Learning Structured Twin-Incoherent Twin-Projective Latent Dictionary Pairs for Classification

Zhao Zhang1,2,3, Yulin Sun1, Zheng Zhang4, Yang Wang2,3, Guangcan Liu5 and Meng Wang2,3
1 School of Computer Science and Technology, Soochow University, Suzhou 215006, China
2 Key Laboratory of Knowledge Engineering with Big Data (Ministry of Education), Hefei University of Technology
3 School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China
4 Bio-Computing Research Center, Harbin Institute of Technology (Shenzhen), Shenzhen, China
5 School of Information and Control, Nanjing University of Information Science and Technology, Nanjing, China
e-mail: cszhangi@gmail.com, daitusun@gmail.com

Abstract—In this paper, we extend the popular dictionary pair learning (DPL) into the scenario of twin-projective latent flexible DPL under a structured twin-incoherence. Technically, a novel framework called Twin-Projective Latent Flexible DPL (TP-DPL) is proposed, which minimizes the twin-incoherence constrained flexibly-relaxed reconstruction error to avoid the possible over-fitting issue and produce accurate reconstruction. In this setting, TP-DPL integrates the twin-incoherence based latent flexible DPL and the joint embedding of codes as well as salient features by twin-projection into a unified model in an adaptive neighborhood-preserving manner. Therefore, TP-DPL can unify the procedures of salient feature representation and classification. The twin-incoherence constraint on coefficients and features can explicitly ensure high intra-class compactness and inter-class separation over them. TP-DPL also integrates the adaptive weighting to preserve local neighborhood of both coefficients and salient features within each class explicitly. For efficiency, TP-DPL selects the Frobenius-norm and abandons the costly l0-norm for group sparse representation. Another byproduct is that TP-DPL can directly apply the class-specific twin-projective reconstruction residual to compute the label of data. Extensive results on public databases show that TP-DPL can deliver the state-of-the-art performance.

Index Terms—Structured twin-incoherence, twin-projective latent dictionary pair learning, structured adaptive weighting, discriminative classification

I. INTRODUCTION

Compact representation learning is an important topic in the communities of data mining and pattern classification for a wide variety of complex data. For the high-dimensional data understanding, lots of compact learning methods have been developed, e.g., subspace learning, matrix factorization and sparse representation. Due to the great success to a variety of real emerging applications, e.g., denoising [14], clustering [8], visual saliency and image classification [1-13][36-40], sparse representation through Dictionary Learning (DL) has received much attention in recent years. DL aims to compute the compact representation of data by a linear combination of a few highly correlated atoms in a dictionary [15][41-45]. Thus, the discriminating ability of the dictionary atoms will determine the accuracy of the linear reconstruction over the atoms. It is also worth noticing that the dictionary size, i.e., number of atoms, also has direct effect on the complexity of the compact representation of data. Thus, learning a good dictionary with the strong distinguishing power is crucial for the data representation and classification [1-12][41-45].

One most popular compact DL method is the K-Singular Value Decomposition (KSVD) [3]. By the alternates between the sparse coding over dictionary and a process of updating atoms to better fit data, KSVD can represent data effectively, but it does not consider the abilities for discrimination and classification. To use label information for data classification, many discriminative methods were proposed, which can be roughly divided into two kinds, i.e., learning overall shared dictionary of all classes and specific sub-dictionaries over each subject class. The overall shared DL usually applies the discriminative regularization to force the coefficients to be discriminant, among which Discriminative KSVD (D-KSVD) [6] and Label Consistent KSVD (LC-KSVD) [1] are classical models. D-KSVD adds the classification error into KSVD to enhance classification, while LC-KSVD further adds a label consistency to make the coefficients discriminative.

The category-specific sub-dictionary learning encourages each sub-dictionary to correspond to one single class and the sub-dictionaries of different classes to be as independent as possible. Several representative methods are Dictionary Learning with Structured Incoherence (DLSI) [8], Fisher Discrimination Dictionary Learning (FDDL) [7], Projective Dictionary Pair Learning (DPL) [9], Structured Analysis Discriminative Dictionary Learning (ADDL) [11], Latent Label Consistent Dictionary Learning (LLC-DL) [10] and Low-rank Shared Dictionary Learning (LRSPL) [16]. For representation learning, DPL is clearly different from DLSI, FDDL and LRSPL, since it extends the regular dictionary learning to dictionary pair learning. The analysis dictionary analytically codes data, and the synthesis dictionary is used to reconstruct given data [9]. Based on DPL, ADDL further includes the analysis incoherence promoting function and extends the dictionary pair learning into the joint analysis multi-class classifier training over extracted coefficients. By the analysis DL, DPL and ADDL are efficient for encoding both inside and outside data, but they both cannot encode the salient features of samples jointly to make the reconstruction more accurate for representations. To solve this issue, recent LLC-DL aims at decomposing given data into a latent sparse reconstruction over a structured latent weighted dictionary, a salient feature part and an error part fitting noise. That is, LLC-DL integrates the compact representation and salient feature extraction into a unified model. Although LLC-DL can extract salient features of samples jointly, it suffers from an obvious drawback that it cannot change the dictionary size flexibly. LLC-DL also cannot ensure the high inter-class separation and intra-class compactness of the learnt salient features over different classes, because there is no explicit constraint on the projection for feature extraction. Thus, it would be better to define an explicit incoherence constraint on the sub-projection over each class l so that it computes a salient feature subspace where the training points from class
Structured Twin-Incoherence based Twin-Projective Latent DPL

Fig. 1: The schematic flow-diagram of our proposed RA-DPL framework for image representation and recognition.

We will briefly introduce the closely related DL approaches.

A. Review of Overall Dictionary Learning

Let $X = [x_1, \ldots, x_c] \in \mathbb{R}^{N \times c}$ be a set of training samples from $c$ classes, where $x_i$ is a sample vector, $N$ is the dimension of the original data and $c$ is the number of samples. Then, DL can compute a dictionary $D = [d_1, \ldots, d_k] \in \mathbb{R}^{N \times K}$ of $K$ atoms and the coding coefficients matrix $S = [s_1, \ldots, s_c] \in \mathbb{R}^{K \times c}$ of $X$ from the following general formulation:

$$\hat{S} = \arg\min_D \|X - DS\|_F^2 + \lambda \|S\|_1,$$  

where $\|X - DS\|_F^2$ is the reconstruction error over the data $X$, $K$ is the total number of atoms over all classes, and $\lambda > 0$ is a scalar constant. $\|S\|_1$ is a $l_1$-norm regularization, where $\lambda = 0/1$ corresponds to $l_0/l_1$-norm, which is widely-used to ensure the sparse properties of $S$, but such an operation usually incurs a heavy computation burden in reality.

B. Structured Dictionary Learning by DPL

To avoid the computation burden caused by the costly $l_0$ or $l_1$-norm, DPL was recently derived to calculate a synthesis dictionary $D$ and an analysis dictionary $L_i$ jointly by solving the following dictionary pair learning problem:

$$\hat{L}_i, \hat{D} = \arg\min_{L_i, D} \sum_{i=1}^{c} \|x_i - L_i d_i\|_2^2 + \lambda \|L_i\|_1, \text{ s.t. } \|k\|_1 \leq 1,$$

where $X_c$ is the complementary data matrix of $X_i \in \mathbb{R}^{N \times c}$ in $X$, i.e., excluding $X_i$ itself from $X$. $D = [d_1, \ldots, d_k] \in \mathbb{R}^{N \times K}$ is the synthesis dictionary of class $i$, $L_i \in \mathbb{R}^{c \times K}$ is the analysis sub-dictionary of the class $i$, $K$ is the number of dictionary atoms according to class $i$, $K$ is the total number of atoms, $L_i = [L_{i1}, \ldots, L_{ic}] \in \mathbb{R}^{c \times K}$ is the synthesis dictionary of class $i$, $L_i \in \mathbb{R}^{c \times K}$ is the analysis sub-dictionary of the class $i$, $K$ is the number of dictionary atoms according to class $i$, $K$ is the total number of atoms, $L_i = [L_{i1}, \ldots, L_{ic}] \in \mathbb{R}^{c \times K}$. $D = [D_1, \ldots, D_k] \in \mathbb{R}^{N \times K}$ and $N$ is the number of samples in class $i$. The constraint $\|k\|_1 \leq 1$ can avoid the trivial solution of $L_i$ and make the computation stable. Note that DPL applies Frobenius-norm rather than costly $l_0/l_1$-norm to impose the group sparsity on coefficients.
matrix \( LX \) (i.e., \( LX \) is nearly block-diagonal).

In the test phase, if the query sample \( y \) is from the class \( k \), its projective coding vector by \( L_k \) will be more likely to be significant, while its projective coding vectors by \( L_i, \ i \neq k \), tend to be small. Consequently, the reconstruction residual \( \|y - DL_ky\|_2 \) will be much smaller than \( \|y - DL_iy\|_2, \ i \neq k \). Thus, the class-specific reconstruction residual can be used to identify the label of \( y \) by the following criterion:

identity(\( y \)) = \arg \min_{l} \|y - DL_ly\|_2. \quad (3)

That is, the new sample \( y \) will be assigned to class \( l \) that minimizes the reconstruction error \( \|y - DL_ly\|_2 \).

III. STRUCTURED TWIN-INCOHERENCE BASED TWIN-PROJECTIVE LATENT ADAPTIVE DPL (TP-DPL)

A. The Objective Function

To make the representation more accurately and enable the model to extract salient features form given samples jointly, TP-DPL discusses the twin-projective latent flexible DPL problem under the structured twin-incoherence. Specifically, TP-DPL encodes the projective flexible-relaxed latent reconstruction error \( \|X_i + a \cdot e^T - DL_iX_i - P_iX_i\|_2^2 \), which avoids the potential over-fitting issue and makes the representation more accurate, where \( a \in \mathbb{R}^{m \times n} \) is a bias, \( e \in \mathbb{R}^{1 \times n} \) is a column vector of all ones, \( L_iX_i \) encodes the coefficients of samples in class \( i \) and \( P_iX_i \) are salient features of \( X_i \). That is, our TP-DPL integrates the sparse representation and feature extraction into a unified framework similarly as [10]. The difference between TP-DPL and LLC-DPL will be discussed shortly. In addition, a structured twin-incoherence function \( r(L_k, P) \) on codes \( L_kX \) and features \( P_kX \) is used to ensure intra-class compactness and inter-class separation over \( L_kX \) and \( P_kX \) jointly. Specifically, if \( L_kL_kX \) is the projective analysis latent sub-dictionary and \( P_kP_kX \) is sub-projection corresponding to class \( k \), we hope that both \( L_k \) and \( P_k \) can project the representation and features of training samples of class \( \ell \) (\( \ell \neq k \)) to a nearly null space similarly as [9], i.e.,

\[ L_kX_\ell = 0, P_kX_\ell = 0, \forall \ell \neq k. \quad (4) \]

Thus, the salient features \( P_kX \) will be discriminant in terms of high intra-class compactness and inter-class separation, and the coefficients \( L_kX \) will be nearly block diagonal. Then, we can formulate the following structured twin-incoherence function to improve the discriminating power of TP-DPL:

\[ r(L_k, P) = \|L_kX_\ell + P_kX_\ell\|_2^2, \quad (5) \]

where \( X_\ell \) is the complementary data matrix of \( X_\ell \) over \( X \). To make the coding efficient, TP-DPL uses the Frobenius-norm and abandon the costly \( l_0/l_1 \)-norm for group sparsity [9-10].

In addition, we also investigate how to preserve the local neighborhood information of both \( L_kX \) and \( P_kX \) within each class \( k \) clearly in an adaptive manner. Specifically, a structured twin-projection based adaptive reconstruction weighting function \( g(L_k, P_k, W_k) \) is involved by computing an adaptive reconstruction weight matrix \( W_k \in \mathbb{R}^{m \times n} \) of each class \( k \), where \( X_k \) is the number of samples in class \( k \). Since we hope that the processes of extracting salient features \( P_kX \) and learning coefficients \( L_kX \) can keep neighborhood information of samples within each class, we minimize the neighborhood reconstruction errors \( \|P_kX - P_kXW_k\|_2^2 \) and \( \|L_kX - L_kXW_k\|_2^2 \) jointly. Thus, we can define the adaptive reconstruction weighting function as follows:

\[ g(L_k, P_k, W_k) = \|P_kX - P_kXW_k\|_2^2 + \|L_kX - L_kXW_k\|_2^2 + PP_k. \quad (6) \]

That is, the weight matrix \( W_k \) is shared in the projective feature space spanned by \( L_k \) and projective representation space spanned by \( L_k \). In addition, the adaptive weighting function \( g(L_k, P_k, W_k) \) clearly correlates the coefficients and salient features, i.e., certain important information can be shared in the feature and representation spaces.

By combining Eq.(4), Eq.(5), and Eq.(6), the final objective function of our TP-DPL can be reformulated as

\[ \min_{D, P, \alpha, e} \sum_{i,j} \left( \|X_i + a \cdot e^T - DL_iX_i - P_iX_i\|_2^2 + \alpha \left( \|L_iX_i\|_2^2 + \|P_iX_i\|_2^2 \right) \right) + \beta \left( \|P_iX_i - P_iX_iW_i\|_2^2 + \|L_iX_i - L_iX_iW_i\|_2^2 + PP_i \right) \]

\[ \quad \quad \quad \quad + \|e\|_2^2 + \|P_i\|_2^2 \quad (7) \]

where \( X_i, D, (L_k, P) \) and \( P_i \) are training data, sub-dictionary, twin-projection pair and adaptive weighting matrix over the class \( i \) respectively. In this paper, we omit the constraint \( \|P_i\|_2 \leq 1 \) and the main reason will be described later. \( \alpha \) and \( \beta \) are two positive weighting factors. Note that the joint minimization of \( \alpha (L_k, P) + \beta g(L_k, P_k, W_k) \) can produce the neighborhood preserving discriminative coding coefficients and salient features clearly to enhance the results.

Since the above objective function in Eq. (7) is generally non-convex, we introduce a variable matrix \( S_i \) (\( S_i = L_iX_i \)) to relax the problem as DPL. But note that we apply a flexible term \( \|L_iX_i + b \cdot e^T - S_i\|_2^2 \) to avoid the over-fitting and make the representation more accurate, where \( b \in \mathbb{R}^{m \times n} \) is also a bias, since directly minimizing \( \|L_iX_i - S_i\|_2^2 \) to approximate \( S_i \) using \( L_iX_i \) is too hard, which may cause over-fitting. While the term \( \|P_iX_i + b \cdot e^T - S_i\|_2^2 \) provides a flexible approximation error clearly. The relaxed optimization problem is

\[ \min_{D, P, \alpha, e} \sum_{i,j} \left( \|X_i + a \cdot e^T - DL_iX_i - P_iX_i\|_2^2 + \alpha \left( \|L_iX_i\|_2^2 + \|P_iX_i\|_2^2 \right) \right) + \beta \left( \|P_iX_i - P_iX_iW_i\|_2^2 + \|L_iX_i - L_iX_iW_i\|_2^2 + PP_i \right) \]

\[ \quad \quad \quad \quad + \|e\|_2^2 + \|P_i\|_2^2 \]   \quad (8)

Note that the schematic flow-diagram of our framework for image recognition is illustrated in Fig. 1, where we show the principles of training and test phases. Next, we detail the optimization procedures of our TP-DPL algorithm.

B. Optimization

Since the optimization of involved variables depend on each other, the problem cannot be solved directly. Following the common way, we solve the problem by an alternate strategy. Let \( \varphi \) be the objective function of our TP-DPL, by taking the derivatives of \( \varphi \) with respect to \( a_i \) and \( b_i \) and setting the derivatives to zeros, we can obtain

\[ \frac{\partial \varphi}{\partial a_i} = a_i e^T + X_i e - D_i S_i - P_i X_i e = 0 \]

\[ \Rightarrow a_i = (D_i S_i + P_i X_i e - e^T X_i e) / N \quad (9) \]

\[ \frac{\partial \varphi}{\partial b_i} = b_i e^T + L_i X_i e - S_i e = 0 \Rightarrow b_i = (S_i e - L_i X_i e) / N \quad (10) \]

By the above equations, we can rewrite the flexible errors \( x_j + a_j e^T - D_j S_j - P_j X_j \) and \( L_j X + b_j e^T - S_j \) as follows:

\[ L_j X + b_j e^T - S_j = L_j X + \left( S_j e^T - L_j X_i e \right) / N - S_j \]

\[ = \left( L_j X - L_j X e^T \right) / (N - S_j) \]

\[ = L_j X_i H_j - S_j H_j \]   \quad (11)
where $H_f = I - ee^T / N$ is “centering matrix”. By substituting the above equations into Eq.(8), we obtain the following equivalent optimization problem for TP-DPL:

$$
\min_{D \in \mathbb{R}^{K \times n}, P \in \mathbb{R}^{n \times n}, S \in \mathbb{R}^{n \times n}} \sum_{i=1}^{n} \|X_i - D_i S_i - P_i X_i H_i \|_F^2 + 
g \|X_i H_i - S_i H_i \|_F^2 + \beta \|P_i X_i - P_i X_i W_i \|_F^2 + \gamma \|X_i H_i - S_i H_i \|_F^2.
$$

(13)

Then, the above minimization problem can be alternated among the following steps:

1. **Fix $P$, $L$, $W$, update $D$ and $S$:** By removing the terms irrelevant to $S$ and $D$ from the problem in Eq.(13), we have

$$
\langle D, S \rangle = \arg \min_{D, S} \sum_{i=1}^{n} \|X_i H_i - D_i S_i - P_i X_i H_i \|_2^2 + \gamma \|X_i H_i - S_i H_i \|_2^2.
$$

(14)

Note that the above centered matrix by $H_f$ corresponds to the normalized matrix of original one, which plays a similar role as that of constraint $\|D\|_F \leq 1$ in DPL, which can also make the computation of our formulation stable. By taking the derivative w.r.t. $S$ and setting it to zero, we can easily update the coding coefficients $S_i$ of class $i$ as follows:

$$
S_i = (D_i^T D_i + \gamma I)^{-1} A_i (H_i H_i^T)^{-1},
$$

(15)

By taking the derivative w.r.t. $D$ and setting it to zero, we can update the sub-dictionary $D_i$ of class $i$ as

$$
D_i = (X_i H_i H_i^T S_i^T - P_i X_i H_i^T S_i^T) (S_i H_i H_i^T S_i^T)^{-1}.
$$

(16)

2. **Fix $D$, $S$, $W$, update $P$:** By removing terms irrelevant to $P$, we can have the following reduced problem:

$$
P = \arg \min_{P} \sum_{i=1}^{n} \|X_i - D_i S_i - P_i X_i H_i \|_F^2 + \alpha \|P_i X_i - P_i X_i W_i \|_F^2.
$$

(17)

By taking the derivative w.r.t. $P$ and setting it to zero, we can update the projection matrix $P$ as

$$
P_i = (X_i H_i H_i^T X_i^T - D_i S_i H_i H_i^T X_i^T) (B_i + C_i)^{-1},
$$

(18)

3. **Fix $S$ and $W$, update $L$:** By removing terms irrelevant to $L$ from Eq.(13), we have the following reduced problem:

$$
L = \arg \min_{L} \sum_{i=1}^{n} \|X_i H_i - S_i H_i \|_F^2 + \beta \|X_i L_i - L_i X_i W_i \|_F^2
$$

(19)

By taking the derivative w.r.t. $L$ and setting it to zero, we can update the analysis dictionary $L$ as

$$
L_i = (\gamma S_i H_i H_i^T X_i^T) (F_i + \beta X_i W_i W_i^T X_i^T - \beta X_i W_i X_i^T)^{-1},
$$

(20)

4. **Fix $P$ and $L$, optimize $W$:** By removing terms irrelevant to $W$ from Eq.(13), we have the following reduced problem:

$$
W = \arg \min_{i=1}^{n} \beta \|X_i - L_i X_i W_i \|_F^2 + \alpha \|X_i - L_i X_i W_i \|_F^2.
$$

(21)

By taking the derivative w.r.t. $W$ and setting it to zero, we can update the adaptive weighting matrix $W$ as

$$
W_i = (X_i^T P_i^T P_i X_i + X_i^T X_i L_i L_i X_i + 1)^{-1} (X_i^T P_i^T P_i X_i + X_i^T X_i L_i L_i X_i).
$$

(22)

For complete presentation of the method, we summarize the optimization procedures of TP-DPL in Table I, where the iteration will stop when the difference between consecutive objective function values in adjacent iterations is less than 0.0001, and the dictionary $D$ is initialized by applying the identical number of training samples of each class.

### Table I: Optimization procedures of TP-DPL

| **Input:** | Training data matrix $X$, class label set $Y$, dictionary size $K$, parameters $\alpha$, $\beta$ and $\gamma$. |
| **Output:** | $D$, $L$, $P$, $W$, $S$ |
| **Initialization:** | Initialize $P^{(0)}$, $S^{(0)}$ and $W^{(0)}$ to be random matrices with unit F-norm; Initialize $D^{(0)}$ by the cosine similarity between training samples; Initialize $D^{(0)}$ using training samples; $t=0$ |
| **while** not converge **do** |
| 1. Update the sparse coefficients $S^{(t)}$ by Eq.(15) |
| 2. Update the analysis dictionary $D^{(t)}$ by Eq.(16) |
| 3. Update the salient feature coefficients $P^{(t)}$ by Eq.(18) |
| 4. Update the synthesis latent dictionary $L^{(t)}$ by Eq.(20) |
| 5. Update the adaptive weight matrix $W^{(t)}$ by Eq.(22) |
| 6. $t = t+1$ |
| **end while** |

### C. Convergence Analysis

The problem of TP-DPL is solved alternately, so we want to analyze its convergence. Note that TP-DPL is essentially an alternate convex search (ACS) algorithm [20-22], so we can have the following remarks [20-22] to assist the analysis.

**Theorem 1** [22]. If $B \in \mathbb{R}^{n \times n}$, $f : B \rightarrow \mathbb{R}$ is bounded and the optimization of variables in each iteration are solvable, the generated sequence $\{f(z)\}_{z \in \mathbb{B}}$ by using the ACS algorithm will converge monotonically.

**Theorem 2** [22]. Let $X \subseteq \mathbb{R}^n$, $Y \subseteq \mathbb{R}^m$ be the closed set and let $f : X \times Y \rightarrow \mathbb{R}$ be continuous. Let the optimization of each variable in each iteration be solvable, then we can have: (1) If the sequence $(z)_{z \in \mathbb{B}}$ by ACS is contained within a compact set, the sequence will contain at least one accumulation point. (2) For each accumulation point $z$ of sequence $(z)_{z \in \mathbb{B}}$: (a) if the optimal solution of one variable with others fixed in each iteration is unique, then all accumulation points will be the local optimal solutions and have the same function value; (b) if the optimal solution of each variable is unique, then we have $\lim_{z \rightarrow z_0} \|z - z_0\| = 0$, and the accumulation points can form a compact continuum $C$.

Based on Theorem 1 and Theorem 2, we can present three remarks on the convergence of our TP-DPL.

**Remark 1.** The generated sequence $\{f(D, S, P, L, W)\}_{z \in \mathbb{N}}$ by our TP-DPL converges monotonically in each iteration. **Proof.** For the objective function of TP-DPL in Eq.(13), $W$, $P$, $S$, $D$, $L$ are major variables. From the procedures, if $D$, $L$ and $W$ are fixed, the objective function is convex for $S$ and $P$; if $S$ and $P$ are fixed, the function is convex for $D$, $L$ and $W$. In other words, the objective function of TP-DPL is a bi-convex problem for $\{D, L, W\}_{(S, P)}$ and the proposed optimization method is actually an alternate convex search.
and since they are small feature values for the function \( F \) will be potentially small over class \( k \). The sequence of \( \{D, S, P, E, W\} \) generated by our TP-DPL has at least one accumulation point. All the accumulation points are local the optimal solutions of \( f \) and moreover have the same function value.

**Proof.** It is easy to check the problem of TP-DPL satisfies \( f(D, L, S, P, W) \) for \( \|D\|_1 \rightarrow \infty \), \( \|S\|_1 \rightarrow \infty \), \( \|P\|_1 \rightarrow \infty \) and \( \|W\|_1 \rightarrow \infty \). Thus, the generated sequence \( \{D, S, P, E, W\} \) is bounded in finite dimensional space, and the compact set condition in Theorem 2 (Condition 1) is satisfied. Thus, the sequence has at least one accumulation point. By Theorem 2 (Condition 2a), all accumulation points are local optimal and have the same functional value.

**Remark 3.** If \( D, S, P \) and \( L \) have unique solutions, then the sequence \( \{D, E, S, P, W\} \) generated by TP-DPL satisfies:

\[
\lim_{l \rightarrow \infty} \|D^{l+1} - D^l\|_1 = 0, \quad \|E^{l+1} - E^l\|_1 = 0, \quad \|P^{l+1} - P^l\|_1 = 0, \quad \|W^{l+1} - W^l\|_1 = 0.
\]

**Proof.** Based on Remark 2, the Condition 1 and 2a in the Theorem 2 are satisfied in TP-DPL, if we have the unique optimal solutions of \( L \) and \( W \), then we have the conclusion Eq. (23) based on the Condition 2b in Theorem 2 [22]. So, it is easy to check that our TP-DPL is a reasonable approach.

**D. Classification Approach**

We discuss the classification approach using TP-DPL. After convergence of TP-DPL, the synthesis dictionary \( D^{l} \) and analysis dictionary \( L^{l} \) can be obtained to produce small coefficients of samples from classes other than \( k \), since they can only generate significant coefficients for the samples of the class \( k \). Moreover, the projection \( P^{l} \) is also trained to reconstruct the samples of class \( k \) to produce the significant salient feature values \( P^{l}X^{l} \) and small feature values for the samples of classes other than \( k \). As a result, the residual \( \|Y_l - D^{l}L^{l}X_l - P^{l}X^{l} - W^{l}\|_1 \) will be potentially small over class \( k \).

On the other hand, although \( L_l, P_l \) and \( D_l \) are not trained to reconstruct \( X_l \) in the feature space, \( D^{l}X^{l} \) and \( P^{l}X^{l} \) are small, i.e., the residual \( \|Y_l - D^{l}L^{l}X_l - P^{l}X^{l} - W^{l}\|_1 \) will be large.

In the testing phase, if a query sample \( y \) is a sample from class \( k \), its twin-projective dictionaries \( D^{l} \) and \( L^{l} \), and salient feature vector by \( P^{l} \) will be more likely to be significant, while its latent dictionary vector by \( L_l(i \neq k) \) and salient feature coding vector by \( P_l(i \neq k) \) tend to be small. As a result, the reconstruction residual \( \|Y_l - D^{l}L^{l}X_l - P^{l}X^{l} - W^{l}\|_1 \) will be much smaller than residual \( \|Y_l - D^{l}L^{l}X_l - P^{l}X^{l} - W^{l}\|_1 \). Thus, the class-specific reconstruction residual can be used to identify the label of sample \( y \). Thus, similarly as [4][9] we can naturally define the following classifier associated with our TP-DPL:

\[
id(y) = \text{argmin}\ \|y - D^{l}L^{l}X - P^{l}\|_2.
\]

where \( D_l \) is the trained synthesis dictionary of class \( l \), \( L_l \) is the analysis sub-dictionary, \( P_l \in \mathbb{R}^{m \times n} \) is the sub-projection.

**IV. DISCUSSION: RELATIONSHIP ANALYSIS**

**A. Connection to the DPL algorithm [9]**

The most related model to our TP-DPL is DPL, and we will show that DPL is a special case of our TP-DPL. Recalling the objective function of TP-DPL in Eq.(7), suppose that we constrain \( \beta=0 \) and \( P \to 0 \), the problem can be reduced to:

\[
\min_{D, L, S, P, W} \sum_{i=1}^{n} \|Y_i - a_i - D_iX_i + P_i\|_1, \quad \text{s.t.} \quad \|a_i\|_1 \leq 1,
\]

which is just the flexibly-relaxed formulation of DPL, since the above problem changes to DPL if \( a_i = 0 \). But setting \( \beta=0 \) means that local information of coefficients and salient features cannot be preserved any more and setting \( a_i = 0 \) to minimize \( \|Y_i - D_iX_i\|_1 \) may make the reconstruction from the over-fitting. Thus, our TP-DPL will be superior to DPL.

**B. Connection to the LLC-DL algorithm**

We also discuss the connection between our TP-DPL and LLC-DL [10]. To facilitate the comparison, we first present the objective function of LLC-DL as follows:

\[
\min_{D, L, S, P, W} \sum_{i=1}^{n} \|Y_i - D_S X_i - P_X\|_1 + a_i \|y_i - x_i\|_1 + \|A_S L\|_0,
\]

where \( H \) is class label set and \( \|\hat{W}^\perp\|_1 \) is the \( l_1 \)-norm based robust classifier. By comparing the above problem with our TP-DPL, we can find that: (1) LLC-DL discusses latent DL under a structured discriminative sparse code error, while we discuss the twin-projective latent flexible adaptive learning of dictionary pairs (a synthesis dictionary \( D \) and an analysis dictionary \( L \)) under a structured twin-incoherence. Moreover, the function of the adaptive weighting function \( g(x, y, W) \) in TP-DPL correlates the coefficients and salient features, while LLC-DL cannot mine shared vital information in the feature and representation spaces; (2) For the salient feature extraction, LLC-DL learns a shared projection \( P \) for all the classes, while TP-DPL learns sub-projections \( P_l \) for various classes. More importantly, TP-DPL imposes an incoherence constraint on salient features over different classes to ensure high inter-class separation by encouraging each \( L_l \) to project the training samples of class \( j (j \neq l) \) to a nearly null space. But LLC-DL cannot ensure this issue clearly; (3) LLC-DL computes the representation matrix \( S_j \) of each class directly, so it needs the extra time-consuming sparse reconstruction process with well-trained dictionary \( D \) to obtain the codes of each new data. In contrast, our TP-DPL learns an analysis dictionary \( L_l \) jointly, which can be used to extract the codes from inside and outside data efficiently. In addition, TP-DPL applies the Frobenius-norm for preserving the group sparse properties similarly as DPL [9], while LLC-DL applies the \( l_1 \)-norm on the coefficients, but the optimization of \( l_1 \)-norm is usually time-consuming; (4) LLC-DL incorporates the classification error over the extracted features for the joint optimization by involving an extra tuning parameter, but the optimal parameter selection is usually difficult in reality. While our TP-DPL will not suffer from this issue, since our method minimizes the reconstruction residual to determine the label of each outside new data directly.

**V. EXPERIMENTAL RESULTS AND ANALYSIS**

We mainly evaluate our TP-DPL for data representation and classification, and the performance of TP-DPL is compared with those of closely related SRC [4], DLSI [8], KSVD [3], D-KSVD [6], LC-KSVD [1], COPAR [32], FDDL [7], DPL [9], ADDL [11], LLC-DL[10] and LSRLD [16]. Since DLSI and KSVD did not define an explicit classification method, we apply the same classification approach of SRC for them.

Five face databases (i.e., ORL [25], YaleB [26], UMIST
images in each individual are randomly chosen for training and the number of atoms is set to 100, corresponding to an average of 5 items per person. For ETH80, we follow [11] to use discriminant features [33], select 6 images from each class for training, test on the rest and select the number of atoms corresponding to an average of 6 items per class. For each pair of parameters, we average the results over varied parameters from $\{5 \times 10^4, 5 \times 10^5, 5 \times 10^3, 5 \times 10^2, 5, 5 \times 10^1, 5 \times 10^0, 5 \times 10^{-1}, 5 \times 10^{-2}\}$. The parameter selection results are shown in Fig.3. As can be seen, our TP-DPL performs well in a wide range of parameters in each group, i.e., it is insensitive to the model parameters.

![Parameter sensitivity analysis of TP-DPL on ETH80](image1)

![Parameter sensitivity analysis of TP-DPL on AR](image2)

### B. Parameter Selection Analysis

The parameter selection issue still remains an open problem, thus we apply a heuristic way to select the most important parameters. Note that our TP-DPL has three parameters (i.e., $\alpha$, $\beta$ and $\gamma$), so we fix one of the parameters and explore the effects of other two on the performance by the grid search strategy. AR face and ETH80 object databases are used as examples. For AR, we use the convolutional features and 20

**TABLE II.**

| Dataset Name     | # Samples | # Dim | # Classes |
|------------------|-----------|-------|-----------|
| ORL face         | 400       | 1024  | 40        |
| YaleB face       | 2414      | 504   | 38        |
| AR face          | 2600      | 540   | 100       |
| CMU PIE face     | 11554     | 1024  | 68        |
| UMIST face       | 1012      | 1024  | 20        |
| ETH80 object     | 3280      | 1024  | 80        |
| Fifteen scene categories | 4485 | 3800 | 15 |

![Convergence behavior of TP-DPL](image3)

**Fig. 2:** Convergence behavior of TP-DPL, where the x-axis is the number of iterations and the y-axis is the objective function value.

**C. Application to Image Recognition**

We evaluate each method for representing and recognizing three kinds of image databases, i.e., face images (i.e., YaleB, AR, CMU PIE, and UMIST), ETH80 object database, and the fifteen nature scene categories database. Some image examples of these databases are shown in Fig.4. For each method, we choose the model parameters carefully. Since KSDV, D-KSVD and LC-KSVD applies the $l_0$-norm based sparsity constraint for DL, we still use the $l_0$-norm for them for fair comparison. The averaged recognition results are reported as the evaluation metric of each algorithm.

![Sample images of the evaluated real-world databases](image4)

**Fig. 4:** Sample images of the evaluated real-world databases.

**Face Recognition on YaleB.** We use random face features [1-3][13][18-19] in this study, i.e., each image is projected onto a 504-dimensional vector by a generated matrix from a zero-mean normal distribution, and each row of matrix is $l_2$ normalized. We clearly follow the setting in [9] for the fair comparison, i.e., half of the images per class are randomly selected for training and the rests are used for testing. The dictionary contains 570 items, corresponding to an average of 15 items of each class. The averaged recognition rates are reported in Table III, where $\alpha=0.0005$, $\beta=500$ and $\lambda=0.5$. 

![Sample images](image5)
are set in TP-DPL. The results of other compared methods are adopted directly from [9]. We find that TP-DPL delivers higher accuracy than its competitors under the same setting.

**Face Recognition on AR.** In this study, by following the common procedures [1-3][12-14], the face set that contains 2600 images of 50 males and 50 females is evaluated. We clearly follow [1-3][13][19] to use 540-dimensional random face features. We also randomly choose 20 images from each person for training and test on the rest. The dictionary contains 500 items, corresponding to an average of 5 items per category. $a=0.0005$, $b=50000$ and $γ=0.5$ are used for our TP-DPL. The results are described in Table IV, where the results of compared methods are adopted from [1][11]. We find from the results that our TP-DPL can deliver enhanced results than its competitors under the same setting.

| Evaluated Methods | Accuracy |
|-------------------|----------|
| SRC(all train. sample) | 96.5% |
| K-SVD(15 items) | 93.1% |
| D-KSVD(15 items) | 94.1% |
| LC-KSVD(15 items) | 94.5% |
| LC-KSVD(15 items) | 95.0% |
| DLSI (15 items) | 97.0% |
| COPAR (15 items) | 96.0% |
| FDDL(15 items) | 96.7% |
| DPL(15 items) | 97.5% |
| LRSID (15 items) | 97.3% |
| ADDL(15 items) | 97.8% |
| Our TP-DPL(15 items) | 98.2% |

**Face Recognition on CMU PIE.** This database contains 68 persons with 41368 face images as a whole. Follow the common procedures in [2][29], 170 near frontal images per person are employed for the evaluations. This face subset consists of five near frontal pose (C05, C07, C09, and C29) and all the images have different illuminations, lighting and expression. We also adopt random face features [4][17] and set the dimension to 256. We train on 20, 30, and 40 images per person and test the rest, and set the dictionary size to the number of training images in each study. The averaged results are described in Table V, where $a=50$, $b=500$ and $γ=0.005$ are used in TP-DPL. We find that: (1) the accuracy increases as the training number increases; (2) our TP-DPL is superior to its competitors in investigated cases.

| Evaluated Methods | Accuracy |
|-------------------|----------|
| SRC(30 items, 100 labels) | 91.8% |
| KSVD(30 items, 100 labels) | 86.7% |
| D-KSVD(30 items, 100 labels) | 89.1% |
| LC-KSVD(30 items, 100 labels) | 90.4% |
| LC-KSVD2(30 items, 100 labels) | 92.9% |
| DLSI(30 items, 100 labels) | 92.5% |
| COPAR(30 items, 100 labels) | 92.9% |
| DPL(30 items, 100 labels) | 93.1% |
| LRSID (30 items, 100 labels) | 97.1% |
| ADDL(30 items, 100 labels) | 98.1% |
| Our TP-DPL(30 items, 100 labels) | 98.8% |

**Scene Recognition on fifteen categories database.** The nature scene categories database includes fifteen scenes, i.e., suburban, open country, mountain, coast, forest, store, kitchen, office, industrial, living room, tall building, bedroom, street, highway and inside city. Each category contains 200 to 400 images, and each scene image has about 250 x 300 pixels. By following [1][9][11], the spatial pyramid features by using a four-level spatial pyramid and a SIFT-descriptor codebook with size 200 are computed for the simulations. The final spatial pyramid features are reduced to 3000 by using PCA [34]. Following the settings in [1][11], we select 100 images per category for training and test on the rest. The dictionary size is set to 450 items, corresponding to an average of 30 items for each category. $a=5\times10^4$, $b=5000$ and $γ=0.5$ are used in TP-DPL. We describe the averaged results in Table VII, where directly adopt the results of compared methods from [1][11]. We can find that TP-DPL obtains better results than other models under the same experimental setting.

**Object Recognition on ETH80.** ETH80 object database has 3280 images of 80 subcategories from 8 big categories.
[30]. Each big category contains 10 subcategories, each of which has 41 images. We follow [11] to use the discriminant features [33]. We similarly select 6 images from each class for training and test on the rest. $\alpha=50$, $\beta=50$ and $\gamma=0.05$ are used in TP-DPL. We show the averaged results in Table VIII. We find that TP-DPL achieves the enhanced result than other evaluated models. DPL, LLC-DL, LSDL and ADDL also deliver promising results that are highly comparative with TP-DPL. In addition, we also evaluate the recognition rates for individual classes and show some image examples in the eight classes having high accuracy rate in Fig.5.

| Evaluated Methods | Mean ± Std (%) |
|-------------------|---------------|
| SRC (6 items, 6 labels) | 89.6 ± 0.8 |
| KSVD (6 items, 6 labels) | 91.2 ± 0.8 |
| D-KSVD (6 items, 6 labels) | 91.2 ± 0.4 |
| LK-SVD (6 items, 6 labels) | 90.7 ± 0.7 |
| LK-KSVD (6 items, 6 labels) | 91.5 ± 0.8 |
| DLSDL (6 items, 6 labels) | 92.7 ± 0.9 |
| COPAL (6 items, 6 labels) | 93.1 ± 0.7 |
| FDDL (6 items, 6 labels) | 93.2 ± 0.3 |
| DPL0 (6 items, 6 labels) | 97.7 ± 0.2 |
| DPL5 (6 items, 6 labels) | 97.7 ± 0.2 |
| Our TP-DPL (6 items, 6 labels) | 98.3 ± 0.2 |

Results on AR. In this study, we randomly choose 5, 10, 15 and 20 images from each person for training and test on the rest. The number of dictionary atoms is set to be the size of the training set for our TP-DPL, which corresponds to an average of 5, 10, 15 and 20 items from each category. $\alpha=50$, $\beta=50$ and $\gamma=0.05$ are used in our TP-DPL. We report the mean clustering AC of different runs in Table IX. We find that the clustering result on $DLY$+$PY$ is obviously superior to those on $Y$, $DLY$ and $PY$, i.e., $Y$, $DL$ or $PY$ fails to capture all important features. In other words, $DLY$+$PY$ can describe given data better. As a result, existing models that directly apply the reconstruction $DS$, coefficients or salient features for data classification or clustering may produce inaccurate results. On the contrary, our TP-DPL minimizes a structured twin-incoherence based twin-projective flexible latent reconstruction error for representation learning and data classification, which is potentially more reasonable.

| Atom number | Y | DLY | DLY+$PY$ |
|-------------|---|-----|---------|
| 5(per class) | 71.55% | 51.73% | 81.17% | 86.63% |
| 10(per class) | 76.11% | 36.68% | 55.74% | 87.51% |
| 15(per class) | 65.65% | 30.08% | 30.41% | 88.44% |
| 20(per class) | 66.47% | 32.52% | 31.43% | 89.22% |

Results on CMU PIE face database. For this database, we randomly choose 10, 15, 20, 25, 30, 35 and 40 images per person for training and test on the rest. The number of dictionary atoms is set to be the size of training set for our TP-DPL. $\alpha=50$, $\beta=50$ and $\gamma=0.005$ are set in TP-DPL. The averaged clustering results are shown in Table X. We can find that the clustering AC on $DLY$+$PY$ can still deliver the enhanced performance over $DL$ or $PY$ based on different numbers of training samples, which can once again prove that $DLY$+$PY$ is able to describe the original data better than $DL$ or $PY$. It can also be found that the increasing number of training samples can improve the result on $DLY$+$PY$.

| Train number | Y | DLY | DLY+$PY$ |
|--------------|---|-----|---------|
| 10(per class) | 81.62% | 31.12% | 31.03% | 85.53% |
| 15(per class) | 80.85% | 31.32% | 30.87% | 86.42% |
| 20(per class) | 81.03% | 29.59% | 28.89% | 86.69% |
| 30(per class) | 80.86% | 23.52% | 24.07% | 87.74% |
| 35(per class) | 81.45% | 23.05% | 23.96% | 88.28% |
| 40(per class) | 81.87% | 23.05% | 22.35% | 88.78% |

Visualizaiton of features on CMU PIE. We also aim at visualizing the feature set $Y$ and $DLY$+$PY$ for observation. Based on the convolutional features of database, we choose 40 images per person for training and use the rest to form $Y$. We simply set the number of atoms to the number of training samples. To better compare the original feature set $Y$ and our $DLY$+$PY$ features, the top-10 class are selected from the test set for this visualization task based on the clustering results. Note that we aim to visualize the first 9-dimensions of the features $Y$ and $DLY$+$PY$ in Fig.6, from which we observe that the features $DLY$+$PY$ by our TP-DPL are clearly better than convolutional features $Y$ by obtaining high intra-class compactness and high inter-class separation, which is a very good message for the feature representation, data clustering and classification in reality.
E. Quantitative Evaluation of Dictionaries

We mainly evaluate the performance of learned dictionary $D$ of each method. We show the quantitative evaluation results of recognition against varying dictionary sizes. Three image databases, i.e., MIT CBCL, CMU PIE and fifteen nature scene categories databases are evaluated. For CMU PIE, we still use random face features of dimension 256, choose 30 samples per class for training and evaluate each method with varying dictionary sizes $K$ of dictionary, i.e., $K=340, 680, 1020, 1360, 1700$ and 2040 in Fig. 7a. For fifteen scene categories, we also follow [1][5] to choose 100 samples per class for training and evaluate each method with varying dictionary sizes $K$, i.e., $K=75, 150, 225, 300, 375$ and 450 in Fig. 7b. For MIT CBCL, we choose 6 samples per class as training set, test on the rest, evaluate each model with varying sizes $K$ of the dictionary, i.e., $K=20, 30, 40, 50$ and 60, and the averaged recognition results are illustrated in Fig. 7c. We can find that: (1) the recognition accuracy of each method can be increased when the number of atoms increases; (2) TP-DPL obtains better results than its competitors across all dictionary sizes. DPL, ADDL and LRSDL can also perform well by delivering higher accuracies than other remaining methods. KSV is the worst method in most cases.

VI. CONCLUDING REMARKS

We proposed a structured twin-incoherence constrained twin-projective latent adaptive DPL model for classification. TP-DPL unifies the salient feature extraction, representation and classification in an adaptive locality-preserving manner via minimizing a twin-incoherent flexible reconstruction error. The twin-incoherence constraints on coefficients and salient features can produce discriminative coefficients and features with high inter-class separation and high intra-class compactness, which is benefit for enhancing classification.

We evaluated the effectiveness of our method on several public databases for data classification and clustering. The obtained results demonstrated the superior performance of our model. The clustering and visualization of features also verified the effectiveness of the twin-incoherence based twin-projective latent reconstruction to deliver the salient features jointly. In future, we will investigate how to extend our model to the semi-supervised scenario [12] to handle the cases that the number of labeled samples is limited to enhance the latent dictionary pair learning. Extending our method to the other application areas, such as the content or text based image retrieval, is also interesting.
ACKNOWLEDGMENT

This work is partially supported by the National Natural Science Foundation of China (61672365, 61730008, 61725203, 61622305, 61871444, 61572339), High-Level Talent of the "Six Talent Peak" Project of Jiangsu Province of China (XYDX-055), and the Fundamental Research Funds for the Central Universities (17LX034, 12Z1019FGPA-0102). Dr. Zhao Zhang is the corresponding author.

REFERENCES

[1] Z. Jiang, L. Z. Jin, and S. Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 35, pp. 2651-2664, 2013.
[2] Z. Zhang, F. Z. Li, T. W. Chow, L. Zhang, and S. C. Yan, "Sparse Codes Auto-Extractor for Classification: A Joint Embedding and Dictionary Learning Framework for Representation," IEEE Trans. Signal Processing, vol. 64, no.14, pp. 3790-3805, 2016.
[3] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," IEEE Trans. Signal Processing, vol. 54, no. 1, pp. 4311-4322, 2006.

[10] J. Wright, M. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust Face Recognition via Sparse Representation," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 31, no. 2, pp. 210-227, 2009.
[11] Z. Li, Z. Lai, Y. Xu, J. Yang, and D. Zhang, "A Locality-Constrained Label Embedding Dictionary Learning Algorithm for Image Classification," IEEE Trans. on Neural Networks and Learning Systems, vol.28, no.2, pp.278-293, 2017.

[19] M. Zhang, L. Zhang, X. Feng, and D. Zhang, "Sparse Representation based Fisher Discriminant Dictionary Learning for Image Classification," International Journal of Computer Vision, vol.109, no.3, pp. 209-232, 2014.
[20] I. Ramirez, P. Sprechmann, G. Sapino, "Classification and Clustering via dictionary learning with structured incoherence and shared features," In: Proc. IEEE CVPR, CA USA, pp. 3501-3508, 2010.

[22] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Projective dictionary pair learning for pattern classification," In: Proceedings of the Neural Information Processing Systems, Montreal, QC, Canada, 2014.

[25] Z. Zhang, W. Jiang, F. Li, M. Zhao, B. Li and L. Zhang, "Structured Latent Label Consistent Dictionary Learning for Salient Machine Faults Representation based Robust Classification," IEEE Trans. on Industrial Informatics, vol.13, iss.2, pp.642-654, 2017.
[26] Z. Zhang, W. Jiang, J. Qin, L. Zhang, F. Li, M. Zhang, and S. Yan, "Joint label and structured analysis discriminative dictionary and analysis multiclass classifier," IEEE Trans. on Neural Networks and Learning Systems, vol.29, iss.8, pp.3798-3814, 2018.

[35] W. Jiang, Z. Zhang, F. Li, L. Zhang, M. Zhao, and X. Jin, "Joint label consistent dictionary learning and adaptive label prediction for semi-supervised machine fault classification," IEEE Transactions on Industrial Informatics, vol. 12, no. 1, pp. 248-256, 2016.

[44] Qiu, Z. Li, Jiang, and R. Chellappa, "Sparse Dictionary-Based Representation and Recognition of Action Attributes," In: Proc. IEEE Int’l Conf. Computer Vision, 2011.
[45] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” IEEE Transactions on Image Processing, vol. 14, no. 12, pp. 4263-4278, 2005.

[46] R.Rubinstein, A. Bruckstein, and M. Elad, “Dictionaries for sparse representation modeling,” Proceedings of the IEEE, vol.98, no.6, pp.1045-1057, 2010.

[47] T. H. Vu, and V. Monga, “Fast low-rank shared dictionary learning for image classification,” IEEE Trans. on Image Processing, vol. 26, no. 11, pp. 5160-5175, 2017.
[48] Z. Zhang, Y. Zhang, F. Z. Li, M. B. Zhao, L. Zhang and S. C. Yan, “Discriminative Sparse Flexible Manifold Embedding with Novel Graph for Robust Visual Representation and Label Propagation,” Pattern Recognition, vol. 61, pp. 492–510, Jan 2017.

[49] J. Wen, X. Fang, J. Cui, L. Fei, K. Yan, K. Yan, Y. Chen, Y. Xu, “Robust Sparse Linear Discriminant Analysis,” IEEE Transactions on Circuits and Systems for Video Technology, 2018.

[50] Z. Jiang, G. Zhang, and L. Davis, “Submodular Dictionary Learning for Sparse Coding," In: Proc. IEEE CVPR, 2012.
[51] J. Leeu, “Block-relaxation algorithms in statistics," In: Information systems and data analysis, Springer, pp. 308–324, 1994.