JOINT KERNEL DICTIONARY AND CLASSIFIER LEARNING FOR SPARSE CODING VIA LOCALITY PRESERVING K-SVD

Weiyang Liu[†], *Zhiding Yu*[‡], *Meng Yang*[§*], *Lijia Lu*[†], *and Yuexian Zou*[†]

[†]School of ECE, Peking University [‡]Dept. of ECE, Carnegie Mellon University
[§]College of Computer Science & Software Engineering, Shenzhen University

ABSTRACT

We present a locality preserving K-SVD (LP-KSVD) algorithm for joint dictionary and classifier learning, and further incorporate kernel into our framework. In LP-KSVD, we construct a locality preserving term based on the relations between input samples and dictionary atoms, and introduce the locality via nearest neighborhood to enforce the locality of representation. Motivated by the fact that locality-related methods works better in a more discriminative and separable space, we map the original feature space to the kernel space, where samples of different classes become more separable. Experimental results show the proposed approach has strong discrimination power and is comparable or outperforms some state-of-the-art approaches on public databases.

Index Terms— Discriminative Dictionary Learning, Locality Preserving K-SVD, Kernel Space.

1. INTRODUCTION

Sparse coding has served an important role in a wide variety of vision problems, ranging from image restoration [1], image denoising [2] to image classification [3], etc. The technique linearly represents a query image $\boldsymbol{y}$ by a few atoms from a dictionary $\boldsymbol{D}$, i.e., $\boldsymbol{y}=\boldsymbol{D}\boldsymbol{x}$ or $\boldsymbol{y}\approx\boldsymbol{D}\boldsymbol{x}$, where $\boldsymbol{x}$ is sparse coefficients. In image classification, one classifies a query based on the sparse codes. Thus, the discrimination power of the dictionary is of considerable importance. Wright et al. [3] employ the entire set of training samples as the dictionary and achieve promising results on face recognition. However, the method is time-consuming, since it needs to solve the l_1 minimization problem, which reduces its scalability for large scale databases. Alternatively, some off-the-shelf bases (e.g. Fourier, wavelets) can also be used as dictionaries [4], yet these dictionaries might not be the optimal choices in certain tasks like image classification or face recognition. Learning the desired dictionary from training samples has been reported to bring additional performance gain to classification tasks [5–11]. Dictionary learning methods boosts the performance from perspective of coding coefficients, dictionary, or both. Group sparse coding [9, 10] regularizes the coding coefficients to be similar within the same class. Graph-regularized sparse coding [11] combines geometrical information into coding coefficients and dictionary in order to preserve the local manifold structure. Moreover, dictionaries with labels are learned to exploit the class-specific representation residual for classification [3, 7, 8]. For instance, [8] learns a structured dictionary via Fisher criteria and enforce coding coefficients to have small within-class scatter but big between-class scatter.

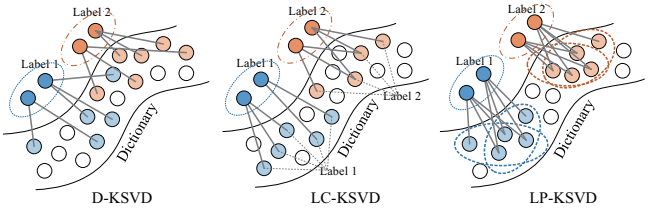

Fig. 1. Intuitive comparison among D-KSVD, LC-KSVD and LP-KSVD. The selected atoms for representation are highlighted.

Of great interest is a series of recently proposed dictionary learning approaches [5, 6, 12] via K-SVD algorithm [13]. Since K-SVD algorithm focuses on the representation power without considering the discrimination power, [12] proposes to jointly learn a classifier based on coding coefficients for classification tasks. Pham et al. [12] combine the classification error of the linear classifier, the representation error of data as well as the regularization terms into the objective function, minimizing it by iteratively updating the variables while preserving the sparsity. However, this approach involves multiple additional optimizations, which is inefficient and easy to be trapped into local minima. To address this, discriminative K-SVD (D-KSVD) [5] eliminates the representation error term of unlabeled data and formulates the objective function into the K-SVD framework. In order to further enhance the discrimination power of the learned dictionary, label consistent K-SVD (LC-KSVD) [6] constructs a label consistent term and applies similar method to formulate the optimization into the K-SVD framework. The label consistent term supervises each class has similar sparse codes, exhibiting discrimination.

Inspired by recent progress obtained via enforcing locality [14, 15], we propose the locality preserving K-SVD (LP-

Corresponding Author {yang.meng@szu.edu.cn;wyliu@pku.edu.cn}.
This work is partially supported by the National Natural Science Foundation of China under Grant No. 61402289, and Shenzhen Scientific Research and Development Funding Program under Grant JCYJ20140509172609171.

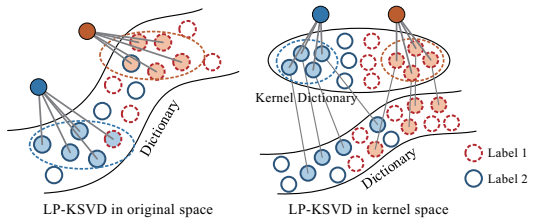

Fig. 2. Comparison between LP-KSVD in original space and LP-KSVD in kernel space. For LP-KSVD, the label information (label) in dictionary is not necessarily a prior. We show the label just to imply the hidden unsupervised label consistency when applying LP-KSVD in kernel space.

KSVD) approach in which dictionary and classifier are jointly learned similar to [5, 6]. Specifically, we first construct a penalty term that enforces the representation by local dictionary atoms similar to the input sample and penalises the representation by nonlocal (dissimilar) atoms. The classification error term, representation error term and locality penalty term are then jointly optimized by the K-SVD algorithm. An intuitive comparison among D-KSVD, LC-KSVD and LP-KSVD is provided in Fig. 1. We can see that the locality-preserving term could keep the relationship between input samples and dictionary atoms in dictionary learning, making the nearby samples (usually with the same label) represented by similar dictionary atoms. In fact, Laplacian sparse coding [16] also uses the local information to construct graph Laplacian for feature quantization. It preserves the consistence in sparse representation of similar local features with graph Laplacian and apply one-vs-all SVM as classifier, while we enforce locality into coding coefficients and dictionary as suggested in [14] and jointly optimize a classifier via K-SVD.

The label consistent term in LC-KSVD serves a similar role to the classification error term. They both penalize the dictionary representation from classes different from the input class. On the other hand, different from classification error, the locality preserving term incorporates the neighborhood information into dictionary and coding coefficients, thus retaining more crucial information. We believe the locality preserving term is endowed with more discriminative power than label consistent term. The representation obtained via the proposed dictionary learning can be further enhanced by exploiting the nonlinear structure within the data [17]. [18] also shows the benefit of representation power from kernel dictionaries in classification tasks. Moreover, the same benefit is reported in [19, 20]. Different from [18], however, we specifically aim at the image classification by jointly learning the optimal linear classifier and simultaneously taking locality into consideration. Further, mapping the original feature space into kernel space implies more separable representation between different classes can be achieved as Fig. 2 illustrates, since the locality preserving term is imposed in kernel space.

2. PRELIMINARIES

Given an input image set matrix $\boldsymbol{Y}=\{\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_k\}\in\mathbb{R}^{n\times k}$ in which each column vector $\boldsymbol{y}_i$ represents a n-dimensional input sample, we let $\boldsymbol{D}=\{\boldsymbol{d}_1,\boldsymbol{d}_1,\cdots,\boldsymbol{d}_m\}\in\mathbb{R}^{n\times m}$ be the dictionary that is composed of m training samples, and $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_k\}\in\mathbb{R}^{m\times k}$ be the coefficient matrix where each column is a sparse code for an input sample. Learning a dictionary to sparsely represent $\boldsymbol{Y}$ can be accomplished by solving the minimization problem:

$$\langle \boldsymbol{D},\boldsymbol{X}\rangle = \arg\min_{\boldsymbol{D},\boldsymbol{X}} \|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}\|_2^2 \quad \text{s.t. } \forall i, \|\boldsymbol{x}_i\|_0 \le c \tag{1}$$

where $\|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}\|_2^2$ denotes the representation error, and c is a sparsity constraint parameter which enforces each sparse code should have fewer than c nonzero values. This objective function is exactly what K-SVD algorithm optimizes with, and it emphasizes more on representation power instead of discrimination power. Therefore, [5, 6] jointly learn a classifier at the same time and combine the classification error in the objective function. Towards this end, a classifier $g(\boldsymbol{x},\boldsymbol{W})$ is constructed with parameter matrix $\boldsymbol{W}\in\mathbb{R}^{\times m}$. An optimal classifier should satisfy

$$\boldsymbol{W} = \arg\min_{\boldsymbol{W}} \sum_i \mathcal{L}\{l_i, g(\boldsymbol{x}_i,\boldsymbol{W})\} + \lambda\|\boldsymbol{W}\|_F^2 \tag{2}$$

where $\mathcal{L}$ is the classification loss function that is usually defined as logistic loss function or quadratic loss function, l_i is the label of y_i, and λ is a regularization parameter. Combining Eq.(1) and Eq.(2) can jointly learn the dictionary and classifier as in [5, 6, 12] by defining objective function:

$$\begin{aligned}\langle \boldsymbol{D},\boldsymbol{W},\boldsymbol{X}\rangle = \arg &\min_{\boldsymbol{D},\boldsymbol{W},\boldsymbol{X}} \\ &\|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}\|_2^2 + \sum_i \mathcal{L}\{l_i, g(\boldsymbol{x}_i,\boldsymbol{W})\} + \lambda\|\boldsymbol{W}\|_F^2. \\ &\text{s.t. } \forall i, \|\boldsymbol{x}_i\|_0 \le c\end{aligned} \tag{3}$$

When using the l_2 norm as the loss function, the case becomes [12] without the unlabeled data therein, and it is exactly the same case in D-KSVD [5]. For LC-KSVD [6], it further enforces a label consistent term into the objective function, making the coding coefficients concentrate on its own class.

3. KERNEL DICTIONARY AND CLASSIFIER LEARNING VIA LOCALITY PRESERVING K-SVD

3.1. Locality Preserving K-SVD

As suggested in locality-constrained linear coding (LLC) [14] and local coordinate coding (LCC) [15], locality plays a significant role in classification. We therefore incorporate locality preserving constraint into the objective function:

$$\begin{aligned}\langle \boldsymbol{D},\boldsymbol{W},\boldsymbol{T},\boldsymbol{X}\rangle = \arg &\min_{\boldsymbol{D},\boldsymbol{W},\boldsymbol{T},\boldsymbol{X}} \\ &\|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}\|_2^2 + \alpha\|\boldsymbol{H}-\boldsymbol{W}\boldsymbol{X}\|_2^2 \\ &\qquad + \beta\|\boldsymbol{P}-\boldsymbol{T}\boldsymbol{X}\|_2^2 + \lambda\|\boldsymbol{W}\|_F^2 \\ &\text{s.t. } \forall i, \|\boldsymbol{x}_i\|_0 \le c\end{aligned} \tag{4}$$

where α, β are scaling parameters that control the contribution of classification error term and locality preserving term respectively, and $\|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}\|_2^2$ and $\|\boldsymbol{H}-\boldsymbol{W}\boldsymbol{X}\|_2^2$ are representation error term and classification error term respectively. $\boldsymbol{H}=\{\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_k\} \in \mathbb{R}^{v \times k}$ denotes the class labels of the input images $\boldsymbol{Y}$. Each column of $\boldsymbol{H}$ is a label vector, formulated as $\boldsymbol{h}_i=\{0, 0, \cdots, 1, \cdots, 0, 0\}$ in which 1 indicates the positive label in the according category. $\boldsymbol{W}$ denotes a set of parameters for a linear classifier. Most importantly, locality preserving term $\|\boldsymbol{P}-\boldsymbol{T}\boldsymbol{X}\|_2^2$ is added to penalize the representation which is not locally concentrated. To design $\boldsymbol{P}$, we first calculate the q_i nearest neighbors of each input image $\boldsymbol{y}_i$ in dictionary, and then use $\omega(q, \boldsymbol{y}_i)$ to denote the positions of these q nearest neighbors in dictionary, where nonzero elements stand for the positions of nearest neighbors in dictionary. For example, in $\omega(q, \boldsymbol{y}_i)=\{1, 0, \cdots, 0, 1, 0, \cdots, 1, 1\}^T$, 1 represents the positions of nearest neighbors. Therefore, we constuct the matrix $\boldsymbol{P}$ via

$$\boldsymbol{P}=\left\{\omega(q, \boldsymbol{y}_1), \omega(q, \boldsymbol{y}_2), \cdots, \omega(q, \boldsymbol{y}_k)\right\}. \tag{5}$$

Consider an example that $\boldsymbol{D}$ has 4 samples from 2 categories, each category with 2 samples. Given $\boldsymbol{Y}$ of 4 input samples, 2-nearest neighbors of $\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3$ and $\boldsymbol{y}_4$ are $(\boldsymbol{d}_1, \boldsymbol{d}_3), (\boldsymbol{d}_1, \boldsymbol{d}_2), (\boldsymbol{d}_3, \boldsymbol{d}_4)$ and $(\boldsymbol{d}_2, \boldsymbol{d}_4)$ respectively. Then $\boldsymbol{P}$ is constructed as

$$\boldsymbol{P}=\begin{Bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{Bmatrix}. \tag{6}$$

However, the desired $\boldsymbol{D}$ is not given in the first place, so we optimize a $\boldsymbol{D}$ with the locality preserving term removed and use it to initialize $\boldsymbol{P}$. $\boldsymbol{P}$ contains all the locality information for the input samples, which is essential to preserve locality. $\boldsymbol{T}$ is a linear transformation matrix, which transforms the original sparse codes $\boldsymbol{x}$ to be more locally concentrated. Particularly, we construct the locality term in a similar form to the classification error so as to conveniently formulate the objective function for K-SVD algorithm. Furthermore, we believe the representation codes should still be sparse since locality must lead to sparsity but not necessary vice versa.

In order to apply the K-SVD algorithm to efficiently find the optimal solution, we drop the regularization term and further reformulate the objective function in Eq.(4) as

$$\begin{aligned} \langle \boldsymbol{D}, \boldsymbol{W}, \boldsymbol{T}, \boldsymbol{X} \rangle &= \arg \min_{\boldsymbol{D}, \boldsymbol{W}, \boldsymbol{T}, \boldsymbol{X}} \\ &\left\| \begin{Bmatrix} \boldsymbol{Y} \\ \alpha^{\frac{1}{2}} \boldsymbol{H} \\ \beta^{\frac{1}{2}} \boldsymbol{P} \end{Bmatrix} - \begin{Bmatrix} \boldsymbol{D} \\ \alpha^{\frac{1}{2}} \boldsymbol{W} \\ \beta^{\frac{1}{2}} \boldsymbol{T} \end{Bmatrix} \boldsymbol{X} \right\|_2^2 \\ &\text{s.t. } \forall i, \|\boldsymbol{x}_i\|_0 \leq c \end{aligned} \tag{7}$$

which is the generalized form optimized in [13]. We can directly use the K-SVD algorithm to solve the problem.

3.2. Objective Function

As formulated in Eq.(4), the objective function for dictionary learning contains four penalty terms, which are the representation error $\|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}\|_2^2$, classification error $\|\boldsymbol{H}-\boldsymbol{W}\boldsymbol{X}\|_2^2$, locality preserving term $\|\boldsymbol{P}-\boldsymbol{T}\boldsymbol{X}\|_2^2$ and parameter regularizer $\|\boldsymbol{W}\|_F^2$ respectively.

Representation error: It is a fundamental constraint in sparse coding, which is used to keep the estimation close to the input sample in order to keep the representation power.

Classification error: As the main contribution to the discrimination power, this constraint penalize misclassifications. It makes sparse codes more discriminative and also learns a linear classifier for classification in a supervised way, avoiding to use dictionary classifier [3], since K-SVD algorithm may disable such classifier.

Locality preserving term: Motivated by the fact that the samples in the same class usually have similar local dictionary atoms, we construct this term to preserve the locality of representation and bring more discrimination power. The distance metric used to find neighbors of input samples is flexible, but we use simple l_1 distance in experiments. The direct intuition of locality preserving term is that the sparse codes become locally concentrated after a linear invertible transformation. In fact, the locality information is preserved within sparse codes via such constraint. We let $\boldsymbol{X}^N=\boldsymbol{T}\boldsymbol{X}$ and further rewrite $\boldsymbol{D}^N=\boldsymbol{D}\boldsymbol{T}^{-1}$, $\boldsymbol{W}^N=\boldsymbol{W}\boldsymbol{T}^{-1}$. Thus the penalty terms become $\|\boldsymbol{Y}-\boldsymbol{D}^N\boldsymbol{X}^N\|_2^2$, $\|\boldsymbol{H}-\boldsymbol{W}^N\boldsymbol{X}^N\|_2^2$ and $\|\boldsymbol{P}-\boldsymbol{X}^N\|_2^2$ respectively. From this formulation, we can clearly see the representation error and classification error stay the same form as in Eq.(4), while the locality preserving term make the sparse codes represented by local atoms.

3.3. Learning in Kernel Space

We can easily generalize a linear classifier into a nonlinear one via kernel function [21, 22], enhancing the discrimination power of the original linear classifier. To this end, we use kernel function to create a nonlinear mapping mechanism $\boldsymbol{y} \in \mathbb{R} \mapsto \phi(\boldsymbol{y}) \in \mathbb{H}$ where $\mathbb{H}$ is a unique associated reproducing kernel Hilbert space (RKHS). We map every sample into higher dimensional kernel space via transformation ϕ and use the kernel function $K(\boldsymbol{y}', \boldsymbol{y}'')=\phi(\boldsymbol{y}')^T\phi(\boldsymbol{y}'')$, where $\boldsymbol{y}', \boldsymbol{y}''$ are different samples and ϕ denotes the implicit nonlinear mapping associated with the kernel function $K(\boldsymbol{y}', \boldsymbol{y}'')$. The kernel dictionary is written as

$$\phi(\boldsymbol{D})=\left\{\phi(\boldsymbol{d}_1), \phi(\boldsymbol{d}_2), \cdots, \phi(\boldsymbol{d}_m)\right\} \in \mathbb{R}^{s \times m} \tag{8}$$

where s is the dimension of the kernel space that is much larger than the dimension of the original feature space. Similarly, the input image set $\boldsymbol{Y}$ is kernelized to $\phi(\boldsymbol{Y})$. Thus, the representation error term is rewritten as $\|\phi(\boldsymbol{Y})-\phi(\boldsymbol{D})\boldsymbol{X}\|$.

However, the high dimensional kernel space may lead to high computational complexity, so we will perform dimen-

sionality reduction in kernel space, similar to [19,20]. Specifically, the projection matrix $\boldsymbol{R}\in\mathbb{R}^{s\times u}(u<s)$ is constructed following the similar approach in KPCA [22]. Combining the projection matrix $\boldsymbol{R}$, we derive the new representation error term $\|\boldsymbol{R}^T\phi(\boldsymbol{Y})-\boldsymbol{R}^T\phi(\boldsymbol{D})\boldsymbol{X}\|$. Further, we assume that each column vector in $\boldsymbol{R}$ is a linear combination of samples in kernel space, and decompose $\boldsymbol{R}$ as

$$\begin{aligned}\boldsymbol{R}&=\{\boldsymbol{R}_1,\cdots,\boldsymbol{R}_s\}=\phi(\boldsymbol{D})\cdot\boldsymbol{\Psi}\\&=\{\phi(\boldsymbol{d}_1),\cdots,\phi(\boldsymbol{d}_m)\}\cdot\{\boldsymbol{\psi}_1,\cdots,\boldsymbol{\psi}_u\}\end{aligned}\tag{9}$$

in which $\boldsymbol{\psi}_i$ is a m-dimensional column vector that is also linear projection coefficients of columns in $\phi(\boldsymbol{D})$, satisfying $\boldsymbol{R}_i=\phi(\boldsymbol{D})\cdot\boldsymbol{\psi}_i$. Moreover, $\boldsymbol{\Psi}$ is also called pseudo-transformation matrix [19]. Putting Eq.(9) into the representation error, we can derive $\|\boldsymbol{\Psi}^TK(\boldsymbol{D},\boldsymbol{Y})-\boldsymbol{\Psi}^T\boldsymbol{G}\boldsymbol{X}\|$ where

$$\begin{aligned}K(\boldsymbol{D},\boldsymbol{Y})&=\{K(\boldsymbol{D},\boldsymbol{y}_1),\cdots,K(\boldsymbol{D},\boldsymbol{y}_k)\}\\&=\begin{Bmatrix}K(\boldsymbol{d}_1,\boldsymbol{y}_1)&\cdots&K(\boldsymbol{d}_1,\boldsymbol{y}_k)\\\vdots&\ddots&\vdots\\K(\boldsymbol{d}_m,\boldsymbol{y}_1)&\cdots&K(\boldsymbol{d}_m,\boldsymbol{y}_k)\end{Bmatrix},\end{aligned}\tag{10}$$

and $\boldsymbol{G}$ ($\boldsymbol{G}_{ij}=K(\boldsymbol{d}_i,\boldsymbol{d}_i)$), also equal to $\phi(\boldsymbol{D})^T\cdot\phi(\boldsymbol{D})$. Particularly, $\boldsymbol{G}$ is defined as the kernel gram matrix that should be symmetric and positive semi-definite based on Mercer's theorem. Since $\boldsymbol{G}$ and $K(\boldsymbol{D},\boldsymbol{Y})$ are known as a prior, dimensionality reduction now focuses on finding $\boldsymbol{\Psi}$ instead of $\boldsymbol{R}$. Various methods to find $\boldsymbol{\Psi}$ are provided in [23] [1] . Therefore, the kernel dictionary learning can be formulated as

$$\begin{aligned}\langle\boldsymbol{G},\boldsymbol{W},\boldsymbol{T},\boldsymbol{X}\rangle&=\arg\min_{\boldsymbol{G},\boldsymbol{W},\boldsymbol{T},\boldsymbol{X}}\\&\left\|\begin{Bmatrix}\boldsymbol{\Psi}^TK(\boldsymbol{D},\boldsymbol{Y})\\\alpha^{\frac{1}{2}}\boldsymbol{H}\\\beta^{\frac{1}{2}}\boldsymbol{P}\end{Bmatrix}-\begin{Bmatrix}\boldsymbol{\Psi}^T\boldsymbol{G}\\\alpha^{\frac{1}{2}}\boldsymbol{W}\\\beta^{\frac{1}{2}}\boldsymbol{T}\end{Bmatrix}\boldsymbol{X}\right\|_2^2\\&\text{s.t. }\forall i,\|\boldsymbol{x}_i\|_0\le c\end{aligned}\tag{11}$$

from which we can learn a locality-preserving kernel dictionary and a corresponding optimal linear classifier. The dictionary learned this way addresses the desirable locality property and is able to adapt to the underlying structure of the input samples for better representation and classification.

3.4. Optimization

Following the basic protocol in the original K-SVD algorithm, we can efficiently find the optimal solution for Eq.(11). For conciseness, we let $\boldsymbol{Y}'=\{\boldsymbol{\Psi}^TK(\boldsymbol{D},\boldsymbol{Y}),\alpha^{\frac{1}{2}}\boldsymbol{H},\beta^{\frac{1}{2}}\boldsymbol{P}\}^T$ and $\boldsymbol{D}'=\{\boldsymbol{\Psi}^T\boldsymbol{G},\alpha^{\frac{1}{2}}\boldsymbol{W},\beta^{\frac{1}{2}}\boldsymbol{T}\}^T$, rewriting Eq.(11) as

$$\langle\boldsymbol{D}',\boldsymbol{X}\rangle=\arg\min_{\boldsymbol{D}',\boldsymbol{X}}\|\boldsymbol{Y}'-\boldsymbol{D}'\boldsymbol{X}\|_2^2\quad\text{s.t. }\forall i,\|\boldsymbol{x}_i\|_0\le c\tag{12}$$

[1] We find no advantages of using complex method to construct $\boldsymbol{\Psi}$ in our experiments, so we simply use identity matrix as $\boldsymbol{\Psi}$ in order to retain the intuitive interpretation.

which is identical to the problem that K-SVD algorithm solves. K-SVD algorithm updates the dictionary atom by atom until convergence by solving the following problem:

$$\langle\boldsymbol{d}'_k,\boldsymbol{x}_T^k\rangle=\arg\min_{\boldsymbol{d}'_k,\boldsymbol{x}_T^k}\|\boldsymbol{E}_k-\boldsymbol{d}'_k\boldsymbol{x}_T^k\|_F^2\tag{13}$$

where $\boldsymbol{x}_T^k$ is the kth row in $\boldsymbol{X}$ ($\boldsymbol{x}_k$ is the kth column), and $\boldsymbol{E}_k=\boldsymbol{Y}-\sum_{j\neq k}(\boldsymbol{d}'_j\boldsymbol{x}_T^j)$. Note that, $\boldsymbol{E}_k$ denotes the error for all input samples while the kth atom is removed. Then we discard the zero entries in $\boldsymbol{x}_T^k$ and obtain the row vector $\boldsymbol{x}_R^k$. We also define $\boldsymbol{E}_k^R$ to stand for the selection of error column vectors that correspond to samples that use the atom $\boldsymbol{d}_k$. Thus Eq.(13) is equivalent to the following minimization:

$$\langle\boldsymbol{d}'_k,\boldsymbol{x}_R^k\rangle=\arg\min_{\boldsymbol{d}'_k,\boldsymbol{x}_R^k}\|\boldsymbol{E}_k^R-\boldsymbol{d}'_k\boldsymbol{x}_R^k\|_F^2.\tag{14}$$

Singular value decomposition (SVD) can be utilized to solve Eq.(14). For the error matrix $\boldsymbol{E}_k^R$, SVD decomposes it to $\boldsymbol{E}_k^R=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$. According to the decomposition results, the solution for $\boldsymbol{d}'_k$ and the representation coefficients $\boldsymbol{x}_R^k$ is $\boldsymbol{d}'_k=\boldsymbol{U}(:,1),\boldsymbol{x}_R^k=\boldsymbol{\Sigma}(1,1)\cdot\boldsymbol{V}(:,1)$. Then the nonzero entries in $\boldsymbol{x}_T^k$ are replaced by $\boldsymbol{x}_R^k$. The updating is done iteratively until convergence. The algorithm is given in Algorithm 1.

Algorithm 1 Kernel Dictionary and Classifier Learning via LP-KSVD

Input: $\boldsymbol{Y},\boldsymbol{P}^{(0)},\boldsymbol{H},\boldsymbol{\Psi},K(\cdot),\alpha,\beta,c,r,i=0$
Output: $\boldsymbol{D}^{ke}(\boldsymbol{D}^{ke}=\boldsymbol{\Psi}^T\boldsymbol{G}),\boldsymbol{T},\boldsymbol{W}$

1: Obtain $\boldsymbol{D}^{(0)}$ by constructing the dictionary with training atoms. It does not matter whether atoms of the same label are grouped together.
2: Kernelize the dictionary by computing $\boldsymbol{D}^{ke}=\boldsymbol{\Psi}^T\boldsymbol{G}$ where $\boldsymbol{G}=\phi(\boldsymbol{D}^{(0)})^T\cdot\phi(\boldsymbol{D}^{(0)})$. In the kernel dictionary, K-SVD algorithm is used to initialize $\boldsymbol{D}^{ke(0)}$.
3: Kernelize the input samples $\boldsymbol{Y}$ via $\boldsymbol{\Psi}^TK(\boldsymbol{D},\boldsymbol{Y})$.
4: Compute sparse codes $\boldsymbol{X}^{(0)}$ for $\boldsymbol{\Psi}^TK(\boldsymbol{D},\boldsymbol{Y})$ via Eq.(1) with $\boldsymbol{D}^{ke(0)}$ as dictionary.
5: Initialize $\boldsymbol{T},\boldsymbol{W}$ with $\boldsymbol{T}^{(0)},\boldsymbol{W}^{(0)}$ via Eq.(15).
6: Initialize $\boldsymbol{Y}'$, $\boldsymbol{D}'$ which are defined in Eq.(14).
7: Update each column and the corresponding representation coefficient by iteratively solving Eq.(14) via K-SVD algorithm.
8: Obtain $\boldsymbol{\Psi}^T\boldsymbol{G},\boldsymbol{T},\boldsymbol{W}$ via Eq.(16).
9: Let $i\leftarrow i+1$ and use the newly learned dictionary to recompute $\boldsymbol{P}^{(i)}$.
10: If $i=r$, output $\boldsymbol{\Psi}^T\boldsymbol{G},\boldsymbol{T},\boldsymbol{W}$; Else, go to Step 1;

For initialization, we first compute $\boldsymbol{P}^{(0)}$ via the $\boldsymbol{D}^*$ obtained from Eq(11) with $\beta=0$. It is equivalent to removing the locality preserving term to obtain $\boldsymbol{D}^*$. Then we use $\boldsymbol{D}^*$ to compute $\boldsymbol{P}^{(0)}$. For $\boldsymbol{W}^{(0)},\boldsymbol{T}^{(0)}$, we use multivariate regression to initialize:

$$\begin{aligned}\boldsymbol{W}^{(0)}&=\boldsymbol{H}(\boldsymbol{X}^{(0)})^T\left((\boldsymbol{X}^{(0)})(\boldsymbol{X}^{(0)})^T+\mu_1\boldsymbol{I}\right)^{-1}\\\boldsymbol{T}^{(0)}&=\boldsymbol{P}(\boldsymbol{X}^{(0)})^T\left((\boldsymbol{X}^{(0)})(\boldsymbol{X}^{(0)})^T+\mu_1\boldsymbol{I}\right)^{-1}\end{aligned}\tag{15}$$

where $\boldsymbol{H}, \boldsymbol{P}$ are given a prior, μ_1, μ_2 are two small parameters, and $\boldsymbol{I}$ represents an identical matrix.

Since $\boldsymbol{P}$ is updated iteratively, the stopping criteria needs to be discussed. We perform a experiment to evaluate the convergence of $\boldsymbol{P}$, as shown in Fig 3(a). The recognition rate corresponding to $\boldsymbol{P}^{(i)}$ in each iteration is also shown in Fig 3(a). From Fig. 3(a), we can see the matrix $\boldsymbol{P}$ becomes stable after approximately 5 times iterations. In fact, we have tested on multiple database and found out 5 times iterations are enough for an appropriate $\boldsymbol{P}$. Moreover, we sum up $\boldsymbol{P}$ from each iteration (totally 100 iterations), and obtain $\boldsymbol{P}^{\Sigma}=\sum_i \boldsymbol{P}^{(i)}$. We binarize $\boldsymbol{P}^{\Sigma}$ and draw it in Fig. 3(b) where black is 0 and white is 1. It shows that locality preserving term has hidden label consistent property, since label consistent matrix [6] is a block-diagonal matrix. LP-KSVD is adaptive to the local relations between atoms and training samples. Interestingly, in an ideal scenario that samples of each label are perfectly separated, LP-KSVD becomes LC-KSVD.

3.5. Classification Strategy

We compute the sparse codes of the query with the learned kernel dictionary, and then use the linear classifier to get the label. The original K-SVD algorithm requires $\boldsymbol{D}'$ to be normalized column-wise, so $\boldsymbol{W}$ is learned with the unnormalized $\boldsymbol{D}^{ke}=\boldsymbol{\Psi}^T\boldsymbol{G}=\{\boldsymbol{d}_1^{ke},\cdots,\boldsymbol{d}_m^{ke}\}$, making it impossible to directly use dictionary $\boldsymbol{D}^{ke}$ and classifier parameters $\boldsymbol{W}=\{\boldsymbol{w}_1,\cdots,\boldsymbol{w}_m\}$ for classification. We do the following operations to obtain $\hat{\boldsymbol{D}}^{ke}$ and $\hat{\boldsymbol{W}}$:

$$\begin{aligned}\hat{\boldsymbol{D}}^{ke} &= \{\hat{\boldsymbol{d}}_1^{ke},\cdots,\hat{\boldsymbol{d}}_m^{ke}\} = \left\{\frac{\boldsymbol{d}_1^{ke}}{\|\boldsymbol{d}_1^{ke}\|_2},\cdots,\frac{\boldsymbol{d}_m^{ke}}{\|\boldsymbol{d}_m^{ke}\|_2}\right\}\\ \hat{\boldsymbol{W}} &= \{\hat{\boldsymbol{w}}_1,\cdots,\hat{\boldsymbol{w}}_m\} = \left\{\frac{\boldsymbol{w}_1}{\|\boldsymbol{d}_1^{ke}\|_2},\cdots,\frac{\boldsymbol{w}_m}{\|\boldsymbol{d}_m^{ke}\|_2}\right\}\end{aligned}. \quad (16)$$

The relation between the desired $\langle\hat{\boldsymbol{D}}^{ker},\hat{\boldsymbol{W}}\rangle$ and the obtained $\langle\boldsymbol{D}^{ker},\boldsymbol{W}\rangle$ is shown as follows ($\bar{\boldsymbol{T}},\boldsymbol{T}$ are left out):

$$\begin{aligned}\boldsymbol{y} &\approx \boldsymbol{D}^{ke}\boldsymbol{x} = \sum x_j\|\boldsymbol{d}_j^{ke}\|_2\frac{\boldsymbol{d}_j^{ke}}{\|\boldsymbol{d}_j^{ke}\|_2} = \sum x_j^*\hat{\boldsymbol{d}}_j^{ke} = \hat{\boldsymbol{D}}^{ke}\boldsymbol{x}^*\\ \boldsymbol{h} &\approx \boldsymbol{W}\boldsymbol{x} = \sum x_j\|\boldsymbol{d}_j^{ke}\|_2\frac{\boldsymbol{w}_j}{\|\boldsymbol{d}_j^{ke}\|_2} = \sum x_j^*\hat{\boldsymbol{w}}_j = \hat{\boldsymbol{W}}\boldsymbol{x}^*\end{aligned} \quad (17)$$

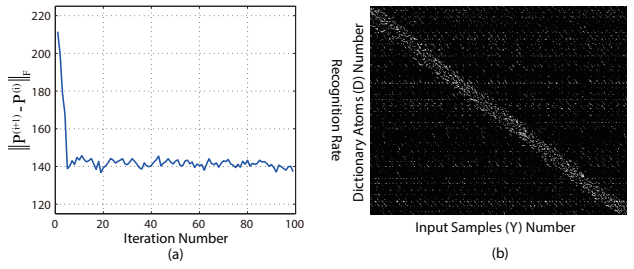

Fig. 3. (a) An example of the convergence of $\boldsymbol{P}$ on extended YaleB database. (b) Binary Representation of matrix $\boldsymbol{P}^{\Sigma}$. This experiment is performed on extended YaleB database and under the same settings in [6].

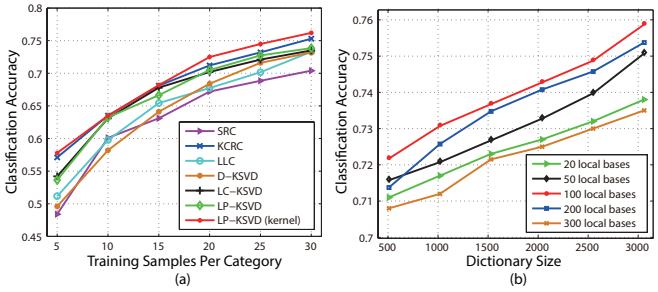

Fig. 4. (a) Classification accuracy on Caltech101 database. (b) Evaluation of different local bases under varying dictionary size.

where $\boldsymbol{x}^*$ denotes a new set of sparse codes, $\boldsymbol{y}$ is a query sample and $\boldsymbol{h}$ is the label vector. We obtain the label by first coding the query over dictionary $\hat{\boldsymbol{D}}^{ker}$, and the label of the query refers to the index of the largest element in $\boldsymbol{h}=\hat{\boldsymbol{W}}\boldsymbol{x}^*$.

4. EXPERIMENTS AND RESULTS

This section uses $\alpha=4, \beta=4$ and adopts RBF kernel with parameter 0.5 in all experiments. Note that, The matrix $\boldsymbol{P}$ is obtained by 5 iterations. α and β are determined by fivefold cross validation on the training dataset.

4.1. Caltech101 Database

Caltech101 database [24] contains 9144 images from 102 classes with significant variance in shape. We train on 5, 10, 15, 20, 25, 30 samples per category and test on the rest. Dictionary size is set as 510, 1020, 1530, 2040, 2550 and 3060 for 5, 10, 15, 20, 25 and 30 samples per category respectively. Note that, we adopt the spatial pyramid features. The results are averaged for 10 times and shown in Fig. 4(a). We also evaluate the classification performance with different local bases under varying dictionary size trained in LP-KSVD. Note that, we use 30 samples per category to train the dictionary. Fig. 4(a) shows the proposed LP-KSVD (kernel) performs well in image classification tasks and surpasses most competitive approaches. LP-KSVD is 0.3% to 0.4% better than LC-KSVD in 25 and 30 training samples per category. LP-KSVD with kernel outperforms LP-KSVD and LC-KSVD by approximate 2.5%. Fig. 4(b) shows the proposed method with 100 local atoms has the best performance, indicating the number of local bases has an optimal range in which the classification accuracy reaches its best.

4.2. Extended YaleB Database

Extended YaleB database [25] contains 2414 frontal face under different illumination conditions. We randomly select 32 images as training samples, the rest as testing samples, and adopt the 504 dimensional randomface features. For K-SVD based algorithms, We use 32 images per person to train a dictionary of size 570. The number of local bases for LP-KSVD

is set as 40. We run the experiment for 10 times and obtain the average recognition rate in Table 1. The running time[2] for dictionary training and query testing is shown in Table 2. We can see the proposed LP-KSVD outperforms LC-KSVD in kernel space and other competitive approaches. As for the efficiency of LP-KSVD, the time to classify a query is approximate 0.3ms, which is fast enough for applications.

Table 1. Recognition results (%) on extended YaleB database. For SRC, we use a dictionary of 570 atoms, same as D-KSVD, LC-KSVD and LP-KSVD.

Method	Accuracy	Method	Accuracy
SRC [3]	81.7	LLC [14] (25 bases)	83.1
D-KSVD [5]	93.5	LLC [14] (50 bases)	89.7
LC-KSVD [6]	95.0	LC-KSVD [6] (kernel)	94.4
LP-KSVD	94.8	LP-KSVD (kernel)	94.9

Table 2. Average dictionary training time and running time for classifying a testing image on extended YaleB database.

Method	Testing Time (ms)
SRC [3]	19.72
LC-KSVD [6]	0.298
LP-KSVD	0.328
LP-KSVD (kernel)	0.536

4.3. 15 Scenes Database

15 scenes categories database [26] contains 200 to 400 images per category, including kitchen, mountain and store. We randomly select 100 images per category for training and the rest for testing. The size of learned dictionary is set as 450. Note that, we adopt the spatial pyramid features. Results are shown in Table 3. Our proposed method achieves better performance than SRC, LLC, D-KSVD and LC-KSVD. We can see the kernel indeed boosts the performance for both LC-KSVD and LP-KSVD. LP-KSVD performs better than LC-KSVD and LP-KSVD with kernel achieves the best classification accuracy in some competitive methods.

Table 3. Classification results (%) on 15 scenes database.

Method	Accuracy	Method	Accuracy
SRC [3]	91.8	LLC [14] (30 bases)	90.1
D-KSVD [5]	89.2	LSC [16]	89.9
LC-KSVD [6]	92.9	LC-KSVD [6] (kernel)	93.5
LP-KSVD	93.4	LP-KSVD (kernel)	94.0

5. CONCLUDING REMARKS

This paper proposes the locality preserving K-SVD algorithm and elaborates the gain that locality brings especially in kernel space. Specifically, a locality preserving term is designed to enforce locality into dictionary and coding coefficients. In order to make locality information more discriminative, we map the original feature space to the kernel space, making samples of different classes more separable. LP-KSVD approach in kernel space and its specific algorithm are presented and comprehensively discussed. Experimental results validate the superiority of LP-KSVD in kernel space and also show that our approach achieves state-of-the-art results on extended YableB database, Caltech101 database and 15 scenes database.

[2]We use a PC with 3.4 GHz dual-core CPU and 16GB RAM. All experiments presented in the paper are performed in this PC.

6. REFERENCES

[1] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *ICCV*, 2009. 1

[2] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE TIP*, 2006. 1

[3] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE TPAMI*, 2009. 1, 3.2, 1, 2, 3

[4] K. Huang and S. Aviyente, "Sparse representation for signal classification," in *NIPS*, 2006. 1

[5] Q. Zhang and B. Li, "Discriminative k-svd for dictionary learning in face recognition," in *CVPR*, 2010. 1, 1, 2, 2, 2, 1, 3

[6] Z. Jiang, Z. Lin, and L. S. Davis, "Label consistent k-svd: learning a discriminative dictionary for recognition," *IEEE TPAMI*, 2013. 1, 1, 2, 2, 2, 3.4, 3, 1, 2, 3

[7] M. Yang, D. Dai, L. Shen, and L. V. Gool, "Latent dictionary learning for sparse representation based classification," in *CVPR*, 2014. 1

[8] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Sparse representation based fisher discrimination dictionary learning for image classification," *IJCV*, 2014. 1

[9] Y. Sun, Q. Liu, J. Tang, and D. Tao, "Learning discriminative dictionary for group sparse representation," *IEEE TIP*, 2014. 1

[10] S. Bengio, F. Pereira, Y. Singer, and D. Strelow, "Group sparse coding," in *NIPS*, 2009. 1

[11] M. Zheng, J. Bu, C. Chen, C. Wang, L. Zhang, G. Qiu, and D. Cai, "Graph regularized sparse coding for image representation," *IEEE TIP*, 2011. 1

[12] D. Pham and S. Venkatesh, "Joint learning and dictionary construction for pattern recognition," in *CVPR*, 2008. 1, 2, 2

[13] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE TSP*, 2006. 1, 3.1

[14] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *CVPR*, 2010. 1, 3.1, 1, 3

[15] K. Yu, T. Zhang, and Y. Gong, "Nonlinear learning using local coordinate coding," in *NIPS*, 2009. 1, 3.1

[16] S. Gao, I. W. Tsang, L. Chia, and P. Zhao, "Local features are not lonely–laplacian sparse coding for image classification," in *CVPR*, 2010. 1, 3

[17] S. Gao, I. W. Tsang, and L. T. Chia, "Kernel sparse representation for image classification and face recognition," in *ECCV*. 2010. 1

[18] H. Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Kernel dictionary learning," in *ICASSP*, 2012. 1

[19] W. Liu et al., "A novel kernel collaborative representation approach for image classification," in *ICIP*, 2014. 1, 3.3, 3.3

[20] W. Liu and Z. Yu et al., "Kcrc-lcd: Discriminative kernel collaborative representation with locality constrained dictionary for visual categorization," *arXiv:1410.4673*, 2014. 1, 3.3

[21] B. Scholkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*, MIT press, 2001. 3.3

[22] B. Schölkopf, A. Smola, and K. Müller, "Kernel principal component analysis," in *ICANN*. 1997. 3.3, 3.3

[23] L. Zhang, W. Zhou, P. Chang, J. Liu, Z. Yan, T. Wang, and F. Li, "Kernel sparse representation-based classifier," *IEEE TSP*, 2012. 3.3

[24] Li F., R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *CVIU*, 2007. 4.1

[25] A. S. Georghiades, P. N. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE TPAMI*, 2001. 4.2

[26] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006. 4.3