Nonlinear Kernel Dictionary Learning Algorithm Based on Analysis Sparse Model

ZHUOYUN MIAO, HONGJUAN ZHANG, AND SHUANG MA
Department of Mathematics, Shanghai University, Shanghai 200444, China
Corresponding author: Hongjuan Zhang (zhanghongjuan@shu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 11501351.

ABSTRACT In the past decades, relevant sparse representation models and their corresponding dictionary learning algorithms have been explored extensively as they could be applied in various fields. However, most of them are focusing the linear model and nonlinear one is still less touched, although there are plenty of nonlinear scenarios in real applications. To further address this kind of the unmet challenge, in this work we mainly focus the following two works (i) propose a kernel transformation based method directly transforming the nonlinear analysis problem into a linear one, which is exactly the standard sparse analysis form but implies all nonlinear information of the original problem; (ii) present a nonlinear dictionary learning algorithm by leveraging the kernel trick and the KSVD-like manner, which has its root in analysis sparse model rather than synthesis model. Then, the proposed methods are employed to address the classification problem. Benchmark experimental results on three well-known datasets show that the proposed algorithm in (ii) outperforms some related linear algorithms and other existing nonlinear dictionary learning algorithms. Moreover, when the data is interfered by noise or some pixels are missing in the data, the algorithm is also effective, which proves its theoretical advantages owing to analysis sparse model’s merits of the equality of all atoms and the much smaller dimensionality for signal representation to some extent. And the classification accuracy of the proposed method in (i) is slightly lower than that of (ii), but better than that of other state of art methods.

INDEX TERMS Nonlinear classification problem, analysis sparse model, analysis KSVD algorithm.

I. INTRODUCTION
Sparse and redundant signal representation is a research hotspot in recent years. Many scholars have developed strong interests in many domains [1], [2]. Among them, signal models are fundamental for handling various processing tasks, such as denoising [3], solving inverse problems [4], compression [5], interpolation [6], sampling [7], and more. In general, there are two sparse representation models, namely the synthesis sparse model and the analysis sparse model. In the synthesis sparse model, a signal \( y \in \mathbb{R}^n \) is modeled as being the outcome of the multiplication \( y = D\alpha \), where \( D \in \mathbb{R}^{n \times K} \) is a dictionary – its columns are signal prototypes (atoms) that are used to build the signal [8]. As for the analysis sparse model, it relies on a linear operator (a matrix) \( \Omega \in \mathbb{R}^{p \times n} \), which we will refer to as the analysis dictionary, and whose rows constitute analysis atoms [8]. Note that there are both connections and differences between these two models. While the analysis model is referred to as the dual form of the synthesis model when dealing with a square (and invertible) matrix \( \Omega \), it is in fact very different when dealing with a redundant dictionary \( p > n \) [8]. And their key difference is that they pursue the signal’s sparsity in different fields. Specifically, the synthesis sparse model seeks for the sparsity in the space spanned by the dictionary, while the analysis sparse model pursues the sparsity in the range of the sparsity transform.

Furthermore, a fundamental ingredient in the definition of both signal models and its deployment to applications is the dictionary \( D \) or \( \Omega \). Researches have already demonstrated that dictionary learning model usually has a better signal reconstruction and classification performance than directly utilizes original training samples [9], [10]. And more and more dictionary learning algorithms based on the above two models have been proposed for classification tasks, such as face recognition [11], image classification [12], object recognition [13], ear recognition [14].

A. DICTIONARY LEARNING ALGORITHM FOR SYNTHESIS SPARSE MODEL
In this part, we will describe in detail one of the most popular models over the past decades – the synthesis sparse model

\[
\min_{a,D} \| y - D\alpha \|_2 + \| \alpha \|_0
\]
where $\alpha \in \mathbb{R}^K$ is the sparse representation vector, $\|\alpha\|_0$ is the $l_0$ sparsity measure that counts the number of nonzero elements in the vector $\alpha$, and $\|y - D\alpha\|_2$ is the mean squared error resulting from sparse approximation.

However, we all know that the $l_0$ norm problem is NP-hard, which usually needs to be transformed into other problems, such as $l_1$ norm problem, to approximate its solution. Then the optimization problem (1) can be transformed into the following problem

$$
\min_{\alpha,D} \|y - D\alpha\|_2 + \|\alpha\|_1 \quad (2)
$$

It is worth noting that the selection of the dictionary $D$ in model (1) and (2) is crucial during the processing of synthesis sparse model. There are mainly two approaches for its selection, including a predefined dictionary and the dictionary learning method. Indeed, it has been observed that learning a dictionary directly from the training data rather than using a predetermined dictionary (e.g. wavelet) usually leads to a more compact representation and hence can provide better results in many image processing applications, such as recovery [1] and classification [15], [16]. Therefore, over the past decades, researchers have developed many effective dictionary learning methods both for the original problem (1) and its relaxation problem (2), such as KSVD algorithm [14], online dictionary learning algorithm [17], sparse representation-based classifier (SRC) algorithm [18], and so on. Among them, KSVD is the most representative algorithm, which performs singular value decomposition (SVD) on $K$ submatrices and is the generalization of the K-means clustering process.

It is noting that all the above methods are aimed at linear signals processing, but in practice the linear problem is too simple and idealized. Therefore, the nonlinear signal processing problem has become a hot research topic recently. Generally speaking, its basic optimization problem is as follows

$$
\min_{x,\Omega} \|y - x\|_2 + \|\Omega x\|_0 \quad (5)
$$

Here $x \in \mathbb{R}^n$ is the reconstructed signal. The objective is to find a suitable operator $\Omega$ so that the analysis coefficients operator $\Omega x$ is sparse. Note that when both the synthesis sparse model and the analysis sparse model depart, the analysis model becomes more interesting and powerful. This case of analysis dictionary training is a challenging problem [8]. In the past few years, model (5), as one kind of linear case, has been mainly studied [8], [30], [31]. For example, the analysis KSVD algorithm [8] aims to learn an analysis dictionary, which is parallel to the synthesis KSVD in its rationale and structure-analysis KSVD algorithm.

Just like the above synthesis sparse model, the optimization problem (5) can be also relaxed to the following equation

$$
\min_{x,\Omega} \|y - x\|_2 + \|\Omega x\|_1 \quad (6)
$$

Many scholars have also given some results on this kind of relaxation problems [32]–[34].

However, most of the above-mentioned works about the analysis sparse model are all focusing on the dictionary learning method for the linear case. Compared with the linear case, there is less work to discuss nonlinear analysis-based dictionary learning methods, although we all know that nonlinear case is more practical in real problems. Therefore, in this work, we creatively combine the nonlinear kernel approach with the analysis sparse model, and propose two effective methods. The first one is about the kernel transformation method, which transforms the nonlinear analysis model into the linear form, and then is solved directly by linear dictionary learning algorithms. The second focuses on the nonlinear dictionary learning algorithm, which can effectively obtain the analysis dictionary from given datasets by utilizing the sparsity of data in high dimensional feature. Moreover, the validity of both methods are demonstrated in a series of experiments on some datasets.

The manuscript is organized as follows. In Section II, we will introduce the kernel transformation method.
Section III presents the two-stage method which comprises an analysis pursuit step using kernel backward-greedy algorithm, and a dictionary update step based on analysis kernel KSVD. Section IV gives many experimental results to illustrate the effectiveness of our approaches. The final Section concludes this work.

II. KERNEL SPARSE REPRESENTATION TRANSFORMATION METHOD

In this section, we will propose the first work – a kernel transformation method (KTM) that can directly transform a nonlinear problem based on an analysis model into a linear one, but which implies all nonlinear information of the original problem.

Firstly, we will introduce a nonlinear mapping \( \Phi(\cdot) \), which maps the data from the original space \( \mathbb{R}^n \) into a high-dimensional Hilbert feature space \( \mathcal{H} \)

\[
\Phi: \mathbb{R}^n \rightarrow \mathcal{H} \subset \mathbb{R}^\tilde{n}
\]

(7)

here \( \tilde{n} \) is generally much larger than \( n \), and could even be infinite. Therefore, the nonlinear-analysis-based sparse representation in feature space \( \mathcal{H} \) can be represented as follows

\[
\min_{\Phi(x), \Phi(\Omega)} \| \Phi(y) - \Phi(x) \|_2^2 + \| \Phi(\Omega)^T \Phi(x) \|_1
\]

(8)

where \( \Phi(y) \in \mathbb{R}^{\tilde{n}} \) is the given signal, \( \Phi(x) \in \mathbb{R}^{\tilde{n}} \) is the reconstructed signal corresponding to sample \( \Phi(y) \), \( \Phi(\Omega)^T \in \mathbb{R}^{K \times \tilde{n}} \) is the sought analysis dictionary. Here we define the analysis dictionary as \( \Phi(\Omega)^T \) instead of the usual \( \Phi(\Omega) \) for the convenience of the following calculations.

Next, we will use the nuclear technique, so we will briefly introduce the nuclear function firstly. The nuclear function \( \kappa(x, y) \) satisfies the Mercer’s condition [35]: For all data \( \{y_i\}_{i=1}^N \), the function gives rise to a positive semidefinite matrix \( \mathcal{K}_{ij} = \kappa(x_i, y_j) \). Then the nuclear function in the Hilbert feature space \( \mathcal{H} \) can be expressed as \( \kappa(x, y) = \phi(x)^T \phi(y) \). Nuclear functions are often used to implicitly specify the mapping \( \Phi \). It helps to avoid the huge computations required to map of data into the high-dimensional feature space. Generally speaking, some commonly used kernels include Gaussian kernels \( \kappa(x, y) = \exp(-\frac{\|x-y\|^2}{\epsilon}) \) and polynomial kernels \( \kappa(x, y) = (x^T y + c)^d \), where \( c \) and \( d \) are the parameters.

Using the above nuclear technique, given the SVD of \( \mathcal{K}(\Omega, \Omega) \) is \( U \Sigma U^T \), \( U \Sigma^T = I \), we extend the reconstruction term in (8) to

\[
\| \Phi(x) - \Phi(y) \|_2^2 = \| \Phi(x)^T \Phi(x) - \Phi(y)^T \Phi(y) \|
\]

\[
= \| \Phi(x)^T \Phi(x) - \Phi(x)^T \Phi(y) - \Phi(y)^T \Phi(x) + \Phi(y)^T \Phi(y) \|
\]

\[
= \| \kappa(x, x) - 2\kappa(x, y) + \kappa(y, y) \|
\]

\[
= \| \kappa(\Omega, \Omega)^T U \Sigma^{-1/2} \Sigma^{-1/2} U^T \kappa(\Omega, x) - 2\kappa(\Omega, \Omega)^T U \Sigma^{-1/2} \Sigma^{-1/2} U^T \kappa(\Omega, y) + \kappa(\Omega, \Omega)^T U \Sigma^{-1/2} \Sigma^{-1/2} U^T \kappa(\Omega, y) \|
\]

\[
= \| \Sigma^{-1/2} U^T \kappa(\Omega, y) - \Sigma^{-1/2} U^T \kappa(\Omega, x) \|_2^2
\]

(9)

Let \( \Sigma^{-1/2} U^T \kappa(\Omega, y) = \tilde{y}, \Sigma^{-1/2} U^T \kappa(\Omega, x) = \tilde{x} \), \( \Sigma \Sigma^T = \tilde{\Omega} \). \( \mathcal{K}(\Omega, \Omega) \) is a kernel matrix whose elements are \( \kappa(\omega_i, \omega_j) = \phi(\omega_i)^T \phi(\omega_j) \). \( \kappa(\Omega, \Omega) \) and \( \kappa(\Omega, \Omega) \) are kernel vectors.

From the above derivation, (8) can then be transformed into the following standard linear optimization problem

\[
\min_{\tilde{y}, \tilde{x}} \| \tilde{y} - \tilde{x} \|_2^2 + \| \tilde{\Omega} \tilde{x} \|_1
\]

(10)

As a consequence, we can use the linear dictionary learning algorithm to solve the original problem, such as analysis KSVD.

III. KERNEL DICTIONARY LEARNING ALGORITHM

In this section, we are no longer limited to just giving a transformation method, but give a complete nonlinear dictionary learning algorithm. We will first define and formulate the kernel dictionary learning problem. And then we present the kernel backward-greedy algorithm for analysis pursuit in the feature space. Finally, we will propose the analysis kernel KSVD algorithm for learning the analysis dictionary \( \Omega \) in detail.

A. PROBLEM FORMULATION

The definitions of the nonlinear mapping and kernel functions were introduced in the previous section and will not be repeated here. Therefore, we directly give the sparse representation based on nonlinear analysis model in space \( \mathcal{H} \):

\[
\min_{x, \Omega} \| \Phi(Y) - X \|_F^2 + \| \Omega x_i \|_0, \quad \forall i
\]

(11)

where \( \Phi(Y) = [\Phi(y_1), \ldots, \Phi(y_N)] \), \( X \in \mathbb{R}^{\tilde{n} \times N} \) is a matrix whose \( i \)-th column is the vector \( x_i \) corresponding to the sample \( \Phi(y_i) \), \( \Omega \in \mathbb{R}^{K \times \tilde{n}} \) is the sought analysis dictionary.

Since the dimension of feature space \( \mathcal{H} \) would be infinite, traditional optimization methods (such as MOD or KSVD) are not feasible for this kind of problem. In order to solve it, we will set \( X = DZ \) firstly, here \( D \in \mathbb{R}^{K \times \tilde{n}} \) and \( Z \in \mathbb{R}^{\tilde{n} \times N} \) is the coefficient matrix that we want to solve for. Then the following proposition will contribute the reformulation of (11).

**Proposition 1:** There exists an optimal solution \( D^* \) to (11) that has the following form

\[
D^* = \Phi(Y) A
\]

(12)

for some \( A \in \mathbb{R}^{N \times K} \).

**Proof.** Using standard orthogonal decomposition, we can display \( D^* \) as follows

\[
D^* = \tilde{D} + D_\perp
\]

(13)

where \( \tilde{D} = \Phi(Y) A \) and \( \Phi(Y)^T D_\perp = 0 \). The optimal cost is equivalent to

\[
\| \Phi(Y) - D^* Z^* \|_F^2 = \| \Phi(Y) - (\tilde{D} + D_\perp) Z^* \|_F^2
\]

\[
= \| \Phi(Y)(I - AZ^*) - D_\perp Z^* \|_F^2
\]

\[
= tr[\Phi(Y)(I - AZ^*)^T (I - AZ^*)]
\]

\[
= tr[I - AZ^*]^T \Phi(Y)^T \Phi(Y)(I - AZ^*)
\]

\[
+ tr[Z^*^T D^*_\perp D^*_\perp Z^* \]

(14)
The second term can be expressed as
\[
tr[(Z^*)^TD^\perp_iD_iZ^*] = \sum_{i=1}^{N} z_i^* D_i^T D_i z_i \geq 0 \quad (15)
\]
The inequality is because \( D^\perp_i D_i \) is a positive semidefinite matrix. Therefore, in order for the cost function to be optimal, the second term can only be 0. In short, \( D_\perp = 0 \) and \( D^* = D = \Phi(Y)A \) is an optimal solution.

As a result, we can substitute for \( X = \Phi(Y)AZ \), then the problem (11) can be rewritten as follows
\[
\min \limits_{Z, \Omega} \|\Phi(Y) - \Phi(Y)AZ\|_F^2 + \|\Omega Dz_i\|_0, \quad \forall i = 1, \ldots, N \quad (16)
\]
Here, we define \( A \) as a sparse matrix [36]. Note that the optimal problem (16) will be minimized through the use of Mercer kernels. To introduce kernel function more smoothly, we carry out some algebraic manipulations on the objective function based on the original, and the cost function can be rewritten as
\[
\|\Phi(Y) - \Phi(Y)AZ\|_F^2 = \|\Phi(Y)(I - AZ)\|_F^2 = tr((I - AZ)^T \Phi(Y)^T \Phi(Y)(I - AZ)) = tr((I - AZ)^T K(Y,Y)(I - AZ)) \quad (17)
\]
Denote
\[
\Omega Dz_i = K(\Omega^T, D)z_i = \Theta z_i, \quad i = 1, \ldots, N \quad (18)
\]
Note that in space \( \mathcal{H} \), employing the kernel trick avoids mapping data to high-dimensional feature spaces that require a lot of computation. Therefore, it is more efficient to optimize the problem (16) since it only involves kernel matrices of finite-dimensional, avoiding dealing with infinite dimensional matrices that might exist in the original formula (11). Specifically, we convert the problem of the analysis dictionary \( \Omega \) to its alternate \( \Theta \). In the following, we will focus on the learning of \( \Theta \) instead of \( \Omega \).

**B. KERNEL BACKWARD-GREEDY ALGORITHM (KBG Algorithm)**

In this stage, a task called analysis sparse coding or analysis pursuit will be considered, i.e., seeking for the coefficient matrix \( Z \) while fixing \( \Theta \). The formula (16) can be rewritten as
\[
\|\Phi(Y) - \Phi(Y)AZ\|_F^2 = \sum_{i=1}^{N} \|\Phi(y_i) - \Phi(Y)Az_i\|_2^2 \quad (19)
\]
so it can be solved by the following \( N \) different problems
\[
\min_{z_i, \Theta} \|\Phi(y_i) - \Phi(Y)Az_i\|_2^2 + \|\Theta z_i\|_0 \quad (20)
\]
Next we will incorporate the kernel trick into the backward-greedy algorithm to solve the above problem (20). Note that in the following we will omit the subscript \( i \) for simplification.

Since in the analysis sparse model, we mainly focus on the number of zero of \( \Theta z \), which is defined as co-sparsity \( \ell \)
\[
\|\Theta z\|_0 = K - \ell \quad (21)
\]
and meanwhile define the co-support \( \Lambda \) of \( z \) as the set of \( \ell = |\Lambda| \) rows that are orthogonal to it. In other words, \( \Theta_\Lambda z = 0 \), where \( \Theta_\Lambda \) is a sub-matrix of \( \Theta \) that only contains the rows indexed in \( \Lambda \). For a given dictionary \( \Theta \), we define the co-rank of \( z \) as \( K - r \) and co-support \( \Lambda \) as the rank of \( \Theta_\Lambda \), and we assume that \( r (r \ll K) \) represents the dimension of the subspace that \( z \) belongs to. Therefore, the problem that we need to solve is
\[
[\hat{z}, \hat{\Lambda}] = \text{Argmin}_{\hat{z}, \hat{\Lambda}} \|\Phi(y) - \Phi(Y)\hat{A}\hat{z}\|_2 \quad \text{s.t.} \quad \Theta_\Lambda \hat{z} = 0 \quad (22)
\]
or
\[
[\hat{z}, \hat{\Lambda}] = \text{Argmin}_{\hat{z}, \hat{\Lambda}} \text{Rank}(\Theta_\Lambda) \quad \text{s.t.} \quad \Theta_\Lambda \hat{z} = 0 \quad \|\Phi(y) - \Phi(Y)\hat{A}\hat{z}\|_2 \leq \epsilon \quad (23)
\]
Here, \( \epsilon \) represents a pre-determined minimum threshold. Note that if there is a correct correspondence between \( r \) and \( \epsilon \), both of problems (22) and (23) are equivalent, and the choice between them depends on the available information about the process of generating \( \Phi(y) \). Here, we refer to both of them as analysis sparse-coding or analysis pursuit problems.

In an oracle setup, the true co-support \( \Lambda \) is known, and \( z \) can be expressed initially as
\[
z = ((\Phi(Y)A)^T (\Phi(Y)A)^{-1}(\Phi(Y)A)^T \Phi(y) = (A^T K(Y,Y)A)^{-1}(K(Y,Y)A)^T \Phi(y))
\]
thus which can be used to obtain \( z \)
\[
\hat{z} = (I - \Theta_\Lambda^T \Theta_\Lambda)(A^T K(Y,Y)A)^{-1}(K(Y,Y)A)^T \quad (24)
\]
here \( I - \Theta_\Lambda^T \Theta_\Lambda \) is a projection matrix onto a \( r \)-dimensional space. The proposed KBG algorithm parallels to the synthesis greedy pursuit algorithms [37], however, in which the kernel trick is introduced for solving the infinite dimensional problems that synthesis one has no ability to deal with. The pseudo-code for KBG algorithm is given in the following. In practice, the proposed algorithm can be implemented efficiently by accumulating an orthogonalized set of the co-support rows, which means that once \( \hat{k}_j \) has been found and the row \( \theta_\ell \) is about to join the co-support.

**C. THE ANALYSIS KERNEL KSVD ALGORITHM**

In this part, we will give the main algorithm – the analysis kernel KSVD algorithm, which include the updates of coefficient matrix \( Z \) and the analysis dictionary \( \Theta \). Since we have introduced KBG algorithm for updating \( Z \), now we will turn to the update of \( \Theta \) in the following.

Firstly, we will denote some symbols beforehand. Suppose \( Z \) is known in advance. we will first learn \( \theta_p \) for \( p \) row in \( \Theta \), whose update should be only affected by the \( Z \) columns.
Algorithm 1 KBG Algorithm

Input: Signal y, a set of signals Y, analysis dictionary Θ, coefficient matrix A, target co-rank K − r and kernel function k.
Initialize: j = 0, z0 = (AT K(y, Y)A)−1(K(y, Y)A)T, Λ0 = ∅
For j = 1, . . . , K − r do
1) Sweep: ˆkj = Argmin{θT k zj−1}
2) Update Co-Support: Λj = Λj−1 ∪ ˆkj
3) Project:
   ˆzj = (J − Θ ⊤ ΛjΘΛj)(A⊤ K(y, Y)A)−1(K(y, Y)A)T
4) Refine Co-Support: Λj = {k|1 ≤ k ≤ K, |θT k ˆzj| < ϵ0}
end For
return ˆz = ˆzk−r
Output: Vector ˆz ∈ R with co-rank K − r and minimizing ∥Φ(y) − Φ(y)A∥2.

Algorithm 2 Analysis Kernel KSVD Algorithm

Input: A set of Signals Y, coefficient matrix A, kernel function k, analysis dictionary Θ, target co-rank K − r and the maximum iteration number MaxIter.
Initialize: Set Θ = Θ0, the number of iterations J = 1.
While J ≤ MaxIter, do
Stage 1: Analysis Pursuit

{zj, Λj} = Argmin zj,Λj ∥Φ(y) − Φ(y)A  zj ∥2 s.t. Θ (J−1) zj = 0
Rank(Θ (J−1)) = K − r (28)

Use KBG Algorithm to obtain the coefficient matrix Zj, and at the same time fixed Θ(J−1).
Stage 2: Dictionary Update
For p = 1, . . . , K, do
1) Extract relevant examples: P ⊆ indices of the columns of Zj orthogonal to θp−1.
2) Compute row:
   ˆθp := Argminθ∥θT ZP∥2 s.t. ∥θ∥2 = 1.
3) Update row: ΘJ(p − th row) := ˆθpT.
end For
Set J = J + 1.
end While
Output: Z and Θ.

which are orthogonal to it. Let P denote the indices of the columns of Z orthogonal to θp. Therefore, ZP can represent the sub-matrix of Z, which contains columns orthogonal to θp. Meanwhile, θi is used to denote the sub-matrix of Θ containing the rows from Θ that zi is currently orthogonal to, excluding the row of θp. The update step of θp can be given as follows

{θp, ˆzP} = Argminθp, ˆzP ∥Φ(Y) − Φ(Y)AZ∥2 F
s.t. ΘzP = 0, ∀i ∈ P
θT p ZP = 0, ∀p = 1, . . . , K
∥θp∥2 = 1 (26)

Here the normalization constraint on the rows of θp aims to avoid degeneracy, but has no any practical influence on the results. Since Z is known in advance, we will approximate the above optimization problem as

ˆθp = Argminθp ∥θT p ZP∥2 2 s.t. ∥θp∥2 = 1 (27)

Its solution is the singular vector corresponding to the smallest singular value of θp, which can be efficiently computed based on the SVD of θp or using some inverse methods. We all know that one advantage of this specific approximation method is that it disjoins the updates of the rows in Θ, enabling all rows to be updated in parallel. Another desirable feature of the resulting algorithm is that it adopts a structure similar to the KSVD algorithm, in which the minimum eigenvalue is calculated instead of the maximum eigenvalue. The proposed analysis kernel KSVD is described in the following.

IV. EXPERIMENTS
In this section, we will demonstrate the classification performances of the two algorithms we proposed above on some common datasets, and the reconstruction error will be taken as our classification index.

- KTM+ Analysis KSVD: We first use KTM to preprocess training data and testing data, and then the existing analysis KSVD algorithm is applied for dictionary learning. In this method, the reconstruction error is

rS = ∥ ˆy − zS ∥2 2 ∀S = 1, . . . , m (29)

- Analysis Kernel KSVD: The algorithm adopts a two-stage iterative method, including two stages of analysis pursuit and dictionary learning. And the reconstruction error of this algorithm is shown in the following equation

rS = ∥Φ(y) − Φ(Y)A1zS ∥2
= K(y, y) − 2K(y, Ys)A1zS
⋯ + zS T A1 K(Ys, Ys)A1zS, ∀S = 1, . . . , m (30)

where m is the number of classes. Note that the smallest reconstruction error determines which class the data belongs to. And we will present a set of experimental results based on the proposed methods, which can verify the core competence of the proposed methods and reveal the potential of the analysis sparse model in combination into the dictionary learning method. Specifically, we will firstly show the classification
results of USPS digital dataset, and give the classification results under the condition of data absence and noise. And then we will present classification results on the MNIST dataset and Extended YaleB (E-YaleB) dataset to validate the efficiency of the proposed methods.

A. USPS DATASET

The USPS dataset [38] contains 10 classes of 256-dimensional handwritten digits, i.e. \( n = 256 \). For each class, we randomly select \( N_{\text{train}} = 500 \) samples for training and \( N_{\text{test}} = 200 \) samples for testing. Specifically, we need to choose the appropriate parameters for different algorithms to learn the dictionaries of all classes. The co-sparsity \( r \) of the analysis kernel KSVD algorithm is 5, and for KTM, \( r = 7 \). The selection of other parameters is the same, that is, each class dictionary is learned with the dictionary atoms \( K = 200 \), the maximum number of training iterations is set to 20, and the degree of the kernel of the polynomial is 2. The construction error is calculated by (29) and (30) respectively.

In order to visualize the learned kernel dictionary atoms, we find images that lie on the input space and best approximate these atoms in terms of the mean square errors. Fig.1 shows the pre-images of the analysis kernel KSVD dictionary atoms for 10 digits. Next, we start to verify the classification performances of the proposed methods. In the first set of experiments, we give the results for the situation where the test samples are randomly removed with different percentages of image pixels (replaced by zero values), as shown in Fig.2(a). Similarly, Fig.2(b) shows the results of random gaussian noise with different standard deviations to the test samples. In these two simulations, the classification accuracy obtained by the analysis kernel KSVD algorithm proposed in this work is always better than other linear algorithms, moreover the gap between our method and other linear algorithms grows with the increase of distortion degree. We can also see that the trend of our results is similar to that of kernel KSVD and the classification accuracy is even slightly better than kernel KSVD. This shows that the analysis kernel KSVD algorithm has good robustness. For KTM, although the calculated results are much better than the traditional KSVD algorithm in the case of missing pixels, there is still a certain gap compared with other nonlinear kernel methods. However, in the case of noise interference, the classification results are poor. This shows that the robustness of this method is general and there are some limitations in practical application.

Note that in these experiments, some important parameters, such as the suitable polynomial degree, the optimal dictionary atom number and the co-sparsity, should be selected properly. Therefore, in the second set of experiments we will further investigate the influence of parameter selections on the overall recognition performance, thus indicating the reasons why choosing the above parameter values. Fig.3 shows the classification accuracy of KTM and the analysis kernel KSVD algorithm without interference while we vary the polynomial degree. Since the classification accuracy of KTM is poor at degree \( = 1 \), which is only 21.5%, this result is not considered in Fig.3 and is discussed from degree \( = 2 \). From it, we can see that, without external influence, the classification accuracies of the USPS dataset change with the increasing of the polynomial degree. When the degree is 2, the classification accuracies reach the highest value, and then decrease and fluctuate with the increase of the polynomial degree.

Next, we further study the effect of polynomial degree on classification accuracies under external interference. Due to the poor robustness of KTM, the classification accuracies under the condition of interference are not satisfactory, we only use the analysis kernel KSVD algorithm to do

![FIGURE 1. The pre-images of the analysis kernel KSVD dictionary atoms.](image1)

![FIGURE 2. Comparison of digit recognition accuracies for different methods in the presence of Gaussian noise and missing-pixel effects.](image2)

(a) Missing pixels. (b) Gaussian noise.
these experiments. Fig.4 shows the specific results of the experiments. The resulting trend of the broken line is basically the same as that in Fig.3. Therefore, we will fix the polynomial degree at 2 on the USPS dataset.

After determining the polynomial degree, we do the same experiments for the dictionary size and co-sparsity selections. Fig.5 shows the performance of the analysis kernel KSVD algorithms when the dictionary size changes within the range of 50 to 400. The sample dimension of the USPS dataset $n = 256$. If the number of dictionary atom $K > n$, then the dictionary is not redundant, which will make the classification accuracy worse. So for KMT, we limit the number of atoms in the dictionary to 50–250.

Fig.6 shows the specific results of the influence of the dictionary size on the classification accuracies of the analysis kernel KSVD algorithm under interference.

From it, we can see that, when the number of atoms in the dictionary reaches 200, the classification accuracy of KTM reaches the highest level, so we chose the optimal number of atoms in the dictionary at 200. For the analysis kernel KSVD algorithm, when the number of the dictionary atoms reaches 200, the classification accuracy tends to be relatively stable. And then it starts to drop a little bit after the number of atoms is greater than 300. Since the increase of the number of dictionary atoms will lead to the increased computational burden of the algorithm, we select the optimal number of dictionary atoms 200 while considering the balance between the classification accuracy and the operation time. From Fig.5 we can also see that when the dictionary size is 200, the best error rate obtained by the analysis kernel KSVD algorithm is 2.2%, and the error rate of KTM is only 3.2%, which devote to the beautiful results.

In addition, in Fig.7, we compare the performance of the two algorithms proposed in this paper when the co-sparsity varies from 3 to 15. For analysis kernel KSVD, the change of the co-sparsity has no significant influence on classification accuracy. When the co-sparsity is 5, the classification accuracy reaches its maximum value. After that, although there is a small fluctuation, the overall trend is decreasing. But for KTM, the influence of co-sparsity on classification accuracy is more obvious. The classification accuracy reaches the highest when the co-sparsity is 7, and is much higher than other cases.

Furthermore, Fig.8 shows the influence of the co-sparsity on classification accuracies under external interference. In the cases of large external interference, such as 70% pixel missing, the influence of the co-sparsity on classification accuracy fluctuates greatly. But the overall trend of the image is still similar to Fig.7, so we still set the co-sparsity of the analysis kernel KSVD algorithm for the experiment in this paper as 5.

In the following, in order to demonstrate the performance of the proposed methods, we compare them with some state-of-the-art methods, for example, SRC, FDDL [39], KSVD, LC-KSVD [40], and kernel PCA, kernel KSVD [22], LKDL [41], DLE [42]. We list the results in Table 1. From this table, we can see that the classification accuracy of analysis kernel KSVD algorithm is up to 97.8%, which is far superior to all other methods. The classification accuracy of KTM is 96.8%, which is second only to the analysis kernel KSVD algorithm.

In addition, in order to explain the efficiency of KTM, we further calculate the classification results on the USPS dataset based on the analysis KSVD algorithm only, whose classification accuracy rate is only 62.0%. Compared with the experimental result 96.8% using KTM combining
analysis KSVD, the classification accuracy is very low, which indicates that the KTM is an effective preprocessing transformation in improving the classification accuracy.

B. MNIST DATASET

For further demonstrating the performances of the proposed methods, in this part, we will conduct experiments on the MNIST dataset, which is also a handwritten dataset, but is larger and more complex than the USPS. The MNIST dataset is composed of $28 \times 28$ images of the 10 handwritten digits. There are 60,000 training images with 10,000 testing images in this benchmark. Just like the experiments in USPS, the digits is stacked in vectors of dimension $n = 784$. Fig.9 shows the sample images from MNIST.

Note that in these experiments, we also use (29) and (30) to calculate the reconstruction errors. By analyzing the experimental results on the USPS dataset, we can know that the polynomial degree $d$ and the co-sparsity $r$ have less influence on the classification accuracy, while the dictionary size has a greater influence. Therefore, in the experiments of MNIST dataset, we follow the values of $d$ and $r$ on USPS dataset experiments, and carry out further experiments on the selection of the dictionary size. Fig.10 shows the classification results of the two methods when the dictionary size varies from 200 to 400. Thus, we select the dictionary atomic number $K = 300$ for the analysis kernel KSVD algorithm. For KTM, the classification accuracy gradually flattens out with the increase of the number of dictionary atoms. As the increase of $K$ will greatly increase the operation time, we choose $K = 400$.

Furthermore, Table 2 gives the comparisons about the recognition accuracy of our algorithms with other common methods and the latest methods. We find that even for a large handwritten digital dataset, the analysis kernel KSVD algorithm has a higher amount of improvement for the classification accuracy. However, the result of KTM is not so optimistic, and its classification result is not outstanding among all algorithms. The reason we speculate may be that the error of the dataset with large dimension will increase after the kernel transformation.

Besides, we also do the same classification experiment on the MNIST dataset similar to USPS dataset, in which the classification accuracy rate of analysis KSVD is still very low, only 54.8%, which also proves that the KTM method’s effectiveness.

C. E-YaleB DATASET

In this part, we will use the E-YaleB dataset to verify the proposed methods further. The E-YaleB dataset contains 2,414 frontal face images of 38 subjects [43] captured under various laboratory-controlled lighting conditions. There are about 64 images for each person. The original images are cropped to $192 \times 168$ pixels. This dataset is challenging due to varying illumination conditions and expressions.
FIGURE 8. Comparison of digit recognition accuracies for different co-sparsity in the presence of Missing pixels and Gaussian noise. The polynomial degree is set to 2 and the dictionary size is set to 200 in for this experiments. (a) Missing pixels. (b) Gaussian noise.

TABLE 1. Comparison of recognition results on USPS dataset.

| Algorithms    | Accuracy |
|---------------|----------|
| SRC           | 91.2%    |
| FDDL [39]     | 96.3%    |
| KsVd          | 96.1%    |
| LC-KSVD [41]  | 95.8%    |
| kernel PCA    | 96.0%    |
| kernel KSVD   | 96.9%    |
| LKDL [41]     | 96.0%    |
| DLE [42]      | 93.2%    |
| analysis kernel KSVD | 97.8%  |
| KTM + analysis KSVD | 96.8%  |

FIGURE 9. Sample images from MNIST.

TABLE 2. Comparison of recognition results on MNIST dataset.

| Algorithms    | Accuracy |
|---------------|----------|
| SRC           | 90.3%    |
| FDDL [42]     | 93.7%    |
| KsVd          | 93.4%    |
| LC-KSVD [42]  | 89.6%    |
| kernel PCA    | 89.0%    |
| kernel KSVD   | 94.5%    |
| LKDL [42]     | 94.1%    |
| DLE [42]      | 94.7%    |
| analysis kernel KSVD | 96.5%  |
| KTM + analysis KSVD | 91.3%  |

FIGURE 10. Classification accuracy versus the dictionary size of the MNIST dataset.

Fig.11 shows some sample images from E-YaleB. Following the common setting, the random features from a 504 dimensional input domain will be exploited to evaluate the algorithms [44]. For this dataset, half of the samples per class will be randomly picked as the training samples while the rest are used as the testing samples. We still use the reconstruction errors (29) and (30) as the criteria for classifications in this experiment. For analysis kernel KSVD, let the co-sparsity r = 5, the dictionary atoms K = 250, and the polynomial degree d = 2. For KTM, we set the co-sparsity r = 7, the dictionary atoms K = 200, and the polynomial degree d = 2. The classification accuracies of all methods are summarized in Table 3. Besides the algorithms already mentioned in the above experiments, we also add five state-of-the-art algorithms for comparison, which are EasyDL [45], MCDL [46], IC-DDL [47], SADL [48] and K-LSDSR [49]. We find that the analysis kernel KSVD algorithm can achieve the 96.8% classification accuracy, which shows that our method is competitive among all compared methods. However, the result of KTM is as high as 98.3%, which exceeds all other algorithms, indicating that although this method is not very robust, it still owns some merits, such as the perfect classification accuracy in E-YaleB.
Just like the above mentioned two datasets, we also use the analysis KSVD algorithm for the classification experiment on the E-YaleB dataset, and the classification accuracy is 97.7%. Compared with the experimental result 98.3% using KTM, the classification accuracy is still slightly lower although it is already very high. Compared with USPS and MINST dataset, the performance of the analysis KSVD algorithm in E-YaleB has improved greatly, which may be caused by the advantage of analysis-based method, such as a very common structure in image processing.

V. CONCLUSION

In this work, we first propose a kernel sparse representation transformation method (KTM), which tactfully transforms nonlinear problems into a linear one. After this kind of transformation, the original nonlinear problem possesses the optimization model similar to the standard analysis sparse problem. Therefore, linear dictionary learning algorithms, such as the analysis KSVD algorithm, can be applied to solve the original problem. However, it is noting that all of the variables after KTM imply all nonlinear information of the original problem rather than linear. Maybe this is one of the reasons why the efficiency of the proposed algorithm is higher than the original analysis KSVD algorithm. Moreover, experimental results also indicate that exploiting nonlinear sparsity via learning dictionaries in a nonlinear feature space can provide better discrimination than the corresponding linear dictionary learning methods.

However, this method has some limitations, for example, it lacks robustness for the classification of the datasets with noise, therefore, another nonlinear dictionary learning algorithm is introduced, which is our main contribution, i.e., the analysis kernel KSVD. More specifically, first of all, the proposed algorithm adopts a two-stage iterative way for obtaining the analysis dictionary from a given dataset in a KSVD-like manner. However, due to the different concerns of the initial model — analysis sparse model, the aim of the proposed method is to find a suitable dictionary $\Omega$ so that the analysis coefficient $\mathbf{x}$ is sparse, which is other than the related synthesis-based methods essentially. Furthermore, the presented algorithm is also parallel to the analysis KSVD in its rationale and structure, for example, both of them are for learning the $l_0$ analysis dictionary, however, its root is in the nonlinear high-dimensional dataset processing problem rather than the linear problem. In more details, we first incorporate the appropriate kernel functions into the kernel backward-greedy algorithm for analysis pursuit and the analysis dictionary learning stage, then design the final analysis kernel KSVD algorithm to meet the challenges from nonlinear and high-dimensional data. It is noting that the introduction of kernel tricks makes the curse of dimensionality better settled, which possess higher adaptability for solving nonlinear signal problems than original analysis KSVD algorithm. Benchmark experimental results on three well-known datasets, USPS, MNIST and E-YaleB, show that the proposed method outperforms some related linear algorithms and other existing nonlinear dictionary learning algorithms. Moreover, its robustness is further proved when the data is interfered by noise or some pixels are absent from data, which makes up for the lack of robustness in the kernel transformation method.

Therefore the proposed nonlinear kernel dictionary learning algorithms’ superiority on datasets validates its effectiveness to a larger extent. Meanwhile, its superiority also verifies its theoretical advantages owing to analysis sparse model’s merits of the equality of all atoms and the much smaller dimensionality for signal representation to some extent.

ACKNOWLEDGMENT

Thank all the referees and the editorial board members for their insightful comments and suggestions, which improved this article significantly.

REFERENCES

[1] R. Rubinstein, A. M. Bruckstein, and M. Elad, “Dictionaries for sparse representation modeling,” Proc. IEEE, vol. 98, no. 6, pp. 1045–1057, Jun. 2010.
[2] S. Wang and W. Guo, “Sparse multigraph embedding for multimodal feature representation,” IEEE Trans. Multimed., vol. 19, no. 7, pp. 1454–1466, Jul. 2017.
[3] M. Aharon, M. Elad, and A. Bruckstein, “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation,” IEEE Trans. Signal Process., vol. 54, no. 11, pp. 4311–4322, Nov. 2006.
[4] G. Yu, G. Sapiro, and S. Mallat, “Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity,” IEEE Trans. Signal Process., vol. 21, no. 5, pp. 2481–2499, May 2012.
[5] V. Cevher, P. Indyk, L. Carin, and R. Baraniuk, “Sparse signal acquisition and recovery with graphical models,” IEEE Signal Process. Mag., vol. 26, no. 2, pp. 92–103, Nov. 2010.
[6] M. B. Mashhadi, M. Fallah, and F. Marvasti, “Interpolation of sparse graph signals by sequential adaptive thresholds,” in Proc. Int. Conf. Sampling Theory Appl. (SampTA), Jul. 2017, pp. 266–270.
[7] S. Wang and W. Guo, “Sparse multigraph embedding for multimodal feature representation,” IEEE Trans. Multimed., vol. 19, no. 7, pp. 1454–1466, Jul. 2017.
[8] R. Rubinstein, T. Peleg, and M. Elad, “Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model,” IEEE Trans. Signal Process., vol. 61, no. 3, pp. 661–677, Feb. 2013.
[9] K.-K. Huang, D.-Q. Dai, C.-X. Ren, and Z.-R. Lai, “Learning kernel extended dictionary for face recognition,” IEEE Trans. Neural Netw. Learn. Syst., vol. 28, no. 5, pp. 1082–1094, May 2017.
[10] Y. Chen and J. Su, “Sparse embedded dictionary learning on face recognition,” Pattern Recognit., vol. 64, pp. 51–59, Apr. 2017.
[11] X. Shu, J. Tang, Z. Li, H. Lai, L. Zhang, and S. Yan, “Personalized age progression with bi-level aging dictionary learning,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 4, pp. 905–917, Apr. 2018.
[12] T. H. Vu and V. Monga, “Fast low-rank shared dictionary learning for image classification,” IEEE Trans. Image Process., vol. 26, no. 11, pp. 5160–5175, Nov. 2017.
[13] Z. Wang, Y. Wang, H. Liu, and H. Zhang, “Structured kernel dictionary learning with correlation constraint for object recognition,” IEEE Trans. Image Process., vol. 26, no. 9, pp. 4578–4590, Sep. 2017.

[14] S. Banerjee and A. Chatterjee, “Image set based ear recognition using novel dictionary learning and classification scheme,” Eng. Appl. Artif. Intell., vol. 55, pp. 37–46, Oct. 2016.

[15] J. Mairal, F. Bach, and J. Ponce, “Task-driven dictionary learning,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 4, pp. 791–804, Apr. 2012.

[16] Y. Huang, Y. Quan, T. Liu, and Y. Xu, “Exploiting label consistency in structured sparse representation for classification,” Neural Comput. Appl., vol. 31, pp. 6509–6520, Apr. 2018.

[17] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online dictionary learning for sparse coding,” in Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML), Jun. 2009, pp. 689–696.

[18] J. Wright, A. Y. Yang, A. Ganesh, S. Shankar Sastry, and Y. Ma, “Robust face recognition via sparse representation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 2, pp. 210–227, Feb. 2009.

[19] H. Jiang, Q. Tian, J. Farrell, and B. A. Wandell, “Learning the image processing pipeline,” IEEE Trans. Image Process., vol. 26, no. 10, pp. 5032–5042, Oct. 2017.

[20] Y. T. Hoang and T. Van, “Artificial neural network algorithm for acoustic echo cancellation applications,” IRJET, vol. 5, no. 10, pp. 344–349, Oct. 2018.

[21] B. Song, P. Li, J. Li, and A. Plaza, “One-class classification of remote sensing images using kernel sparse representation,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 9, no. 4, pp. 1613–1623, Apr. 2016.

[22] H. Van Nguyen, V. M. Patel, N. M. Nasrabadhi, and R. Chellappa, “Design of non-flat linear kernel dictionaries for object recognition,” IEEE Trans. Image Process., vol. 22, no. 12, pp. 5123–5135, Dec. 2013.

[23] Z. Chen, W. Zuo, Q. Hu, and L. Lin, “Kernel sparse representation for time series classification,” Inf. Sci., vol. 292, pp. 15–26, Jan. 2015.

[24] Y. Quan, C. Bao, and H. Ji, “Equiangular kernel dictionary learning with applications to dynamic texture analysis,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 308–316.

[25] A. Shirivastava, V. M. Patel, and R. Chellappa, “Non-linear dictionary learning with partially labeled data,” Pattern Recognit., vol. 48, no. 11, pp. 3283–3292, Nov. 2015.

[26] C.-H. Chen, V. M. Patel, and R. Chellappa, “Learning from ambiguously labeled face images,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 7, pp. 1653–1667, Jul. 2018.

[27] M. Elad, P. Milanfar, and R. Rubinstein, “Analysis versus synthesis in signal priors,” Inverse problems, vol. 23, no. 5, pp. 947–968, Apr. 2007.

[28] Y. Xu, Z. Li, J. Yang, and D. Zhang, “A survey of dictionary learning algorithms for face recognition,” IEEE Access, vol. 5, pp. 8502–8514, Apr. 2017.

[29] Y. Zhang, T. Yu, and W. Wang, “An analysis dictionary learning algorithm under a noisy data model with orthogonality constraint,” Sci. World J., vol. 2014, Jul. 2014, Art. no. 852978.

[30] J. Guo, Y. Guo, X. Kong, M. Zhang, and R. He, “Discriminative analysis learning,” in Proc. 30th AAAI Conf., Feb. 2016, pp. 1617–1623.

[31] Z. Zhang, W. Jiang, J. Qin, L. Zhang, F. Li, M. Zhang, and S. Yan, “Jointly learning structured analysis discriminative dictionary and analysis multiclass classifier,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 8, pp. 3798–3814, Aug. 2018.

[32] M. Yaghoobi, S. Nam, R. Gribonval, and M. E. Davies, “Constrained overcomplete analysis operator learning for copsparse signal modelling,” IEEE Trans. Signal Process., vol. 61, no. 9, pp. 2341–2355, May 2013.

[33] E. M. Eksioglu and O. Bayir, “Overcomplete sparsifying transform learning algorithm using a constrained least squares approach,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2014, pp. 7158–7162.

[34] S. Ravishankar and Y. Bresler, “Learning sparsifying transforms,” IEEE Trans. Signal Process., vol. 61, no. 5, pp. 1072–1086, Mar. 2013.

[35] J. Yin, Z. Liu, Z. Jin, and W. Yang, “Kernel sparse representation based classification,” Neurocomputing, vol. 77, no. 1, pp. 120–128, Feb. 2012.

[36] R. Rubinstein, M. Zibulevsky, and M. Elad, “Double sparsity: Learning sparse dictionaries for sparse signal approximation,” IEEE Trans. Signal Process., vol. 58, no. 3, pp. 1553–1564, Mar. 2010.

[37] J. A. Tropp, “Greed is good: Algorithmic results for sparse approximation,” IEEE Trans. Inf. Theory, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

[38] J. J. Hull, “A database for handwritten text recognition research,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 16, no. 5, pp. 550–554, May 1994.

[39] M. Yang, L. Zhang, X. Feng, and D. Zhang, “Fisher discrimination dictionary learning for sparse representation,” in Proc. Int. Conf. Comput. Vis., Nov. 2011, pp. 543–550.

[40] Z. Jiang, Z. Lin, and L. S. Davis, “Learning a discriminative dictionary for sparse coding via label consistent K-SVD,” in Proc. CVPR, Jun. 2011, pp. 1697–1704.

[41] A. Golts and M. Elad, “Linearized kernel dictionary learning,” IEEE J. Sel. Topics Signal Process., vol. 10, no. 4, pp. 726–739, Jun. 2016.

[42] A. Abdli, M. Rahmati, and M. M. Ebadzadeh, “Dictionary learning enhancement framework: Learning a non-linear mapping model to enhance discriminative dictionary learning methods,” Neurocomputing, vol. 357, pp. 135–150, Sep. 2019.

[43] A. S. Georgiades, P. N. Belhumeur, and D. J. Kriegman, “From few to many: Illumination cone models for face recognition under variable lighting and pose,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 6, pp. 643–660, Jun. 2001.

[44] Z. Jiang, Z. Lin, and L. S. Davis, “Label consistent K-SVD: Learning a discriminative dictionary for recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 11, pp. 2651–2664, Nov. 2013.

[45] Y. Quan, Y. Xu, Y. Sun, Y. Huang, and H. Ji, “Sparse coding for classification via discrimination ensemble,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 5839–5847.

[46] Y. Quan, Y. Xu, Y. Sun, and Y. Huang, “Supervised dictionary learning with multiple classifier integration,” Pattern Recognit., vol. 55, pp. 247–260, Jul. 2016.

[47] Z. Li, Z. Zhang, Z. Fan, and J. Wen, “An interactively constrained discriminative dictionary learning algorithm for image classification,” Eng. Appl. Artif. Intell., vol. 72, pp. 241–252, Jun. 2018.

[48] W. Tang, A. Panahi, H. Krim, and L. Dai, “Analysis dictionary learning based classification: Structure for robustness,” IEEE Trans. Image Process., vol. 28, no. 12, pp. 6035–6046, Dec. 2019.

[49] B. Benoua, B. Ghansah, and E. K. Ansah, “Kernel based locality–Sensitive discriminative sparse representation for face recognition,” Sci. Afr., vol. 7, Mar. 2020, Art. no. e00249.

ZHOUYUN MA received the B.S. degree in mathematics from Shanghai Maritime University, China, in 2018. She is currently pursuing the degree with the Department of Mathematics, Shanghai University. Her research interests include sparse representation and image classification.

HONGJIAN ZHANG received the M.S. degree from the Department of Applied Mathematics, Dalian University of Technology, Dalian, China, in 2005, and the Ph.D. degree in mathematics from the Dalian University of Technology, in 2009. In 2012, she was a Visiting Scholar at the University of Aizu, Japan. From 2016 to 2017, she visited Rochester University, USA, where she engaged in research in music data’s separation problem. She is currently an Associate Professor with the Department of Mathematics, Shanghai University. Her research interests include blind signal processing, pattern recognition, and systems optimization.

SHUANG MA received the B.S. degree in mathematics from Shandong Agricultural University, China, in 2018. She is currently pursuing the degree with the Department of Mathematics, Shanghai University. Her research interests include blind signal processing and pattern recognition.