Pattern Recognition Letters 78 (2016) 64–69

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Greedy dictionary learning for kernel sparse representation based classifier☆

Vinayak Abrol*, Pulkit Sharma, Anil Kumar Sao

SCEE, School of Computing and Electrical Engineering, Indian Institute of Technology, Mandi, India

ARTICLE INFO

Article history:
Received 27 August 2015
Available online 23 April 2016

Keywords:
Classification
Kernel sparse representations
Dictionary learning
Sparse coding

ABSTRACT

We present a novel dictionary learning (DL) approach for sparse representation based classification in kernel feature space. These sparse representations are obtained using dictionaries, which are learned using training exemplars that are mapped into a high-dimensional feature space using the kernel trick. However, the complexity of such approaches using kernel trick is a function of the number of training exemplars. Hence, the complexity increases for large datasets, since more training exemplars are required to get good performance for most of the pattern classification tasks. To address this, we propose a hierarchical DL approach which requires the kernel matrix to update the dictionary atoms only once. Further, in contrast to the existing methods, the dictionary is learned in a linearly transformed/coefficient space involving sparse matrices, rather than the kernel space. Compared to the existing state-of-the-art methods, the proposed method has much less computational complexity, but performs similar for various pattern classification tasks.

1. Introduction

In recent years kernel sparse representation based classifier (KSRC) has been widely explored in various pattern classification and recognition tasks [30,32,34]. These kernel algorithms were actually proposed to improve the performance of sparse representation based classifier (SRC) [27,33] which uses the linear modeling framework, by exploiting the advantages of projecting data in some high-dimensional space [15]. However, linear representations are inadequate for representing non-linear structures of the data which arise in many practical applications [32]. Hence, if an appropriate kernel function is utilized, then there is a high probability that similar features are grouped together in the high-dimensional space [34]. Linear modeling in the projected feature space can thus address the issue of non-linearity and provide better discrimination than their traditional counterparts in the original space. In addition, all KSRC based approaches can operate in a high-dimensional space without explicitly transforming the data in that space [15].

In KSRC, the aim is to obtain the sparse representation $\mathbf{a}_t$ of a test feature vector $\mathbf{x}_t$ which is mapped to some high-dimensional space, and classify it to the class that gives the smallest reconstruction error [15]. Given a dictionary $\ddot{\mathbf{D}}$ in higher dimensions (learned/build using class specific training data), the sparse vector is obtained by solving the following sparse coding problem [34]:

$$\underset{\mathbf{a}_t}{\operatorname{argmin}} \|\phi(\mathbf{x}_t) - \ddot{\mathbf{D}}\mathbf{a}_t\|_2^2 \text{ s.t } \forall_i \ \|\mathbf{a}_i\|_0 \leq T_0, \tag{1}$$

where $\|.\|_0$ is the l_0-norm (convex surrogates can also be used), T_0 denotes the imposed limit on the cardinality of the sparse vector. Here, the transformation function $\phi : \mathbb{R}^n \rightarrow \mathcal{S}$ maps the input space to a high-dimensional Hilbert space $\mathcal{S}$. However, in most cases the transformation ϕ is not known, and the optimization of problem in (1) is infeasible using traditional methods. This issue can be addressed by using a kernel similarity function κ, which avoids the explicit mapping of training data to space $\mathcal{S}$ [30]. Let $\mathcal{K}(\ddot{\mathbf{D}}, \ddot{\mathbf{D}})$ be a kernel matrix whose elements are computed using kernel κ as $\kappa(\mathbf{d}_i, \mathbf{d}_j) = \phi(\mathbf{d}_i)^T\phi(\mathbf{d}_j)$. Similarly let $\mathcal{K}(\ddot{\mathbf{D}}, \mathbf{x}_t)$ be a vector with elements $\kappa(\mathbf{d}_i, \mathbf{x}_t)$. Hence, (1) can be written as [34]:

$$\underset{\mathbf{a}_i}{\operatorname{argmin}} \kappa(\mathbf{x}_t, \mathbf{x}_t) + \mathbf{a}_t^T\mathcal{K}(\ddot{\mathbf{D}}, \ddot{\mathbf{D}})\mathbf{a}_t - 2\mathbf{a}_t^T\mathcal{K}(\ddot{\mathbf{D}}, \mathbf{x}_t) \quad \text{s.t } \|\mathbf{a}_t\|_0 \leq T_0. \tag{2}$$

The challenge with such a formulation is that the dictionary atom $\mathbf{d}_i$ in the original signal space corresponding to atom $\phi(\mathbf{d}_i)$ in space $\mathcal{S}$ is unknown [30]. To address this issue, the training matrix (or its subset) can be used as the dictionary [34]. However, using $\phi(\mathbf{X})$ as the dictionary $\ddot{\mathbf{D}}$ in KSRC is inefficient, especially when the number of training exemplars is very large [30,32]. For instance, (i) testing phase will be very slow, as determining sparse codes for

☆ This paper has been recommended for acceptance by Y. Liu

* Corresponding author. Tel.: +91 9646315212.
E-mail address: vinayak-abrol@students.iitmandi.ac.in (V. Abrol).

http://dx.doi.org/10.1016/j.patrec.2016.04.014
0167-8655/

Table 1
Matrix and vector dimensions.

$\mathbf{X}$	$\phi(\mathbf{X})$	$\ddot{\mathbf{D}}$	$\mathbf{B}$	$\mathbf{A}$	$\ddot{\mathbf{d}}_i$	$\mathbf{x}_i$	$\mathbf{a}_i$
$n \times l$	$\tilde{n} \times l$	$\tilde{n} \times m$	$l \times m$	$m \times l$	$\tilde{n} \times 1$	$n \times 1$	$m \times 1$

dictionaries with more number of atoms is computationally expensive, prohibiting real-time application, and (ii) manually selecting a subset of the training data to seed the dictionary is not only tedious but also sub-optimal since there is no guarantee that such selection form the best dictionary [18].

In order to address these issues, recent studies suggest in favor of learning a dictionary instead of using the training data itself [1,18,28,30,32]. The dictionary for each class is learned from its training signal set $\mathbf{X} \in \mathbb{R}^{n \times l}$ by minimizing the reconstruction error and satisfying the sparsity constraints [30]. Existing dictionary learning (DL) algorithms available in the literature solves the following optimization problem:

$$\underset{\ddot{\mathbf{D}},\mathbf{A}}{\operatorname{argmin}}\, g(\mathbf{A}) \text{ subject to } \|\phi(\mathbf{X}) - \ddot{\mathbf{D}}\mathbf{A}\|_F^2 \leq \epsilon, \tag{3}$$

where $\|.\|_F$ is the Frobenius norm, $g()$ is a function that promotes sparsity (e.g, l_0-norm), ϵ is the error tolerance constant, $\ddot{\mathbf{D}} \in \mathbb{R}^{\tilde{n} \times m}$ is a dictionary in space $\mathcal{S}$ and $\mathbf{A}$ is the sparse coefficient matrix corresponding to the transformed training set $\phi(\mathbf{X})$ [31]. This non-convex problem is solved via alternative minimization in two steps i.e., sparse coding and dictionary update. The sparse coding problem can be solved for each training similar to (2). Once the sparse code for each exemplar is calculated, the dictionary can be updated such that the error, $\|\phi(\mathbf{X}) - \ddot{\mathbf{D}}\mathbf{A}\|_F^2$ is minimized. DL gives a suitable number of discriminative atoms for each class spanning its signal space, but learning an optimal dictionary in space $\mathcal{S}$ is not straight forward as compared to signal space [30]. In [32], it has been proved that the optimal solution for the dictionary $\ddot{\mathbf{D}}$ has the form $\ddot{\mathbf{D}} = \mathbf{PB} = \phi(\mathbf{X})\mathbf{B}$. Here, $\mathbf{P} = \phi(\mathbf{X})$ is a better choice as compared to relying on any manual selection. Further, $\mathbf{P}$ acts as a known prior and a regularizer to reduce over-fitting and instability while DL. In fact with this formulation, the dictionary atoms are linear combinations of the training exemplars instead of the training data itself. Thus, one can tune the dictionary of a class via modifying its corresponding sparse matrix $\mathbf{B}$. To understand this, consider the objective function in (3) as:

$$\begin{aligned} &\|\phi(\mathbf{X}) - \phi(\mathbf{X})\mathbf{BA}\|_F^2 \\ &= \|\phi(\mathbf{X})(\mathbf{I} - \mathbf{BA})\|_F^2 \\ &= \mathbf{tr}((\mathbf{I} - \mathbf{BA})^T \mathcal{K}(\mathbf{X}, \mathbf{X})(\mathbf{I} - \mathbf{BA})) \end{aligned} \tag{4}$$

where $\mathcal{K}(\mathbf{X}, \mathbf{X})$ is the kernel matrix whose elements are computed as $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$. Such a formulation is more efficient to solve, since it involves a kernel matrix of finite dimensions and a sparse matrix $\mathbf{B}$, instead of dealing with a possibly very large or some high-dimensional dictionary $\ddot{\mathbf{D}}$. Here, some popular kernels such as linear, Gaussian and polynomial kernels can be employed [15]. For the reader's convenience, Table 1 summarizes the dimensions of all important matrices and vectors. Parameters involving $\tilde{n}$ are associated with the high-dimensional feature space.

In order to solve (4), both $\mathbf{B}$ and $\mathbf{A}$ are alternatively optimized with respect to the whole dictionary $\ddot{\mathbf{D}}$ using the kernel trick. Hence, existing algorithms have large time complexity due to: (i) size of the dataset, (ii) overcompleteness of the dictionary, and (iii) density and size of the kernel matrix. In some cases extracted features are such that the kernel matrix is sparse, and hence sparse matrix manipulation methods may be used. However, in most cases the kernel matrix is dense, which leads to increased time complexity mainly for large scale learning tasks. This is because kernel methods typically construct a kernel matrix $\mathcal{K} \in \mathbf{R}^{l \times l}$ where l is the number of training instances. Nevertheless, the complexity of dictionary optimization using the kernel trick is therefore a function of the number of training exemplars, instead of the dimensionality of the input exemplars. Further, optimizing $\mathbf{B}$ require the kernel matrix in each iteration of existing DL algorithms [32]. Although, to deal with large kernel matrices many methods have focused on computing its low-rank approximation [2,9], but the focus of this paper is to propose an effective approach only for alleviating memory and computational cost for DL.

In this work using the kernel trick, we show that alternative to (4), one can define an objective function such that learning $\mathbf{B}$ (separately for each class) is efficient and independent of any computations involving the kernel matrix. This is achieved by learning the matrix $\mathbf{B}$ in the coefficient space rather than the signal or kernel space. For this, the kernel DL problem is transformed into a more suitable form to find a numerically stable solution, a process referred to as *preconditioning* [25]. Note that, the sparse coding stage will still require the use of kernel matrix. In order to update the matrix $\mathbf{B}$, the proposed algorithm uses a hierarchical subset selection procedure. In each iteration, a column/atom of $\mathbf{B}$ is selected in accordance to its energy contribution, from the transformed training exemplars in the coefficient domain. It is done in such a way that the information learned by the previously updated atoms can be used to guide an adaptive design of subsequent atoms. Thus, after each update the modified residual serves as the new training set for the next update i.e., any atom is learned in accordance to what was not learned using previous atoms.

1.1. Related work

In earlier works of [15,20,34], the dictionaries used to compute the kernel sparse representation (based on l_1-norm minimization) consists of exemplars from the transformed training exemplars. In addition, $\mathcal{S}$ being a very high-dimensional space, in [20], the transformed exemplars are projected on to a reduced dimensionality subspace e.g., using principal component analysis (PCA). In contrast, works in [32] and [30], proposed to learn the dictionary by solving (4) using conventional DL approaches. In [30], the matrix $\mathbf{B}$ is updated using multilevel dictionary learning (MDL) method [29]. While in [32], $\mathbf{B}$ is updated based on Method of optimal directions (MOD) [12] or K-singular value decomposition (KSVD) [3], and the sparse coding stage is solved using the modified kernel OMP (KOMP) algorithm. In [32] $\mathbf{B}$ is optimized separately for different classes, while in [30] a single $\mathbf{B}$ is learned for all classes, using an ensemble of kernel matrices, such that the class discrimination is maximum. It is important to note that, the kernel MDL (KMDL), kernel MOD (KMOD) or kernel KSVD (KKSVD) formulation is highly non-convex and hard to solve in a moderate amount of time [32]. KMOD/KKSVD suffers from similar drawbacks as MOD/KSVD i.e., high complexity of the matrix inversion and lack of convergence guarantees, respectively. Moreover, in all the existing methods optimizing $\mathbf{B}$ require the kernel matrix in each iteration.

These issues are addressed in the proposed DL approach since: (i) it does not involve the kernel matrix to update the dictionary in each iteration of the algorithm, (ii) dictionary update is efficient as it is performed in the coefficient domain involving sparse matrices, and (iii) dictionary update does not involve any computationally intensive operations such as SVD or matrix inversion which makes it faster.

1.2. Organization of the paper

The rest of the paper is organized as follows: In Section 2 we propose an efficient algorithm for kernel sparse DL problem.

Section 3 analyzes the computational complexity and the convergence behavior of the proposed algorithm. The experimental results for digit classification, spoken letter classification and Parkinson speech classification tasks are presented in Section 4. The summary of the paper is given in Section 5.

2. Proposed Kernel Sparse Greedy Dictionary: KSGD

The proposed kernel DL algorithm involve two stages: sparse coding (SC) and dictionary update (DU). The SC and DU procedure are alternatively optimized iteratively until convergence. Convergence can be achieved by either fixing number of iterations or when the reconstruction error at the current iteration is below a tolerance level (ϵ). The SC stage can be solved by KOMP algorithm as proposed in [32]. In DU stage, $\mathbf{B}$ is updated such that the error, $\|\phi(\mathbf{X}) - \phi(\mathbf{X})\mathbf{BA}\|_F^2$ is minimized. The overall design process to update the dictionary will now be presented in two parts: (i) Section 2.1 shows, how we can exploit the kernel trick to learn a dictionary in the coefficient domain, and (ii) the proposed approach of updating the matrix $\mathbf{B}$ is explained in Section 2.1.1.

2.1. Proposed approach for updating dictionary atoms

It has been suggested in the literature that it is preferable to minimize the error associated with the estimation of the original vector $\mathbf{x}$, instead of the error associated with the estimation of its sparse representation $\mathbf{a}$ [31]. However, in [6], authors have shown that the error in the coefficient or the representation domain upper bounds the error in the signal domain. In particular for an overcomplete dictionary $\mathbf{D}$ we have:

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 = \|\mathbf{Da} - \mathbf{D}\hat{\mathbf{a}}\|_2^2 \leq \lambda^2(\mathbf{D})\|\mathbf{a} - \hat{\mathbf{a}}\|_2^2 \tag{5}$$

where $\hat{\mathbf{x}}$ is some estimate of $\mathbf{x}$ and $\lambda(\mathbf{D})$ is the largest singular value of $\mathbf{D}$ [6]. Hence, for kernel DL problem (involving transformation ϕ or equivalently kernel κ), it is better to manipulate or process the signal's information in the coefficient domain instead of the signal domain. In order to achieve this, the objective function (4) can be expressed as:

$$\begin{aligned}&\|\phi(\mathbf{X}) - \phi(\mathbf{X})\mathbf{BA}\|_F^2 \\ &= \|\phi(\mathbf{X}) - \phi(\mathbf{X})\mathbf{C}\|_F^2 \text{ s.t. } diag(\mathbf{C}) = 0\end{aligned} \tag{6}$$

Here, matrix $\mathbf{C}$ can be seen as the coefficient matrix for representing each training exemplar in space $\mathcal{S}$ as a linear combination of other training exemplars. This can be interpreted as an affinity transformation, where training exemplars that lie in the same subspace utilize one another in their sparse representations [11]. The coefficient matrix $\mathbf{C}$ is computed such that the error is bounded i.e., $\|\phi(\mathbf{X}) - \phi(\mathbf{X})\mathbf{C}\|_F^2 < \delta$ [1]. To maintain this bound, the product $\mathbf{BA}$ should be close to $\mathbf{C}$ for a sparse coefficient matrix $\mathbf{A}$, or equivalently we can update $\mathbf{B}$ by alternatively solving the problem

$$\underset{\mathbf{B},\mathbf{A}}{\operatorname{argmin}} \|\mathbf{C} - \mathbf{BA}\|_F^2 \text{ s.t } \forall_i \ \|\mathbf{a}_i\|_0 \leq T_0, \ \forall_j \ \|\mathbf{b}_j\|_2 = 1, \tag{7}$$

with respect to $\mathbf{B}$ and $\mathbf{A}$, instead of one defined in (4). Further, note that the kernel matrix is required only once to compute $\mathbf{C}$.

In other words, the problem of learning $\ddot{\mathbf{D}}$ (or equivalently $\mathbf{B}$) given $\mathbf{X}$ and $\mathbf{A}$ is now simplified to the problem of learning $\mathbf{B}$ given $\mathbf{C}$ and $\mathbf{A}$. This can also be seen as a sub-DL or matrix factorization problem solvable by existing methods such as KSVD [3]. The main difference lies in use of $\mathbf{C}$ instead of $\mathbf{X}$ as the training set. Further, any dictionary update (DU) step indeed converges (if it at all converges) much faster, since all the matrices involved i.e $\mathbf{C}$, $\mathbf{B}$ and $\mathbf{A}$ are sparse or near-sparse.

[1] (6) is a simple SC problem solvable using methods such as KOMP. Pseudo-code of KOMP algorithm is provided in Appendix A.

2.1.1. KSGD update step

It should be noted that with the proposed formulation in (7), one can employ any DU step, such as MOD or KSVD to update the matrix $\mathbf{B}$. In contrast, we exploit the sparse nature of matrices involved in (7). This problem can be addressed by using an exemplar based approach, where the training exemplars (or equivalently columns of $\mathbf{C}$) are chosen as columns/atoms of $\mathbf{B}$ [22]. The aim is to use the training data itself, so as to sparsely represent each exemplars in the dataset as a linear combination of a small subset of exemplars [23]. In addition, it is desired that the information learned by the previous atoms can be used to guide an adaptive selection of subsequent atoms, which lead to a faster DU. Thus, to achieve this we propose a greedy iterative hierarchical subset selection approach so as to minimize the overall error/residual matrix in the coefficient domain i.e.,

$$\mathbf{E} = \mathbf{C} - \mathbf{BA} = \mathbf{C} - (\mathbf{W_1} + \mathbf{W_2} + \cdots) \ \forall_k : \mathbf{W}_k = \mathbf{b}_k\mathbf{a}_{[k]}, \tag{8}$$

where, $\mathbf{b}_k$ and $\mathbf{a}_{[k]}$ denotes the kth column and row of $\mathbf{B}$ and $\mathbf{A}$ respectively.

At start the error/residual $\mathbf{E}$ is initialized to $\mathbf{C}$ i.e., considering $\mathbf{B} = \emptyset$. Now $\mathbf{B}$ is updated atom-by-atom hierarchically, by sequentially extracting a new column $\mathbf{e}_k$ from the current error matrix $\mathbf{E}$ based on the criteria minimum$(\frac{l_1(\mathbf{e}_k)}{l_2(\mathbf{e}_k)})$ ratio[2], computed over its columns [3]. To motivate such selection, we should mention that similar measures (e.g., squared column norm) have been extensively employed for subspace clustering, low-rank approximation and column subset selection (CSS) problems, where the aim is to select a subset of training data which spans its entire column space [8,14,17]. Following this, the coefficient vector $\mathbf{a}_{[k]}$ is updated by solving the following optimization problem

$$\mathbf{a}_{[k]} = \underset{\mathbf{a}_{[k]}}{\operatorname{argmin}} \|\mathbf{E} - \mathbf{b}_k\mathbf{a}_{[k]}\|_F^2 \tag{9}$$

Instead of solving this problem using rank-1 decomposition via SVD, one can use an alternate formulation as:

$$\begin{aligned}\|\mathbf{E} - \mathbf{W}_k\|_F^2 &= tr\{(\mathbf{E} - \mathbf{b}_k\mathbf{a}_{[k]})(\mathbf{E} - \mathbf{b}_k\mathbf{a}_{[k]})^T\} \\ &= \|\mathbf{E}\|_F^2 - 2\mathbf{b}_k^T\mathbf{E}\mathbf{a}_{[k]}^T + \|\mathbf{b}_k\|^2\|\mathbf{a}_{[k]}\|^2\end{aligned} \tag{10}$$

which after differentiation gives the solution: $\mathbf{a}_{[k]} = \mathbf{b}_k^T\mathbf{E}$, provided the atom $\mathbf{b}_k$ is normalized after selection. After each such update, the error $\mathbf{E}$ is minimized by subtracting from it the updated atom's energy contribution i.e., $\mathbf{b}_k\mathbf{a}_{[k]}$. Now, the modified error matrix serves as the new training set for the next atom update. Thus, the next atom is not only incoherent to previously selected atoms but is selected in accordance to what was not learned previously. This is because after each atom selection, updating $\mathbf{E}$ is equivalent to orthogonalizing it to the atoms chosen so far. To prove this, consider the residual update step as the projection of the current residual onto the dictionary atom space, as done in case of Gram–Schmidt procedure [17]:

$$\begin{aligned}\mathbf{E}_{i+1} &= \mathcal{P}_{\mathbf{B}}\mathbf{E}_i = \left(\mathbf{I} - \frac{\mathbf{b}_k\mathbf{b}_k^T}{\mathbf{b}_k^T\mathbf{b}_k}\right)\mathbf{E}_i \\ &= \mathbf{E}_i - \frac{\mathbf{b}_k\mathbf{b}_k^T\mathbf{E}_i}{\mathbf{b}_k^T\mathbf{b}_k} = \mathbf{E}_i - \frac{\mathbf{b}_k\mathbf{a}_{[k]}}{\mathbf{b}_k^T\mathbf{b}_k}\end{aligned} \tag{11}$$

Here, $\mathcal{P}_{\mathbf{B}}$ denotes the projection matrix, $\mathbf{I}$ denotes identity matrix of appropriate dimensions, $\mathbf{b}_k$ denotes the kth normalized chosen atom and $\mathbf{a}_{[k]}$ denotes the coefficient row vector obtained using

[2] l_1/l_2 ratio is a special case of pq-mean ($p \leq 1$, $q > 1$) sparsity metric which has been shown to satisfy all the important sparsity attributes [16].

[3] Since $\phi(\mathbf{X})$ accounts for the most of the information in atoms of dictionary $\ddot{\mathbf{D}} = \phi(\mathbf{X})\mathbf{B}$, the matrix $\mathbf{B}$ is assumed to be sparse. Hence function *minimum* is used to chose sparse columns in order to promote sparsity over columns of $\mathbf{B}$.

(11). Because of atom normalization, the denominator term in the right-hand-side of (11) is equal to 1, and therefore this equation corresponds to the proposed residual update step. However, note that updating $\mathbf{a}_{[k]}$ using (11) will not preserve the sparsity constraint in (7). To address this, in KSGD any dictionary atom is updated using only those columns in $\mathbf{E}$ having similar indices as columns in $\mathbf{X}$, whose sparse representations use the current atom. Denoting by Ω the indices of the columns in $\mathbf{X}$ that use the kth atom, the updated residual matrix is obtained as

$$\mathbf{E}^{\Omega} - \mathbf{W}_k^{\Omega} = \mathbf{E}^{\Omega} - \mathbf{b}_k \mathbf{a}_{[k]}^{\Omega} = \mathbf{E}^{\Omega} - \mathbf{b}_k \mathbf{b}_k^T \mathbf{E}^{\Omega} \tag{12}$$

where, $\mathbf{b}_k$ is chosen as before. The main improvement in doing so is that the atoms of $\ddot{\mathbf{D}}$ will now have a larger support, since (12) will only update the residual columns corresponding to set Ω, and the sparsity constraint in $\mathbf{a}_k$ will be preserved. In addition, one achieves good coherence bounds for the dictionary, while reducing redundant or correlated atoms. Algorithm 1 shows the pseudo-code of the proposed approach.

Algorithm 1 Kernel sparse greedy dictionary learning algorithm.

Inputs: Training data matrix $\mathbf{X} \in \mathbb{R}^{n \times l}$ containing l exemplars
Outputs: Sparse dictionary coefficient matrix $\mathbf{B} \in \mathbb{R}^{l \times m}$

Initialization: ϵ, Kernel $\mathcal{K}$, sparsity level T_0 and random matrix $\mathbf{B}$
Compute $\mathbf{C}$ via (6)
Perform outer iterations
$\mathbf{A} \leftarrow KOMP(\mathbf{X}, \ddot{\mathbf{D}}, T_0)$, $\mathbf{E} \leftarrow \mathbf{C}$
Perform inner iterations: $k = 1$ to m
  Ω : indices of the columns in $\mathbf{E}$ whose reps. use $\mathbf{b}_k$
  $i \leftarrow min\left(\frac{l_1(\mathbf{e}_i)}{l_2(\mathbf{e}_i)}\right) \ \forall_{i \in \Omega}$
  $\mathbf{b}_k \leftarrow \mathbf{e}_i$
  $\mathbf{b}_k \leftarrow \mathbf{b}_k / \|\mathbf{b}_k\|_2$
  $\mathbf{a}_{[k]}^{\Omega} \leftarrow \mathbf{b}_k^T \mathbf{E}^{\Omega}$
  $\mathbf{E}^{\Omega} \leftarrow \mathbf{E}^{\Omega} - \mathbf{b}_k \mathbf{a}_{[k]}^{\Omega}$
Until m **columns**
Update dictionary as $\ddot{\mathbf{D}} = \phi(\mathbf{X})\mathbf{B}$
Until convergence

At first sight, the proposed DU procedure looks similar to the MDL update step [30], but there are significant differences between them. In MDL, the modified error matrix also serves as the new training data for the next level. However, the sparsity of the representation in each level of MDL algorithm is fixed at 1 which is not the case with the proposed method. Also, in MDL various sub-dictionaries (of a single dictionary) are updated in each level (using 1-D subspace clustering), while in KSGD only one atom is updated per iteration by a simple atom choosing criteria, and that too in the coefficient domain.

3. Computational complexity and convergence study

The advantages of the proposed KSGD algorithm are in its low computational complexity and convergence guarantees. Provided the kernel matrix is precomputed, the computational cost per-iteration (of SC and DU steps) of the proposed algorithm scales as $\mathcal{O}(n^2 L)$ for learning a square dictionary (with $m = n$) from L training exemplars. This cost is much lower than the cost in case of existing approaches e.g., KKSVD or KMOD, which scales as $\mathcal{O}(n^3 L)$ [32]. Further, the proposed greedy approach is stable and has proven convergence guarantees.

Proposition 1. *While updating variables* $\mathbf{B}$ *and* $\mathbf{A}$ *alternatively, the generalized minimization problem of (7) (in the coefficient domain) is asymptotically regular.*

Proof. Objective function defined in (7) is strictly convex when either $\mathbf{B}$ or $\mathbf{A}$ is fixed, and being quadratic has a bounded curvature. It is well known that, when objective function is continuous its epigraph is compact [7]. Therefore, solution set for minimization based on $\mathbf{B}$ or $\mathbf{A}$ is bounded. This is also because, the dictionary is updated in the coefficient domain (obtained via kernel mapping over ϕ) from a closed set i.e., a subset of current residual columns. Further, such a formulation constraints the maximum value of $\|\ddot{\mathbf{D}}\|_F$ and $g(\mathbf{A})$ to be bounded. Hence, for a Euclidean space, boundedness and closedness are sufficient to prove that the minimization problem in the coefficient domain is asymptotically regular.

Proposition 2. *The objective function (or residual error) during DU reduces monotonically in each iteration and thus the convergence of the proposed algorithm is guaranteed using Lyapunov's 2nd theorem [9].*

Proof. The convergence of the proposed algorithm follows from the fact that using (12) the updated residual energy reduces each time a new atom is updated i.e., $\|\mathbf{E}_{i+1}\|_F^2 < \|\mathbf{E}_i\|_F^2$, since $\|\mathbf{b}_k \mathbf{a}_{[k]}\|_F^2 > 0$[4]. Thus considering that $\mathbf{b}_k$ has unit l_2 norm, the final error matrix $\mathbf{E}_M$ after updating M dictionary atoms iteratively, can be expressed as:

$$\begin{aligned} \|\mathbf{E}_M\|_F^2 &= \|\mathbf{E}\|_F^2 - \sum_{k=1}^{M} \|\mathbf{b}_k \mathbf{a}_{[k]}\|_F^2 \\ &= \|\mathbf{E}\|_F^2 - \sum_{k=1}^{M} \|\mathbf{b}_k\|_2^2 \|\mathbf{a}_{[k]}\|_2^2 = \|\mathbf{E}\|_F^2 - \sum_{k=1}^{M} \|\mathbf{a}_{[k]}\|_2^2 \end{aligned} \tag{13}$$

It can be observed that, the overall reduction in objective function mainly depends on sum of squares of coefficient vector $\mathbf{a}_{[k]}$. This shows that performance with KSGD algorithm is sensitive to SC step and one can tune the dictionary to obtain better results, using appropriate sparsity constraints (e.g., smooth penalty functions such as convex l_1-norm).

4. Experimental results

The performance of the proposed method is evaluated along with the existing methods (e.g., KMOD, KKSVD) using classification tasks for three different datasets namely USPS digit, isolated spoken letter (ISOLET) and Parkinson's Disease (PD) speech datasets [4]. As proposed in [32], we used the generative approach to do the classification. In particular, a test example is classified to the classes that give the smallest reconstruction error. We first concatenate dictionaries from all (say q) classes as:

$$\ddot{\mathbf{D}}_f = [\ddot{\mathbf{D}}^1 \cdots \ddot{\mathbf{D}}^q] = [\phi(\mathbf{X}^1)\mathbf{B}^1 \cdots \phi(\mathbf{X}^q)\mathbf{B}^q] \tag{14}$$

The final dictionary $\ddot{\mathbf{D}}_f$ is then used to solve for the sparse decomposition $\mathbf{a}_t = [\mathbf{a}_t^1, \ldots, \mathbf{a}_t^q]$ of a given test example $\mathbf{x}_t$, where $\mathbf{a}_t^j$ contains the sparse coefficients associated with respect to the dictionary $\ddot{\mathbf{D}}^j$ of the jth class. Finally the reconstruction error with respect to j^{th} class is computed as:

$$\begin{aligned} r^j &= \|\phi(\mathbf{x}_t) - \phi(\mathbf{X}^j)\mathbf{B}^j \mathbf{a}_t^j\|_2^2 \quad \forall_j \quad j = 1, \ldots, q \\ &= \mathcal{K}(\mathbf{x}_t, \mathbf{x}_t) - 2\mathcal{K}(\mathbf{x}_t, \mathbf{X}^j)\mathbf{B}^j \mathbf{a}_t^j + \mathbf{a}_t^{jT} \mathbf{B}^{jT} \mathcal{K}(\mathbf{X}^j, \mathbf{X}^j)\mathbf{B}^j \mathbf{a}_t^j \end{aligned} \tag{15}$$

In order to have a fair comparison, all the experiments are performed under similar conditions. It should be noted that one is free to use any kernel function or parameter setting to optimize for best performance. In all the experiments the SC stage is solved using KOMP algorithm. Parameters for each kernel are computed

[4] $\mathbf{a}_{[k]}$ is sparse with zeros except at locations indexed by Ω

Table 2
Comparison of classification accuracies on USPS dataset for different methods over 50 trials.

Method	Accuracy %
KKSVD [32]	98.42
KMDL [30]	98.40
Proposed-KSGD	98.40
KSR [15]	97.80
KNN [19]	94.40
ISVM [26]	97.00
SDL [21]	96.46

Table 3
Comparison of classification accuracies in %, on noisy USPS dataset for different methods over 50 trials.

Method	Noise standard deviation σ			
	0.3	0.9	1.2	1.5
KKSVD [32]	97.6	94.5	87.6	83.6
KMOD [32]	97.4	94.1	86.3	82.8
Proposed-KSGD	97.5	94.4	87.6	83.6

Table 4
Comparison of classification accuracies in %, on USPS dataset in absence of some pixels for different methods over 50 trials.

Method	% of missing pixels				
	10	30	50	70	90
KKSVD [32]	97.3	96.5	95.1	87.2	65.8
KMOD [32]	97.1	95.8	94.3	85.1	65.2
Proposed-KSGD	97.3	96.6	95.3	87.2	65.8

using a 5-fold cross validation to achieve best testing performance for the KKSVD approach. Then, the same settings are used in all the other algorithms.

4.1. Digit classification

We used the USPS database which contains 10 classes of 256-dimensional handwritten digits. For each class, 70% training and 30% testing samples were selected randomly. Specifically, dictionary for each class using KSGD, KMOD and KKSVD is learned with the following parameters: 300 atoms, $T_0 = 5$, polynomial kernel of degree 4, error tolerance $\epsilon = 10^{-4}$ as stopping criteria and number of maximum iterations to 200. For the KMLD approach a single dictionary is learned where we set the number of levels to 7, affinity matrix having neighborhood size within the class to 20 samples and between classes to 40 samples.

Table 2 shows the classification accuracies obtained for clean samples using KSGD approach along with existing state-of-the-art approaches. It can be observed that the obtained accuracies for sparse representation based KKSVD, KSR, KMDL and KSGD algorithms are close, but much higher than approaches such as K-nearest neighbor (KNN) [19], invariant support vectors (ISVM) [26] and supervised dictionary learning (SDL) [21]. In addition, we present the results for two cases where the test samples are corrupted by randomly removing different fractions of image pixels or by adding Gaussian noise with different standard deviations in Tables 3 and 4, respectively. One can observe that the proposed KSGD dictionary performs similar to the KKSVD dictionary even in different degradation scenarios.

4.2. Isolated spoken letter classification

We used the ISOLET database which contains spoken letters recorded from 120 training subjects with 52 examples from each speaker and 30 different test subjects. Classification is done using 200-dimensional feature which includes spectral coefficients, contour features, sonorant features, pre-sonorant features, and post-sonorant features etc (see [13] for more details on feature extraction). Dictionary for each class using KSGD, KMOD and KKSVD is learned with the following parameters: Gaussian kernel, 200 atoms, $T_0 = 5$, error tolerance $\epsilon = 10^{-4}$ as stopping criteria and number of maximum iterations to 200. For the KMLD approach a single dictionary is learned where we set the number of levels to 15, affinity matrix having neighborhood size within the class to 10 samples and between classes to 25 samples. In case of KDA approach, we used the spectral regression based KDA method with incremental Cholesky decomposition. The kernel parameters are same as in case of KSGD and the classification is done using support vector machines (SVM).

Table 5
Comparison of classification accuracies on ISOLET dataset for different methods over 50 trials.

Method	Accuracy %	
	Best	Average
KKSVD [32]	97.3	95.8
KMDL [30]	97.3	95.9
Proposed-KSGD	97.2	96.1
KDA [5]	97.4	96.3

Table 6
Comparison of classification accuracies on PD dataset for different methods over 50 trials.

Method	Accuracy %	
	Best	Average
KKSVD [32]	99.4	98.2
KMDL [30]	99.4	99.2
Proposed-KSGD	99.4	98.2

Table 5 shows the classification accuracies on ISOLET dataset. The best classification accuracy is achieved in case of speed up kernel discriminant analysis (KDA) [5] approach, while the performance is similar for the proposed and existing KSRC approaches.

4.3. Parkinson speech classification

We used the PD database which contains multiple types of sound recordings (26 voice samples including sustained vowels, numbers, words and short sentences) from 68 subjects. Specifically, the training data belongs to 20 Parkinson patients (6 female, 14 male) and 20 healthy individuals (10 female, 10 male) while testing data belongs to 28 different Parkinson patients. Classification is done using 26-dimensional feature which includes linear and time-frequency based features e.g., voicing, jitter, pitch, harmonics, pulse and amplitude parameters (see [24] for more details on feature extraction). Dictionary for each class using KKSVD and KSGD is learned with following parameters: Gaussian kernel, 400 atoms, $T_0 = 5$, error tolerance $\epsilon = 10^{-4}$ as stopping criteria and number of maximum iterations to 200. For the KMLD approach a single dictionary is learned where we set the number of levels to 4, affinity matrix having neighborhood size within the class to 18 samples and between classes to 35 samples. For KDA approach, the kernel parameters are same as in case of KSGD and the classification is done using SVM.

Table 6 also shows the classification accuracies on PD dataset. It can be observed that the proposed and existing KSRC approaches achieve much higher accuracies than existing methods based on K-

nearest neighbor (KNN) and SVM approaches i.e., 65.1% and 72.5%, respectively. Further, in [24], authors reported the best classification accuracies of 82.5% and 85% respectively, when KNN and SVM (with linear kernel) classifiers were trained using only some particular voice samples (e.g., vowel 'o' and number 'four'). As explained earlier, this dataset has multiple types of speech recordings and hence existing approaches were not able to generalize well on the training data leading to poor classification accuracies as compared to KSRC based approaches.

5. Summary

We have proposed a kernel sparse DL algorithm, which exploit sparsity of data for classification tasks, in high-dimensional space through an appropriate choice of kernel. The proposed approach learns the dictionary in the coefficient domain rather than the signal domain. This makes the DU procedure efficient, requiring the kernel matrix only once. The proposed DL approach uses a hierarchical method to update atoms of the dictionary in accordance to their energy contribution in representing the training set. The efficiency of the proposed DL algorithm is experimentally demonstrated in various classification tasks. We have also demonstrated the stability and convergence behavior of the proposed DL algorithm. Experimental results demonstrate that the proposed approach has similar performance as in case of existing approaches, but with a huge gain in computational complexity.

Appendix A. The KOMP algorithm

The KOMP algorithm was originally proposed in [32], to generalize OMP using kernels to solve the sparse coding problem. Given a vector $\mathbf{z}$, KOMP finds a coefficient vector $\mathbf{a}$ with at most T_0 non-zero coefficients such that the error $\|\phi(\mathbf{z}) - \ddot{\mathbf{D}}\mathbf{a}\|_2^2$ is minimum, where $\ddot{\mathbf{D}} = \phi(\mathbf{X})\mathbf{B}$ is the dictionary in the high-dimensional space. Algorithm 2 shows the pseudo-code of the KOMP method.

Algorithm 2 KOMP algorithm for obtaining sparse representation in high-dimensional feature space.

Inputs: Signal $\mathbf{z} \in \mathbb{R}^n$, kernel function κ, matrix $\mathbf{X}$, matrix $\mathbf{B}$, and sparsity level T_0

Outputs: Sparse vector $\mathbf{a} \in \mathbb{R}^m$

Initialization: $s = 0$, $I^0 = \emptyset$, $\mathbf{a}^0 = 0$ and $\mathbf{v}^0 = \mathbf{B}\mathbf{a}^0 = 0$

Perform iterations

1: $\tau^i = (\mathcal{K}(\mathbf{z}, \mathbf{X}) - (\mathbf{v}^s)^T \mathcal{K}(\mathbf{X}, \mathbf{X}))\mathbf{a}^i, \forall_i \notin I^{s-1}$

2: $i_{max} = \text{argmax}|\tau^i|, \forall_i \notin I^{s-1}$

3: $I^s \leftarrow I^{s-1} \cup i_{max}$

4: $\mathbf{a}^s = ((\mathbf{B}^{I^s})^T \mathcal{K}(\mathbf{X}, \mathbf{X})\mathbf{B}^{I^s})^{-1}(\mathcal{K}(\mathbf{z}, \mathbf{X})\mathbf{B}^{I^s})^T$

5: $\mathbf{v}^s \leftarrow \mathbf{B}^{I^s}\mathbf{a}^s$, $s \leftarrow s + 1$

Until T_0 **times**

References

[1] V. Abrol, P. Sharma, A.K. Sao, Voiced/nonvoiced detection in compressively sensed speech signals, Speech Commun. 72 (0) (2015) 194–207, doi:10.1016/j.specom.2015.06.001.

[2] D. Achlioptas, F. Mcsherry, Fast computation of low-rank matrix approximations, J. ACM 54 (2) (2007), doi:10.1145/1219092.1219097.

[3] M. Aharon, M. Elad, A. Bruckstein, K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation, IEEE Trans. Signal Process. 54 (11) (2006) 4311–4322, doi:10.1109/TSP.2006.881199.

[4] K. Bache, M. Lichman, UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences, 2013 http://archive.ics.uci.edu/ml.

[5] D. Cai, X. He, J. Han, Speed up kernel discriminant analysis, VLDB J. 20 (1) (2011) 21–33, doi:10.1007/s00778-010-0189-3.

[6] W. Chen, M.R.D. Rodrigues, I.J. Wassell, Projection design for statistical compressive sensing: a tight frame based approach, IEEE Trans. Signal Process. 61 (8) (2013) 2016–2029, doi:10.1109/TSP.2013.2245661.

[7] J. Dattorro, Convex Optimization and Euclidean Distance geometry, Meboo Press, 2008.

[8] P. Drineas, M. Magdon-Ismail, M.W. Mahoney, D.P. Woodruff, Fast approximation of matrix coherence and statistical leverage, J. Mach. Learn. Res. 13 (1) (2012) 3475–3506.

[9] P. Drineas, M.W. Mahoney, On the Nyström method for approximating a gram matrix for improved kernel-based learning, J. Mach. Learn. Res. 6 (2005) 2153–2175.

[10] M. Elad, Sparse and Redundant Representations - From Theory to Applications in Signal and Image Processing, Springer, 2010.

[11] E. Elhamifar, G. Sapiro, R. Vidal, See all by looking at a few: Sparse modeling for finding representative objects, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 1600–1607, doi:10.1109/CVPR.2012.6247852.

[12] K. Engan, S.O. Aase, J. Hakon Husoy, Method of optimal directions for frame design, in: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 5, 1999, pp. 2443–2446, doi:10.1109/ICASSP.1999.760624.

[13] M. Fanty, R.A. Cole, Spoken letter recognition, in: Advances in Neural Information Processing Systems (NIPS), 3, 1990, pp. 220–226.

[14] A.K. Farahat, A. Elgohary, A. Ghodsi, M.S. Kamel, Greedy column subset selection for large-scale data sets, Springer Knowl. Inf. Syst. 45 (1) (2015) 1–34, doi:10.1007/s10115-014-0801-8.

[15] S. Gao, I.W. Tsang, L. Chia, Sparse representation with kernels, IEEE Trans. Image Process. 22 (2) (2013) 423–434, doi:10.1109/TIP.2012.2215620.

[16] N. Hurley, S. Rickard, Comparing measures of sparsity, IEEE Trans. Inf. Theory 55 (10) (2009) 4723–4741, doi:10.1109/TIT.2009.2027527.

[17] M.G. Jafari, M.D. Plumbley, Fast dictionary learning for sparse representations of speech signals, IEEE J. Sel. Top. Signal Process. 5 (5) (2011) 1025–1031, doi:10.1109/JSTSP.2011.2157892.

[18] Z. Jiang, Z. Lin, L.S. Davis, Label consistent K-SVD: learning a discriminative dictionary for recognition, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2651–2664, doi:10.1109/TPAMI.2013.88.

[19] D. Keysers, J. Dahmen, T. Theiner, H. Ney, Experiments with an extended tangent distance, in: Proceedings of the IEEE International Conference on Pattern Recognition (ICPR), 2, 2000, pp. 2038–2042, doi:10.1109/ICPR.2000.906014.

[20] Z. Li, W. Zhou, P. Chang, J. Liu, Z. Yan, T. Wang, F. Li, Kernel sparse representation-based classifier, IEEE Trans. Signal Process. 60 (4) (2012) 1684–1695, doi:10.1109/TSP.2011.2179539.

[21] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, F.R. Bach, Supervised dictionary learning, in: Advances in Neural Information Processing Systems (NIPS), 2009, pp. 1033–1040.

[22] S. Mandal, A. Bhavsar, A.K. Sao, Hierarchical example-based range-image super-resolution with edge-preservation, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), 2014, pp. 3867–3871, doi:10.1109/ICIP.2014.7025785.

[23] S. Mandal, A.K. Sao, Edge preserving single image super resolution in sparse environment, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), 2013, pp. 967–971, doi:10.1109/ICIP.2013.6738200.

[24] B.E. Sakar, M.E. Isenkul, C.O. Sakar, A. Sertbas, F. Gurgen, S. Delil, H. Apaydin, O. Kursun, Collection and analysis of a parkinson speech dataset with multiple types of sound recordings, IEEE J. Biomed. Health Inf. 17 (4) (2013) 828–834, doi:10.1109/JBHI.2013.2245674.

[25] K. Schnass, P. Vandergheynst, Dictionary preconditioning for greedy algorithms, IEEE Trans. Signal Process. 56 (5) (2008) 1994–2002, doi:10.1109/TSP.2007.911494.

[26] B. Schölkopf, P. Simard, A. Smola, V. Vapnik, Prior knowledge in support vector kernels, in: Advances in Neural Information Processing Systems (NIPS), 1998, pp. 640–646.

[27] P. Sharma, V. Abrol, A.D. Dileep, A.K. Sao, Sparse coding based features for speech units classification, in: Proceedings of the 16th INTERSPEECH, ISCA, 2015, pp. 712–715.

[28] P. Sharma, V. Abrol, A.K. Sao, Learned dictionaries for sparse representation based unit selection speech synthesis, in: Proceedings of the IEEE Twenty Second National Conference on Communications (NCC), 2016, pp. 1–5.

[29] J.J. Thiagarajan, K.N. Ramamurthy, A. Spanias, Multilevel dictionary learning for sparse representation of images, in: Proceedings of the IEEE Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE), 2011, pp. 271–276, doi:10.1109/DSP-SPE.2011.5739224.

[30] J.J. Thiagarajan, K.N. Ramamurthy, A. Spanias, Multiple kernel sparse representations for supervised and unsupervised learning, IEEE Trans. Image Process. 23 (7) (2014) 2905–2915, doi:10.1109/TIP.2014.2322938.

[31] I. Tosic, P. Frossard, Dictionary learning, IEEE Signal Process. Mag. 28 (2) (2011) 27–38, doi:10.1109/MSP.2010.939537.

[32] H. Van Nguyen, V.M. Patel, N.M. Nasrabadi, R. Chellappa, Design of non-linear kernel dictionaries for object recognition, IEEE Trans. Image Process. 22 (12) (2013) 5123–5135, doi:10.1109/TIP.2013.2282078.

[33] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 210–227, doi:10.1109/TPAMI.2008.79.

[34] J. Yin, Z. Liu, Z. Jin, W. Yang, Kernel sparse representation based classification, Neurocomputing 77 (1) (2012) 120–128, doi:10.1016/j.neucom.2011.08.018.