Jointly Learning Non-negative Projection and Dictionary with Discriminative Graph Constraints for Classification

Weiyang Liu[12]
wyliu@pku.edu.cn

Zhiding Yu[3]
yzhiding@andrew.cmu.edu

Yandong Wen[3]
yandongw@andrew.cmu.edu

Rongmei Lin[4]
rongmei.lin@emory.edu

Meng Yang*[1]
yang.meng@szu.edu.cn

[1] College of Computer Science & Software Engineering, Shenzhen University, China

[2] School of ECE, Peking University, China

[3] Dept. of ECE, Carnegie Mellon University, USA

[4] Dept. of Math & Computer Science, Emory University, USA

Abstract

Sparse coding with dictionary learning (DL) has shown excellent classification performance. Despite the considerable number of existing works, how to obtain features on top of which dictionaries can be better learned remains an open and interesting question. Many current prevailing DL methods directly adopt well-performing crafted features. While such strategy may empirically work well, it ignores certain intrinsic relationship between dictionaries and features. We propose a framework where features and dictionaries are jointly learned and optimized. The framework, named "joint non-negative projection and dictionary learning" (JNPDL), enables interaction between the input features and the dictionaries. The non-negative projection leads to discriminative parts-based object features while DL seeks a more suitable representation. Discriminative graph constraints are further imposed to simultaneously maximize intra-class compactness and inter-class separability. Experiments on both image and image set classification show the excellent performance of JNPDL by outperforming several state-of-the-art approaches.

1 Introduction

Sparse coding has been widely applied in a variety of computer vision problems where one seeks to represent a signal as a sparse linear combination of bases (dictionary atoms). Dictionary plays an important role as it is expected to robustly represent components of the query signal. [22] proposed the sparse representation-based classification (SRC) in which the entire training set is treated as a structured dictionary. Methods taking off-the-shelf bases (e.g., wavelets) as the dictionary were also proposed [7]. While such strategy is straight forward and convenient, research also indicates that it may not be optimal for classification
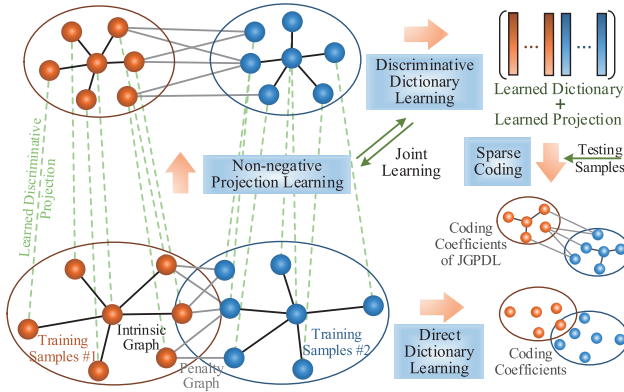
Figure 1. An illustration of JNPDL. The dictionary and non-negative projection are jointly learned with discriminative graph constraints.

tasks. Many currently prevailing approaches choose to learn dictionaries directly from training data and have leaded to good results in face recognition [8, 10, 24, 26, 29] and image classification [25]. These dictionary updating strategies are referred to as dictionary learning (DL). DL received significant attention for its excellent representation power. Such advantage mainly comes from the fact that allowing the update of dictionary atoms often results in additional flexibility to discover better intrinsic signal patterns, therefore leading to more robust representations.

DL methods can be broadly divided into unsupervised DL and supervised DL. Due to the lack of label information, unsupervised DL only guarantees to discover signal patterns that are representative, but not necessarily discriminative. Supervised DL exploits the information among classes and requires the learned dictionary to be discriminative. Related literatures include the discriminative K-SVD [29] and label-consistent K-SVD [8]. Dictionary regularization terms which takes label information into account were also introduced [1, 16, 17]. For stronger discrimination, [24, 26] tried to learn a discriminative dictionary via the Fisher discrimination criteria. More recently, [25] proposed a latent DL method by jointly learning a latent matrix to adaptively build the relationship between dictionary atoms and class labels.

Different from the conventional wide variety of discriminative DL literatures, our work casts an alternative view on this problem. One major purpose of this paper is to jointly learn a feature projection that improves DL. Instead of keep exploiting additional discrimination from the dictionary representation, we consider optimizing the input feature to further improve the learned dictionary. We believe such process can considerably influence the quality of learned dictionary, while a better learned dictionary may directly improve subsequent classification performance.

Given that mid-level object parts are often discriminative for classification, we aim to learn a feature projection that mines these discriminative patterns. It is well-known that non-negative matrix factorization (NMF) [11] can learn similar part-like components. In the light of NMF and projective NMF (PNMF) [13], we consider the projective self-representation (PSR) model where the set of training samples Y is approximately factorized as: $Y \approx MPY$. The model jointly learns both the intermediate basis matrix M and the projection matrix P with non-negativity such that the additive (non-subtractive) combinations leads to learned projected features PY accentuating spatial object parts. In the paper, we propose a novel

NMF-like feature projection learning framework on top of the PSR model to simultaneously incorporate label information with discriminative graph constraints. One shall see, our proposed framework can be viewed as a tradeoff between NMF and feature learning [30].

The dictionary representation is further discriminatively learned given the projected input features. An overview of the joint non-negative projection and dictionary learning (JNPDL) framework is illustrated in Fig. 1. The construction of discriminative graph constraints in both non-negative projection and dictionary learning follows the graph embedding framework [23]. While the inputs of graph constraints are essentially the same, they form different regularization terms for the convenience of optimization. Finally, a discriminative reconstruction constraint is also adopted so that coding coefficients will only well represent samples from their own classes but poorly represent samples from other classes. We test JNPDL in both image classification and image set classification with comprehensive evaluations, showing the excellent performance of JNPDL.

2   Related Works

Only a few works [4, 15, 28] have discussed similar ideas but have all reported more competitive performance than conventional DL methods. [15] proposed a framework which simultaneously learns feature and dictionary. Their work focused on learning a reconstructive feature (filterbank) with similar idea in [30]. The work in [4] jointly learns a dimensionality reduction matrix and a dictionary for face recognition. Unlike JNPDL, the model focused on low-dimensional representation without discriminative constraints. The source of discrimination purely comes from a Fisher-like constraint on coding coefficients. [28] presented a simultaneous projection and dictionary learning method with carefully designed discriminative sigmoid reconstruction error. Their method represents the input samples with the multiplication of projection matrix and dictionary, which differs significantly from JNPDL. [5] presents a novel dictionary pair learning approach for pattern classification, which also partially resembles our framework.

3   The Proposed JNPDL Model

Let Y be a set of s-dimensional training samples, i.e., $Y = \{Y_1, Y_2, \cdots, Y_K\}$ where Y_i denotes the training samples from class i. The learned structured (class-specific) dictionary is denoted by $D = \{D_1, D_2, \cdots, D_K\}$ and the corresponding sparse representation of the training samples over dictionary D is defined as $X = \{X_1, X_2, \cdots, X_K\}$. M is the intermediate non-negative basis matrix while $P \in \mathbb{R}^{s_p \times s}$ denotes the projection matrix. In order to avoid high inter-class correlations and benefit subsequent sparse coding, JNPDL first projects training samples to a more discriminative space before dictionary encoding. Similar process applies for testing. At training, the projection, dictionary and encoded coefficients are jointly learned with the following model:

$$\langle D, M, P, X \rangle = \arg \min_{D, M \geq 0, P \geq 0, X} \left\{ R(D, P, X) + \alpha_1 G_p(P, M) + \alpha_2 G_c(X) + \alpha_3 \|X\|_1 \right\} \quad , \quad (1)$$

where α_1, α_2 and α_3 are scalar constants, and $R(D, P, X)$ is the discriminative reconstruction error. $G_p(P, M)$ is the graph-based projection term learning the NMF-like feature projection, while $G_c(X)$ is the graph-based coding coefficients term imposing discriminative label information to DL.

3.1   Discriminative Reconstruction Error

Discriminative reconstruction error targets the three following objectives: minimizing the global reconstruction error, minimizing the local reconstruction error[1] and maximizing the non-local reconstruction error. The term is defined as:

$$R(\boldsymbol{D}, \boldsymbol{P}, \boldsymbol{X}) = \|\boldsymbol{PY} - \boldsymbol{DX}\|_F^2 + \sum_{i=1}^{K} \|\boldsymbol{PY}_i - \boldsymbol{D}_i \boldsymbol{X}_i^i\|_F^2 + \sum_{i=1}^{K} \sum_{j=1, j \neq i}^{K} \|\boldsymbol{D}_j \boldsymbol{X}_i^j\|_F^2, \tag{2}$$

where $\boldsymbol{X}_i^j$ denotes the coding coefficients of samples $\boldsymbol{Y}_i$ associated with the sub-dictionary $\boldsymbol{D}_j$. $\sum_{i=1}^{K} \|\boldsymbol{Y}_i - \boldsymbol{D}_i \boldsymbol{X}_i^i\|_F^2$ denotes the local reconstruction error that requires the local sub-dictionaries $\boldsymbol{D}_i^i$ well represent samples from class i. $\|\boldsymbol{D}_j \boldsymbol{X}_i^j\|_F^2$ is further minimized such that inter-class coefficients $\boldsymbol{X}_i^j, i \neq j$ are relatively small compared with $\boldsymbol{X}_i^i$.

3.2   Graph-based Coding Coefficients Term

The term seeks to constrain the intra-class coding coefficients to be similar while the inter-class ones to be significantly dissimilar. We first construct an intrinsic graph for intra-class compactness and a penalty graph for inter-class separability. Rewriting the coding coefficients $\boldsymbol{X}$ as $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N\}$ where N is the number of training samples, the similarity matrix of the intrinsic graph is defined as:

$$\{\boldsymbol{W}_c\}_{ij} = \begin{cases} 1, & \text{if } i \in S_{k_1}^+(j) \text{ or } j \in S_{k_1}^+(i) \\ 0, & \text{otherwise} \end{cases}, \tag{3}$$

where $S_{k_1}^+(i)$ indicates the index set of the k_1 nearest intra-class neighbors of sample $\boldsymbol{x}_i$. Similarly, the similarity matrix of the penalty graph is defined as:

$$\{\boldsymbol{W}_c^p\}_{ij} = \begin{cases} 1, & j \in S_{k_2}^-(i) \text{ or } i \in S_{k_2}^-(j) \\ 0, & \text{otherwise} \end{cases}, \tag{4}$$

where $S_{k_2}^-(i)$ denotes the index set of the k_2 nearest inter-class neighbors of $\boldsymbol{x}_i$. the coefficients term is defined as:

$$G_c(\boldsymbol{X}) = \text{Tr}(\boldsymbol{X}^T \boldsymbol{L}_c \boldsymbol{X}) - \text{Tr}(\boldsymbol{X}^T \boldsymbol{L}_c^p \boldsymbol{X}), \tag{5}$$

where $\boldsymbol{L}_c = \boldsymbol{B}_c - \boldsymbol{W}_c$ in which $\boldsymbol{B}_c = \sum_{j \neq i} \{\boldsymbol{W}_c\}_{ij}$ and $\boldsymbol{L}_c^p = \boldsymbol{B}_c^p - \boldsymbol{W}_c^p$ in which $\boldsymbol{B}_c^p = \sum_{j \neq i} \{\boldsymbol{W}_c^p\}_{ij}$. Imposing the graph-based discrimination makes the coding coefficients more discriminative. Interestingly, most existing discriminative coding coefficients term, such as in [15, 26], are special cases of the graph-based discrimination constraint.

3.3   Graph-based Non-negative Projection Term

This term aims to learn a non-negative projection that maps the training samples to a discriminative space. Inspired by [13], we design a structured projection matrix by dividing the projection matrix $\boldsymbol{P}$ into two parts:

$$\boldsymbol{Y}^{proj} = \begin{bmatrix} \hat{\boldsymbol{Y}}^{proj} \\ \tilde{\boldsymbol{Y}}^{proj} \end{bmatrix} = \boldsymbol{PY} = \begin{bmatrix} \hat{\boldsymbol{P}} \\ \tilde{\boldsymbol{P}} \end{bmatrix} \boldsymbol{Y} \tag{6}$$

---

[1]Stands for the reconstruction with atoms from the same class.

where $\hat{\boldsymbol{Y}}^{proj} = \{\hat{\boldsymbol{y}}_1^{proj}, \cdots, \hat{\boldsymbol{y}}_N^{proj}\} \in \mathbb{R}^{q \times N}$ that serves for the certain purpose of graph embedding, and $\tilde{\boldsymbol{Y}}^{proj} = \{\tilde{\boldsymbol{y}}_1^{proj}, \cdots, \tilde{\boldsymbol{y}}_N^{proj}\} \in \mathbb{R}^{(s_p-q) \times N}$ that contains the additional information for data reconstruction ($\hat{\boldsymbol{Y}}^{proj}$ is a relaxed matrix that compensates the information loss in $\hat{\boldsymbol{Y}}^{proj}$). Note that $\hat{\boldsymbol{Y}}^{proj}$ preserves the discriminative graph properties while the whole $\boldsymbol{Y}^{proj}$ is used for data reconstruction purpose. Therefore the purposes of data reconstruction and graph embedding coexist harmoniously and do not mutually compromise like conventional formulations with multiple objectives. The basis matrix $\boldsymbol{M}$ is correspondingly divided into two parts $\boldsymbol{M} = \{\hat{\boldsymbol{M}}, \tilde{\boldsymbol{M}}\}$ in which $\hat{\boldsymbol{M}} \in \mathbb{R}^{s \times q}$ and $\tilde{\boldsymbol{M}} \in \mathbb{R}^{s \times (s_p-q)}$. $\tilde{\boldsymbol{M}}$ can be considered as the complementary space of $\hat{\boldsymbol{M}}$.

We first define $\boldsymbol{y}_j^{proj}$ as the jth column vector of $\boldsymbol{Y}^{proj}$ and then construct the intrinsic graph and the penalty graph using the same procedure as graph-based coding coefficients term. The construction of the similarity matrix $\boldsymbol{W}_p$ and $\boldsymbol{W}_p^p$ for intrinsic graph and the penalty graph is identical to $\boldsymbol{W}_c$ and $\boldsymbol{W}_c^p$, except that $\boldsymbol{W}_p$ and $\boldsymbol{W}_p^p$ measure the similarities among features and adopt different parameters. Differently, $\boldsymbol{W}_c$ and $\boldsymbol{W}_c^p$ measure similarities among coding coefficients. As [24] suggests, we have two objectives to preserve graph properties and enhance discrimination:

$$\begin{cases} \max_{\hat{\boldsymbol{Y}}^{proj}} \sum_{i \neq j} \|\hat{\boldsymbol{y}}_i^{proj} - \hat{\boldsymbol{y}}_j^{proj}\|_2^2 \{\boldsymbol{W}_p^p\}_{ij} \\ \min_{\hat{\boldsymbol{Y}}^{proj}} \sum_{i \neq j} \|\hat{\boldsymbol{y}}_i^{proj} - \hat{\boldsymbol{y}}_j^{proj}\|_2^2 \{\boldsymbol{W}_p\}_{ij} \end{cases}. \tag{7}$$

$\sum_{i \neq j} \|\boldsymbol{y}_i^{proj} - \boldsymbol{y}_j^{proj}\|_2^2 \{\boldsymbol{W}_p\}_{ij} = \sum_{i \neq j} \|\hat{\boldsymbol{y}}_i^{proj} - \hat{\boldsymbol{y}}_j^{proj}\|_2^2 \{\boldsymbol{W}_p\}_{ij} + \sum_{i \neq j} \|\tilde{\boldsymbol{y}}_i^{proj} - \tilde{\boldsymbol{y}}_j^{proj}\|_2^2 \{\boldsymbol{W}_p\}_{ij}$, for a specific $\boldsymbol{Y}^{proj}$, minimizing the objective function with respect to $\tilde{\boldsymbol{Y}}^{proj}$ is equivalent to maximizing the objective function associated with the complementary part, i.e., $\hat{\boldsymbol{Y}}^{proj}$. Thus we constrain the projection matrix with the following equivalent objective:

$$\begin{cases} \min_{\hat{\boldsymbol{P}}} \text{Tr}(\hat{\boldsymbol{P}} \boldsymbol{Y} \boldsymbol{L}_p \boldsymbol{Y}^T \hat{\boldsymbol{P}}^T) \\ \min_{\tilde{\boldsymbol{P}}} \text{Tr}(\tilde{\boldsymbol{P}} \boldsymbol{Y} \boldsymbol{L}_p^p \boldsymbol{Y}^T \tilde{\boldsymbol{P}}^T) \end{cases} \tag{8}$$

where $\boldsymbol{L}_p = \boldsymbol{B}_p - \boldsymbol{W}_p$ in which $\boldsymbol{B}_p = \sum_{j \neq i} \{\boldsymbol{W}_p\}_{ij}$ and $\boldsymbol{L}_p^p = \boldsymbol{B}_p^p - \boldsymbol{W}_p^p$ in which $\boldsymbol{B}_p^p = \sum_{j \neq i} \{\boldsymbol{W}_p^p\}_{ij}$. we formulate the graph-based projection term as follows:

$$G_p(\boldsymbol{P}, \boldsymbol{M}) = \|\boldsymbol{Y} - \boldsymbol{M} \boldsymbol{P} \boldsymbol{Y}\|_F^2 + \beta \cdot \text{Tr}(\hat{\boldsymbol{P}} \boldsymbol{Y} \boldsymbol{L}_p \boldsymbol{Y}^T \hat{\boldsymbol{P}}^T) + \beta \cdot \text{Tr}(\tilde{\boldsymbol{P}} \boldsymbol{Y} \boldsymbol{L}_p^p \boldsymbol{Y}^T \tilde{\boldsymbol{P}}^T) + \|\boldsymbol{M} - \boldsymbol{P}^T\|_F^2 \quad, \tag{9}$$

where β is a scaling constant and each column of $\boldsymbol{M}$ is normalized to unit l_2 norm. We use $\|\boldsymbol{Y} - \boldsymbol{M} \boldsymbol{P} \boldsymbol{Y}\|_F^2 + \|\boldsymbol{M} - \boldsymbol{P}^T\|_F^2$ to incorporate the projection matrix into a NMF-like framework, and these two terms can further ensure the reconstruction ability of the projection $\boldsymbol{P}$ and avoid the trivial solutions of $\boldsymbol{P}$. they serve similar role to the auto-encoder style reconstruction penalty term in [18, 30, 31]. Other terms in $G_p(\boldsymbol{P})$ preserve the graph properties and enhance discrimination.

4 Optimization

We adopt standard iterative learning to jointly learn $\boldsymbol{X}$, $\boldsymbol{P}$, $\boldsymbol{M}$ and $\boldsymbol{D}$. The proposed algorithm is shown in Algorithm 1. Because of the limited space, we present the detailed optimization framework in **Appendix A**. We also provide its convergence analysis in **Appendix B**. All the appendixes are included in the supplementary material which can be found at http://wyliu.com/.

Algorithm 1 Training Procedure of JNPDL

Input: Training samples $\boldsymbol{Y} = \|\boldsymbol{Y}_1, \cdots, \boldsymbol{Y}_N\|$, intrinsic graph $\boldsymbol{W}_c, \boldsymbol{W}_p$, penalty graph $\boldsymbol{W}_c^p, \boldsymbol{W}_p^p$, parameters $\alpha_1, \alpha_2, \alpha_3, \beta$, iteration number T.

Output: Non-negative projection matrix $\boldsymbol{P}$, dictionary $\boldsymbol{D}$, coding coefficient matrix $\boldsymbol{X}$.

   Step1: Initialization
 1: $t = 1$.
 2: Randomly initialize columns in $\boldsymbol{D}^0, \boldsymbol{M}^0$ with unit l_2 norm.
 3: Initialize $\boldsymbol{x}_{i,1 \leq i \leq N}$ with $((\boldsymbol{D}^0)^T(\boldsymbol{D}^0) + \lambda_2 \boldsymbol{I})^{-1}(\boldsymbol{D}^0)^T \boldsymbol{y}_i$ where $\boldsymbol{y}_i$ is the ith training sample (regardless of label).

   Step2: Search local optima
 4: **while** not convergence or $t < T$ **do**
 5:     Solve $\boldsymbol{P}^t, \boldsymbol{M}^t$ iteratively with fixed $\boldsymbol{D}^{t-1}$ and $\boldsymbol{X}^{t-1}$ via non-negative projection learning.
 6:     Solve $\boldsymbol{X}^t$ with fixed $\boldsymbol{M}^t, \boldsymbol{D}^{t-1}$ and $\boldsymbol{P}^t$ via non-negative projection learning.
 7:     Solve $\boldsymbol{D}^t$ with fixed $\boldsymbol{M}^t, \boldsymbol{P}^t$ and $\boldsymbol{X}^t$ via discriminative DL.
 8:     $t \leftarrow t + 1$.
 9: **end while**
   Step3: Output
10: Output $\boldsymbol{P} = \boldsymbol{P}^t$, $\boldsymbol{D} = \boldsymbol{D}^t$ and $\boldsymbol{X} = \boldsymbol{X}^t$.

---

5 Classification Strategy

When the projection and the dictionary have been learned, we need to project the testing image via learned projection, code the projected sample over the learned dictionary and eventually obtain its coding coefficients which is expected to be discriminative. We first project the testing sample and then code it over the learning dictionary via

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \{\|\boldsymbol{P}\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}\|_2^2 + \lambda_1 \|\boldsymbol{x}\|_1\} \tag{10}$$

where λ is a constant. After obtaining the coding coefficients $\hat{\boldsymbol{x}}$, we classify the testing sample via

$$label(\boldsymbol{y}) = \arg\min_i \left\{ \|\boldsymbol{P}\boldsymbol{y} - \boldsymbol{D}_i \delta_i(\boldsymbol{x})\|_2^2 + \sigma \|\boldsymbol{x} - \boldsymbol{m}_i\|_2^2 \right\} \tag{11}$$

where σ is a weight to balance these two terms, and $\delta_i(\cdot)$ is the characteristic function that selects the coefficients associated with ith class. $\boldsymbol{m}_i$ is the mean vector of the learned coding coefficient matrix of class i, i.e., $\boldsymbol{X}_i$. Incorporating the term $\|\boldsymbol{x} - \boldsymbol{m}_i\|_2^2$ is to make the best of the discrimination within the dictionary, because the dictionary is learned to make coding coefficients similar from the same class and dissimilar among different classes.

6 Experiments

6.1 Implementation Details

We evaluate JNPDL by both image classification and image set classification. We construct $\boldsymbol{W}_p$ and $\boldsymbol{W}_p^p$ with correlation similarity, and set the number of nearest neighbors as $\min(n_c - 1, 5)$ where n_c is the training sample number in class c. The number of shortest pairs from different classes is 20 for each class. For $\boldsymbol{W}_c$ and $\boldsymbol{W}_c^p$, we first remove the graph-based coding coefficients term in Eq. 1 and solve the optimization. The Euclidean distances among the coding coefficients of training samples are used as initial neighbor metric. We set $k_1 = \min(n_c - 1, 5), k_2 = 30$. For all experiments, we fix $\alpha_1 = 1, \alpha_2 = 1$ and $\alpha_3 = 0.05$.

Table 1. Recognition accuracy (%) on extended Yale B and AR.

Method	ExYaleB	AR Face	Method	ExYaleB	AR Face
SVM	88.8	87.1	JDDRDL [4]	90.1	90.9
SRC [22]	91.0	88.5	FDDL [26]	91.9	92.1
D-KSVD [29]	75.3	85.4	DSRC [28]	89.6	88.2
LC-KSVD [8]	86.8	90.2	**JNPDL**	**94.1**	**94.7**

Table 2. Accuracy (%) vs. feature dimension on AR dataset.

Dimension	100	200	300	500
SRC [22]	84.0	87.3	88.5	89.7
FDDL [26]	85.7	88.5	92.0	92.2
DSRC [28]	84.8	86.9	88.2	89.1
JDDRDL [4]	82.5	87.7	90.9	91.6
JNPDL	**88.3**	**92.4**	**94.7**	**95.1**

Other parameters for JNPDL are obtained via 5-fold cross validation to avoid over-fitting. Specifically, we use $\beta = 0.7, \lambda_1 = 5 \times 10^{-6}$ and $\sigma = 0.05$ for image classification. For image set based face recognition, we set $\beta = 0.8, \lambda_2 = 0.001 \times N/700$ where N is the number of training samples. For all baselines, we usually use their original settings or carefully implement them following the paper. All results are the average value of 10 times independent experiments.

6.2   Application to Image Classification

6.2.1   Face recognition

We evaluate JNPDL on the Extended Yale B (ExYaleB)[2] and AR Face Dataset[3]. For both ExYaleB and AR Face, we exactly follow the same training/testing set selection in [24]. The dictionary size is set to half of the training samples. SRC [22] uses all training samples as the dictionary.

   Comparison with state-of-the-art approaches. We compare JNPDL with state-of-the-art DL approaches including D-KSVD [29], LC-KSVD [8] and FDDL [25]. DSRC [28] and JDDRDL [4] which share similar philosophy are also compared. JNPDL, DSRC and JDDRDL uses the original images for training and set the feature dimension after projection as 300. All the other methods use the 300-dimensional Eigenface feature. SRC and Linear SVM are used as baselines. Results are shown in Table 1. One can see that JNPDL achieves promising recognition accuracy on both datasets, respectively achieving 2.2% and 2.6% improvement over the second best approaches.

   Accuracy vs. Feature dimensionality. We vary the feature dimension after projection to evaluate the performance of JNPDL on AR dataset. For SRC and FDDL, the dimensionality reduction is performed by Eigenface. Table 2 indicates that jointly learned projection can preserve much discriminative information even at low feature dimension.

   Joint projection learning vs. Separate projection. Projection and dictionary can also be learned separately. We compare it with joint learning to validate our motivation for JNPDL. We also remove projection learning from JNPDL and use Eigenface features with

---

[2]The extended Yale B dataset consists of 2,414 frontal face images from 38 individuals. All images are normalized to 54×48.

[3]The AR dataset consists of over 4,000 frontal images for 126 individuals. For each individual, 26 pictures were taken in two separated sessions. All images in AR are normalized to 60×43.

Table 3. Accuracy (%) of different projection.

Method	ExYaleB	AR Face
JNPDL (with Eigenface)	91.8	92.2
JNPDL (Separate Learning)	92.1	92.5
JNPDL (Joint Learning)	**94.1**	**94.7**

Table 4. Recognition Accuracy (%) on LFWa Dataset.

Method	Acc.	Method	Acc.	Method	Acc.	Method	Acc.
SVM	63.0	COPAR [10]	72.6	D-KSVD	65.9	LDL [25]	77.2
SRC	72.7	FDDL	74.8	LC-KSVD	66.0	**JNPDL**	**78.1**

dimension set to 300. Results are shown in Table 3. Results show that JNPDL jointly learning projection and dictionary achieves the best accuracy.

Face recognition in the wild. we apply JNPDL in a more challenging face recognition task with LWFa dataset [21] which is an aligned version of LFW. We use 143 subject with no less than 11 samples per subject in LFWa dataset (4174 images in total) to perform the experiment. The first 10 samples are selected as the training samples and the rest is for testing. Following [25], histogram of uniform-LBP is extracted by partitioning face into 10×8 patches. PCA is used to reduce the dimension to 1000. Results are shown in Table 4, with JNPDL achieving the best.

6.2.2 Object categorization

We perform the object categorization experiment on 17 Oxford Flower dataset and use the default experiment setup as in [26]. We compare JNPDL with MTJSRC [27], COPAR, JDDRDL, DSRC, FDDL, SDL, LLC and two baseline: SRC, SVM. For fair comparison, we use the Frequent Local Histogram (FLH) feature to generate a kernel-based feature descriptor the same as [26]. Table 5 shows that JNPDL is slightly worse than FDDL but is better than most competitive approaches. We believe that it is because the kernel features are already quite discriminative and projection does not help much.

6.3   Application to Image Set Classification

6.3.1   Classification strategy for image set classification

Applying the classification in Section 5 to each video frame altogether with a voting strategy, JNPDL can be easily extended to image set classification. Given a testing video $Y^{te} = \{\mathbf{y}_1^{te}, \mathbf{y}_2^{te}, \cdots, \mathbf{y}_{K_t}^{te}\}$ in which $\mathbf{y}_j^{te}$ is the jth frame and K_t is the number of image frames in the video, we project each frame to a feature via the learned non-negative projection $\mathbf{P}$ and obtain its coding coefficients with Eq. (10). Thus the label of a video frame can be obtain by Eq. (11). After getting all the labels of frames, we perform a majority voting to decide the label of the given image set. For testing efficiency, we replace the l_1 norm $\|\mathbf{x}\|_1$ with a l_2 norm $\|\mathbf{x}\|_2^2$ and derive the decision:

$$label(\mathbf{y}_j^{te}) = \arg\min_i \{\|\mathbf{P}\mathbf{y}_j^{te} - \mathbf{D}_i\delta_i(\mathbf{D}^\dagger \mathbf{y}_j^{te})\|_2^2\}. \tag{12}$$

where $\mathbf{D}^\dagger = (\mathbf{D}^T\mathbf{D} + \lambda_2\mathbf{I})^{-1}\mathbf{D}^T$. Eventually we use the majority voting to decide the label of a video (image set).

Table 5. Recognition Accuracy (%) on 17 Oxford Flower Dataset.

Method	Accuracy	Method	Accuracy
SVM	88.6	MTJSRC	88.4
SRC	88.4	COPAR	88.6
LLC (20 bases)	89.7	SDL	91.0
JDDRDL	87.7	FDDL	91.7
DSRC	88.9	**JNPDL**	**92.1**

Table 6. Recognition acc. (%) on Honda, MoBo, YTC datasets.

Method	Honda	MoBo	YTC	Method	Honda	MoBo	YTC
DCC [9]	94.9	88.1	64.8	SANP [6]	93.6	96.1	68.3
MMD [20]	94.9	91.7	66.7	LMKML [14]	98.5	96.3	**78.2**
MDA [19]	97.4	94.4	68.1	SFDL [15]	100	96.7	76.7
CHISD [2]	92.5	95.8	67.4	**JNPDL**	**100**	**97.1**	77.4

6.3.2   Image set based face recognition

Three video face recognition benchmark dataset, including Honda/UCSD [12][4], CMU MoBo [4][5] and YouTube Celebrities (YTC)[6] are used to evaluate the proposed JNPDL. For fair comparison, we follows the experimental setup in [15]. We use Viola-Jones face detector to capture faces and then resize them to 30×30 intensity image. Each image frame is cropped into 30×30 according to the provided eye coordinates. Thus each video is represented as an image set. Following standard experiment protocol as in [2, 15], the detected face images are histogram equalized but no further preprocessing, and the image features are raw pixel values.

   Comparison with state-of-the-art approaches. For both the Honda/UCSD and CMU MoBo datasets, we randomly select one face video per person as the training samples and the rest as testing samples. For YTC dataset, we equally divide the whole dataset into five folds, and each fold contains 9 videos per person. In each fold, we randomly select 3 face videos per person for training and use the rest for testing. We compare JNPDL with DCC [9], MMD [20], MDA [19], CHISD [2], SANP [6], LMKML [14] and SFDL [15]. The settings of these approaches are basically the same as [15]. We select the best accuracy that JNPDL achieves with projected dimensions from 50, 100, 150, 200 and 300. Results in Table 6 show the superiority of the proposed method.

   Accuracy vs. Frame number. We evaluate the performance of JNPDL when videos contain different number of image frames on YTC dataset. We randomly select 50, 100 and 200 frames from each face image set for training and select another 50, 100, 200 for testing. If an image set does not have enough frames, all frames are used. We select the best accuracy JNPDL achieves with projected dimension equal to 50, 100, 150, 200 and 300. Fig. 2 shows JNPDL achieves better accuracy than the other approaches. It can be learned that the discrimination power of JNPDL is strong even with small training set.

   Efficiency. The computational time for JNPDL to recognize a query image set is approximately 5 seconds with a 3.4GHz Dual-core CPU and 16GB RAM, which is comparable to [2, 15] and slightly higher than [9, 19, 20].

---

[4]Contains 59 face video of 20 individuals with large pose and expression variations, and average length of 400 frames.

[5]Consists of 96 videos from 24 subjects, each containing 5 videos of different walking patterns.

[6]Contains 1910 video sequences of 47 celebrities from YouTube. Most videos contain noisy and low-resolution image frames.
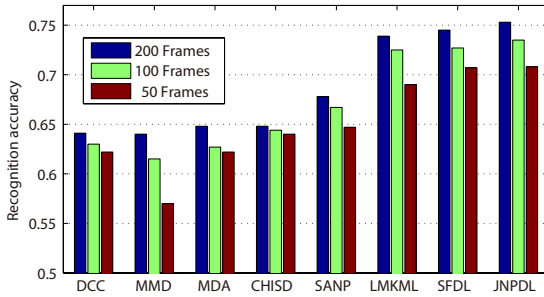
Figure 2. Recognition accuracy with different number of frames.

7   Concluding Remarks

In this paper, we proposed a novel joint non-negative projection and dictionary learning framework where non-negative feature projection and dictionary are simultaneously learned with discriminative graph constraints. The graph constraints guarantee the discrimination of projected training samples and coding coefficients. We also proposed a multiplicative non-negative updating algorithm for the projection learning with convergence guarantee. The learned feature projection considerably improves the quality learned dictionary, leading to better classification performance. Experimental results have validated the excellent performance of JNPDL on both image classification and image set classification.

Possible future work includes handling nonlinear cases using methods like kernel trick or other non-linear mapping algorithms, adding more discriminative regularizations to learn the projection matrix and considering to learn a multiple-layered (hierarchical) projection.

8   Acknowledgement

References

[1] Alexey Castrodad and Guillermo Sapiro. Sparse modeling of human actions from motion imagery. *IJCV*, 100(1):1–15, 2012.

[2] Hakan Cevikalp and Bill Triggs. Face recognition based on image sets. In *CVPR*, 2010.

[3] Zhizhao Feng, Meng Yang, Lei Zhang, Yan Liu, and David Zhang. Joint discriminative dimensionality reduction and dictionary learning for face recognition. *Pattern Recognition*, 46(8): 2134–2143, 2013.

[4] Ralph Gross and Jianbo Shi. The cmu motion of body (mobo) database. 2001.

[5] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Projective dictionary pair learning for pattern classification. In *NIPS*, 2014.

[6] Yiqun Hu, Ajmal S Mian, and Robyn Owens. Sparse approximated nearest points for image set classification. In *CVPR*, 2011.

[7] Ke Huang and Selin Aviyente. Sparse representation for signal classification. In *NIPS*, 2006.

[8] Zhuolin Jiang, Zhe Lin, and Larry S Davis. Label consistent k-svd: learning a discriminative dictionary for recognition. *IEEE T. PAMI*, 35(11):2651–2664, 2013.

[9] Tae-Kyun Kim, Josef Kittler, and Roberto Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *IEEE T. PAMI*, 29(6):1005–1018, 2007.

[10] Shu Kong and Donghui Wang. A dictionary learning approach for classification: Separating the particularity and the commonality. In *ECCV*. 2012.

[11] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[12] Kuang-Chih Lee, Jeffrey Ho, Ming-Hsuan Yang, and David Kriegman. Video-based face recognition using probabilistic appearance manifolds. In *CVPR*, 2003.

[13] Xiaobai Liu, Shuicheng Yan, and Hai Jin. Projective nonnegative graph embedding. *IEEE T. IP*, 19(5):1126–1137, 2010.

[14] Jiwen Lu, Gang Wang, and Pierre Moulin. Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning. In *ICCV*, 2013.

[15] Jiwen Lu, Gang Wang, Weihong Deng, and Pierre Moulin. Simultaneous feature and dictionary learning for image set based face recognition. In *ECCV*. 2014.

[16] Qiang Qiu, Vishal Patel, and Rama Chellappa. Information-theoretic dictionary learning for image classification. *IEEE T. PAMI*, 36(11):2173–2184, 2014.

[17] Ignacio Ramirez, Pablo Sprechmann, and Guillermo Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *CVPR*, 2010.

[18] Anran Wang, Jiwen Lu, Gang Wang, Jianfei Cai, and Tat-Jen Cham. Multi-modal unsupervised feature learning for rgb-d scene labeling. In *ECCV*. 2014.

[19] Ruiping Wang and Xilin Chen. Manifold discriminant analysis. In *CVPR*, 2009.

[20] Ruiping Wang, Shiguang Shan, Xilin Chen, and Wen Gao. Manifold-manifold distance with application to face recognition based on image set. In *CVPR*, 2008.

[21] Lior Wolf, Tal Hassner, and Yaniv Taigman. Effective unconstrained face recognition by combining multiple descriptors and learned background statistics. *IEEE T. PAMI*, 33(10):1978–1990, 2011.

[22] John Wright, Allen Y Yang, Arvind Ganesh, Shankar S Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE T. PAMI*, 31(2):210–227, 2009.

[23] Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin. Graph embedding and extensions: a general framework for dimensionality reduction. *IEEE T. PAMI*, 29(1):40–51, 2007.

[24] Meng Yang, David Zhang, and Xiangchu Feng. Fisher discrimination dictionary learning for sparse representation. In *ICCV*, 2011.

[25] Meng Yang, Dengxin Dai, Lilin Shen, and Luc Van Gool. Latent dictionary learning for sparse representation based classification. In *CVPR*, 2014.

[26] Meng Yang, Lei Zhang, Xiangchu Feng, and David Zhang. Sparse representation based fisher discrimination dictionary learning for image classification. *IJCV*, 109(3):209–232, 2014.

[27] Xiao-Tong Yuan, Xiaobai Liu, and Shuicheng Yan. Visual classification with multitask joint sparse representation. *IEEE T. IP*, 21(10):4349–4360, 2012.

[28] Haichao Zhang, Yanning Zhang, and Thomas S Huang. Simultaneous discriminative projection and dictionary learning for sparse representation based classification. *Pattern Recognition*, 46(1): 346–354, 2013.

[29] Qiang Zhang and Baoxin Li. Discriminative k-svd for dictionary learning in face recognition. In *CVPR*, 2010.

[30] Will Zou, Shenghuo Zhu, Kai Yu, and Andrew Y Ng. Deep learning of invariant features via simulated fixations in video. In *NIPS*, 2012.

[31] Zhen Zuo, Gang Wang, Bing Shuai, Lifan Zhao, Qingxiong Yang, and Xudong Jiang. Learning discriminative and shareable features for scene classification. In *ECCV*. 2014.

Supplementary Material: Jointly Learning Non-negative Projection and Dictionary with Discriminative Graph Constraints for Classification

Weiyang Liu[1], Zhiding Yu[2], Yandong Wen[2], Rongmei Lin[3], Meng Yang[4]
[1]Peking University, [2]Carnegie Mellon University
[3]Emory University, [4]Shenzhen University

Summary

The JNPDL model is well motivated by the current drawbacks of dictionary learning approaches, while each constraints are also well designed (the novel discriminative graph constraints are proposed, and all constrains are designed to be easily optimized). Aiming to bridge the gap between features and dictionary, we do think the proposed idea of learning a projection for features jointly with the dictionary is worth noticing. Consider that the given training data is usually not naturally discriminative, yhus the discrimination of the learned dictionary will be limited by the training data. Jointly learning the discriminative feature projection and discriminative dictionary together helps to improve each other. The key of JNPDL is to learn a discriminative projection (non-negativity is one way to discrimination) that can better work with the dictionary.

In the supplementary material, we give the detailed optimization framework for the JNPDL model in **Appendix A**, and then provide a detailed proof in **Appendix B** to show the convergence of the updating rules for P and M is theoretically guaranteed. In other words, the proposed multiplicative updating algorithm for the non-negative projection P in the paper is proved to be convergent.

Appendix A: Optimization Framework

We adopt a standard iterative learning framework to jointly learn the sparse representation X, the non-negative projection matrix P, the intermediate non-negative basis matrix M and the dictionary D. The proposed algorithm is shown in Algorithm 1. The non-negative projection learning converges as we prove in **Appendix B**.

Non-negative projection learning

To learn the non-negative projection, we optimize P, M with D, X fixed. Thus the JNPDL model is rewritten as

$$
\min_{P \geq 0, M \geq 0} \big\{ \|PY - DX\|_F^2 + \sum_{i=1}^{K} \|PY_i - D_i X_i^i\|_F^2
$$
$$
+ \alpha_1 \|Y - MPY\|_F^2 + \alpha_1 \beta \cdot \mathrm{Tr}(\hat{P} Y L_p Y^T \hat{P}^T)
$$
$$
+ \alpha_1 \beta \cdot \mathrm{Tr}(\tilde{P} Y L_p^p Y^T \tilde{P}^T) + \alpha_1 \|M - P^T\|_F^2 \big\}
\tag{1}
$$

which is essentially a projective non-negative matrix factorization problem [6, 2]. We use the multiplicative iterative solution [6, 2, 4] to solve Eq. (1). Specifically, we transform it into tractable sub-problems and optimize M and P by a multiplicative non-negative iterative procedure.

Because M is the basis matrix, it is necessary to require each column m_i to have unit l_2 norm, i.e., $\|m_i\| = 1$. This extra constraint makes the optimization more complicated, so we compensate the norms of the basis matrix into the coefficient matrix as in [4] and replace $\alpha_1 \beta \mathrm{Tr}(\hat{P} Y L_p Y^T \hat{P}^T) + \alpha_1 \beta \mathrm{Tr}(\tilde{P} Y L_p^p Y^T \tilde{P}^T)$ with

$$
\alpha_1 \beta \cdot \big(\mathrm{Tr}(\hat{Q}\hat{P} Y L_p Y^T \hat{P}^T \hat{Q}^T) + \mathrm{Tr}(\tilde{Q}\tilde{P} Y L_p^p Y^T \tilde{P}^T \tilde{Q}^T) \big)
\tag{2}
$$

where $\hat{Q}$ equals $\mathrm{diag}\{\|m_1\|, \cdots, \|m_q\|\}$ and $\tilde{Q}$ equals $\mathrm{diag}\{\|m_{q+1}\|, \cdots, \|m_s\|\}$.

On optimizing M with P, D, X fixed. We can further rewrite Eq. (18) as $\text{Tr}(MG_m M^T)$ where

$$
\begin{aligned}
G_m &= G_{m+} - G_{m-} \\
&= \begin{bmatrix} \hat{P}Y(\alpha_1\beta B_p)Y^T\hat{P}^T & 0 \\ 0 & \tilde{P}Y(\alpha_1\beta B_p^p)Y^T\tilde{L}^T \end{bmatrix} \odot I \\
&\quad - \begin{bmatrix} \hat{P}Y(\alpha_1\beta W_p)Y^T\hat{P}^T & 0 \\ 0 & \tilde{P}Y(\alpha_1\beta W_p^p)Y^T\tilde{L}^T \end{bmatrix} \odot I
\end{aligned}
\tag{3}
$$

where $\odot$ denotes the element-wise matrix multiplication, and I is an identity matrix. Then we put the non-negative constraints into the objective function with respect to M, and define ψ_{ij} as the Lagrange multiplier for $M \geq 0$. With $\Psi = [\psi_{ij}]$, the Lagrange $\mathcal{L}(W)$ is defined as

$$
\begin{aligned}
\mathcal{L}(M) =& \|PY - DX\|_F^2 + \sum \|PY_i - D_iX_i^i\|_F^2 + \\
& \alpha_1\|Y - MPY\|_F^2 + \text{Tr}(MG_m M^T) \\
& + \alpha_1\|M - P^T\|_F^2 + \text{Tr}(\Psi M^T)
\end{aligned}
\tag{4}
$$

Thus the partial derivative of $\mathcal{L}$ with respect to M is

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial M} =& -2\alpha_1 YY^T P^T + 2\alpha_1 MPYY^T P^T \\
& + 2MG_m + 2\alpha_1 M - 2\alpha_1 P^T + \Psi
\end{aligned}
\tag{5}
$$

According to the Karush-Kuhn-Tucker (KKT) condition ($\psi_{ij}M_{ij} = 0$) and $\frac{\partial \mathcal{L}}{\partial M} = 0$, we can obtain the update rule:

$$
M_{ij}^{(t+1)} = M_{ij}^{(t)} \frac{(\alpha_1 YY^T P^T + MG_{m-} + \alpha_1 P^T)_{ij}}{(\alpha_1 MPYY^T P^T + MG_{m+} + \alpha_1 M)_{ij}}.
\tag{6}
$$

On optimizing P with M, D, X fixed. After updating M, we normalize the column vectors of M and multiply the norm to the projective matrix P, namely $P_i \leftarrow P_i \times \|m_i\|_2$, $m_i \leftarrow m_i/\|m_i\|_2, \forall i$. By using the normalized M, we simplify Eq. (18) as $\alpha_1\beta\text{Tr}(\hat{P}YL_pY^T\hat{P}^T) + \alpha_1\beta\text{Tr}(\tilde{P}YL_p^pY^T\tilde{P}^T)$. Thus the Lagrange $\mathcal{L}(P)$ is

$$
\begin{aligned}
\mathcal{L}(P) =& \|PY - DX\|_F^2 + \sum \|PY_i - D_iX_i^i\|_F^2 + \\
& \alpha_1\beta\text{Tr}(\hat{P}YL_pY^T\hat{P}^T) + \alpha_1\beta\text{Tr}(\tilde{P}YL_p^pY^T\tilde{P}^T) + \\
& \alpha_1\|Y - MPY\|_F^2 + \alpha_1\|M - P^T\|_F^2 + \text{Tr}(\Phi P^T)
\end{aligned}
\tag{7}
$$

where ϕ is the Lagrange multiplier for constraint $P_{ij} \geq 0$ and $\Phi = [\phi_{ij}]$. After setting $\frac{\partial \mathcal{L}}{\partial P} = 0$ and applying KKT condition ($\phi_{ij}P_{ij} = 0$), we obtain the update rule for P:

$$
P_{ij}^{(t+1)} = P_{ij}^{(t)} \frac{\left(\begin{array}{c} DXY^T + \sum D_iX_i^iY_i^T + \alpha_1 M^TYY^T \\ + \alpha_1 M^T + \alpha_1\beta\begin{bmatrix} \hat{P}^tYW_pY^T \\ \tilde{P}^tYW_p^pY^T \end{bmatrix} \end{array} \right)_{ij}}{\left(\begin{array}{c} P^tYY^T + \sum P^tY_iY_i^T + \alpha_1 P^t + \\ \alpha_1 M^TMP^tYY^T + \alpha_1\beta\begin{bmatrix} \hat{P}^tYB_pY^T \\ \tilde{P}^tYB_p^pY^T \end{bmatrix} \end{array} \right)_{ij}}.
\tag{8}
$$

Now both Eq. (6) and Eq. (41) are non-negative update. We prove that the convergence of the updating rule for P and M can be guaranteed. Detailed proof can refer to the supplementary material (Appendix A).

Discriminative dictionary learning

The discriminative dictionary learning is optimized using a standard iterative optimization procedure that is widely adopted in sparse coding.

On optimizing X with D, P, M fixed. With D, P fixed, the optimization of the JNPDL model becomes

$$
\begin{aligned}
\min_X \Big\{ & \|PY - DX\|_F^2 + \sum_{i=1}^K \|PY_i - D_iX_i^i\|_F^2 + \alpha_3\|X\|_1 \\
& + \sum_{i=1}^K \sum_{j=1, j\neq i}^K \|D_jX_i^j\|_F^2 + \alpha_2\text{Tr}(X^T(L')X) \Big\}
\end{aligned}
\tag{9}
$$

Algorithm 1 Training Procedure of JNPDL

Input: Training samples $\boldsymbol{Y} = \|\boldsymbol{Y}_1, \cdots, \boldsymbol{Y}_N\|$, intrinsic graph $\boldsymbol{W}_c$, $\boldsymbol{W}_p$, penalty graph $\boldsymbol{W}_c^p$, $\boldsymbol{W}_p^p$, parameters $\alpha_1, \alpha_2, \alpha_3, \beta$, iteration number T.

Output: Non-negative projection matrix $\boldsymbol{P}$, dictionary $\boldsymbol{D}$, coding coefficient matrix $\boldsymbol{X}$.

    Step1: Initialization
1: $t = 1$.
2: Randomly initialize columns in $\boldsymbol{D}^0$, $\boldsymbol{M}^0$ with unit l_2 norm.
3: Initialize $\boldsymbol{x}_{i,1 \leq i \leq N}$ with $((\boldsymbol{D}^0)^T(\boldsymbol{D}^0) + \lambda_2 \boldsymbol{I})^{-1}(\boldsymbol{D}^0)^T \boldsymbol{y}_i$ where $\boldsymbol{y}_i$ is the ith training sample (regardless of label).
    Step2: Search local optima
4: **while** not convergence or $t < T$ **do**
5:     Solve $\boldsymbol{P}^t$, $\boldsymbol{M}^t$ iteratively with fixed $\boldsymbol{D}^{t-1}$ and $\boldsymbol{X}^{t-1}$ via Eq. (1).
6:     Solve $\boldsymbol{X}^t$ with fixed $\boldsymbol{M}^t$, $\boldsymbol{D}^{t-1}$ and $\boldsymbol{P}^t$ via Eq. (10).
7:     Solve $\boldsymbol{D}^t$ with fixed $\boldsymbol{M}^t$, $\boldsymbol{P}^t$ and $\boldsymbol{X}^t$ via Eq. (12).
8:     $t \leftarrow t + 1$.
9: **end while**
    Step3: Output
10: Output $\boldsymbol{P} = \boldsymbol{P}^t$, $\boldsymbol{D} = \boldsymbol{D}^t$ and $\boldsymbol{X} = \boldsymbol{X}^t$.

---

where $\boldsymbol{L}' = \boldsymbol{L}_c - \boldsymbol{L}_c^p$. Eq. (9) can be solved using feature sign search algorithm [1] after certain formulation based on [7, 3]. We optimize $\boldsymbol{X}$ class by class. Following [1], we update $\boldsymbol{X}_i$ one by one in the ith class. We define $\boldsymbol{x}_{i,j}$ as the coding coefficients of the jth sample in the ith class and reformulated the problem as

$$
\min_{\boldsymbol{X}} \big\{ \|\boldsymbol{PY}_i - \boldsymbol{Dx}_{i,j}\|_F^2 + \|\boldsymbol{PY}_i - \boldsymbol{D}_i \boldsymbol{x}_{i,j}^i\|_F^2
$$
$$
+ \sum_{n=1, n \neq i}^{K} \|\boldsymbol{D}_n \boldsymbol{x}_{i,j}^n\|_F^2 + \alpha_2 Q(\boldsymbol{x}_{i,j}) + \alpha_3 \|\boldsymbol{x}_{i,j}\|_1 \tag{10}
$$

where $Q(\boldsymbol{x}_{i,j}) = \alpha_2(\boldsymbol{x}_{i,j}^T \boldsymbol{X}_i \boldsymbol{L}_j' + (\boldsymbol{X}_i \boldsymbol{L}_j')^T \boldsymbol{x}_{i,j} - \boldsymbol{x}_{i,j}^T \boldsymbol{L}_{jj}')$ in which $\boldsymbol{L}_i'$ is the ith column of $\boldsymbol{L}$, and $\boldsymbol{L}_{ii}'$ is the entry in the ith row and ith column of $\boldsymbol{L}$. $\boldsymbol{x}_{i,j}$ can be solved via feature sign search algorithm as in [7, 3].

On optimizing D with P, X, M fixed. By fixing $\boldsymbol{P}$, $\boldsymbol{M}$ and $\boldsymbol{X}$, the JNPDL model is rewritten as

$$
\min_{\boldsymbol{D}} \big\{ \|\boldsymbol{PY} - \boldsymbol{DX}\|_F^2 + \sum_{i=1}^{K} \|\boldsymbol{PY}_i - \boldsymbol{D}_i \boldsymbol{X}_i^i\|_F^2
$$
$$
+ \sum_{i=1}^{K} \sum_{j=1, j \neq i}^{K} \|\boldsymbol{D}_j \boldsymbol{X}_i^j\|_F^2 \big\} \tag{11}
$$

for which we update $\boldsymbol{D}$ class by class sequentially. When we update $\boldsymbol{D}_v$, the sub-dictionaries $\boldsymbol{D}_i, i \neq v$ associated to the other classes will be fixed. Thus Eq. (11) can be further rewritten to

$$
\min_{\boldsymbol{D}_i, i \in \{1,2,\cdots,K\}} \big\{ \|\boldsymbol{PY}_i - \boldsymbol{D}_i \boldsymbol{X}_i\|_F^2 + \|\boldsymbol{PY}_i - \boldsymbol{D}_i \boldsymbol{X}_i^i\|_F^2
$$
$$
+ \sum_{j=1, j \neq i}^{K} \|\boldsymbol{D}_j \boldsymbol{X}_i^j\|_F^2 \big\} \tag{12}
$$

which is essentially a quadratic programming problem and can be directly solved by the algorithm presented in [5] (update $\boldsymbol{D}_i$ atom by atom). Note that each atom in the dictionary should have unit l_2 norm.

Appendix B: Proof of the convergence of updating rules for P and M

Proof. Before proving the convergence of the updating rules, we first introduce some necessary preliminaries.

Definition 1. *Function $\mathcal{G}(\boldsymbol{A}, \boldsymbol{A}')$ is an auxiliary function for function $\mathcal{F}(\boldsymbol{A})$ if the conditions*

$$
\mathcal{G}(\boldsymbol{A}, \boldsymbol{A}') \geq \mathcal{F}(\boldsymbol{A}), \quad \mathcal{G}(\boldsymbol{A}, \boldsymbol{A}) = \mathcal{F}(\boldsymbol{A}) \tag{13}
$$

are satisfied.

Lemma 1. *If $\mathcal{G}(\boldsymbol{A}, \boldsymbol{A}')$ is an auxiliary function for $\mathcal{F}(\boldsymbol{A})$, then $\mathcal{F}(\boldsymbol{A})$ is non-increasing to $\boldsymbol{A}$ under the update*

$$\boldsymbol{A}^{t+1} = \arg\min_{\boldsymbol{A}} \mathcal{G}(\boldsymbol{A}, \boldsymbol{A}^t) \tag{14}$$

where t denotes the tth iteration.

Proof. From Eq.(14), we construct the following relation:

$$\mathcal{F}(\boldsymbol{A}^t) = \mathcal{G}(\boldsymbol{A}^t, \boldsymbol{A}^t) \geq \mathcal{G}(\boldsymbol{A}^{t+1}, \boldsymbol{A}^t). \tag{15}$$

Because $\mathcal{G}(\boldsymbol{A}, \boldsymbol{A}')$ is an auxiliary function for $\mathcal{F}(\boldsymbol{A})$, we can obtain the following inequality from Eq. (13):

$$\mathcal{G}(\boldsymbol{A}^{t+1}, \boldsymbol{A}^t) \geq \mathcal{F}(\boldsymbol{A}^{t+1}) \tag{16}$$

which leads to

$$\mathcal{F}(\boldsymbol{A}^{t+1}) \leq \mathcal{F}(\boldsymbol{A}^t). \tag{17}$$

Thus $\mathcal{F}(\boldsymbol{A})$ is non-increasing with respect to $\boldsymbol{A}$ under the updating rule in Eq. (14). The lemma is proved $\qquad\square$

We first consider the scenario when $\boldsymbol{P}$ is fixed. With $\boldsymbol{P}$ fixed, we rewrite the optimization objective (Eq. (10) in the paper) as

$$\begin{aligned} \mathcal{F}(\boldsymbol{M}) =& \alpha_1 \|\boldsymbol{Y} - \boldsymbol{MPY}\|_F^2 + Tr(\boldsymbol{MG}_m\boldsymbol{M}^T) \\ & + \alpha_1 \|\boldsymbol{M} - \boldsymbol{P}^T\|_F^2 \end{aligned}. \tag{18}$$

We denote $\mathcal{F}_{ij}$ as the part of $\mathcal{F}(\boldsymbol{M})$ relevant to $\boldsymbol{M}_{ij}$, and then compute the first-order and the second-order derivative as follows:

$$\begin{aligned} \mathcal{F}'_{ij}(\boldsymbol{M}) =& \alpha_1(-2\boldsymbol{YY}^T\boldsymbol{P}^T + 2\boldsymbol{MPYY}^T\boldsymbol{P}^T)_{ij} \\ & + (2\boldsymbol{MG}_m)_{ij} + \alpha_1(2\boldsymbol{M} - 2\boldsymbol{P}^T)_{ij} \end{aligned}, \tag{19}$$

$$\mathcal{F}''_{ij}(\boldsymbol{M}) = 2\alpha_1(\boldsymbol{PYY}^T\boldsymbol{P}^T + \boldsymbol{G}_m + \boldsymbol{I})_{jj} \tag{20}$$

where $\boldsymbol{I}$ denotes an identity matrix with matched size. We construct the function $\mathcal{G}(\boldsymbol{M}_{ij}, \boldsymbol{M}_{ij}^t)$ as

$$\begin{aligned} \mathcal{G}(\boldsymbol{M}_{ij}, \boldsymbol{M}_{ij}^t) =& \mathcal{F}_{ij}(\boldsymbol{M}_{ij}^t) + \mathcal{F}'_{ij}(\boldsymbol{M}_{ij}^t)(\boldsymbol{M}_{ij} - \boldsymbol{M}_{ij}^t) \\ & + \frac{\alpha_1(\boldsymbol{M}^t\boldsymbol{PYY}^T\boldsymbol{P}^T + \boldsymbol{M}^t\boldsymbol{G}_{m+} + \boldsymbol{M}^t)_{ij}}{\boldsymbol{M}_{ij}^t}(\boldsymbol{M}_{ij} - \boldsymbol{M}_{ij}^t)^2 \end{aligned}. \tag{21}$$

Lemma 2. $\mathcal{G}(\boldsymbol{M}_{ij}, \boldsymbol{M}_{ij}^t)$ *in Eq. (21) is an auxiliary function for the function $\mathcal{F}_{ij}(\boldsymbol{M})$.*

Proof. Because it is easily obtained that $\mathcal{G}(\boldsymbol{M}_{ij}, \boldsymbol{M}_{ij}) = \mathcal{F}_{ij}(\boldsymbol{M}_{ij})$, we only need to prove that $\mathcal{G}(\boldsymbol{M}_{ij}, \boldsymbol{M}_{ij}^t) \geq \mathcal{F}_{ij}(\boldsymbol{M}_{ij})$. We first compute the Taylor series expansion of $\mathcal{F}_{ij}(\boldsymbol{M})$ as

$$\begin{aligned} \mathcal{F}_{ij}(\boldsymbol{M}_{ij}) =& \mathcal{F}_{ij}(\boldsymbol{M}_{ij}^t) + \mathcal{F}'_{ij}(\boldsymbol{M}_{ij}^t)(\boldsymbol{M}_{ij} - \boldsymbol{M}_{ij}^t) \\ & + \frac{1}{2}\mathcal{F}''_{ij}(\boldsymbol{M}_{ij}^t)(\boldsymbol{M}_{ij} - \boldsymbol{M}_{ij}^t)^2 \end{aligned}. \tag{22}$$

Because the following inequalities are satisfied:

$$\begin{aligned} (\boldsymbol{M}^t\boldsymbol{PYY}^T\boldsymbol{P}^T)_{ij} =& \sum_v \left(\boldsymbol{M}_{iv}^t(\boldsymbol{PYY}^T\boldsymbol{P}^T)_{vj}\right) \\ \geq& \boldsymbol{M}_{ij}^t(\boldsymbol{PYY}^T\boldsymbol{P}^T)_{jj}, \end{aligned} \tag{23}$$

$$\begin{aligned} (\boldsymbol{M}^t\boldsymbol{G}_{m+})_{ij} =& \sum_v \left(\boldsymbol{M}_{iv}^t(\boldsymbol{G}_{m+})_{vj}\right) \\ \geq& \boldsymbol{M}_{ij}^t(\boldsymbol{G}_m)_{jj}, \end{aligned} \tag{24}$$

$$\boldsymbol{M}_{ij}^t \geq \boldsymbol{M}_{ij}^t\boldsymbol{I}_{jj}, \tag{25}$$

we can let the following relation hold:

$$\begin{aligned} \frac{\alpha_1(\boldsymbol{M}^t\boldsymbol{PYY}^T\boldsymbol{P}^T + \boldsymbol{M}^t\boldsymbol{G}_{m+} + \boldsymbol{M}^t)_{ij}}{\boldsymbol{M}_{ij}^t} \\ \geq (\boldsymbol{PYY}^T\boldsymbol{P}^T + \boldsymbol{G}_m)_{jj}. \end{aligned} \tag{26}$$

Therefore, we can prove that $\mathcal{G}(\boldsymbol{M}_{ij}, \boldsymbol{M}_{ij}^t) \geq \mathcal{F}_{ij}(\boldsymbol{M}_{ij})$ holds. The lemma is proved. $\qquad\square$

Theorem 1. *The updating rule for $\boldsymbol{M}$ can be obtained by minimizing the auxiliary function $\mathcal{G}(\boldsymbol{M}_{ij}, \boldsymbol{M}_{ij}^t)$.*

Proof. We let the derivative of $\mathcal{G}(\boldsymbol{M}_{ij}, \boldsymbol{M}_{ij}^t)$ with respect to $\boldsymbol{M}_{ij}$ equal to zero, namely

$$
\begin{aligned}
&\frac{\partial \mathcal{G}(\boldsymbol{M}_{ij}, \boldsymbol{M}_{ij}^t)}{\partial \boldsymbol{M}_{ij}} \\
&= \frac{2\alpha_1 (\boldsymbol{M}^t \boldsymbol{P}\boldsymbol{Y}\boldsymbol{Y}^T \boldsymbol{P}^T + \boldsymbol{M}^t \boldsymbol{G}_{m+} + \boldsymbol{M}^t)_{ij}}{\boldsymbol{M}_{ij}^t}(\boldsymbol{M}_{ij} - \boldsymbol{M}_{ij}^t) \\
&\quad + \mathcal{F}_{ij}'(\boldsymbol{M}_{ij}^t). \\
&= 0
\end{aligned}
\tag{27}
$$

from which we can derive

$$
\boldsymbol{M}_{ij}^{t+1} = \boldsymbol{M}_{ij}^t \frac{(\alpha_1 \boldsymbol{Y}\boldsymbol{Y}^T \boldsymbol{P}^T + \boldsymbol{M}^t \boldsymbol{G}_{m-} + \alpha_1 \boldsymbol{P}^T)_{ij}}{(\alpha_1 \boldsymbol{M}^t \boldsymbol{P}\boldsymbol{Y}\boldsymbol{Y}^T \boldsymbol{P}^T + \boldsymbol{M}^t \boldsymbol{G}_{m+} + \alpha_1 \boldsymbol{M}^t)_{ij}}.
\tag{28}
$$

which is identical to the updating rule that we use in the paper. Thus the lemma is proved $\qquad \square$

Then we consider the other scenario when $\boldsymbol{M}$ is fixed. After updating the matrix $\boldsymbol{M}$ via Eq. (28), we normalize the column vectors $\boldsymbol{m}_i$ of $\boldsymbol{M}$ and consequently convey the norm to the projective matrix $\boldsymbol{P}$, namely

$$
\begin{aligned}
\boldsymbol{P}_i &\leftarrow \boldsymbol{P}_i \times \|\boldsymbol{m}_i\| \\
\boldsymbol{m}_i &\leftarrow \boldsymbol{m}_i / \|\boldsymbol{m}_i\|
\end{aligned}
\tag{29}
$$

where $\boldsymbol{P}_i$ is the ith column vector of the projection matrix $\boldsymbol{P}$. Considering Eq. (29) and the fixed $\boldsymbol{M}$, we can rewrite the optimization objective (Eq. (10) in the paper) as

$$
\begin{aligned}
\mathcal{F}(\boldsymbol{P}) = \; &\|\boldsymbol{P}\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F^2 + \sum \|\boldsymbol{P}\boldsymbol{Y}_i - \boldsymbol{D}_i \boldsymbol{X}_i^i\|_F^2 + \\
&\alpha_1 \beta \mathrm{Tr}(\hat{\boldsymbol{P}}\boldsymbol{Y}\boldsymbol{L}_p \boldsymbol{Y}^T \hat{\boldsymbol{P}}^T) + \alpha_1 \beta \mathrm{Tr}(\tilde{\boldsymbol{P}}\boldsymbol{Y}\boldsymbol{L}_p^p \boldsymbol{Y}^T \tilde{\boldsymbol{P}}^T). \\
&+ \alpha_1 \|\boldsymbol{Y} - \boldsymbol{M}\boldsymbol{P}\boldsymbol{Y}\|_F^2 + \alpha_1 \|\boldsymbol{M} - \boldsymbol{P}^T\|_F^2)
\end{aligned}
\tag{30}
$$

By denoting $\mathcal{F}_{ij}$ as the part of $\mathcal{F}(\boldsymbol{P})$ relevant to $\boldsymbol{P}_{ij}$, we have the following derivatives:

$$
\begin{aligned}
\mathcal{F}_{ij}'(\boldsymbol{P}) =\; &2(\boldsymbol{P}\boldsymbol{Y}\boldsymbol{Y}^T)_{ij} - 2(\boldsymbol{D}\boldsymbol{X}\boldsymbol{Y}^T))_{ij} + 2(\sum \boldsymbol{P}\boldsymbol{Y}_i \boldsymbol{Y}_i^T)_{ij} \\
&- 2\sum (\boldsymbol{D}_i \boldsymbol{X}_i^i \boldsymbol{Y}_i^T)_{ij} - 2\alpha_1 (\boldsymbol{M}^T \boldsymbol{Y}\boldsymbol{Y}^T)_{ij} + 2\alpha_1 (\boldsymbol{M}^T \boldsymbol{M}\boldsymbol{P}\boldsymbol{Y}\boldsymbol{Y}^T)_{ij}, \\
&+ 2\alpha_1 \beta \begin{bmatrix} \hat{\boldsymbol{P}}\boldsymbol{Y}\boldsymbol{L}_p \boldsymbol{Y}^T \\ \tilde{\boldsymbol{P}}\boldsymbol{Y}\boldsymbol{L}_p^p \boldsymbol{Y}^T \end{bmatrix}_{ij} + \alpha_1 (2\boldsymbol{P} - 2\boldsymbol{M}^T)_{ij}
\end{aligned}
\tag{31}
$$

$$
\begin{aligned}
\mathcal{F}_{ij}''(\boldsymbol{P}) =\; &2(\boldsymbol{Y}\boldsymbol{Y}^T)_{jj} + 2(\sum \boldsymbol{Y}_i \boldsymbol{Y}_i^T)_{jj} + 2\alpha_1 (\boldsymbol{M}^T \boldsymbol{M})_{ii}(\boldsymbol{Y}\boldsymbol{Y}^T)_{jj} \\
&+ 2\alpha_1 \beta \begin{bmatrix} \boldsymbol{Y}\boldsymbol{L}_p \boldsymbol{Y}^T \\ \boldsymbol{Y}\boldsymbol{L}_p^p \boldsymbol{Y}^T \end{bmatrix}_{jj} + 2\alpha_1 \boldsymbol{I}_{jj}
\end{aligned}
\tag{32}
$$

The auxiliary function of $\mathcal{F}_{ij}(\boldsymbol{P})$ is designed as

$$
\begin{aligned}
\mathcal{G}(\boldsymbol{P}_{ij}, \boldsymbol{P}_{ij}^t) =\; &\mathcal{F}_{ij}(\boldsymbol{P}_{ij}^t) + \mathcal{F}_{ij}'(\boldsymbol{P}_{ij}^t)(\boldsymbol{P}_{ij} - \boldsymbol{P}_{ij}^t) \\
&+ \frac{\left(\begin{aligned} &\boldsymbol{P}^t \boldsymbol{Y}\boldsymbol{Y}^T + \sum \boldsymbol{P}^t \boldsymbol{Y}_i \boldsymbol{Y}_i^T + \alpha_1(\boldsymbol{P}^t) + \\ &\alpha_1 \boldsymbol{M}^T \boldsymbol{M}\boldsymbol{P}\boldsymbol{Y}\boldsymbol{Y}^T + \alpha_1 \beta \begin{bmatrix} \hat{\boldsymbol{P}}^t \boldsymbol{Y}\boldsymbol{B}_p \boldsymbol{Y}^T \\ \tilde{\boldsymbol{P}}^t \boldsymbol{Y}\boldsymbol{B}_p^p \boldsymbol{Y}^T \end{bmatrix} \end{aligned} \right)_{ij}}{\boldsymbol{P}_{ij}^t}(\boldsymbol{P}_{ij} - \boldsymbol{P}_{ij}^t)^2.
\end{aligned}
\tag{33}
$$

Lemma 3. *$\mathcal{G}(\boldsymbol{P}_{ij}, \boldsymbol{P}_{ij}^t)$ in Eq. (33) is an auxiliary function for the function $\mathcal{F}_{ij}(\boldsymbol{P})$.*

Proof. Because obviously $\mathcal{G}(\boldsymbol{P}_{ij}, \boldsymbol{P}_{ij}) = \mathcal{F}_{ij}(\boldsymbol{P}_{ij})$, we only need to prove that $\mathcal{G}(\boldsymbol{P}_{ij}, \boldsymbol{P}_{ij}^t) = \mathcal{F}_{ij}(\boldsymbol{P}_{ij})$. We first obtain the Taylor series expansion of $\mathcal{F}_{ij}(\boldsymbol{P})$ as

$$
\begin{aligned}
\mathcal{F}_{ij}(\boldsymbol{P}_{ij}) =\; &\mathcal{F}_{ij}(\boldsymbol{P}_{ij}^t) + \mathcal{F}_{ij}'(\boldsymbol{P}_{ij}^t)(\boldsymbol{P}_{ij} - \boldsymbol{P}_{ij}^t) \\
&+ \frac{1}{2}\mathcal{F}_{ij}''(\boldsymbol{P}_{ij}^t)(\boldsymbol{P}_{ij} - \boldsymbol{P}_{ij}^t)^2
\end{aligned}
\tag{34}
$$

Since the following relations hold:

$$
\begin{aligned}
(\boldsymbol{P}^t \boldsymbol{Y}\boldsymbol{Y}^T)_{ij} &= \sum_v \left(\boldsymbol{P}_{iv}^t (\boldsymbol{Y}\boldsymbol{Y}^T)_{vj} \right) \\
&\geq \boldsymbol{P}_{ij}^t (\boldsymbol{Y}\boldsymbol{Y}^T)_{jj}
\end{aligned}
\tag{35}
$$

$$\left(\sum_i \boldsymbol{P}^t \boldsymbol{Y}_i \boldsymbol{Y}_i^T\right)_{ij} = \sum_v \left(\boldsymbol{P}_{iv}^t \left(\sum \boldsymbol{Y}_i \boldsymbol{Y}_i^T\right)_{vj}\right) \tag{36}$$

$$\geq \boldsymbol{P}_{ij}^t \left(\sum \boldsymbol{Y}_i \boldsymbol{Y}_i^T\right)_{jj}$$

$$(\boldsymbol{M}^T \boldsymbol{M} \boldsymbol{P}^t \boldsymbol{Y} \boldsymbol{Y}^T)_{ij} = \sum_v \left((\boldsymbol{M}^T \boldsymbol{M} \boldsymbol{P}^t)_{iv}(\boldsymbol{Y}\boldsymbol{Y}^T)_{vj}\right)$$

$$\geq (\boldsymbol{M}^T \boldsymbol{M} \boldsymbol{P}^t)_{ij}(\boldsymbol{Y}\boldsymbol{Y}^T)_{jj}$$

$$= \sum_v \left((\boldsymbol{M}^T \boldsymbol{M})_{iv} \boldsymbol{P}_{vj}^t\right)(\boldsymbol{Y}\boldsymbol{Y}^T)_{jj} \tag{37}$$

$$\geq \boldsymbol{P}_{ij}^t (\boldsymbol{M}^T \boldsymbol{M})_{ii}(\boldsymbol{Y}\boldsymbol{Y}^T)_{jj}$$

$$\begin{bmatrix} \hat{\boldsymbol{P}}^t \boldsymbol{Y} \boldsymbol{B}_p \boldsymbol{Y}^T \\ \tilde{\boldsymbol{P}}^t \boldsymbol{Y} \boldsymbol{B}_p^p \boldsymbol{Y}^T \end{bmatrix}_{ij} = \begin{cases} \sum_v \left(\hat{\boldsymbol{P}}_{iv}^t (\boldsymbol{Y}\boldsymbol{B}_p \boldsymbol{Y}^T)_{vj}\right), & \text{if } j \leq q \\ \sum_v \left(\tilde{\boldsymbol{P}}_{iv}^t (\boldsymbol{Y}\boldsymbol{B}_p^p \boldsymbol{Y}^T)_{vj}\right), & \text{otherwise} \end{cases}$$

$$\geq \begin{cases} \hat{\boldsymbol{P}}_{ij}^t (\boldsymbol{Y}\boldsymbol{B}_p \boldsymbol{Y}^T)_{jj}, & \text{if } j \leq q \\ \tilde{\boldsymbol{P}}_{ij}^t (\boldsymbol{Y}\boldsymbol{B}_p^p \boldsymbol{Y}^T)_{jj}, & \text{otherwise} \end{cases}$$

$$\geq \begin{cases} \hat{\boldsymbol{P}}_{ij}^t (\boldsymbol{Y}\boldsymbol{L}_p \boldsymbol{Y}^T)_{jj}, & \text{if } j \leq q \\ \tilde{\boldsymbol{P}}_{ij}^t (\boldsymbol{Y}\boldsymbol{L}_p^p \boldsymbol{Y}^T)_{jj}, & \text{otherwise} \end{cases} \tag{38}$$

$$= \boldsymbol{P}_{ij}^t \begin{bmatrix} \boldsymbol{Y} \boldsymbol{L}_p \boldsymbol{Y}^T \\ \boldsymbol{Y} \boldsymbol{L}_p^p \boldsymbol{Y}^T \end{bmatrix}_{jj}$$

$$\boldsymbol{P}_{ij}^t \geq \boldsymbol{P}_{ij}^t \boldsymbol{I}_{jj}, \tag{39}$$

we can have $\mathcal{G}(\boldsymbol{P}_{ij}, \boldsymbol{P}_{ij}^t) \geq \mathcal{F}(\boldsymbol{P}_{ij})$. Therefore the lemma is proved. □

Theorem 2. *The updating rule for $\boldsymbol{P}$ can be obtained by minimizing the auxiliary function $\mathcal{G}(\boldsymbol{P}_{ij}, \boldsymbol{P}_{ij}^t)$.*

Proof. Let $(\partial \mathcal{G}(\boldsymbol{P}_{ij}, \boldsymbol{P}_{ij}^t))/(\partial \boldsymbol{P}_{ij}) = 0$, and we have

$$2 \frac{\left(\begin{array}{l} \boldsymbol{P}^t \boldsymbol{Y} \boldsymbol{Y}^T + \sum \boldsymbol{P}^t \boldsymbol{Y}_i \boldsymbol{Y}_i^T + \alpha_1(\boldsymbol{P}^t) + \\ \alpha_1 \boldsymbol{M}^T \boldsymbol{M} \boldsymbol{P} \boldsymbol{Y} \boldsymbol{Y}^T + \alpha_1 \beta \begin{bmatrix} \hat{\boldsymbol{P}}^t \boldsymbol{Y} \boldsymbol{B}_p \boldsymbol{Y}^T \\ \tilde{\boldsymbol{P}}^t \boldsymbol{Y} \boldsymbol{B}_p^p \boldsymbol{Y}^T \end{bmatrix} \end{array}\right)_{ij}}{\boldsymbol{P}_{ij}^t} (\boldsymbol{P}_{ij} - \boldsymbol{P}_{ij}^t)$$
$$+ \mathcal{F}_{ij}'(\boldsymbol{P}_{ij}^t) = 0 \tag{40}$$

from which we can derive the updating rule for $\boldsymbol{P}$

$$\boldsymbol{P}_{ij}^{(t+1)} = \boldsymbol{P}_{ij}^{(t)} \frac{\left(\begin{array}{l} \boldsymbol{D} \boldsymbol{X} \boldsymbol{Y}^T + \sum \boldsymbol{D}_i \boldsymbol{X}_i^i \boldsymbol{Y}_i^T + \alpha_1 \boldsymbol{M}^T \boldsymbol{Y} \boldsymbol{Y}^T \\ + \alpha_1 \boldsymbol{M}^T + \alpha_1 \beta \begin{bmatrix} \hat{\boldsymbol{P}}^t \boldsymbol{Y} \boldsymbol{W}_p \boldsymbol{Y}^T \\ \tilde{\boldsymbol{P}}^t \boldsymbol{Y} \boldsymbol{W}_p^p \boldsymbol{Y}^T \end{bmatrix} \end{array}\right)_{ij}}{\left(\begin{array}{l} \boldsymbol{P}^t \boldsymbol{Y} \boldsymbol{Y}^T + \sum \boldsymbol{P}^t \boldsymbol{Y}_i \boldsymbol{Y}_i^T + \alpha_1 \boldsymbol{P}^t + \\ \alpha_1 \boldsymbol{M}^T \boldsymbol{M} \boldsymbol{P}^t \boldsymbol{Y} \boldsymbol{Y}^T + \alpha_1 \beta \begin{bmatrix} \hat{\boldsymbol{P}}^t \boldsymbol{Y} \boldsymbol{B}_p \boldsymbol{Y}^T \\ \tilde{\boldsymbol{P}}^t \boldsymbol{Y} \boldsymbol{B}_p^p \boldsymbol{Y}^T \end{bmatrix} \end{array}\right)_{ij}}. \tag{41}$$

Thus the theorem is proved. □

According to **Lemma 1**, **Theorem 1** and **Theorem 2**, we have proved that the convergence of the updating rules for $\boldsymbol{P}$ and $\boldsymbol{M}$ can be theoretically guaranteed. □

References

[1] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *NIPS*, 2006.

[2] Xiaobai Liu, Shuicheng Yan, and Hai Jin. Projective nonnegative graph embedding. *IEEE T. IP*, 19(5):1126–1137, 2010.

[3] Jiwen Lu, Gang Wang, Weihong Deng, and Pierre Moulin. Simultaneous feature and dictionary learning for image set based face recognition. In *ECCV*. 2014.

[4] Changhu Wang, Zheng Song, Shuicheng Yan, Lei Zhang, and Hong-Jiang Zhang. Multiplicative nonnegative greph embedding. In *CVPR*, 2009.

[5] Meng Yang, Lei Zhang, Jian Yang, and David Zhang. Metaface learning for sparse representation based face recognition. In *ICIP*, 2010.

[6] Zhirong Yang and Erkki Oja. Linear and nonlinear projective nonnegative matrix factorization. *IEEE T. NN*, 21(5):734–749, 2010.

[7] Miao Zheng, Jiajun Bu, Chun Chen, Can Wang, Lijun Zhang, Guang Qiu, and Deng Cai. Graph regularized sparse coding for image representation. *IEEE T. IP*, 20(5):1327–1336, 2011.