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

Weiyang Liu
wyliu@pku.edu.cn
Zhiding Yu
yzhiding@andrew.cmu.edu
Yandong Wen
yandongw@andrew.cmu.edu
Rongmei Lin
rongmei.lin@emory.edu
Meng Yang
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 [8]. While such strategy is straightforward and convenient, research also indicates that it may not be optimal for classification.
tasks. Many currently prevailing approaches choose to learn dictionaries directly from training data and have led 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 [3, 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 [3] 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 $P \in \mathbb{R}^{sp \times s}$ 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:

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

(1)

where $\alpha_1, \alpha_2$ and $\alpha_3$ 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:

\[
R(D, P, X) = \|PY - DX\|_F^2 + \sum_{i=1}^{K} \|PY_i - D_iX_i\|_F^2 + \sum_{i=1}^{K} \sum_{j=1, j \neq i}^{K} \|D_jX_j^i\|_F^2,
\]

where \(X_i^j\) denotes the coding coefficients of samples \(Y_i\) associated with the sub-dictionary \(D_j\). \(\sum_{i=1}^{K} \|Y_i - D_iX_i\|_F^2\) denotes the local reconstruction error that requires the local sub-dictionaries \(D_i\) well represent samples from class \(i\). \(\|D_jX_j^i\|_F^2\) is further minimized such that inter-class coefficients \(X_j^i, i \neq j\) are relatively small compared with \(X_i^i\).

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 \(X\) as \(X = \{x_1, x_2, \ldots, x_N\}\) where \(N\) is the number of training samples, the similarity matrix of the intrinsic graph is defined as:

\[
\{W_c\}_{ij} = \begin{cases} 
1, & \text{if } i \in S^+_{k_1}(j) \text{ or } j \in S^+_{k_1}(i) \\
0, & \text{otherwise} 
\end{cases},
\]

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

\[
\{W_p\}_{ij} = \begin{cases} 
1, & \text{if } j \in S^-_{k_2}(i) \text{ or } i \in S^-_{k_2}(j) \\
0, & \text{otherwise} 
\end{cases},
\]

where \(S^-_{k_2}(i)\) denotes the index set of the \(k_2\) nearest inter-class neighbors of \(x_i\). The coefficients term is defined as:

\[
G_c(X) = \text{Tr}(X^TL_cX) - \text{Tr}(X^TL_pX),
\]

where \(L_c = B_c - W_c\) in which \(B_c = \sum_{j \neq i} \{W_c\}_{ij}\) and \(L_p = B_p - W_p\) in which \(B_p = \sum_{j \neq i} \{W_p\}_{ij}\). Imposing the graph-based discrimination makes the coding coefficients more discriminative. Interestingly, most existing discriminative coding coefficients term, such as in [24, 25], 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 [26], we design a structured projection matrix by dividing the projection matrix \(P\) into two parts:

\[
Y^{proj} = \begin{bmatrix}
\hat{Y}^{proj} \\
\tilde{Y}^{proj}
\end{bmatrix} = PY = \begin{bmatrix}
\hat{P} \\
\tilde{P}
\end{bmatrix}Y
\]

\(^1\) Stands for the reconstruction with atoms from the same class.
where \( \hat{Y}^{proj} = \{ \hat{y}_1^{proj}, \ldots, \hat{y}_N^{proj} \} \in \mathbb{R}^{q \times N} \) that serves for the certain purpose of graph embedding, and \( \hat{Y}^{proj} = \{ \hat{y}_1^{proj}, \ldots, \hat{y}_N^{proj} \} \in \mathbb{R}^{(s_p-q) \times N} \) that contains the additional information for data reconstruction (\( \hat{Y}^{proj} \) is a relaxed matrix that compensates the information loss in \( \hat{Y}^{proj} \)). Note that \( \hat{Y}^{proj} \) preserves the discriminative graph properties while the whole \( Y^{proj} \) 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 \( \mathcal{M} \) is correspondingly divided into two parts \( \mathcal{M} = \{ \hat{\mathcal{M}}, \mathcal{M} \} \) in which \( \hat{\mathcal{M}} \in \mathbb{R}^{q \times q} \) and \( \mathcal{M} \in \mathbb{R}^{q \times (s_p-q)} \). \( \hat{\mathcal{M}} \) can be considered as the complementary space of \( \hat{\mathcal{M}} \).

We first define \( y_i^{proj} \) as the \( j \)th column vector of \( Y^{proj} \) 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 \( W_p \) and \( W_p^p \) for intrinsic graph and the penalty graph is identical to \( W_c \) and \( W_c^p \), except that \( W_p \) and \( W_p^p \) measure the similarities among features and adopt different parameters. Differently, \( W_c \) and \( W_c^p \) measure similarities among coding coefficients. As [23] suggests, we have two objectives to preserve graph properties and enhance discrimination:

\[
\begin{align*}
\max_{\hat{Y}^{proj}} & \sum_{i\neq j} \| \hat{y}_i^{proj} - \hat{y}_j^{proj} \|^2_2 (W_p)_{ij} \\
\min_{\hat{Y}^{proj}} & \sum_{i\neq j} \| \hat{y}_i^{proj} - \hat{y}_j^{proj} \|^2_2 (W_p)_{ij}.
\end{align*}
\]  

(7)

\[ \sum_{i\neq j} \| y_i^{proj} - y_j^{proj} \|^2_2 (W_p)_{ij} = \sum_{i\neq j} \| \hat{y}_i^{proj} - \hat{y}_j^{proj} \|^2_2 (W_p)_{ij} + \sum_{i\neq j} \| \hat{y}_i^{proj} - \hat{y}_j^{proj} \|^2_2 (W_p)_{ij}, \]

for a specific \( Y^{proj} \), minimizing the objective function with respect to \( \hat{Y}^{proj} \) is equivalent to maximizing the objective function associated with the complementary part, i.e., \( \hat{Y}^{proj} \). Thus we constrain the projection matrix with the following equivalent objective:

\[
\begin{align*}
\min_{\hat{P}} & \text{Tr}(\hat{P}Y(L_pY^T\hat{P})^T) \\
\min_{\hat{P}} & \text{Tr}(\hat{P}Y(L_p^pY^T\hat{P})^T).
\end{align*}
\]

(8)

where \( L_p = B_p - W_p \) in which \( B_p = \sum_{j\neq i} (W_p)_{ij} \) and \( L_p^p = B_p^p - W_p^p \) in which \( B_p^p = \sum_{j\neq i} (W_p^p)_{ij} \), we formulate the graph-based projection term as follows:

\[
G_p(P, M) = \| Y - M \mathcal{P} \|_F^2 + \beta \cdot \text{Tr}(\hat{P}Y(L_pY^T\hat{P})^T) + \beta \cdot \text{Tr}(\hat{P}Y(L_p^pY^T\hat{P})^T) + \| M - P^T \|_F^2,
\]

(9)

where \( \beta \) is a scaling constant and each column of \( \mathcal{M} \) is normalized to unit \( l_2 \) norm. We use \( \| Y - M \mathcal{P} \|_F^2 + \| M - P^T \|_F^2 \) to incorporate the projection matrix into a NMF-like framework, and these two terms can further ensure the reconstruction ability of the projection \( \mathcal{P} \) and avoid the trivial solutions of \( P \). They serve similar role to the auto-encoder style reconstruction penalty term in [23, 31, 32]. Other terms in \( G_p(P) \) preserve the graph properties and enhance discrimination.

4 Optimization

We adopt standard iterative learning to jointly learn \( X, P, M \) and \( D \). 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 $Y = [Y_1, \ldots, Y_N]$, intrinsic graph $W_c, W_p$, penalty graph $W^p_c, W^p_p$, parameters $\alpha_1, \alpha_2, \alpha_3, \beta$, iteration number $T$.


Step 1: Initialization
1: $t = 1$.
2: Randomly initialize columns in $D^0, M^0$ with unit $l_2$ norm.
3: Initialize $x_i, 1 \leq i \leq N$ with $((D^0)^T(D^0) + \lambda_2 I)^{-1}(D^0)^Ty_i$ where $y_i$ is the $i$th training sample (regardless of label).

Step 2: Search local optima
4: while not convergence or $t < T$ do
5: Solve $P^t, M^t$ iteratively with fixed $D^{t-1}$ and $X^{t-1}$ via non-negative projection learning.
6: Solve $X^t$ with fixed $M^t, D^{t-1}$ and $P^t$ via non-negative projection learning.
7: Solve $D^t$ with fixed $M^t, P^t$ and $X^t$ via discriminative DL.
8: $t \leftarrow t + 1$.
9: end while

Step 3: Output
10: Output $P = P^t, D = D^t$ and $X = X^t$.

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

$$\hat{x} = \arg\min_x \{ ||Py - Dx||^2_2 + \lambda_1 ||x||_1 \}$$ (10)

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

$$\text{label}(y) = \arg\min_i \{ ||Py - D_i \delta_i(x)||^2_2 + \sigma ||x - m_i||^2_2 \}$$ (11)

where $\sigma$ is a weight to balance these two terms, and $\delta_i(\cdot)$ is the characteristic function that selects the coefficients associated with $i$th class. $m_i$ is the mean vector of the learned coding coefficient matrix of class $i$, i.e., $X_i$. Incorporating the term $||x - m_i||^2_2$ 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 $W_p$ and $W^p_p$ 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 $W_c$ and $W^p_c$, 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>ExYaleB</th>
<th>AR Face</th>
<th>Method</th>
<th>ExYaleB</th>
<th>AR Face</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>88.8</td>
<td>87.1</td>
<td>JDDRDL [3]</td>
<td>90.1</td>
<td>90.9</td>
</tr>
<tr>
<td>SRC [22]</td>
<td>91.0</td>
<td>88.5</td>
<td>FDDL [25]</td>
<td>91.9</td>
<td>92.1</td>
</tr>
<tr>
<td>D-KSVD [29]</td>
<td>75.3</td>
<td>85.4</td>
<td>DSRC [28]</td>
<td>89.6</td>
<td>88.2</td>
</tr>
<tr>
<td>LC-KSVD [8]</td>
<td>86.8</td>
<td>90.2</td>
<td>JNPDL</td>
<td>94.1</td>
<td>94.7</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Dimension</th>
<th>100</th>
<th>200</th>
<th>300</th>
<th>500</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRC [22]</td>
<td>84.0</td>
<td>87.3</td>
<td>88.5</td>
<td>89.7</td>
</tr>
<tr>
<td>FDDL [25]</td>
<td>85.7</td>
<td>88.5</td>
<td>92.0</td>
<td>92.2</td>
</tr>
<tr>
<td>DSRC [28]</td>
<td>84.8</td>
<td>86.9</td>
<td>88.2</td>
<td>89.1</td>
</tr>
<tr>
<td>JDDDRDL [3]</td>
<td>82.5</td>
<td>87.7</td>
<td>90.9</td>
<td>91.6</td>
</tr>
<tr>
<td>JNPDL</td>
<td>88.3</td>
<td>92.4</td>
<td>94.7</td>
<td>95.1</td>
</tr>
</tbody>
</table>

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 [3] 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 $54 \times 48$.

\(^3\)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 $60 \times 43$. 
Table 3. Accuracy (%) of different projection.

<table>
<thead>
<tr>
<th>Method</th>
<th>ExYaleB</th>
<th>AR Face</th>
</tr>
</thead>
<tbody>
<tr>
<td>JNPDL (with Eigenface)</td>
<td>91.8</td>
<td>92.2</td>
</tr>
<tr>
<td>JNPDL (Separate Learning)</td>
<td>92.1</td>
<td>92.5</td>
</tr>
<tr>
<td>JNPDL (Joint Learning)</td>
<td>94.1</td>
<td>94.7</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>63.0</td>
<td>COPAR</td>
<td>72.6</td>
<td>D-KSVD</td>
<td>65.9</td>
<td>LDL</td>
<td>77.2</td>
</tr>
<tr>
<td>SRC</td>
<td>72.7</td>
<td>FDDL</td>
<td>74.8</td>
<td>LC-KSVD</td>
<td>66.0</td>
<td>JNPDL</td>
<td>78.1</td>
</tr>
</tbody>
</table>

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 $10 \times 8$ 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 $y^{te} = \{y^{te}_1, y^{te}_2, \ldots, y^{te}_{K_t}\}$ in which $y^{te}_j$ is the $j$th 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 $P$ 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 $\|x\|_1$ with a $l_2$ norm $\|x\|_2^2$ and derive the decision:

$$label(y^{te}_j) = \arg\min_i \{\|Py^{te}_j - D_i\delta_i(D^\top y^{te}_j)\|_2^2\}. \quad (12)$$

where $D^\top = (D^\top D + \lambda_2 I)^{-1} D^\top$. Eventually we use the majority voting to decide the label of a video (image set).
Table 5. Recognition Accuracy (%) on 17 Oxford Flower Dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>88.6</td>
</tr>
<tr>
<td>SRC</td>
<td>88.4</td>
</tr>
<tr>
<td>LLC (20 bases)</td>
<td>89.7</td>
</tr>
<tr>
<td>JDDRDL</td>
<td>87.7</td>
</tr>
<tr>
<td>DSRC</td>
<td>88.9</td>
</tr>
<tr>
<td>MTJSRC</td>
<td>88.4</td>
</tr>
<tr>
<td>COPAR</td>
<td>88.6</td>
</tr>
<tr>
<td>SDL</td>
<td>91.0</td>
</tr>
<tr>
<td>FDDL</td>
<td>91.7</td>
</tr>
<tr>
<td>JNPDL</td>
<td>92.1</td>
</tr>
</tbody>
</table>

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

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

6.3.2 Image set based face recognition

Three video face recognition benchmark dataset, including Honda/UCSD [4], CMU MoBo [5] and YouTube Celebrities (YTC) [6] are used to evaluate the proposed JNPDL. For fair comparison, we follow the experimental setup in [6]. 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 [4], MMD [21], MDA [3], CHISD [4], 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 [4, 14] and slightly higher than [4, 14, 20].

---

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

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

6Contains 1910 video sequences of 47 celebrities from YouTube. Most videos contain noisy and low-resolution image 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

This work is partially supported by the National Natural Science Foundation for Young Scientists of China (Grant no. 61402289), National Science Foundation of Guangdong Province (Grant no. 2014A030313558)

References


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, thus 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

$$
\min_{P \geq 0, M \geq 0} \left\{ \|PY - DX\|_F^2 + \sum_{i=1}^{K} \|PY_i - D_iX_i\|_F^2 \\
+ \alpha_1 \|Y - MPY\|^2_F + \alpha_1 \beta \cdot \text{Tr}(\hat{P}Y\hat{L}_p^T\hat{P}^T) \\
+ \alpha_1 \beta \cdot \text{Tr}(\hat{Q}Y\hat{L}_p^T\hat{Q}^T) + \alpha_1 \|M - P^T\|^2_F \right\}
$$

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 $\alpha_1 \beta \cdot \text{Tr}(\hat{P}Y\hat{L}_p^T\hat{P}^T)$ with

$$
\alpha_1 \beta \cdot \left( \text{Tr}(\hat{Q}Y\hat{L}_p^T\hat{P}^T\hat{Q}^T) + \text{Tr}(\hat{Q}Y\hat{L}_p^T\hat{Q}^T\hat{Q}^T) \right)
$$

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

1
On optimizing $M$ with $P, D, X$ fixed. We can further rewrite Eq. (18) as $\text{Tr}(MG_mM^T)$ where

$$G_m = G_{m+} - G_{m-} = \begin{bmatrix} \hat{P}Y(\alpha_1\beta B_p)YT\hat{P} & 0 \\ 0 & \hat{P}Y(\alpha_1\beta B_p^T)YT \end{bmatrix} \odot I$$

(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 $\psi_{ij}$ as the Lagrange multiplier for $M \geq 0$. With $\Psi = [\psi_{ij}]$, the Lagrange $\mathcal{L}(W)$ is defined as

$$\mathcal{L}(M) = ||PY - DX||_F^2 + \sum ||PY_i - D_iX_i||_F^2 + \alpha_1||Y - MPY||_F^2 + \text{Tr}(MG_mM^T)$$

$$+ \alpha_1||M - P^T||_F^2 + \text{Tr}(\Psi M^T)$$

(4)

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

$$\frac{\partial \mathcal{L}}{\partial M} = -2\alpha_1YY^T P^T + 2\alpha_1 MPYY^T P^T$$

$$+ 2MG_m + 2\alpha_1 M - 2\alpha_1 P^T + \Psi$$

(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:

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

(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 $\alpha_1 \text{Tr}(\hat{PY}L_pYT P^T) + \alpha_2 \text{Tr}(\hat{PY}L_pYT P^T)$. Thus the Lagrange $\mathcal{L}(P)$ is

$$\mathcal{L}(P) = ||PY - DX||_F^2 + \sum ||PY_i - D_iX_i||_F^2 +$$

$$\alpha_1 \text{Tr}(\hat{PY}L_pYT P^T) + \alpha_2 \text{Tr}(\hat{PY}L_pYT P^T) +$$

$$\alpha_1||Y - MPY||_F^2 + \alpha_1||M - P^T||_F^2 + \text{Tr}(\Phi P^T)$$

(7)

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

$$P_{ij}^{(t+1)} = P_{ij}^{(t)} \left( \frac{DXY^T + \sum D_iX_iY_i^T + \alpha_1 M^TYY^T,}{P^TYY^T + \sum P_i^T Y_i Y_i^T + \alpha_1 P^T} \right)$$

$$\left( \frac{+ \alpha_1 MPYY^T P^T}{+ \alpha_1 M^TMP^TYY^T + \alpha_1 \beta [P^TBY_pYT, P^TBY_p^T]} \right).$$

(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

$$\min_{X} \left\{ ||PY - DX||_F^2 + \sum_{i=1}^{K} ||PY_i - D_iX_i||_F^2 + \alpha_3 ||X||_1ight.$$  

$$+ \sum_{i=1}^{K} \sum_{j=1, j \neq i}^{K} ||D_jX_i||_F^2 + \alpha_2 \text{Tr}(X^T (L^T) X) \}$$

(9)
Before proving the convergence of the updating rules, we first introduce some necessary preliminaries.

Proof. Appendix B: Proof of the convergence of updating rules for $D$ which is essentially a quadratic programming problem and can be directly solved by the algorithm presented in [5] (update are satisfied.

where $L' = L_e - L^p_e$. Eq. (9) can be solved using feature sign search algorithm [1] after certain formulation based on [7, 3]. We optimize $X$ class by class. Following [1], we update $X_i$ one by one in the $i$th class. We define $x_{i,j}$ as the coding coefficients of the $j$th sample in the $i$th class and reformulated the problem as

$$\min_X \left\{ \|PY_i - Dx_{i,j}\|_F^2 + \|PY_i - D_i x_{i,j}\|_F^2 \right\}$$

where $Q(x_{i,j}) = \alpha_2(x_{i,j}^T X_i L_i' + (X_i L_i')^T x_{i,j} - x_{i,j}^T L_{i,i}^T)$ in which $L_i'$ is the $i$th column of $L$, and $L_{i,i}^T$ is the entry in the $i$th row and $i$th column of $L_i$. $x_{i,j}$ can be solved via feature sign search algorithm as in [7, 3].

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

$$\min_D \left\{ \|PY - DX\|_F^2 + \sum_{i=1}^K \|PY_i - D_i X_i\|_F^2 \right\}$$

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

$$\min_{D_i,i \in \{1,2,\ldots,K\}} \left\{ \|PY_i - D_i X_i\|_F^2 + \|PY_i - D_i X_i\|_F^2 \right\}$$

which is essentially a quadratic programming problem and can be directly solved by the algorithm presented in [5] (update $D_i$ 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 $G(A, A')$ is an auxiliary function for function $F(A)$ if the conditions

$$G(A, A') \geq F(A), \quad G(A, A) = F(A)$$

are satisfied.
Lemma 1. If $G(A, A')$ is an auxiliary function for $F(A)$, then $F(A)$ is non-increasing to $A$ under the update

$$A^{t+1} = \arg\min_A G(A, A')$$

(14)

where $t$ denotes the $t$th iteration.

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

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

(15)

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

$$G(A^{t+1}, A^t) \geq F(A^{t+1})$$

(16)

which leads to

$$F(A^{t+1}) \leq F(A^t).$$

(17)

Thus $F(A)$ is non-increasing with respect to $A$ under the updating rule in Eq. (14). The lemma is proved.

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

$$F(M) = \alpha_1||Y - MPY||_F^2 + \text{Tr}(MG_mM^T) + \alpha_2||M - P^T||_F^2.$$  

(18)

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

$$F_{ij}'(M) = \alpha_1(-2YY^TP^T + 2MPY^TP)_{ij} + (2MG_m)_{ij} + \alpha_2(2M - 2P^T)_{ij}$$  

(19)

$$F_{ij}''(M) = 2\alpha_1(PYY^TP + G_m + I)_{jj}$$

(20)

where $I$ denotes an identity matrix with matched size. We construct the function $G(M_{ij}, M_{ij}')$ as

$$G(M_{ij}, M_{ij}') = F_{ij}(M_{ij}') + F_{ij}(M_{ij}) - F_{ij}((M_{ij} - M_{ij}')^2)$$

(21)

Lemma 2. $G(M_{ij}, M_{ij}')$ in Eq. (21) is an auxiliary function for the function $F_{ij}(M)$.

Proof. Because it is easily obtained that $G(M_{ij}, M_{ij}) = F_{ij}(M_{ij})$, we only need to prove that $G(M_{ij}, M_{ij}') \geq F_{ij}(M_{ij})$. We first compute the Taylor series expansion of $F_{ij}(M)$ as

$$F_{ij}(M_{ij}) = F_{ij}(M_{ij}') + F_{ij}(M_{ij}) - F_{ij}((M_{ij} - M_{ij}')^2)$$

(22)

Because the following inequalities are satisfied:

$$(M'PYY^TP)_{ij} = \sum_v (M'_v(PPYY^TP)_{vij})$$

(23)

$$\geq M'_j(PYY^TP)_{jj}$$

$$M'_j \geq M_jI_{jj}.$$  

(24)

(25)

we can let the following relation hold:

$$\frac{\alpha_1(M'PYY^TP + M'G_m + M')_{ij}M_{ij}}{M_{ij}^2} \geq (PYY^TP + G_m)_{jj}.$$  

(26)

Therefore, we can prove that $G(M_{ij}, M_{ij}') \geq F_{ij}(M_{ij})$ holds. The lemma is proved. \qed
Theorem 1. The updating rule for $M$ can be obtained by minimizing the auxiliary function $G(M_{ij}, M_{ij}^t)$.

Proof. We let the derivative of $G(M_{ij}, M_{ij}^t)$ with respect to $M_{ij}$ equal to zero, namely

$$
\frac{\partial G(M_{ij}, M_{ij}^t)}{\partial M_{ij}} = 2\alpha_1(M'PYY^TP + M'G_{m+} + M'^t)_{ij}(M_{ij} - M_{ij}^t)
$$

(27)

+ $\mathcal{F}'_{ij}(M_{ij}^t)$.

from which we can derive

$$
M_{ij}^{t+1} = M_{ij}^t - \frac{\alpha_1 YYY^TP + M'G_{m+} + \alpha_1P^T}_{ij}.$$

(28)

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

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

$$
P_i \leftarrow P_i \times \|m_i\|
$$

(29)

$$
m_i \leftarrow m_i / \|m_i\|
$$

where $P_i$ is the $i$th column vector of the projection matrix $P$. Considering Eq. (29) and the fixed $M$, we can rewrite the optimization objective (Eq. (10) in the paper) as

$$
\mathcal{F}(P) = \|PY - DX\|_F^2 + \sum_i \|PY_i - D_iX_i\|_F^2 + \alpha_1 \beta \text{Tr}(PYPY_iY^TP) + \alpha_1 \beta \text{Tr}(PYPY_iY^TP)
$$

(30)

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

$$
\mathcal{F}'_{ij}(P) = 2(DXX^T)_{ij} - 2(DXX^T)_{ij} + 2\sum_i (PYY_iY_i)_{ij}
$$

(31)

$$
- 2\sum_i (D_iX_iY_i)_{ij} - 2\alpha_1(M^TYY^TP + \alpha_1M^TP^T)_{ij}.
$$

$$
\mathcal{F}_p'(P) = 2(YY^T)_{jj} + 2\sum\sum (YY^T)_{jj} + 2\alpha_1(M^T)_{ij}(YY^T)_{jj}
$$

(32)

+ $2\alpha_1 \beta \begin{bmatrix} P_{ij}P_{ij}^T & 0 \end{bmatrix}$

+ $2\alpha_1 \beta \begin{bmatrix} P_{ij}P_{ij}^T & 0 \end{bmatrix}$

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

$$
G(P_{ij}, P_{ij}^t) = \mathcal{F}_{ij}(P_{ij}^t) + \mathcal{F}_{ij}(P_{ij})(P_{ij} - P_{ij}^t)
$$

(33)

$$
P_{ij}P_{ij}^T + \alpha_1(PP^T) + \left[ \begin{array}{c} \alpha_1M^TP_{ij}YY^TP_{ij} \end{array} + \alpha_1(PP^T) \right]_{ij}
$$

Since the following relations hold:

$$
(P'YY^T)_{ij} = \sum_v (P_{ij}^v(YY^T)_{vjo})
$$

(35)

$$
\geq P_{ij}^v(YY^T)_{jj},
$$

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

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

$$
\mathcal{F}_{ij}(P_{ij}) = \mathcal{F}_{ij}(P_{ij}^t) + \mathcal{F}_{ij}(P_{ij})(P_{ij} - P_{ij}^t)
$$

(34)

$$
+ \frac{1}{2} \mathcal{F}''_{ij}(P_{ij})(P_{ij} - P_{ij}^t)^2.
$$

Since the following relations hold:

$$
(P'YY^T)_{ij} = \sum_v (P_{ij}^v(YY^T)_{vjo})
$$

(35)
\[
\sum_i P^t Y_i Y_i^T = \sum_i (P^t_i, (\sum Y_i Y_i^T)_{ij}), \\
\geq P^t_i (\sum Y_i Y_i^T)_{jj},
\]
(36)
\[
(M^T M P^t Y Y^T)_{ij} = \sum_v ((M^T M P^t)_{i\nu} (Y Y^T)_{\nu j}) \\
\geq (M^T M P^t)_{ij} (Y Y^T)_{jj} \\
= \sum_v ((M^T M)_{i\nu} P^t_{ij}) (Y Y^T)_{jj},
\]
(37)
\[
\left[ \hat{P}^t Y B_p Y^T \right]_{ij} = \begin{cases} 
\sum_v (P^t_{i\nu} (Y B_p Y^T)_{\nu j}), & \text{if } ij \leq q \\
\sum_v (P^t_{i\nu} (Y B_p Y^T)_{\nu j}), & \text{otherwise}
\end{cases}, \\
\geq \begin{cases} 
P^t_{ij} (Y B_p Y^T)_{jj}, & \text{if } ij \leq q \\
0, & \text{otherwise}
\end{cases}, \\
\geq \begin{cases} 
P^t_{ij} (Y L_p Y^T)_{jj}, & \text{if } ij \leq q \\
0, & \text{otherwise}
\end{cases},
\]
(38)
\[
= P^t_{ij} \begin{bmatrix} Y L_p Y^T \\ Y L_p Y^T \end{bmatrix}_{jj}
\]
(39)

we can have \( \mathcal{G}(P_{ij}, P^t_{ij}) \geq \mathcal{F}(P_{ij}) \). Therefore the lemma is proved.

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

**Proof.** Let \((\partial \mathcal{G}(P_{ij}, P^t_{ij}))/\partial P_{ij} = 0\), and we have
\[
2 \left( \begin{array}{c}
P^t Y Y^T + \sum Y_i Y_i^T + \alpha_1 (P^t) + \\
\alpha_1 M^T M P^t Y Y^T + \alpha_1 \beta \left[ \hat{P}^t Y B_p Y^T \right]_{ij}
\end{array} \right)_{ij} \\
\frac{P^t_{ij}}{(P_{ij} - P^t_{ij})}.
\]
(40)
\[
+ \mathcal{F}'(P^t_{ij}) = 0
\]
from which we can derive the updating rule for \( P \)
\[
P^{(t+1)}_{ij} = P^{(t)}_{ij} \\
D X Y^T + \sum D_i Y_i Y_i^T + \alpha_1 M^T Y Y^T + \\
\alpha_1 M^T + \alpha_1 \beta \left[ \hat{P}^t Y B_p Y^T \right]_{ij} \\
\frac{P^t Y Y^T + \sum P^t Y_i Y_i^T + \alpha_1 P^t + \alpha_1 M^T M P^t Y Y^T + \alpha_1 \beta \left[ \hat{P}^t Y B_p Y^T \right]_{ij}}{P^t Y Y^T + \sum P^t Y_i Y_i^T + \alpha_1 P^t + \alpha_1 M^T M P^t Y Y^T + \alpha_1 \beta \left[ \hat{P}^t Y B_p Y^T \right]_{ij}}.
\]
(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 \( P \) and \( M \) can be theoretically guaranteed.

---

**References**


