Deep Latent Low-Rank Fusion Network for Progressive Subspace Discovery

Zhao Zhang\textsuperscript{1,2}, Jiahuan Ren\textsuperscript{2}, Zheng Zhang\textsuperscript{3} and Guangcan Liu\textsuperscript{4}

\textsuperscript{1} Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology, China
\textsuperscript{2} School of Computer Science and Technology, Soochow University, China
\textsuperscript{3} Bio-Computing Research Center, Harbin Institute of Technology, Shenzhen, China
\textsuperscript{4} School of Information and Control, Nanjing University of Information Science and Technology, China
cszzhang@gmail.com, hnzry10086@outlook.com, darren.zhang@uq.edu.au, geliu@uquit.edu.cn

Abstract

Low-rank representation is powerful for recovering and clustering the subspace structures, but it cannot obtain deep hierarchical information due to the single-layer mode. In this paper, we present a new and effective strategy to extend the single-layer latent low-rank models into multiple-layers, and propose a new and progressive Deep Latent Low-Rank Fusion Network (DLRF-Net) to uncover deep features and structures embedded in input data. The basic idea of DLRF-Net is to refine features progressively from the previous layers by fusing the subspaces in each layer, which can potentially obtain accurate features and subspaces for representation. To learn deep information, DLRF-Net inputs shallow features of the last layers into subsequent layers. Then, it recovers the deeper features and hierarchical information by congregating the projective subspaces and clustering subspaces respectively in each layer. Thus, one can learn hierarchical subspaces, remove noise and discover the underlying clean subspaces. Note that most existing latent low-rank coding models can be extended to multilayers using DLRF-Net. Extensive results show that our network can deliver enhanced performance over other related frameworks.

1 Introduction

Representation learning is always a fundamental problem to obtain the underlying explanatory factors and features for the subsequent data classification or clustering tasks [Chen et al., 2019 and 2020] [Zhang et al., 2016 and 2019]. Representation learning is still challenging in reality due to the complexity and diversity of data [Lu et al., 2019] [Su et al., 2019] [Acharya et al., 2019] [Ding et al., 2018].

Since most real data can be characterized using low-rank subspaces, low-rank coding methods can recover the underlying subspaces and obtain notable features [Liu et al., 2019] [Ren et al., 2019]. \textit{Low-Rank Representation (LRR)} [Liu et al., 2013] is one of the most classical algorithms to discover multi-subspaces, but it is essentially a transductive method failing to handle new data efficiently. To address the out-of-sample issue, \textit{Inductive Robust Principal Component Analysis (IRPCA)} [Bao et al., 2012] was recently proposed seeking a low-rank projection to map samples into underlying subspaces. To enable a solution for subspace segmentation and feature extraction, \textit{Latent LRR} (LatLRR) [Liu et al., 2011], was proposed, which decomposed data into a principal feature part, a salient feature part and a sparse error. Although LatLRR resolves the insufficient sampling issue and obtains enhanced performance over LRR, it still suffers from a high computational cost due to using Nuclear-norm to approximate the rank function to constrain the subspaces, while the computation of the Nuclear-norm needs the time-consuming Singular Value Decomposition of matrices at each iteration, especially for large-scale datasets. To improve the efficiency, a \textit{Frobenius-norm based LatLRR} (FLLRR) [Yu et al., 2018] was proposed, which approximates the rank function using Frobenius-norm. But the Frobenius-norm is sensitive to noise and outliers, which may produce inaccurate representations.

It is noteworthy that the above algorithms have a common drawback, i.e., they are “shallow” models using single-layer structures. As a result, they cannot obtain deep information and subspaces. But due to the strong representation ability of the deep neural networks [Kim et al., 2019], deep low-rank coding models equipped with carefully designed hierarchical structures should be able to obtain the enhanced performance. In fact, researchers have also designed some deep low-rank coding models, such as \textit{Weakly-supervised Deep Nonnegative Low-rank Model (WDNL)} [Li et al., 2017a], which finds the intrinsic relations between images and tags by removing noise or irrelevant tags, but it is unclear how to handle images directly and the results are usually incomplete. Another deep model is \textit{Deep Low-Rank Subspace Ensemble} (DLRSE) [Xue et al., 2019], where the Frobenius-norm is used as a low-rank constraint. DLRSE uses the deep matrix factorization to learn the diverse hierarchical structures and obtains the low-rank representations from extracted factors. But DLRSE is originally proposed for multi-view clustering, which clearly differs from our task. To cluster big data effectively, a \textit{Progressive Latent Subspace Clustering via Learning Deep Encoder (PLrSC)} [Li et al., 2017] has been recently proposed.
### 2.2 Projective Low-rank Subspace Clustering via Learning Deep Encoder (PllrSC)

Assuming that \( Y = [Y_1, \ldots, Y_k] \in \mathbb{R}^{n \times N} \) is a big dataset and over-sufficiently drawn from a union of \( k \) subspaces, where \( N \) is the number of samples in all subspaces. PllrSC assumes that \( X = [X_1, \ldots, X_k] \in \mathbb{R}^{n \times N} \) is a small dataset sampled randomly from \( Y \) and \( X \) is still sufficient. First, PllrSC learns a non-iterative deep encoder \( f_{a_k}(X; \theta) \), where \( \theta \) is the learning parameter to approximate low-rank representations. Then, the deep encoder is utilized to obtain the low-rank codes for replacing the costly non-linear Singular Value Thresholding (SVT) [Cai et al., 2010] operations. Thus, the predictive low-rank decomposition of PllrSC can be written as follows:

\[
\min_{Z} \| Z \|_F^2 + \gamma \| E \|_F^2 + \lambda \| Z - f_{a_k}(X; \theta) \|_F^2, \quad s.t. \ X = XZ + LX + E
\]

where \( E \in \mathbb{R}^{n \times N} \) denotes a sparse error, \( \lambda \) is the regularization parameter for \( E \), \( \gamma \) is a control parameter for approximation term \( \| Z - f_{a_k}(X; \theta) \|_F^2 \), \( \theta \) is a learning parameter set, where \( \theta = \{\theta_1, \ldots, \theta_m\} \in \mathbb{R}^{l_1 \times d_1, \ldots, l_m \times d_m} \), \( l_i \) denotes the number of the units in the \( i \)-th layer (\( l = d \) and \( l_m = N \)). Then, PllrSC employs the alternating direction algorithm (ADM) [Liu et al., 2013] and a gradient descent algorithm (GD) [Li et al., 2015] to optimize the above problem. Finally, PllrSC applies the landmark-based spectral clustering (LSC) algorithm to cluster the big dataset \( Y \) [Li et al., 2017].
Deep Coefficients are the deep coefficient matrices, and \( \lambda \)s \( t XZ Z XZ Z Z L XZ Z Z E \) are fed into the \( l \)-th layer, which are further decomposed into a deep principal feature part, a deep salient feature part and a sparse error. The whole framework of our DLRF-Net is shown in Figure 2. Assuming that DLRF-Net has \( M \) layers, the decomposition process of our DLRF-Net framework in the \( l \)-th layer can be presented as follows:

\[
XZ_0Z_1 \cdots Z_{l-1}Z_l = XZ_0Z_1 \cdots Z_{l-1}Z_l + L_1XZ_0Z_1 \cdots Z_{l-1} + E_l^1, \tag{5}
\]

where \( Z_l^1 \) and \( Z_l^2 \) are the deep coefficient matrices, \( L_l \) and \( L_l^1 \) denote the deep projection matrices that are learned from \( XZ_0Z_1 \cdots Z_{l-1} \) and \( L_0, L_1, \ldots, L_{l-1} \) in the \( l \)-th layer, respectively. Note that \( Z_0 \) and \( L_0 \) are included to simplify the descriptions, which are set to the identity matrices, i.e., the input of the first layer is the original data. It should be noted that for the optimization in the \( l \)-th layer, \( Z_0, Z_1, \ldots, Z_{l-1}, L_0, L_1, \ldots, L_{l-1} \) are known variables that are updated in the last layer. As such, intuitively from the multilayer learning process, deep principal features \( XZ_0Z_1 \cdots Z_{l-1} \) and deep salient features \( L_0, L_1, \ldots, L_{l-1} \) are learnt progressively from different layers, i.e., extracting fine-grained features from layer to layer.

Finally, deep principal features \( XZ_0Z_1 \cdots Z_l \) and salient features \( L_0, L_1, \ldots, L_l \) in the \( l \)-th layer can be obtained as

\[
XZ_0Z_1 \cdots Z_l = (XZ_0Z_1 \cdots Z_l^1 + Z_l^1)/2, \tag{6}
\]

\[
L_lX = (L_l + L_l^1)L_lX/2.
\]

The above subspace fusion operation can potentially make the learned representations more accurate by fusing feature information from deep principal and salient features in previous layers. The above averaging operation can also prevent the feature information loss and balance the information from deep principal and salient features in each layer. As such, we have the following model for DLRF-Net in the \( l \)-th layer:

\[
\min_{Z_l, L_l, E_l} \frac{1}{2} \left( \|Z_l^1\|_p + \|Z_l^2\|_p + \|L_l^1\|_p + \|L_l^2\|_p \right) + \lambda_l \left( \|E_l^1\|_p + \|E_l^2\|_p \right) \tag{7}
\]

\[
s.t. XZ_0 \cdots Z_{l-1} = L_{l-1}L_{l-2} \cdots L_0X + E_l^1,
\]

\[
L_{l-1} \cdots L_0X = (L_{l-1} \cdots L_0X)Z_l^1 + L_l^1(L_{l-1} \cdots L_0X) + E_l^2,
\]

where \( \| \cdot \|_p \) is the matrix \( p \)-norm, which can be Nuclear-norm or squared Frobenius-norm. \( \lambda_l \) is a positive tunable parameter that replies on the noise level of data [Liu et al., 2011]. We name the Nuclear-norm based DLRF-Net as nDLRF-Net and name the squared Frobenius-norm based DLRF-Net as fDLRF-Net. The objective function of fDLRF-Net can then be defined as follows for deep subspace discovery:

\[
\min_{Z_l, L_l, E_l} \sum_{l=1}^{M} \left( \frac{1}{2} \left( \|Z_l^1\|_p^p + \|Z_l^2\|_p^p + \|L_l^1\|_p^p + \|L_l^2\|_p^p \right) + \lambda_l \left( \|E_l^1\|_p + \|E_l^2\|_p \right) \right)
\]

\[
s.t. XZ_0 \cdots Z_{l-1} = L_{l-1}L_{l-2} \cdots L_0X + E_l^1,
\]

\[
L_{l-1} \cdots L_0X = (L_{l-1} \cdots L_0X)Z_l^1 + L_l^1(L_{l-1} \cdots L_0X) + E_l^2.
\]

Next, we show the optimization procedures of DLRF-Net.

### 4 Optimization

We mainly describe the optimization of fDLRF-Net in detail, as the optimization of nDLRF-Net is similar. Since \( Z_l, P_l \) and \( E_l \) depend on each other, we update them alternately. We use the inexact Augmented Lagrange Multiplier (Inexact ALM) algorithm [Lin et al., 2009] for efficiency.
Algorithm 1 Solving Eq.(9) by Inexact ALM (l-th layer)
Inputs: Reconstructed data $X_{1,\ldots,m}$, tunable parameters $\lambda, \alpha$.
Initialization: $t = 0$, $Z_0^{(1)} = (Z_0^{(1)})^\top = (0)$, $L_0^{(i)} = (0), (E_0^{(i)})^T = (0)$, $\mu^{+,\alpha} = 0, \mu^* = 10^{-4}, \varrho = 1.0$. 
While not converged do
1. Update the coefficients sub-matrices $(Z_1)^{(i)}$ and $(Z_2)^{(i)}$ by using Eq.(12-16), and obtain $Z_1^{(i)} = (Z_1^{(i)})^T + (Z_1^{(i)})^T/2$;
2. Update the projection sub-matrices $(E_1)^{(i)}$ and $(E_2)^{(i)}$ by using Eq.(14-15), and obtain $E_1^{(i)} = (E_1^{(i)})^T + (E_1^{(i)})^T/2$;
3. Update the sparse errors $(E_1^{(i)})^T$ and $(E_2^{(i)})^T$ by Eq.(16-17);
4. Update the Lagrange multipliers $(\nu^{(i)})^T$ and $(\nu^{(i)})^T$;
5. Update the parameter $\mu$ by $\mu^{(i)} = \min(\mu^{+,\alpha}, \mu^*)$;
6. Check for convergence: Suppose $\|P_{l-1} - P_{l-1}Z_1^{(i)} - L_1^{(i)}, P_{l-1} - E_1^{(i)}\|,\|S_{l-1} - S_{l-1}Z_1^{(i)} - L_1^{(i)}, S_{l-1} - E_1^{(i)}\| \leq \varepsilon$, stop; else $t = t + 1$.
End while
Output: $Z_1^{(i)}, P_{l-1} = P_{l-1}^{(i)}$.

To simplify the descriptions of optimization, we train the model layer by layer. To learn features in the l-th layer ($l=1, 2, \ldots, M$), the target function can be defined as

$$
\min_{Z_1} \frac{1}{2} \left( \|Z_1\|^2 + \|Z_2\|^2 + \|E_1\|^2 + \|E_2\|^2 \right) + \lambda \left( \|E_1\|^2 + \|E_2\|^2 \right)
$$

s.t. $X_{Z_0^{(1)}} = (X_{Z_0^{(1)}}) \nu_1^{(i)} + L_0^{(i)} (X_{Z_0^{(1)}}) + E_1^{(i)}$, $L_0^{(i)} = L_0^{(i+1)} = X_{Z_0^{(1)}} + E_2^{(i)}$.

Denote by $P_{l-1} = X_{Z_0^{(1)}}$ and $S_{l-1} = (P_{l-1})$ two auxiliary matrices, the Lagrange function of Eq.(9) can be obtained as

$$
\varphi(Z_1, L_1, E_1) = \frac{1}{2} \left( \|Z_1\|^2 + \|Z_2\|^2 + \|E_1\|^2 + \|E_2\|^2 \right) + \lambda \left( \|E_1\|^2 + \|E_2\|^2 \right) + \left( Y_1, P_{l-1} - P_{l-1}Z_1^{(i)} - L_1^{(i)}, P_{l-1} - E_1^{(i)} \right) + \left( Y_2, S_{l-1} - S_{l-1}Z_1^{(i)} - L_1^{(i)}, S_{l-1} - E_1^{(i)} \right) + \frac{\mu}{2} \left( \|P_{l-1} - P_{l-1}Z_1^{(i)} - L_1^{(i)}, P_{l-1} - E_1^{(i)}\|^2 + \|S_{l-1} - S_{l-1}Z_1^{(i)} - L_1^{(i)}, S_{l-1} - E_1^{(i)}\|^2 \right)
$$

where $Y_1$ and $Y_2$ are Lagrange multipliers, and $\mu$ denotes a positive parameter. Then, the LRF-Net updates the variables by solving $\varphi$. Note that the optimization procedures of our DLRF-Net in the l-th layer can then be detailed as follows:

Fix others, update $Z_1$: For the optimization of $Z_1$, we need to solve $Z_1$ and $Z_2$. By removing the irrelevant terms, we can update $Z_1$ and $Z_2$ by the following reduced problem:

$$
\varphi(Z_1, Z_2) = \frac{1}{2} \left( \|Z_1\|^2 + \|Z_2\|^2 \right) + \left( Y_1, \Xi_{l-1} - P_{l-1}Z_1^{(i)} \right) + \left( Y_2, \Omega_{l-1} - S_{l-1}Z_1^{(i)} \right) + \frac{\mu}{2} \left( \|\Xi_{l-1} - P_{l-1}Z_1^{(i)}\|^2 + \|\Omega_{l-1} - S_{l-1}Z_1^{(i)}\|^2 \right)
$$

where $\Xi_{l-1} = P_{l-1} - L_1^{(i)}, P_{l-1} - E_1^{(i)}$ and $\Omega_{l-1} = S_{l-1} - L_1^{(i)}, S_{l-1} - E_1^{(i)}$. We first show the optimization of $Z_1$. By taking the derivative of $\varphi(Z_1, Z_2)$ w.r.t. $Z_1$ and zeroing the derivative, we can infer the coefficients matrix $Z_1$ at the $(t+1)$-th iteration as follows:

$$
(Z_1)^{(i+1)} = (I + \mu^* P_{l-1}^T P_{l-1}^{-1})^{-1} \mu^* P_{l-1}^T \left( \Xi_{l-1} + (Y_1)^T / \mu \right),
$$

where $\Xi_{l-1} = P_{l-1} - L_1^{(i)} P_{l-1} - (E_1^{(i)})$. Similar to the optimization of $Z_1$, we can infer $(Z_2)^{(i+1)}$ in the $(t+1)$-th iteration as

$$
(Z_2)^{(i+1)} = (I + \mu^* P_{l-1}^T P_{l-1}^{-1})^{-1} \mu^* P_{l-1}^T \left( \Omega_{l-1} + (Y_2)^T / \mu \right),
$$

where $\Omega_{l-1} = S_{l-1} - L_1^{(i)} S_{l-1} - (E_1^{(i)})$. After optimizing the $(Z_1)^{(i+1)}$ and $(Z_2)^{(i+1)}$, we can obtain $(Z_1)^{(i+1)} = \left( (Z_1)^{(i+1)} + (Z_2)^{(i+1)} \right) / 2$.

Fix others, update $L_1$: By removing the irrelevant terms from $\varphi$, taking the derivatives of $\varphi(Z_1, Z_2)$ w.r.t. $P_{l-1}$ and $E_2$, and zeroing the derivatives, we can similarly obtain

$$
(L_1)^{(i+1)} = \mu^* \left( \Xi_{l-1} + (Y_1)^T / \mu \right) + (I + \mu^* P_{l-1}^T P_{l-1}^{-1})^{-1} \mu^* P_{l-1}^T (E_2)^T,
$$

$$
(L_2)^{(i+1)} = \mu^* \left( \Omega_{l-1} + (Y_2)^T / \mu \right) + (I + \mu^* P_{l-1}^T P_{l-1}^{-1})^{-1} \mu^* P_{l-1}^T (E_2)^T.
$$

which can be easily solved by the shrinkage operator [Lin et al., 2009], where $\Delta_1$ and $\Delta_2$ are auxiliary matrices defined as $\Delta_1 = P_{l-1} (Z_1)^{(i+1)} - (L_1)^{(i+1)}, (Y_1)^T / \mu$ and $\Delta_2 = S_{l-1} (Z_2)^{(i+1)} - (L_2)^{(i+1)}, (Y_2)^T / \mu$. For complete presentation of our model, we summarized the optimization procedures of solving the sub-problem of Eq.(9) in the $l$-th layer in Algorithm 1.

5 Discussion

5.1 Relationship Analysis

We mainly discuss the relations of our DLRF-Net to LatLRR and FLRR. To facilitate the analysis, we consider the special case that $l=1$. We first express this special case as

$$
\min_{Z_1} \frac{1}{2} \left( \|Z_1\|^2 + \|L_1\|^2 \right) + \lambda \|E_1\|^2
$$

s.t. $X_{Z_0^{(1)}} = X_{Z_0^{(1)}} + E_1, L_0X = L_0Z_1 + L_0E_1 + E_1$.

Since $Z_0$ and $L_0$ are initialized to the identity matrices in the optimization, the two constraints are the same. As such, it is clear that when we use the Frobenius-norm to constrain the matrices $Z_1$ and $L_1$, the problem identifies FLRR; while we use the Nuclear-norm as constraints, the resulting problem is identical to LatLRR. That is, both FLRR and LatLRR are the special cases of our DLRF-Net framework.

5.2 Computational Time Complexity

We analyze the complexity of each layer of our deep models. For fDLRF-Net, SVD is not used and the major computation is updating the matrices $Z_1$ and $L_1$. Thus, the time complexity of Algorithm 1 is equal to that of FLRR. Thus, it is easy to
infer that the total time complexity of fDLRF-Net is $M$ times that of each layer, where $M$ is the number of layers, which is usually a small value. For nDLRF-Net, the time complexity of each layer is the same as that of regular LatLRR and the total time complexity is $M$ times that of each layer.

6 Experimental Results and Analysis

We conduct experiments to evaluate the effectiveness of our fDLRF-Net and nDLRF-Net, and show the comparison results with other related methods, including FLLRR, LatLRR, LRR, Robust LatLRR (rLatLRR) [Zhang et al., 2014b], Laplacian Regularized LRR (rLRR) [Zhang et al., 2014a], Sa-LatLRR [Wang et al., 2018] and PLrSC [Li et al., 2017b]. Three real image databases are involved, including two face datasets (i.e., CMU PIE [Sim et al., 2003], UMIST [Graham et al., 1998]) and the Fashion MNIST database [Xiao et al., 2017]. The details of used databases are described in Table 1. We follow the common procedure to resize each face image into 32×32 pixels and images of the Fashion MNIST dataset are resized into 28×28 pixels. We perform all experiments on a PC with Intel (R) Core (TM) i7-7700 CPU @ 3.6 GHz 8G.

6.1 Visual Image Analysis by Visualization

Visualization of coefficient matrix $Z$. To represent data appropriately, $Z$ should have a block-diagonal structure. Each block denotes the coefficients for certain subject so that each sample can be reconstructed by the samples of one class as much as possible. We follow [Liu et al., 2011] to construct 10 independent subspaces $\{S_i\}_{i=1}^{10}$, and apply this artificial data for DLRF-Net. The visualization of $Z$ in the first four layers are illustrated in Figure 3. We see that all coefficient matrices $Z$ have block-diagonal structures. But compared with the $l$-th layer, the results of other layers have less noise and wrong inter-class connections. It can also be found that the subspace structures of $Z$ are improved progressively, i.e., the learned structures from the 3-rd and 4-th layers are better that from the 2-nd layer. But the difference of the structures of $Z$s in the 3-rd and 4-th layers is small, i.e., our DLRF-Net can remove noise contained in features and recovery the subspaces of $Z$ by using small number of layers. That is, the structures of the coefficient matrix $Z$ will not become better, even though we use more layers involving high computational cost. In all the simulations, the results of fDLRF-Net and nDLRF-Net in the $l$-th layer corresponds to FLLRR and LatLRR, respectively.

Visualization of the recovered features $XZ$. We evaluate fDLRF-Net and nDLRF-Net by visualizing recovered deep features $XZ$. Given a data matrix $X$, DLRF-Net decomposes it into principal features $XZ$, salient features $LX$ and a sparse error $E$ in each layer. CMU PIE face dataset is used. This face image database contains 68 persons with 41368 images under varying poses, illuminations and facial expressions. 170 near frontal images per person are employed for CMU PIE, which contains five near frontal poses (C05, C07, C09, C27, and C29). To evaluate the robustness properties, random Gaussian noise with variance 500 is included into the image data. Some original images, noisy images and recovered principal features in the first three layers are shown in Figure 4. We see find that fDLRF-Net and nDLRF-Net can effectively remove the shadow and noise in images in a progressive way, compared with the recovered results of FLLRR and LatLRR.

6.2 Quantitative Clustering Evaluations

We compare each model for clustering images. UMIST and Fashion MNIST are evaluated. UMIST has 1012 images from 20 different individuals. Fashion-MNIST has 10 classes and 70000 unique products. In this study on Fashion-MNIST, we choose 1000 samples per class, i.e., totally 10000 samples. To evaluate the performance, we follow the common procedures and use the coefficient matrix $Z^*$ of each method to construct the weights by $W = (|Z^*|+|Z^{**}|)/2$ and then use the Normalized Cuts (NCut) [Shi et al., 2000] for clustering. For PLrSC and DLRF-Net, we use the coefficients from the final layer. For each number K of clusters, we choose K categories randomly and the results are averaged over 30 initializations.

The clustering accuracy (AC) [Cai et al., 2017] is used as the quantitative metric. The values of AC on evaluated databases are shown in Table 2. We see that: (1) the clustering accuracy of each method goes down as the number of categories increases, since clustering more data is difficult than clustering less; (2) Our fDLRF-Net and nDLRF-Net deliver higher ACs than the other competitors, especially for FLLRR and LatLRR, implying that DLRF-Net can learn more effective representations by mining deep information.

<table>
<thead>
<tr>
<th>Dataset Name</th>
<th># Samples</th>
<th># Dim</th>
<th># Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMU PIE</td>
<td>11554</td>
<td>1024</td>
<td>68</td>
</tr>
<tr>
<td>UMIST face</td>
<td>1012</td>
<td>1024</td>
<td>20</td>
</tr>
<tr>
<td>Fashion MNIST</td>
<td>70000</td>
<td>784</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 1: Descriptions of used image datasets.

![Figure 3: Visualization of the representation coefficient matrices $Z$ of our fDLRF-Net (a-d) and nDLRF-Net (e-h).](image)

![Figure 4: Recovered principal features on CMU PIE face database (Left: fDLRF-Net; Right: nDLRF-Net).](image)
6.3 Noisy Image Clustering Against Corruptions

We investigate the robustness properties against noisy cases that images are corrupted. UMIST and Fashion MNIST are used. Random Gaussian noise with different variance (100, 200, ..., 500) is added to examine the robustness. For each setting, we average the result over 30 random initialization for NCut. We set the number K of clusters as 5. Some noisy images and clustering accuracy are shown in Figure 5, where the clustering results of five layers of our nDLRF-Net and nDLRF-Net are described. We see clearly that: (1) generally speaking the clustering accuracy of each method goes down with the increasing level of noise, as clustering data of high noise level is more difficult than clustering that of low noise level; (2) the best records are usually obtained in the 2-nd layer and the 3-rd layer, by comparing with the other cases.

6.4 Investigation of Parameters

UMIST database is used. DLRF-Net has one parameter λ, so we can select the most important one by a linear search from \(10^4, 10^5, ..., 10^{10}\). We set the number K of clusters to 5 and show the analysis results of the first three layers in Figure 6. For each layer, we select the best parameter, and then we fix it to learn deeper features of next layer. We find that λ usually becomes a larger one with increasing number of layers, which is easy to understand. Since DLRF-Net recovers the subspaces progressively and learnt subspaces become clean layer by layer. Note that similar observations can be found from the other datasets, but the results will not be presented.

7 Conclusion

We proposed a new progressive deep latent low-rank fusion network to uncover deep hidden features and deep clustering structures. DLRF-Net discovers the subspaces by refining the principal and salient features from previous layers progressively and then fusing the subspaces. Specifically, DLRF-Net recovers hierarchical features by congregating the projective subspace and subspaces in each layer. Most existing latent low-rank coding models can also be easily extended to multilayer scenario for learning deep features. In future, more effective deep low-rank fusion strategies will be explored.

Acknowledgments

This work is supported by the NSFC (61672365, 61806035 and U1936217) and the Fundamental Research Funds for the Central Universities of China (JZ2019HGPA0102), and New Generation AI Major Project of Ministry of Science and Technology of China (2018AAA0100601). Zhao Zhang and Zheng Zhang are the corresponding authors of this paper.
References


[Cai et al., 2017] Deng Cai, Xiaofei He, Jiawei Han. Document clustering using locality preserving indexing. IEEE TKDE, 17(12):1624-1637, 2005.


