department (Nos. 20180520215JH and 20180520027JH).

References

[1] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Comput. Electr. Eng. 40 (1) (2014) 16–28.
[2] J. Tang, S. Alelyani, H. Liu, Feature selection for classification: A review, Data Classif.: Algorithms Appl. 37 (2014).
[3] J. Li, K. Cheng, S. Wang, et al., Feature selection: A data perspective, J. ACM Comput. Surv. (CSUR) 50 (6) (2018).
[4] N. Dessì, B. Pes, Similarity of feature selection methods: An empirical study across data intensive classification tasks, Expert Syst. Appl. 42 (10) (2015) 4632–4642.
[5] Y. Li, T. Li, H. Liu, Recent advances in feature selection and its applications, Knowl. Inf. Syst. (2017) 1–27.
[6] S. Solorio-Fernández, J.F. Martínez-Trinidad, J.A. Carrasco-Ochoa, A new unsupervised spectral feature selection method for mixed data: a filter approach, Pattern Recognit. 72 (2017) 314–326.
[7] J. Li, H. Liu, Challenges of feature selection for big data analytics, IEEE Intell. Syst. 32 (2) (2017) 9–15.
[8] J. Gui, Z. Sun, S. Ji, et al., Feature selection based on structured sparsity: a comprehensive study, IEEE Trans. Neural Netw. Learn. Syst. 28 (7) (2017) 1490–1507.
[9] J. Wang, J.M. Wei, Z. Yang, et al., Feature selection by maximizing independent classification information, IEEE Trans. Knowl. Data Eng. 29 (4) (2017) 828–841.
[10] B. Liu, B. Fang, X. Liu, et al., Large margin subspace learning for feature selection, Pattern Recognit. 46 (10) (2013) 2798–2806.
[11] Z. Zhao, L. Wang, H. Liu, et al., On similarity preserving feature selection, IEEE Trans. Knowl. Data Eng. 25 (3) (2013) 619–632.
[12] C. Hou, F. Nie, X. Li, et al., Joint embedding learning and sparse regression: A framework for unsupervised feature selection, IEEE Trans. Cybern. 44 (6) (2014) 793–804.
[13] X. Zhu, X. Li, S. Zhang, et al., Robust joint graph sparse coding for unsupervised spectral feature selection, IEEE Trans. Neural Netw. Learn. Syst. 28 (6) (2017) 1263–1275.
[14] S. Wang, W. Zhu, Sparse graph embedding unsupervised feature selection, IEEE Trans. Syst. Man Cybern. Syst. 48 (3) (2018) 329–341.
[15] B. Ripley, Pattern Recognition and Neural Networks, Cambridge University Press, 2007.
[16] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, in: Advances in Neural Information Processing Systems, Vol. 18, 2005.
[17] Z. Zhao, H. Liu, Spectral feature selection for supervised and unsupervised learning, in: Proceedings of the 24th International Conference on Machine Learning, ACM, 2007, pp. 1151–1157.
[18] D. Cai, C. Zhang, X. He, Unsupervised feature selection for multi-cluster data, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2010, pp. 333–342.
[19] Y. Yang, H. Shen, Z. Ma, et al., $l_{2,1}$-norm regularized discriminative feature selection for unsupervised learning, IJCAI 22 (1) (2011) 1589–1594.
[20] Z. Li, Y. Yang, J. Liu, X. Zhou, H. Lu, Unsupervised feature selection using nonnegative spectral analysis, in: Proceedings of the Twenty-Sixth Conference on Artificial Intelligence, 2012, pp. 1026–1032.

[21] F. Nie, H. Huang, X. Cai, et al., Efficient and robust feature selection via joint $l_{2,1}$-norms minimization, in: Proceedings of Advances in Neural Information Processing Systems, 2010, pp. 1813–1821.
[22] M. Qian, C. Zhai, Robust unsupervised feature selection, in: Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, 2013, pp. 1621–1627.
[23] P. Zhu, W. Zuo, L. Zhang, et al., Unsupervised feature selection by regularized self-representation, Pattern Recognit. 48 (2) (2015) 438–446.
[24] Y. Yi, W. Zhou, Y. Cao, et al., Unsupervised feature selection with graph regularized nonnegative self-representation, Biometric Recognit. (2016) 591–599.
[25] W. Zhou, C. Wu, Y. Yi, et al., Structure preserving non-negative feature self-representation for unsupervised feature selection, IEEE Access 5 (1) (2017) 8792–8803.
[26] Y. Yi, W. Zhou, C. Bi, et al., Inner product regularized nonnegative self representation for image classification and clustering, IEEE Access 5 (1) (2017) 14165–14176.
[27] M. Qi, T. Wang, F. Liu, et al., Unsupervised feature selection by regularized matrix factorization, Neurocomputing 273 (2018) 593–610.
[28] C. Tang, X. Zhu, J. Chen, et al., Robust graph regularized unsupervised feature selection, Expert Syst. Appl. 96 (2018) 64–76.
[29] P. Zhu, W. Zhu, Q. Hu, et al., Subspace clustering guided unsupervised feature selection, Pattern Recognit. 66 (2017) 364–374.
[30] C. Tang, X. Liu, M. Li, et al., Robust unsupervised feature selection via dual self-representation and manifold regularization, Knowl.-Based Syst. 145 (2018) 109–120.
[31] R. Shang, W. Wang, R. Stolkin, et al., Non-negative spectral learning and sparse regression-based dual-graph regularized feature selection, IEEE Trans. Cybern. 48 (2) (2018) 793–806.
[32] Y. Wang, Y. Zhang, Nonnegative matrix factorization: A comprehensive review, IEEE Trans. Knowl. Data Eng. 25 (6) (2013) 1336–1353.
[33] Z. Yang, Y. Xiang, K. Xie, Y. Lai, Adaptive method for nonsmooth nonnegative matrix factorization, IEEE Trans. Neural Netw. Learn. Syst. 99 (2016) 1–13.
[34] G. Lu, Y. Wang, J. Zou, Low-rank matrix factorization with adaptive graph regularizer, IEEE Trans. Image Process. 25 (5) (2016) 2196–2205.
[35] S. Wang, W. Pedrycz, Q. Zhu, et al., Subspace learning for unsupervised feature selection via matrix factorization, Pattern Recognit. 48 (1) (2015) 10–19.
[36] R. Shang, W. Wang, R. Stolkin, et al., Subspace learning-based graph regularized feature selection, Knowl.-Based Syst. 112 (2016) 152–165.
[37] J. Wang, L. Wu, J. Kong, et al., Maximum weight and minimum redundancy: a novel framework for feature subset selection, Pattern Recognit. 46 (6) (2013) 1616–1627.
[38] J. Guo, Y. Guo, X. Kong, et al., Unsupervised feature selection with ordinal locality, in: IEEE International Conference on Multimedia and Expo, 2017.
[39] Z. Zhang, L. Shao, Y. Xu, et al., Marginal representation learning with graph structure self-adaptation, IEEE Trans. Neural Netw. (2017). http://dx.doi.org/10.1109/TNNLS.2017.2772264.
[40] J. Han, Z. Sun, H. Hao, Selecting feature subset with sparsity and low redundancy for unsupervised learning, Knowl.-Based Syst. 86 (2015) 210–222.
[41] C. Ding, T. Li, M. Jordan, Convex and semi-nonnegative matrix factorizations, IEEE Trans. Pattern Anal. Mach. Intell. 32 (1) (2010) 45–55.
[42] Y. Yi, C. Bi, X. Li, et al., Semi-supervised local ridge regression for local matching based face recognition, Neurocomputing 167 (2015) 132–146.
[43] D. Lee, H. Seung, Algorithms for non-negative matrix factorization, in: Advances in Neural Information Processing Systems, 2001, pp. 556–562.
[44] R. Remmert, Theory of Complex Functions, Springer Science & Business Media, 2012.
[45] K. Lee, J. Ho, D. Kriegman, Acquiring linear subspaces for face recognition under variable lighting, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) 684–698.
[46] S. Terence, B. Simon, B. Maan, The CMU pose, illumination, and expression (PIE) database, IEEE Trans. Pattern Anal. Mach. Intell. 25 (12) (2003) 1615–1618.
[47] A. Martinez, The AR face database, in: CVC Technical Report, Vol. 24, 1998.
[48] M. Lyons, S. Akamatsu, M. Kamachi, et al., Coding facial expressions with Gabor wavelets, in: IEEE International Conference on Automatic Face and Gesture Recognition, Vol. 1998, 1998, pp. 200–205.
[49] F.S. Samaria, A.C. Harter, Parameterisation of a stochastic model for human face identification, in: Proceedings of the Second IEEE Workshop on Applications of Computer Vision, Vol. 22, 1995, pp. 138–142.
[50] S. Nene, S. Nayar, H. Murase, Columbia object image library (COIL-20), in: Technical Report, CUCS-005-96, 1996.