Dual Pursuit for Subspace Learning

Shuangyan Yi, Yingyi Liang, Zhenyu He, Yi Li, Yiu-ming Cheung

Computer Science and Technology
Harbin Institute of Technology Shenzhen Graduate School


Sep 10, 2017
Acknowledgement

Biocomputing Lab
Dual Pursuit for Subspace Learning

Shuangyan Yi\textsuperscript{a,1}, Yingyi Liang\textsuperscript{a,1}, Zhenyu He\textsuperscript{a,*}, Yi Li\textsuperscript{a}, Yiu-ming Cheung\textsuperscript{b}

\textsuperscript{a}School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, China
\textsuperscript{b}Department of Computer Science and Institute of Research and Continuing Education, Hong Kong Baptist University, Hong Kong

Abstract

In general, Low-Rank Representation (LRR) aims to find the lowest-rank representation with respect to a dictionary. In fact, the dictionary is a key aspect of low-rank representation. However, a lot of low-rank representation methods usually use the data itself as a dictionary (i.e., a fixed dictionary), which may degrade their performances due to the lack of clustering ability of a fixed dictionary. To this end, we propose to learn a locality-preserving dictionary instead of the fixed dictionary for low-rank representation, where the locality-preserving dictionary is constructed by using a graph regularization technique to capture the intrinsic geometric structure of the dictionary, and hence the locality-preserving dictionary has underlying clustering ability. In this way, the obtained low-rank representation via the locality-preserving dictionary has a better grouping-effect representation. Inversely, a better grouping-effect representation can help to learn a good dictionary. The locality-preserving dictionary and the grouping-effect representation interact with each other, whereby dual pursuit is called. The proposed method, namely, Dual Pursuit for Subspace Learning, provides us a robust method for clustering and classification simultaneously, and compares favorably with the other state-of-the-art methods.

1. Introduction

Low-Rank Representation (LRR) \cite{1, 2, 3}, as a promising subspace clustering method, aims to capture the underlying data structure from a global perspective, which has been reported to be superior to similar methods \cite{4}. Due to the effectiveness of LRR, various methods based on it \cite{5, 6, 7, 8} have been proposed and widely used in motion segmentation \cite{9, 10, 11}, face recognition \cite{12, 13, 14}, visual tracking \cite{15, 16}, saliency detection \cite{17, 18}, and so on.

LRR exploits the self-expressive ability of the data itself via low-rank and finds the underlying low-rank structure. Although LRR provides us with an efficient way to automatically correct the corruptions lying in the original data, it only considers the global structure. Therefore, it is

\textsuperscript{*}Corresponding author.

Email addresses: shuangyanshuangfei@163.com (Shuangyan Yi), liangyingyi002@163.com (Yingyi Liang), zyhe@hitsz.edu.cn (Zhenyu He), ly_res@163.com (Yi Li), ymc@comp.hkbu.edu.hk (Yiu-ming Cheung)

\textsuperscript{1}Both authors contributed equally to this work.
a natural idea to take the local manifold structure into consideration [2, 19, 16]. Lu et al. [2] proposed to incorporate a graph Laplacian into LRR and to enforce the desired low-rank subspace structures. Liu et al. [19] introduced a manifold regularization into LRR and formed a non-negative low-rank representation method.

All of the aforementioned methods use the observation data itself as a dictionary (i.e., a fixed dictionary), however, such a strategy may degrade their performances, especially when the intrinsic geometric structures of observations are hidden in observations. To this end, the idea of constructing a novel dictionary for low-rank representation has appeared in the literature. For example, latent low-rank representation (LatLRR) [6] is proposed to construct a dictionary by using both observed and unobserved data. Similar to LatLRR, the dual low-rank method in [20] is proposed to learn a set of low-rank features as a dictionary for detecting the low-rank salient regions. Moreover, a novel low-rank version is constructed in the transformed data space [21], which uses the transformed data as a dictionary to recover the transformed data itself. In contrast, we argue that learning a locality-preserving dictionary is necessary for low-rank representation, because the locality-preserving dictionary has an underlying clustering ability and this will help learn a good grouping-effect representation. In this paper, the locality-preserving dictionary is therefore constructed by using a graph regularization technique, which is able to capture the intrinsic geometry structure of data and implicitly has a clustering ability [22]. Based on such a locality-preserving dictionary, the grouping-effect representation is likely to be in the dense diagonal blocks due to the stronger discriminative features extracted by locality-preserving dictionary. Here, the proposed method, namely dual pursuit, interacts locality-preserving dictionary and grouping-effect representation each other to simultaneously obtain the most optimal dictionary and representation. Therefore, unlike the traditional subspace learning methods that usually consider clustering and classification independently, the proposed method can perform both clustering and classification simultaneously.

The main contributions of this paper are summarized as follows:

- We propose to learn a locality-preserving dictionary for low-rank representation instead of a fixed dictionary. The learned locality-preserving dictionary can capture the intrinsic geometry structure of data, and thus the learned dictionary has an underlying clustering ability to some extent.
- We propose to simultaneously learn the locality-preserving dictionary and grouping-effect representation. Such a strategy can make the locality-preserving dictionary and grouping-effect representation interact with each other and achieves the most optimal locality-preserving dictionary and grouping-effect representation.

2. Brief Review of Low-Rank Representation and Graph Regularization Technique

In this section, we briefly review low-rank representation and graph regularization technique.

2.1. Low-Rank Representation

Given the observed data $X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{d \times n}$ ($n$ is the number of samples and $d$ is the dimension of features), which is approximately drawn from a mixture of multiple low-rank
subspaces, the LRR method [2] uses the observed data itself to find the lowest-rank representation matrix \( Z \) of all data jointly as follows:

\[
\arg \min_{Z,E} \|Z\|_s + \lambda \|E\|_{2,1},
\]

\[
s.t. \quad X = XZ + E,
\]

where \( \|Z\|_s \), defined as the sum of all singular values of \( Z \), is the so-called nuclear norm [23], and \( \| \cdot \|_{2,1} \) is the \( \ell_{2,1} \)-norm to characterize the error \( E \).

2.2. Graph Regularization Technique

The graph regularization technique is built on a graph and the graph is constructed from the data samples. In the graph, the weight of the edge between data samples \( x_i \) and \( x_j \) is usually defined as:

\[
W_{ij} = \begin{cases} 
1, & \text{if } x_i \in N_K(x_j) \text{ or } x_j \in N_K(x_i) \\
0, & \text{otherwise}
\end{cases}
\]

(2)

where \( N_K(x_i) \) indicates the set of \( K \) nearest neighbours of \( x_i \). The set of \( W_{ij} \) is denoted as \( W \), which is a sparse symmetric \( n \times n \) matrix.

The goal of the graph regularization technique is to preserve the locality relation among samples in the low-dimensional subspace. Assuming that \( \Phi_i \) and \( \Phi_j \) are any two low-dimensional subspace points produced by the original samples \( x_i \) and \( x_j \), the formulation of the graph regularization technique can be generalized as follows:

\[
\arg \min \sum_{ij} \| \Phi_i - \Phi_j \|_2^2 W_{ij}.
\]

(3)

The graph regularization technique is able to weaken unnecessary connections and strengthen the necessary connections. Generally, \( \sum_{ij} \| P^T x_i - P^T x_j \|_2^2 W_{ij} \) and \( \sum_{ij} \| z_i - z_j \|_2^2 W_{ij} \) are two common graph regularization terms, where the former imposes the graph regularization technique on the projection matrix \( P \) while the latter imposes the graph regularization technique on the representation coefficient matrix \( Z \).

3. Dual Pursuit for Subspace Learning

In this section, the objective function of dual pursuit is first constructed and then its optimization algorithm is given.

3.1. Objective Function of the Proposed Method

Low-rank representation uses the data itself as a fixed dictionary to recover the original data. However, using the data directly itself as a dictionary may be blind and may degrade the performance of grouping-effect representation. To this end, it is necessary to improve the performance of grouping-effect representation by learning an effective dictionary. In fact, a priori information (i.e., the grouping-effect) can be used to guide the learning of the effective dictionary. Fortunately, the graph regularization term \( \sum_{ij} \| P^T x_i - P^T x_j \|_2^2 W_{ij} \) can make the transformed
data $P^TX$ have the grouping-effect property. This is because the transformed data, by preserving the neighborhood structure in the transformed space, implicitly emphasizes the data groups that are more correct than the original data groups. More specifically, the graph regularization term $\sum_{ij} \| P^T x_i - P^T x_j \|^2 W_{ij}$ can make the neighboring points in the original space nearer in the transformed data space and the faraway points in the original space further in the transformed data space [22]. Taking Fig. 1 as an example, node3 and node1 are usually classified as one class due to their nearest Euclidean distance. In fact, node3 and node1 are not in the same class. However, the graph regularization term $\sum_{ij} \| P^T x_i - P^T x_j \|^2 W_{ij}$ will make node4 and node3 nearer while make node4 and node1 farther. That is, in the transformed space, node4 and node3 are classified as one class while node4 and node1 are not classified as one class. Naturally, node3 and node1 are not classified as the same class. Therefore, the learned data transformed by the graph regularization technique, which can correctly capture the intrinsic geometric structure of the data, is called the locality-preserving dictionary. Based on such a locality-preserving dictionary, a more optimal grouping-effect representation coefficient matrix is expected.

![Figure 1: Illustration of the locality-preserving dictionary. There are a total of six nodes in the original space, where the red curve line is the neighbourhood of node4 and the blue curved line is the neighbourhood of node2. Usually, node3 and node1 would be classified as one class due to their nearest Euclidean distance. In fact, node3 and node1 are not in the same class. However, the graph regularization technique can correctly capture the neighbourhood structures of data.](image-url)

For the noiseless data, the proposed dual pursuit is written as follows:

$$\arg\min_{Z,P} \|Z\|_* + \frac{\lambda}{2} \sum_{ij} \| P^T x_i - P^T x_j \|^2 W_{ij},$$

s.t. $X = P^T X Z$, \hspace{1cm} (4)

Furthermore, when the data is corrupted by noise (i.e., illumination corruptions or random pixel corruptions), the objective function of dual pursuit is written as follows:

$$\arg\min_{Z,P,E} \|Z\|_* + \frac{\lambda}{2} \sum_{ij} \| P^T x_i - P^T x_j \|^2 W_{ij} + \gamma \|E\|_{2,1},$$

s.t. $X = P^T X Z + E$, \hspace{1cm} (5)
where $\lambda \geq 0$ and $\gamma \geq 0$ are two balance parameters, $Z$ is the low-rank representation coefficient matrix, $P \in \mathbb{R}^{m \times m}$ is the transformation matrix, and $E$ is an error term. Here, $W_{ij}$ defined in Eq. (2) is used to constrain $P^T x_i$ and $P^T x_j$. The graph regularization term $\sum_{ij} \| P^T x_i - P^T x_j \|_2^2 W_{ij}$ is used to find a transformation matrix $P$ such that, under this transformation matrix, the sum of Euclidean distances between data pairs that are local to each other is minimized.

There are two main distinctive aspects of our method: the graph regularization term (i.e. $\sum_{ij} \| P^T x_i - P^T x_j \|_2^2 W_{ij}$) and the constraint (i.e. $X = P^T XZ + E$). In terms of the graph regularization term, the graph weight $W$ is used to constrain the transformed data pairs (e.g., $P^T x_i$ and $P^T x_j$) such that they preserve the local geometric structure well. One important reason for adding $\sum_{ij} \| P^T x_i - P^T x_j \|_2^2 W_{ij}$ into our objective function is to let the learned dictionary $P^T X$ have the grouping-effect property. Based on such a locality-preserving dictionary, the grouping-effect representation will bring a better clustering result than that based on a fixed dictionary. On the other hand, through the constraint $X = P^T XZ + E$, the dual pursuit is produced where the optimized variables $Z$ and $P$ interact with each other. Finally, the optimal $Z^*$ can be used for subspace clustering and the optimal $P^*$ can be used for feature extraction.

3.2. Optimization

Many algorithms have been developed to optimize the low-rank representation methods, such as Singular Value Thresholding (SVT) [24], Augmented Lagrange Multiplier (ALM) [25], Alternating Direction Method (ADM) [26], and Linearized Alternating Direction Method with Adaptive Penalty (LADMAP) [27]. Recently, in order to deal with multi-block variables, the Linearized Alternating Direction Method with Parallel Splitting and Adaptive Penalty (LADMAPSAP) [28] has been proposed with a convergence guarantee.

Using some simple algebraic formulations [29, 19, 30, 31] for $\frac{1}{2} \| P^T x_i - P^T x_j \|_2^2 W_{ij}$, Eq. (5) is converted into the equivalent optimization formulation as follows:

$$\arg \min_{Z; P; E} \| Z \|_*$ + \lambda \text{tr}(P^T XLX^TP) + \gamma \| E \|_{21},$$

s.t. $X = P^T XZ + E$,

$$\text{where } L = D - W \text{ is the graph Laplacian matrix [32], } D \text{ is a diagonal matrix whose entries are column sums of } W, \text{ and } W \text{ is the weight matrix with } W_{ij}.$$

The main part of the objective function in Eq. (6) is convex, and the constraint in Eq. (6) is convex only when the variable $P$ or $Z$ is fixed. For efficiency, we utilize the ALM method [25] to solve the optimization problem. One trick in applying ALM is to make the objective function separable. Therefore, we introduce one auxiliary variable $A$ and reformulate Eq. (6) into the following equivalent formulation:

$$\arg \min_{Z; A; P; E} \| A \|_* + \lambda \text{tr}(P^T XLX^TP) + \gamma \| E \|_{21},$$

s.t. $X = P^T XZ + E, Z = A$.

5
Then, using the ALM method, we can minimize the following augmented Lagrange function:

\[
\arg \min_{Z,A,P,E} \|A\|_* + \lambda \text{tr}(P^T XLX^TP) + \gamma \|E\|_{21} + \text{tr}(Y_1^T(X - P^TXZ - E)) \\
+ \text{tr}(Y_2^T(Z - A)) + \frac{\mu}{2} (\|X - P^TXZ - E\|_F^2 + \|Z - A\|_F^2),
\]

where \(\text{tr}()\) and \(\|\cdot\|_F\) denote the trace and Frobenius norm of a matrix respectively, and \(\mu > 0\) is a penalty parameter. This augmented Lagrange function is unconstrained, and hence it can be minimized with respect to \(Z, A, P\) and \(E\) respectively by fixing other variables. Finally, we update the Lagrange multipliers \(Y_1\) and \(Y_2\). The main solving process is given as follows.

**Step 1:** Fix other variables and update \(Z\). Eq. (8) is reduced to the following formulation:

\[
\arg \min_{Z} \left\| X - P_k^TXZ_k - E_k + \frac{Y_k^k}{\mu} \right\|_F^2 + \left\| Z_k - A_k + \frac{Y_k^k}{\mu} \right\|_F^2.
\]

By derivating Eq. (9) with respect to \(Z\) be 0, this optimization solution \(Z_{k+1} = (\mu I + \mu X^TP_kP_k^T X)^{-1} (X^TP_kY_k^k + \mu X^TP_k(X - E_k) - Y_k^k + \mu A_k)\) is got.

**Step 2:** Fix other variables and update \(A\). Eq. (8) is reduced to the following formulation:

\[
\arg \min_{A} \|A\|_* + \frac{\mu}{2} \left\| A - (Z + \frac{Y_k^k}{\mu}) \right\|_F^2.
\]

This optimization solution \(A_{k+1} = J_{\frac{1}{\mu}}(Z_{k+1} + \frac{Y_k^k}{\mu})\), where \(J\) is the thresholding operator with respect to the singular value \(\frac{1}{\mu}\), can be obtained via SVT operator [24].

**Step 3:** Fix other variables and update \(P\). Eq. (8) is reduced to the following formulation:

\[
\arg \min_{P} \lambda \text{tr}(P^T XLX^TP) + \frac{\mu}{2} \left\| X - P^TXZ - E_k + \frac{Y_k^k}{\mu} \right\|_F^2.
\]

By derivating Eq. (11) with respect to \(P\) be 0, this optimization solution \(P_{k+1} = (2\lambda XLX^T + \mu XZ_{k+1}^TX^T)^{-1}\mu XZ_{k+1}(X - E_k + Y_k^k/\mu)^T\) is got.

**Step 4:** Fix other variables and update \(E\). Eq. (8) is reduced to the following formulation:

\[
\arg \min_{E} \gamma \|E\|_{21} + \frac{\mu}{2} \left\| X - P^TXZ - E + \frac{Y_k^k}{\mu} \right\|_F^2,
\]

which has the closed-form solution \(E_{k+1} = S_{\frac{1}{\mu}}(X - P_{k+1}^TXZ_{k+1}^{} + \frac{Y_k^k}{\mu})\) using the shrinkage operator [25], where \(S\) is the \(\ell_{21}\) minimization operator [7].

The complete solving process is shown in Algorithm 1. The major computation of Algorithm 1 is at Step 2, which requires computation of the Singular Value Decomposition (SVD) of the matrix \(Z_{k+1} + Y_k^k/\mu \in R^{n \times n}\), where \(k\) is the number of iterations. Therefore, the complexity of this algorithm is \(O(kn^3)\). When \(n\) is large, its computational cost is very high. Some references [3, 33] provide a way to reduce the computational cost. Moreover, the most optimized solutions \(Z_{k+1}, A_{k+1}, P_{k+1}\), and \(E_{k+1}\) are denoted as \(Z^*, A^*, P^*,\) and \(E^*\).
Algorithm 1. Optimization Algorithm

Initialize: $Z_0 = A_0 = 0$, $E_0 = 0$, $Y_1^0 = 0$, $Y_2^0 = 0$, $P_0 = I$, $\mu_0 = 10^{-3}$, $\max \mu = 10^6$, $\rho = 1.1$, $\varepsilon = 10^{-6}$, $k = 0$;

While not converged do

1: Fix the others and update $Z$ by setting:
   $$Z_{k+1} = (\mu_k I + \mu_k X^T P_k P_k^T X)^{-1} (X^T P_k Y_1^k + \mu_k X^T P_k (X - E_k) - Y_2^k + \mu_k A_k);$$

2: Fix the others and update $A$ by setting:
   $$A_{k+1} = J \frac{1}{\mu_k} (Z_{k+1} + Y_2^k/\mu_k);$$

where $J$ is the thresholding operator [24] with respect to the singular value $1/\mu_k$.

3: Fix the others and update $P$ by setting:
   $$P_{k+1} = (2X L X^T + \mu_k X Z_{k+1} Z_{k+1}^T X^T)^{-1} \mu_k X Z_{k+1} (X - E_k + Y_1^k/\mu_k)^T;$$

4: Fix the others and update $E$ by setting:
   $$E_{k+1} = S \frac{1}{\mu_k} (X - P_{k+1} X Z_{k+1} + Y_1^k/\mu_k);$$

where $S$ is the $\ell_2$ minimization operator [7].

5: Update the multipliers:
   $$Y_1^{k+1} = Y_1^k + \mu_k (X - P_{k+1} X Z_{k+1} - E_{k+1});$$
   $$Y_2^{k+1} = Y_2^k + \mu_k (Z_{k+1} - A_{k+1});$$

6: Update the parameter $\mu_{k+1}$: $\mu_{k+1} = \min (\rho \mu_k, \max \mu)$

7: Check the convergence conditions:
   $$\| X - P_{k+1} X Z_{k+1} - E_{k+1} \|_\infty < \varepsilon$$ and
   $$\| Z_{k+1} - A_{k+1} \|_\infty < \varepsilon;$$

8: Update $k: k \leftarrow k + 1$;

End while

4. Experiments

Implementation Details: In our experiments, four databases are adopted, i.e., PIE, Extended Yale B, COIL20, and USPS. For each database, all the data is randomly divided into two groups. More specifically, for the PIE database, 15 images per class, that is, a total of $15 \times 68 = 1020$ images, are randomly selected as the first group of data and the remaining 612 images are randomly selected as the second group of data; for Extended Yale B, 30 images per class, that is, a total of $30 \times 38 = 1140$ images, are randomly selected as the first group of data and the remaining 1274 images are randomly selected as the second group of data; for COIL20, 5 images per class, that is, a total of $5 \times 20 = 100$ images, are randomly selected as the first group of data and the remaining 1340 images are randomly selected as the second group of data; for USPS, 20 images per class, that is, a total of $20 \times 10 = 200$ images, are randomly selected as the first group of data and the remaining 9098 images are randomly selected as the second group of data.

In our experiments, the proposed method is optimized on the first group of data of each database, and then, the optimized variables $Z^*$ and $P^*$ are simultaneously obtained for the same database and the same parameters. After the optimized variables $Z^*$ and $P^*$ have been simultaneously obtained under the same parameters, $Z^*$ is used to perform data clustering on the first group of data while $P^*$ is used to perform data classification on the second group of data. Before that, we first discuss the effectiveness of the proposed dual pursuit.
4.1. Experiment on Subspace Clustering

Data clustering aims to group samples into different groups. In this paper, we adopt two metrics, that is, accuracy (AC) and normalized mutual information (NMI) [31], to evaluate the clustering performance.

Given a data point \( x_i \), let \( z \) and \( \hat{z} \) be the ground truth label and the label produced by a clustering approach, respectively. Then the AC measure is defined by

\[
AC = \frac{\sum_{i=1}^{n} \delta(\hat{z}(i), Match(\hat{z}; z)(i))}{n},
\]

where \( n \) is the total number of samples, \( Match(\cdot) \) is the best matching function that permutes \( \hat{z} \) to \( z \), and \( \delta(a, b) \) is set to 1 if and only if \( a = b \).

The NMI measure between two index sets \( K \) and \( K' \) is defined as

\[
NMI(K, K') = \frac{MI(K, K')}{\max(H(K), H(K'))},
\]

where \( H(K) \) and \( H(K') \) denote the entropy of \( K \) and \( K' \), respectively. MI is defined as

\[
MI(K, K') = \sum_{y \in K} \sum_{y' \in K'} p(x, y) \log_2 \left( \frac{p(x, y)}{p(x)p(y)} \right),
\]

where \( p(y) \) and \( p(x) \) denote the marginal probability distribution functions of \( K \) and \( K' \), respectively, and \( p(x, y) \) is the joint probability distribution function of \( K \) and \( K' \). Usually, \( NMI(K, K') \) ranges from 0 to 1, where the value 1 means that the two clusters are identical and the value 0 means that the two clusters are independent.

Experimental Results: The proposed method is compared with three state-of-the-art subspace clustering methods based on low-rank representation LRR, LatLRR, and smooth representation clustering (SMR)[34]. Here, SMR provides two ways to compute the edge weights of an undirected graph, and here we use the popular \( J_1 \) for a fair comparison.

In order to show the robustness of the proposed method, the clustering experiments are implemented with different number of clusters. More specifically, on the first group data of each database, we use its first \( k \) classes for the corresponding data clustering experiments. The detailed clustering results are reported in four tables, where the red typeface indicates the results of our method. From Tables 1 and 3, it can be seen that our method consistently outperforms the other state-of-the-art methods. From Table 2, it can be seen that our method obviously outperforms LRR and LatLRR but gives a comparable result to SMR when the number of clusters is \( k = 38 \). From Table 4, it can be seen that our method compares favourably with LRR. To sum up, our method gives the better clustering results than the other methods overall.

Generally, the clustering performance decreases as the number of clusters increases. As can be seen, the AC result obtained by the proposed method decreases as the number of clusters increases. However, it can be observed that there exists a fluctuation in terms of NMI in our proposed method. The phenomenon could be interpreted by the use of K-means, whose result may be different under different initializations of cluster centers. A similar phenomenon occurred strongly in the other
methods in terms of both AC and NMI. This also shows that our method is more robust than the other methods.

Table 1: **Clustering accuracy (%)** on the first group of data of the PIE database, where Num.# is the number of clusters.

<table>
<thead>
<tr>
<th>Cluster</th>
<th>AC(%)</th>
<th>NMI(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LRR</td>
<td>LatLRR</td>
</tr>
<tr>
<td>Num.#</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>95.00</td>
<td>86.67</td>
</tr>
<tr>
<td>30</td>
<td>63.33</td>
<td>60.89</td>
</tr>
<tr>
<td>56</td>
<td>60.71</td>
<td>60.48</td>
</tr>
<tr>
<td>68</td>
<td>60.85</td>
<td>61.57</td>
</tr>
</tbody>
</table>

Table 2: **Clustering accuracy (%)** on the first group of data of the Extended Yale B database, where Num.# is the number of clusters.

<table>
<thead>
<tr>
<th>Cluster</th>
<th>AC(%)</th>
<th>NMI(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LRR</td>
<td>LatLRR</td>
</tr>
<tr>
<td>Num.#</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>63.67</td>
<td>77.07</td>
</tr>
<tr>
<td>20</td>
<td>59.50</td>
<td>63.67</td>
</tr>
<tr>
<td>30</td>
<td>46.56</td>
<td>50.44</td>
</tr>
<tr>
<td>38</td>
<td>56.14</td>
<td>57.02</td>
</tr>
</tbody>
</table>

Table 3: **Clustering accuracy (%)** on the first group of data of the COIL20 database, where Num.# is the number of clusters.

<table>
<thead>
<tr>
<th>Cluster</th>
<th>AC(%)</th>
<th>NMI(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LRR</td>
<td>LatLRR</td>
</tr>
<tr>
<td>Num.#</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>60.00</td>
<td>64.00</td>
</tr>
<tr>
<td>10</td>
<td>60.00</td>
<td>62.00</td>
</tr>
<tr>
<td>15</td>
<td>62.67</td>
<td>65.33</td>
</tr>
<tr>
<td>20</td>
<td>65.00</td>
<td>62.00</td>
</tr>
</tbody>
</table>

Table 4: **Clustering accuracy (%)** on the first group of data of the USPS database, where Num.# is the number of clusters.

<table>
<thead>
<tr>
<th>Cluster</th>
<th>AC(%)</th>
<th>NMI(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LRR</td>
<td>LatLRR</td>
</tr>
<tr>
<td>Num.#</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>92.50</td>
<td>65.00</td>
</tr>
<tr>
<td>6</td>
<td>78.33</td>
<td>81.67</td>
</tr>
<tr>
<td>8</td>
<td>81.87</td>
<td>72.50</td>
</tr>
<tr>
<td>10</td>
<td>77.00</td>
<td>63.00</td>
</tr>
</tbody>
</table>

Furthermore, the data decomposition abilities of the proposed method and LRR are compared in Fig. 2. By comparing the recovered data of our method (i.e., \((P^*)^T X Z^*)\) with that of LRR (i.e., \(XZ^*)\), we can see that both the two methods have good performance and our method is visually...
better than LRR (see the red and yellow boxes). This is because, being based on a learned locality-preserving dictionary, our method may have a stronger power to push the noise into the error term than LRR based on a fixed dictionary (see the error term $E^*$). It is worth noting that the result of subspace clustering only depends on the optimized representation coefficient matrix $Z^*$. So, we additionally display our $XZ^*$ under the $(P^*)^T X Z^*$ (see the grey arrow). By comparing our $XZ^*$ with that of LRR, it can be seen that our $XZ^*$ can capture the generality characteristic of the intraclass faces/objects and achieves a high clustering accuracy.

Figure 2: Visualization of data decomposition of some samples from both the Extended Yale B and the COIL20 databases. A total of twenty samples from two databases (each database includes two classes and each class includes five images) are selected to display the performance of our method and LRR. For this total of twenty samples, the data decomposition of LRR is shown in the first row while ours is shown in the second row.
Table 5: **Classification accuracy** on four databases. The optimal dimension is shown in brackets.

<table>
<thead>
<tr>
<th>Database</th>
<th>LPP</th>
<th>NPE</th>
<th>LatLRR</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIE</td>
<td>89.38(300D)</td>
<td>90.36(358D)</td>
<td>91.67(1024D)</td>
<td><strong>92.81(1024D)</strong></td>
</tr>
<tr>
<td>Extended Yale B</td>
<td>90.42(350D)</td>
<td>87.44(415D)</td>
<td>85.95(1024D)</td>
<td><strong>90.74(1024D)</strong></td>
</tr>
<tr>
<td>COIL20</td>
<td>74.18(35D)</td>
<td>73.06(47D)</td>
<td><strong>83.06(1024D)</strong></td>
<td>81.04(1024D)</td>
</tr>
<tr>
<td>USPS</td>
<td>76.28(50D)</td>
<td>76.42(68D)</td>
<td><strong>88.70(256D)</strong></td>
<td>84.03(256D)</td>
</tr>
</tbody>
</table>

4.2. **Experiment on Feature Extraction**

In this section, we use $P^*$ to perform classification for the second group of data. That is, for a training sample or testing sample $x$, the transformed feature vector $y$ can be calculated as follows:

$$y = (P^*)^T x.$$  \hfill (13)

It is worth noting that $y$ has the same dimension as $x$, unlike in the general dimension reduction methods. After all the training samples (i.e., the first group data) and testing samples (i.e., the second group of data) have been transformed by $P^*$, the 1-nearest neighbour classifier is used to classify these testing samples in the transformed space.

**Experimental Results:** The proposed method is compared with LatLRR and two popular dimension reduction methods, including Locality Preserving Projection (LPP) [22] and Neighbourhood Preserving Embedding (NPE) [35]. For each database, the first group of data is used as the training samples and the second group of data is used as the testing samples. More specifically, for the PIE database, the selected 15 images per class are used as the training samples and the remaining images are used as the testing samples; on the Extended Yale B database, the selected 30 images per class are used as the training samples and the remaining images are used as the testing samples; for the COIL20 database, the selected five images per class are used as the training samples and the remaining images are used as the testing samples; for the USPS database, the selected 20 images per class are used as the training samples and the remaining images are used as the testing samples.

Table 5 shows that our method gives the best classification accuracy on the Extended Yale B and PIE databases. As is well known, the face images may reside on a nonlinear submanifold [36, 37] and the local feature extraction method may achieve a good classification performance. Therefore, LPP and NPE have achieved a good classification accuracy. However, our method outperforms them by 0.32-3.45%. This is because our method not only uses $P$ to capture the local geometric structure but also uses $Z$ to capture the global structures, and hence our method is likely to achieve the best performance on these two facial databases. On the COIL20 and USPS databases, our method has approximately the same performance to LatLRR. This is because both the COIL20 and USPS databases have a big distance between inter-classes and LatLRR is likely to achieve the best performance as a global feature extraction method.

**Robustness to Noise:** In order to test the robustness of our method to noise, which possibly appears in testing data, we design an additional experiment on the Extended Yale B dataset with 30 training samples per class. For each testing sample, we randomly corrupt some pixels. The value of a corrupted pixel is replaced by a random value that ranges uniformly from 0 to 1. We
implement all methods on this noisy testing data and record the best result of each method. Fig. 3 shows that both our method and LatLRR are robust to noise, but our method outperforms LatLRR.

Figure 3: Testing the robustness of all methods on the Extended Yale B dataset with 30 training samples. The classification accuracy (averaged over 20 runs) of all methods is drawn with different percentage corruption.

4.3. Parameter Settings

There are two parameters in our objective function, namely $\lambda$ and $\gamma$. To demonstrate the effects of these two parameters for experiments, different combinations of these values selected from a reasonable discrete set $\{1e^{-6}, 1e^{-5}, 1e^{-4}, 1e^{-3}, 1e^{-2}, 1e^{-1}, 1e^{-0}, 1e^0, 1e^1, 1e^2, 1e^3, 1e^4\}$ are evaluated on each database. Specifically, the classification accuracy of each combination of parameter values is shown in Fig. 4, in which both the first and second columns represent the clustering performance of subspace clustering, and the third column represents the classification performance of feature extraction. In terms of subspace clustering experiments, there are different optimal parameters with different number of clusters $k$ on the first group of data. Here, we give the clustering performance of each database with the largest number of clusters, that is, the PIE database with $k = 68$, the Extended Yale B database with $k = 38$, the COIL20 database with $k = 20$, and the USPS database with $k = 10$. In the case of the largest number of clusters, the AC performance is given in the first column of Fig. 4, and the NMI performance is given in the second column of Fig. 4. Comparing the first column with the second column, we can see that the AC performance and NMI performance have almost the same optimal parameters. Moreover, we give the classification performance of each database in the third column of Fig. 4. From the third column of Fig. 4, we can see that the classification performance is roughly consistent over a wide parameter range, which overlaps with the optimal parameter for the clustering performance. Therefore, we use the same parameter for subspace clustering and feature extraction. For example, for the PIE database, when $k = 68$, the parameters used are $\lambda = 0.032$ and $\gamma = 10^{-5}$. For the Extended Yale B database, when $k = 38$, the parameters used are $\lambda = 0.0007$ and $\gamma = 0.00003$. On the COIL20 database,
Figure 4: The clustering and classification performances of our method versus the parameters $\lambda$ and $\gamma$ on four databases.
when $k = 20$, the parameters used are $\lambda = 10^{-6}$ and $\gamma = 10^{-4}$. For the USPS database, when $k = 10$, the parameters used are $\lambda = 10^3$ and $\gamma = 10^{-6}$.

5. Conclusion

In this paper, from the view of learning a dictionary, we integrate the traditional subspace learning method into the low-rank representation and produce a dual pursuit for clustering and classification simultaneously. The proposed method provides us a robust unsupervised subspace clustering algorithm as well as a robust unsupervised feature extraction algorithm simultaneously. As an unsupervised feature extraction algorithm, our method shows the robustness to the noise and produces the compared classification results with the previous methods. As an unsupervised subspace clustering algorithm, our method shows a better clustering results than previous methods.

[29] D. Cai, X. He, J. Han, Spectral regression: A unified approach for sparse subspace learning, in: International Conference on Data Mining, 2007, pp. 73–82.