Neural Computing and Applications manuscript No.
(will be inserted by the editor)

Sparse coding and dictionary learning with class-specific group sparsity

Yuping Sun[a] · **Yuhui Quan**[b] · **Jia Fu**[c,*]

Abstract In recent years, sparse coding via dictionary learning has been widely used in many applications for exploiting sparsity patterns of data. For classification, useful sparsity patterns should have discrimination, which cannot be well achieved by standard sparse coding techniques. In this paper, we investigate structured sparse coding for obtaining discriminative class-specific group sparsity patterns in the context of classification. A structured dictionary learning approach for sparse coding is proposed by considering the $\ell_{2,0}$ norm on each class of data. An efficient numerical algorithm with global convergence is developed for solving the related challenging $\ell_{2,0}$ minimization problem. The learned dictionary is decomposed into class-specific dictionaries for the classification that is done according to the minimum reconstruction error among all the classes. For evaluation, the proposed method was applied to classifying both the synthetic data and real-world data. The experiments show the competitive performance of the proposed method in comparison with several existing discriminative sparse coding methods.

Keywords Structured sparsity · Group sparse coding · Discriminative dictionary learning · Classification

1 Introduction

Recent studies have shown the success of sparse modeling in analyzing high dimensional data [15, 51, 41, 48]. The basic assumption in sparse modeling is that signals of interest can be succinctly expressed in a linear manner under some suitable system. The representative elements for expressing signals are often referred to as *atoms* and the total set of all such atoms is called a *dictionary*. The computational method for sparse modeling, which aims at finding both the dictionary and the sparse coefficients from input signals, is usually called *sparse coding*. To be more specific, given a set of signals $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_p] \in \mathbb{R}^{n\times p}$, sparse coding is to determine a collection of dictionary atoms $\boldsymbol{D} = [\boldsymbol{d}_1, \ldots, \boldsymbol{d}_m] \in \mathbb{R}^{n\times m}$, together with a set of sparse coefficients $\boldsymbol{C} = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_p] \in \mathbb{R}^{m\times p}$, such that each signal $\boldsymbol{y}_i$ can be well represented by a linear combination of only a few atoms taken from $\boldsymbol{D}$:

$$\boldsymbol{y}_j \approx \sum_{\ell=1}^{m} \boldsymbol{c}_j(\ell)\boldsymbol{d}_\ell,$$

where $\boldsymbol{c}_j$ is a sparse vector with most entries being zero or close to zero.

The sparse coding is often formulated as the following minimization problem:

$$\min_{\boldsymbol{D}\in\mathcal{X},\boldsymbol{C}} \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{C}\|_F^2 + \lambda\phi(\boldsymbol{C}), \tag{1}$$

where the parameter λ determines the sparsity degree of the codes, and

$$\mathcal{X} = \{\boldsymbol{X} \in \mathbb{R}^{n\times m} : \|\boldsymbol{x}_j\|_2 = 1, 1 \le j \le m\},$$

denotes the feasible set of dictionary, which ensures that all atoms in the dictionary are appropriately normalized. The function ϕ is for encouraging the sparsity of the codes. A natural choice of ϕ is the ℓ_0 norm $\|\cdot\|_0$ that counts the number of non-zero entries. The ℓ_0 norm based model is a non-convex and non-smooth optimization problem which is difficult to solve. Motivated by the theoretical breakthroughs [9, 14] as well as the biological observations [27], the ℓ_1 norm is often used as a convex relaxation of ℓ_0 norm to facilitate the computation.

[*] Corresponding author (scutjfu@outlook.com)
[a] School of Automation Science & Engineering, South China University of Technology, Guangzhou, 510641, China
[b] School of Computer Science & Engineering, South China University of Technology, Guangzhou, 510006, China
[c] School of Journalism & Communication, South China University of Technology, Guangzhou, 510006, China

Many existing sparse coding based recognition systems purely pursue the sparsity of codes, regardless of the high-order information existing in signals (e.g. intrinsic relationships among inner-class signals), which is sub-optimal regarding the discriminability needed for classification and recognition. Given an observation matrix

$$\boldsymbol{X} = [\boldsymbol{X}_{[1]}, \boldsymbol{X}_{[2]}, \ldots, \boldsymbol{X}_{[K]}],$$

where the sub-matrix $\boldsymbol{X}_{[k]}$ denotes the observations from the kth category. The corresponding optimal sparse coefficient matrix $\boldsymbol{C}$ under an ideally discriminative dictionary should be in the following form:

$$\boldsymbol{C}^* \triangleq \begin{bmatrix} \boldsymbol{C}_{1,1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{C}_{2,2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{C}_{K,K} \end{bmatrix}. \tag{2}$$

1.1 Motivation

One key issue in designing sparsity-driven recognition systems is how to construct a useful dictionary such that it can encode the given data in the form of (2). While some recent approaches [53, 56, 3, 57, 22] have attempted to pursue such optimal structured sparsity either explicitly or implicitly, these approaches have obvious disadvantages. For instance, the sparsity constrain and low rank constrain used in [57] are implemented with the ℓ_1 norm and the spectral norm respectively, both of which could lead to biased solutions due to their penalization on large coefficients [3]. The supervised sparse coding methods [34, 56] only consider lowering classification errors regardless of the structures on sparse coefficients, which often result in sub-optimal solutions. The methods that explicitly structure sparse code, such as group Lasso [53] and label consistency [22], need predefinitions on sparsity patterns, which is inflexible for real applications and inaccurate when data has large variations whereas the dictionary size is limited.

The aforementioned issues gave us inspiration to propose an effective structured sparse coding method in the context of classification, which is capable of automatically discovering the underlying structures of data and obtaining the sparsity patterns of the form (2). Our basic idea is to use group sparsity for modeling each class of data, which encourages class-specific group sparsity patterns under the learned dictionary. In other words, the sparsity penalty function ϕ is of the form

$$\phi(\boldsymbol{C}) = \sum_{k=1}^{K} \psi(\boldsymbol{C}_{[k]}),$$

where $\boldsymbol{C}_{[k]}$ is the sparse code of the data from the kth class, and ψ is a penalty function for promoting group-structure sparsity of $\boldsymbol{C}_{[k]}$. In similar spirit, our preliminary work [47] proposed a $\ell_{2,1}$ penalty model with a reweighting scheme, which avoids solving the NP-hard problems regarding sparsity. However, the reweighting scheme is subtle and the convergence of the related algorithm cannot be guaranteed. Furthermore, the design of stable reweighting schemes is challenging for many real applications. Motivated by these facts, in this paper, by using the $\ell_{2,0}$ norm to define ψ, we propose a $\ell_{2,0}$-group sparse coding model, as well as an efficient algorithm with global convergence developed to solve the related challenging minimization problem.

1.2 Contribution

In this paper, we proposed a $\ell_{2,0}$ structured sparse coding method for exploiting the class-specific joint structured sparsity patterns existing in labeled data for classification. The proposed method has several advantages over many existing sparse coding methods. Compared with the reconstructive sparse modeling methods like [1, 3], the proposed one incorporates supervised information into sparse code and hence has improved discrimination. Instead of purely considering low discrimination error like the supervised sparse coding methods [49, 56], the proposed method pursues the sparsity patterns which have not only discrimination but also structures. Compared to many exiting structured sparse coding methods like [42, 22], the proposed method need no predefinition on the sparsity pattern and hence are more adaptive to data. Compared with its closely-related work [47] which is based on reweighted $\ell_{2,1}$ penalty, the proposed method avoids the design of reweighting scheme which can be very difficult for real applications, and meanwhile it yields comparable or even better results. Furthermore, the proposed method has the global convergence in its numerical algorithm. The proposed method was tested on synthetic data as well as in some recognition tasks with real-world data, and in the experiments it has exhibited promising performance in comparison with several recent sparse coding methods.

1.3 Notation and Organization

Throughout the paper, the notations are used as follows. Regular lower letters are used for scalars, bold lower letters for column vectors, bold upper letters for matrices, and calligraphic English alphabets for sets. Let $\boldsymbol{Y} = [Y_{i,j}]_{i,j}$ denote a matrix, $\boldsymbol{y}^i$ and $\boldsymbol{y}_j$ denote the i-th row and j-th column of $\boldsymbol{Y}$ respectively, and $\boldsymbol{Y}_{[k]}$ denote a sub-matrix of $\boldsymbol{Y}$ collecting columns from the k-th category. For a vector $\boldsymbol{y}$, y_i denotes its i-th element. Given a matrix $\boldsymbol{Y}$, its Frobenius norm is defined as $\|\boldsymbol{Y}\|_F = \sqrt{\sum_{i,j} Y_{i,j}^2}$, its $\ell_{2,p}$-norm is defined as $\|\boldsymbol{Y}\|_{2,p} = \sum_i \| \sum_j Y_{i,j}^2 \|_p$ for $0 \leq p \leq 1$, its ℓ_0-norm is defined as $\|\boldsymbol{Y}\|_0 = \sum_{i,j} \|\boldsymbol{Y}_{i,j}\|_0$, where for a scalar x, $\|x\|_0$

equals 0 if $x = 0$ and 1 otherwise. Given a positive constant $\lambda > 0$ and a vector $\boldsymbol{x}$, the isotropic hard shrinkage operator $H_\lambda(\boldsymbol{x})$ is defined by $H_\lambda(\boldsymbol{x}) = \frac{1}{2}(1+sgn(\|\boldsymbol{x}\|_2-\lambda))\boldsymbol{x}$. The proximal operator is defined by

$$\mathrm{Prox}_t^O(\boldsymbol{x}) := \underset{\boldsymbol{u}}{\mathrm{argmin}}\, O(\boldsymbol{u}) + \frac{t}{2}\|\boldsymbol{u}-\boldsymbol{x}\|_F^2,$$

where O is a proper lower semi-continuous function. The indicator function $I_{\mathcal{X}}(\boldsymbol{X})$ is defined as a function of a matrix $\boldsymbol{X}$ that satisfies $I_{\mathcal{X}}(\boldsymbol{X}) = 0$ if $\boldsymbol{X} \in \mathcal{X}$ and $+\infty$ otherwise.

The rest of this paper is organized as follows. Section 2 gives a brief review on the related work, Section 3 presents the details of the proposed structured sparse coding methods, Section 4 is devoted to the experimental evaluation, and the paper is concluded in Section 5.

2 Related work

2.1 Pursuing structured sparsity

As a natural extension of the standard sparsity, structured sparsity has achieved a lot of attention in statistical learning and compressive sensing [19, 42]. Group sparsity is one often-used structured sparsity, which assumes atoms are selected to express input signals in the group-wise manner rather than the singleton-wise one. In group sparse coding, the coding coefficients in the same group tend to be zero or nonzero simultaneously, while the number of active groups is penalized by the sparsity term. Using group sparsity can help to access or enforce the dependencies between dictionary atoms and to control the expressive power of the reconstruction model [23, 20]. There are a few methods combining group sparse coding with dictionary learning for classification, such as [44, 30, 54], which use disjoint groups of atoms for sparse coding. In practice, groups are not necessarily disjoint but can overlap. Using overlapping groups can improve the compactness of dictionary as dictionary atoms can belong to more than one group, and it may bring improvement on performance; see e.g. [20, 2, 13, 39]. In most existing group sparse coding methods, groups are often assumed to be static and fixed a priori.

The aforementioned group sparse coding methods do not consider the relationship between groups. Thus, some approaches seek for more complex grouping strategies for group sparse coding. For encoding hierarchical sparsity that often seen in wavelet representation and pyramid representation of images, the tree sparsity has been investigated in [24, 21, 37]. The graph sparsity [2, 7, 43] is more general, which indicates direct graph structure on the partitions of sparse coefficients. The main disadvantage of using tree sparsity or graph sparsity is that the resultant coding complexity is often very high compared with using standard sparsity.

In many applications, not only the correlations of sparse coefficients, but also the high-order information existing in data and sparse codes, should be considered. With this goal, several collaborative sparse coding approaches [33, 8, 11, 33, 48] have been proposed to exploit structured sparsity in a collaborative manner, which utilize the similarity of intra-class data for structured sparse coding. Furthermore, there exists a growing interest in exploiting block structured sparsity, where all related samples are jointly encoded to share the same sparse groups of features. The block sparse coding can be done either explicitly or implicitly, and implicit methods often introduce some low-rank constrains to sparse coding; see e.g. [16, 57, 12].

2.2 Sparse coding based classification

Sparse coding can be used for classification in different manners. The most simple way is to encode data by unsupervised sparse coding and directly use the obtained sparse coefficients as features for training classifiers; see e.g. [26, 49]. Such methods can faithfully represent the samples, but separating the coding process from classifier learning is often not optimal, as sparse codes are not enforced to be discriminative for classification [1, 34, 5].

To improve the discriminability of sparse coding, plenty of approaches incorporate the classification penalty into the sparse coding process, which could learn a dictionary with enhanced discrimination [56, 38, 29, 6]. Such approaches are often referred to as discriminative sparse coding methods and can be classified into two categories. In the first category, a dictionary shared by all classes is employed or learned with certain discrimination terms assessed on the sparse representation coefficients. The common choice for the discrimination term includes the class separation criterion (e.g. Fisher discriminant criterion [50, 52]), prediction loss (e.g. linear predictive error [56, 22, 35, 36], logistic loss [29] and hinge loss [28]), and label inconsistency penalty [29]. Though the discrimination of sparse code is enhanced, these methods could not guarantee interesting patterns on sparse code.

In the second category of discriminative sparse coding, a dictionary is partitioned into multiple sub-dictionaries whose atoms are related to certain class labels [58, 17], and the reconstruction error on each sub-dictionary is utilized for classification. In this case, the discrimination of sparse code comes from the class-related sub-dictionaries, and it can be further improved by inducing incoherence between dictionaries [38, 25], using Fisher discriminant [50, 52], or incorporating task-driven discrimination terms (e.g. [10]). In this kind of method, the assignment of class labels to sub dictionaries is crucial. Some most recent works focus on jointly sparse coding and revealing the relation between dictionary atoms and class labels [52, 55], and our proposed method falls into this line of work.

3 Our method

To pursue the sparsity patterns of the form (2), we develop a new sparse coding model by using the $\ell_{2,0}$ group sparsity penalty on each category of data. For well solving the related $\ell_{2,0}$ minimization problem, we provide an efficient numerical algorithm with the global convergence property. We also present an effective classification scheme accordingly.

3.1 Model

Recall that $\boldsymbol{Y} = [\boldsymbol{Y}_{[1]}, \boldsymbol{Y}_{[2]}, \dots, \boldsymbol{Y}_{[K]}]$ denotes a set of training samples from K categories with $\boldsymbol{Y}_{[k]}$ denoting the training samples from k-th category. Also recall that $\mathcal{X} = \{\boldsymbol{D} \in \mathbb{R}^{n\times m} : \|\boldsymbol{d}_j\|_2 = 1, 1 \le j \le m\}$ denote the set of dictionary $\boldsymbol{D}$ with ℓ_2-normalized atoms. Our proposal is a group sparse coding model built upon the $\ell_{2,0}$ norm, which is defined as follows:

$$\operatorname*{argmin}_{\boldsymbol{D}\in\mathcal{X},\boldsymbol{C}} \sum_{k=1}^{K} \frac{1}{2}\|\boldsymbol{Y}_{[k]} - \boldsymbol{D}\boldsymbol{C}_{[k]}\|_F^2 + \lambda\|\boldsymbol{C}_{[k]}\|_{2,0}, \tag{3}$$

where λ is a scalar for balancing the contribution of the reconstruction term and the sparsity term, and $\boldsymbol{C}_{[k]}$ is a submatrix of $\boldsymbol{C}$ which collects the sparse codes of signals from the k-th category. In other words, $\boldsymbol{C}_{[k]}$ contains the sparse coefficients of $\boldsymbol{Y}_{[k]}$.

It is worth mentioning that a reweighted $\ell_{2,1}$ norm was proposed in our preliminary work [47]. In comparison, the proposed model (3) uses $\|\boldsymbol{C}_{[k]}\|_{2,0}$ as the sparsity penalty without weighting, which avoids both the design and computation of the reweighting. Moreover, unlike its $\ell_{2,1}$ convex relaxation, the $\ell_{2,0}$ penalty gives an accurate measure on sparsity without bias.

3.2 Numerical algorithm

Owing to the $\ell_{2,0}$ penalty term as well as the ambiguity between the sparse codes $\boldsymbol{C}$ and the dictionary $\boldsymbol{D}$, the problem (3) is a challenging non-smooth and non-convex optimization problem. Motivated by [3], we use the alternating proximal linearized method [?] to develop an effective solver for (3) with the global convergence property. The basic idea is to solve the original problem (3) by breaking it into several subproblems, each of which is transferred to proximal linearized minimization problem. For the convenience of presentation, we first define

$$\begin{cases} E(\boldsymbol{C}_{[k]}) = \lambda\|\boldsymbol{C}_{[k]}\|_{2,0}, \\ J(\boldsymbol{C},\boldsymbol{D}) = \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{C}\|_F^2. \\ L(\boldsymbol{D}) = I_{\mathcal{X}}(\boldsymbol{D}). \end{cases} \tag{4}$$

Using the alternating proximal linearized method, the problem (3) is decomposed into the following subproblems,

$$\begin{aligned} \boldsymbol{C}_{[k]}^{(t+1)} &\in \mathrm{Prox}_{\beta_k^{(t)}}^{E}(\boldsymbol{C}_{[k]}^{(t)} - \frac{1}{\beta_k^{(t)}}\nabla_{\boldsymbol{C}_{[k]}^{(t)}} J(\widetilde{\boldsymbol{C}}_{[k]}^{(t)}, \boldsymbol{D}^{(t)})), \\ \boldsymbol{d}_j^{(t+1)} &\in \mathrm{Prox}_{\mu_j^{(t)}}^{L(\bar{\boldsymbol{D}}_j^{(t)})}(\boldsymbol{d}_j^{(t)} - \frac{1}{\mu_j^{(t)}}\nabla_{\boldsymbol{d}_j} J(\boldsymbol{C}^{(t+1)}, \widetilde{\boldsymbol{D}}^{(t)})), \end{aligned} \tag{5}$$

where $\beta_k^{(t)}$ and $\mu_j^{(t)}$ are step sizes, and

$$\begin{aligned} \widetilde{\boldsymbol{C}}_{[k]}^{(t)} &= [\boldsymbol{C}_{[1]}^{(t+1)}, \cdots, \boldsymbol{C}_{[k-1]}^{(t+1)}, \boldsymbol{C}_{[k]}^{(t)}, \boldsymbol{C}_{[k+1]}^{(t)}, \cdots, \boldsymbol{C}_{[K]}^{(t)}], \\ \bar{\boldsymbol{D}}_j^{(t)} &= [\boldsymbol{d}_1^{(t+1)}, \cdots, \boldsymbol{d}_{j-1}^{(t+1)}, \boldsymbol{d}_j, \boldsymbol{d}_{j+1}^{(t)}, \cdots, \boldsymbol{d}_m^{(t)}], \\ \widetilde{\boldsymbol{D}}_j^{(t)} &= [\boldsymbol{d}_1^{(t+1)}, \cdots, \boldsymbol{d}_{j-1}^{(t+1)}, \boldsymbol{d}_j^{(t)}, \boldsymbol{d}_{j+1}^{(t)}, \cdots, \boldsymbol{d}_m^{(t)}]. \end{aligned}$$

Given an initial dictionary $\boldsymbol{D}^{(0)}$, we update the estimate of $(\boldsymbol{C}, \boldsymbol{D})$ via the following alternating iterative scheme:

Sparse Approximation. At the beginning of the $(t+1)$-th iteration, given dictionary $\boldsymbol{D}^{(t)}$, the update of sparse code $\boldsymbol{C}_{[k]}$ for the k-th category using (5) can be rewritten as

$$\boldsymbol{C}_{[k]}^{(t+1)} \in \operatorname*{argmin}_{\boldsymbol{C}} \|\boldsymbol{C}\|_{2,0} + \frac{\beta_k^{(t)}}{2\lambda}\|\boldsymbol{C} - \boldsymbol{C}_{[k]}^{*}\|_F^2. \tag{6}$$

where

$$\boldsymbol{C}_{[k]}^{*} = \boldsymbol{C}_{[k]}^{(t)} - \frac{1}{\beta_k^{(t)}}\nabla_{\boldsymbol{C}_{[k]}^{(t)}} J(\widetilde{\boldsymbol{C}}_{[k]}^{(t)}, \boldsymbol{D}^{(t)}).$$

The problem (6) has explicit solution given by isotropic hard shrinkage on each row of $\boldsymbol{C}_{[k]}^{*}$, i.e.,

$$\boldsymbol{c}_{[k]}^{i,(t+1)} = H_{\frac{2\lambda}{\mu^{(t)}}}(\boldsymbol{c}_{[k]}^{i,*}),$$

where $\boldsymbol{c}_{[k]}^{i,(t+1)}$ and $\boldsymbol{c}_{[k]}^{i,*}$ correspond to the i-th row of $\boldsymbol{C}_{[k]}^{(t+1)}$ and $\boldsymbol{C}_{[k]}^{*}$ respectively.

Dictionary Refinement. With the sparse codes $\boldsymbol{C}^{(t+1)}$ from the previous step, we update dictionary $\boldsymbol{D}$ atom by atom as follows,

$$\boldsymbol{d}_j^{(t+1)} \in \operatorname*{argmin}_{\|\boldsymbol{d}_j\|_2=1} \|\boldsymbol{d}_j - \boldsymbol{s}_j^{(t)}\|_2, \quad \forall j, \tag{7}$$

where $\boldsymbol{s}_j^{(t)} = \boldsymbol{d}_j^{(t)} - \frac{1}{\mu_j^{(t)}}\nabla_{\boldsymbol{d}_j} J(\boldsymbol{C}^{(t+1)}, \widetilde{\boldsymbol{D}}^{(t)})$. Thus a closed-form solution to the problem (7) is given by

$$\boldsymbol{d}_j^{(t+1)} = \boldsymbol{s}_j^{(t)}/\|\boldsymbol{s}_j^{(t)}\|_2, \quad \forall j. \tag{8}$$

The above algorithm has the global convergence property, which is proved in Section 3.4.

3.3 Initialization scheme and classification strategy

To start the above optimization process, we need to initialize the dictionary. The choice of initial dictionary $\boldsymbol{D}_0$ is crucial, as the algorithm may converge to different critical points with different $\boldsymbol{D}_0$s due to the non-convexity and non-smoothness of the optimization problem. A bad initial dictionary, e.g., Gaussian random dictionary, is likely to yield depressed performance. In our implementation, we initialize $\boldsymbol{D}_0$ by uniformly random sampling from each class of training data, and the sampling number in each class is set equal. This scheme has also been used in many existing approaches [50, 52]. In experience, the classification performance using this scheme is stable regardless the randomness of sampling. It is worth mentioning that analytic dictionaries such as truncated DCT are not good choices for dictionary initialization when input data are image feature vectors but not the original images or image patches.

Once dictionary $\boldsymbol{D}$ is learned, we construct multiple sets of atoms shared by samples from different categories, i.e., $\boldsymbol{i}(k) = \{i : | \, \|\boldsymbol{c}^i_{[k]}\|_2 > \theta, 1 \leq i \leq m\}$, where θ is a sufficiently small positive constant (e.g., 10^{-6}). Each set of atoms $\boldsymbol{D}_{[\boldsymbol{i}(k)]}$ is associated with the corresponding class-specific structured sparsity pattern and thus can be used to classify new samples. Given a test sample $\overline{\boldsymbol{y}}$, we calculate its representation $\overline{\boldsymbol{c}}(k)$ on each set of atoms $\boldsymbol{D}_{[\boldsymbol{i}(k)]}$ as follows,

$$\overline{\boldsymbol{c}}(k) = \underset{\boldsymbol{c}}{\operatorname{argmin}} \frac{1}{2}\|\overline{\boldsymbol{y}} - \boldsymbol{D}_{[\boldsymbol{i}(k)]}\boldsymbol{c}\|_2^2 + \frac{\alpha}{2}\|\boldsymbol{c}\|_2^2, \tag{9}$$

where α is a scalar for stability of solution. The problem (9) could be solved via ridge regression with a closed-form solution given by

$$\overline{\boldsymbol{c}}(k) = \left(\boldsymbol{D}^\top_{[\boldsymbol{i}(k)]}\boldsymbol{D}_{[\boldsymbol{i}(k)]} + \alpha \boldsymbol{I}_m\right)^{-1} \boldsymbol{D}^\top_{[\boldsymbol{i}(k)]}\overline{\boldsymbol{y}}$$

Finally, the class label of $\overline{\boldsymbol{y}}$ is set to be the index which corresponds to the minimum reconstruction error, i.e.,

$$\text{identity}(\overline{\boldsymbol{y}}) = \underset{k}{\operatorname{argmin}} \|\overline{\boldsymbol{y}} - \boldsymbol{D}_{\boldsymbol{i}(k)}\overline{\boldsymbol{c}}(k)\|_2^2. \tag{10}$$

3.4 Convergence analysis

The developed algorithm above for solving the problem in (3) can be proved to have excellent property of global convergence (i.e. whole sequence convergence), which is summarized in the following theorem.

Theorem 1: The solution sequence $\{\boldsymbol{C}^{(t)}, \boldsymbol{D}^{(t)}\}$ which is generated by the iteration procedure in Section 3.2, is a Cauchy sequence and converges to a critical point of (3).

Proof: The proof can be done by checking the conditions of the Theorem 1 in [**?**]. Here we only provide a straightforward sketch proof. Firstly, the objective functions $F(\boldsymbol{C}_{[k]})$, $Q(\boldsymbol{C}_{[k]}, \boldsymbol{D})$ and $G(\boldsymbol{D})$ are obviously semi-algebraic functions. Therefore the whole objective function in (3), which can be written as $\sum_{k=1}^K (F(\boldsymbol{C}_{[k]}) + Q(\boldsymbol{C}_{[k]}, \boldsymbol{D})) + G(\boldsymbol{D})$, is a semi-algebraic function, and hence satisfy the *Kurdyka-Lojasiewicz* property. Secondly, the sequence $\{\boldsymbol{C}^{(t)}, \boldsymbol{D}^{(t)}\}$ is bounded and the step sizes $\mu_j^{(t)}$, $\mu_c^{(t)}$ are bounded. Thirdly, it is easy to verify that $\nabla_{\boldsymbol{C}_{[k]}} Q(\boldsymbol{C}_{[k]}, \boldsymbol{D})$ and $\nabla_{\boldsymbol{d}_j} f(\boldsymbol{C}, \boldsymbol{D})$ have Lipschitz constant on any bounded set. From the Theorem 1 in [**?**], we conclude that the solution sequence is a Cauchy sequence and converges to a critical point of (3).

From Theorem 1, the elements of the solution sequence $\{\boldsymbol{C}^{(t)}, \boldsymbol{D}^{(t)}\}$ become arbitrarily close to each other when the iteration goes on. In other words, global convergence property can be achieved by the developed algorithm. This allows us to stop the algorithm when the changes of solution sequence is sufficiently small.

4 Experiment

In this section, we evaluate the classification performance of the proposed method by using synthetic data as well as real-world datasets.

4.1 Protocol

Several recent sparse coding approaches are used for comparison in the experiments, including SRC [46] [1], DLSI [38], D-KSVD [56], FDDL [50], JDL [58], COPAR [25], LC-KSVD [22], LDL [52], L0DL [4], MCDL [35], and our preliminary work RL21DL [47]. For convenience of presentation, our proposed method in this paper is denoted by L20DL.

There are mainly three parameters to be set in L20DL, including the regularization parameters α and λ, and the dictionary size m. The setting of these three parameters depends on the application and data. In all the experiments, if not specified, a five-fold cross validation scheme is used to optimize α and λ,[2] and the dictionary sizes of all compared methods are set the same. A detailed analysis of the influence of parameter selection on the performance of the proposed method is presented in Section 4.2. As mentioned in 3.3, the initial dictionary of L20DL is generated via randomly selecting the same number of samples from each class as the dictionary atoms.

4.2 Evaluation on synthetic data

The synthetic data is generated as follows. A random orthogonal dictionary $\boldsymbol{D} \in \mathbb{R}^{m \times m}$ is generated by taking the

[1] For fair comparison, the dictionary size of SRC is set the same as our method in all the experiments.

[2] In practice, we found that the parameter α can be simply set to a sufficiently small positive constant.

left orthogonal matrix from the singular value decomposition of a normal matrix. Then three sparse matrices $\boldsymbol{X}(k) \in \mathbb{R}^{m \times q}$, $k = 1, 2, 3$, are generated such that each of them has: (1) T rows of non-zeros with random values drawn from the normal distribution, (2) t out of T rows share the same row indices among these three matrices, and (3) the indices of the remaining $T - t$ rows are totally distinct across different matrices. Finally, the data matrix $\boldsymbol{Y}$ for test is generated by $\boldsymbol{Y} = \boldsymbol{D}\boldsymbol{X} + \boldsymbol{N}$ where $\boldsymbol{X} = [\boldsymbol{X}(1), \boldsymbol{X}(2), \boldsymbol{X}(3)]$ and $\boldsymbol{N}$ is a Gaussian noise matrix with mean 0 and standard deviation 0.05. By the above scheme, we generate three classes of signals, where each class of signals lies at a T-dimensional subspace and the subspaces of all the three classes have a t-dimensional overlap.

We compare the performance of L20DL with several existing sparse coding methods, including SRC, D-KSVD, FDDL and RL21DL, in terms of classification accuracy as well as average running time during both training phase and test phase. For fair comparison, the parameters of all the compared methods are finely tuned up via five-fold cross validation, while the experimental environments and protocols used by these methods are set as the same. The parameters of L20DL are set as follows: $m = 21$, $T = 6$, $t = 3$, and $q = 50$. The results over 10 runs are listed in Table 1. It can be seen that the proposed L20DL method performed on a par with RL21DL and perform better than SRC, D-KSVD and FDDL. Regarding the computational efficiency, in training the proposed L20DL method is the fastest one among all the tested methods, while in testing L20DL and D-KSVD are faster than SRC, RL21DL and FDDL. To verify the correctness of our algorithm, we show the objective function value decay and global convergence behaviors of L20DL in Figure 1(a) and Figure 1(b). Figure 1(c) and(d) show the coding results by L20DL on a synthetic data, which demonstrates the proposed L20DL method has the capability of discovering the low-dimensional subspaces of high-dimensional data and pursuing class-specific structured patterns.

Table 1: The classification accuracy (%) and computational time (ms) of the compared methods on the synthetic data.

Method	Accuracy	Training time per iteration	Test time per sample
SRC [46]	61.33	-	0.08
D-KSVD [56]	51.33	2.13	0.03
FDDL [50]	78.00	17.28	3.22
RL21DL	93.17	1.34	0.15
L20DL	93.17	1.12	0.03

To analyze the influence of the parameters α, λ and m in L20DL, we conducted a test on the synthetic data by alternatively adjusting one of the parameters and fixing the rest ones. The RL21DL algorithm is also included in this test for comparison. In both RL21DL and L20DL, we set the parameters m and λ to be 21 and 0.25 respectively. The parameter α is set to be 0.001 in RL21DL while 10^{-6} in L20DL. The effects of parameter selection on classification performance of L20DL are shown in Figure 2. We can see from the Figure 2 (a) that, the performance of the tested two methods, especially RL21DL, drop much when m is small. The reason is that using too few dictionary atoms cannot well capture the underlying structure of the data, which makes the generated sparse coefficients insufficiently discriminative. For both these two methods, the performance increase gradually until that m is larger than 21, which agrees with the original generation of the synthetic data. It can be seen from Figure 2 (b) and (c) that the performances of both RL21DL and L20DL are not sensitive to λ and α within a small range. On the whole, the proposed L20DL method performs better than RL21DL on the synthetic dataset.

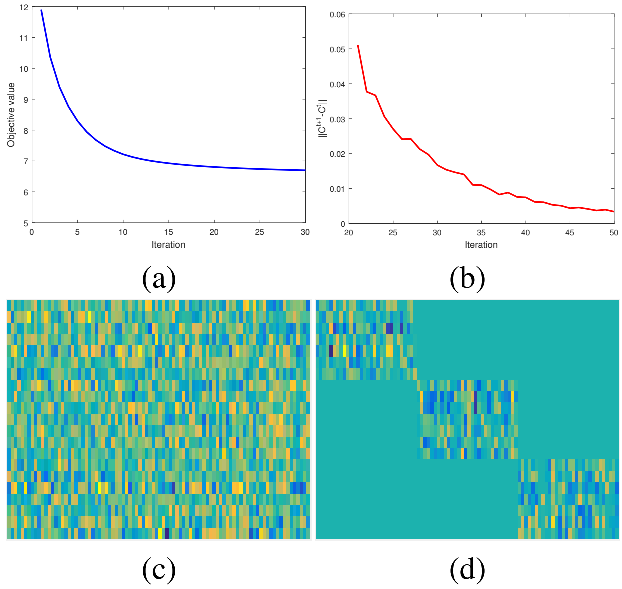
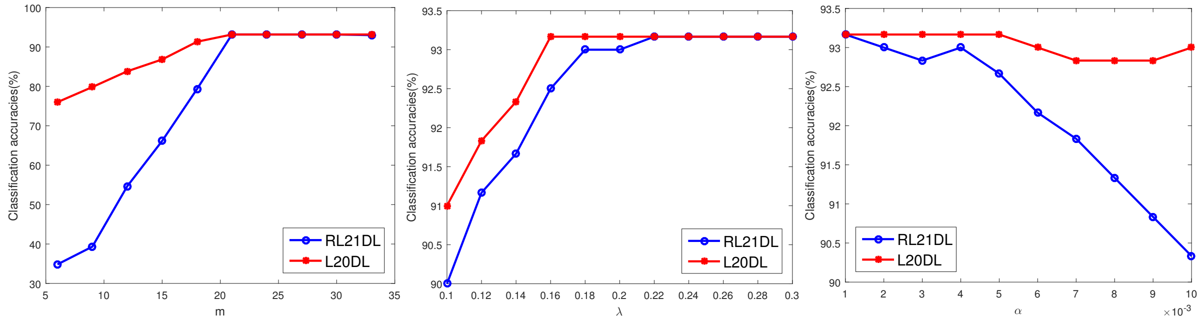


Fig. 1: Some results of L20DL on the synthetic data. (a) Objective function value decay of L20DL; (b) The norms of the increments of the coefficient sequence $\{\boldsymbol{C}^{(t)}\}_t$ generated by L20DL; (c) The synthetic data for training; (d) The coding results of (c) by L20DL.

4.3 Evaluation on real data

4.3.1 Face recognition on Extended Yale B

We conducted face recognition experiments on the Extended Yale B dataset [18]. The Extended Yale B dataset contains $2,414$ frontal face images of 38 persons, some samples of which are shown in Figure 3. There are about 64 images for



(a)

(b)

(c)

Fig. 2: Classification accuracies (%) versus the parameters m, λ, and α of the RL21DL and L20DL methods applied on the synthetic data, which are shown in (a), (b) and (c) respectively.

each person with different illumination conditions and expressions. Following the experimental settings in [56], we used the cropped version (192 × 168 pixels) of the original images and then projected each cropped image into a 504-dimensional feature vector by employing a random matrix from zero-mean normal distribution. To ensure results reliable, we repeated the experiments 10 times and report the average results for comparison. In each run, we randomly split the dataset into two halves, and use one half with 32 images per person for training and the other half for testing.

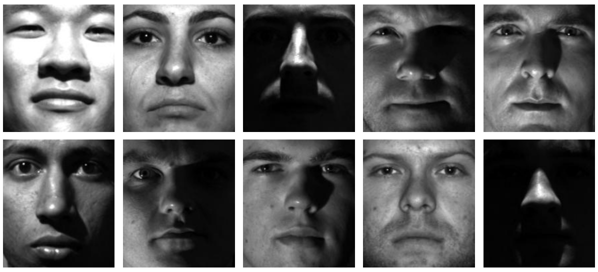


Fig. 3: Sample images from the extended Yale B dataset.

The default dictionary size is set as $m = 570$, and the parameters α and λ and are set as 10^{-6} and 0.01 respectively in L20DL. The performance of LLC, SRC, KSVD, D-KSVD, LC-KSVD and RL21DL are included for comparison. The experimental results of all the compared methods are summarized in Table 2. It can be observed from Table 2 that the proposed method is very competitive among all the compared methods. The proposed L20DL method outperformed many recent approaches such as SRC, KSVD, LLC, D-KSVD and RL21DL.

Table 2: Face recognition results (%) of the compared methods applied on the Extended YaleB dataset.

Method	Accuracy	Method	Accuracy
KSVD [1]	93.10	SRC [46]	80.50
D-KSVD [56]	94.10	LC-KSVD [22]	95.00
LLC [45]	90.70	RL21DL	94.52
L0DL [4]	95.12	L20DL	94.97
MCDL [35]	95.79		

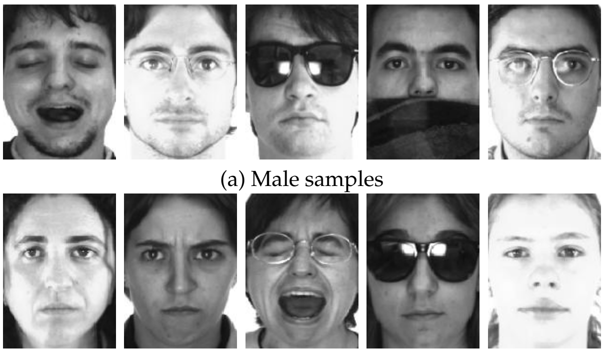
4.3.2 Gender classification on AR Face

We applied the proposed L20DL method to gender classification on the AR face dataset [31]. Following the experimental setting in [50], we chose a non-occluded subset from the original database, which consists of 50 females and 50 males and includes 14 samples per subject. Some selected sample images are shown in Figure 4. The images of the first 25 females and 25 males are chosen to be the training set, while the rest are for test. We reduced the dimension of each image to 300 via PCA.

Table 3: Gender classification results (%) of the compared methods applied on the AR database.

Method	Accuracy	Method	Accuracy
DLSI [38]	93.70	LDL [52]	95.00
COPAR [25]	93.00	RL21DL	94.00
JDL [58]	91.00	L20DL	94.00
FDDL [50]	93.70		

As there are only two classes, the dictionary size in the experiments is set to be relatively small ($m = 50$). The parameters α and λ are set to be 10^{-6} and 2.5 respectively



(a) Male samples

(b) Female samples

Fig. 4: Sample images from the AR face dataset.

in L20DL. We compare our method with DLSI, COPAR, JDL, FDDL, LDL and RL21DL. The gender classification accuracies of the test methods are listed in Table 3. It can been observed that the proposed L20DL method achieved comparable performance in comparison with the competing dictionary learning methods.

4.3.3 Scene classification on Scene-15

We conducted scene classification on the widely used Scene-15 dataset [26] to evaluate the performance of the proposed method. There are totally 15 scene categories on the Scene-15 dataset, including suburb, kitchen, bedroom, living room, industrial, forest, inside cite, highway, coast, mountain, street, tall building, open country, store and office, as shown in Figure 5. The Scene-15 dataset contains 4485 scene images, each of which is of resolution around 250 × 300. The number of images in each class varies from 210 to 410. In the experiments, we extracted the SIFT-based spatial pyramid features [26] from the original images, and used these 3000-dimensional features as the input data of all the compared methods.

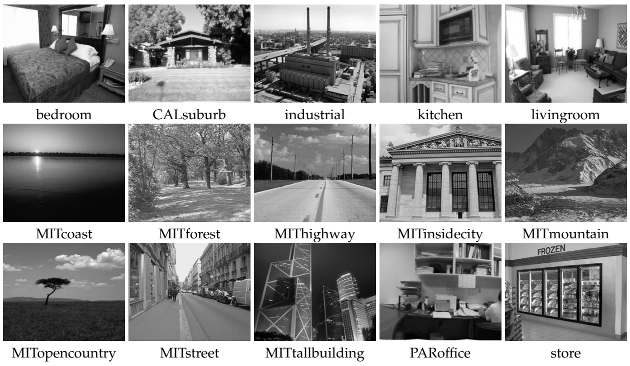
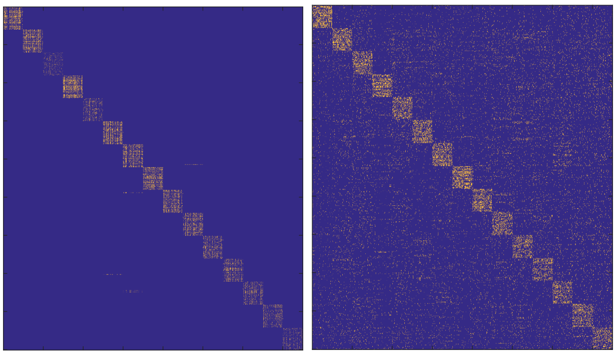


Fig. 5: Sample images from the Scene-15 dataset.



(a) L20DL (b) D-KSVD

Fig. 6: The coding results by L20DL and D-KSVD on the Scene-15 dataset.

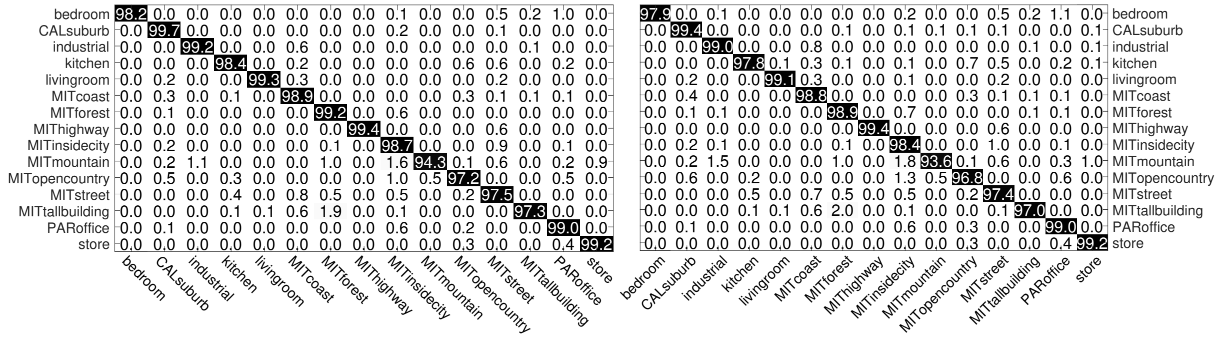
The dictionary size is set as $m = 450$m, and the parameters α and λ are set as 10^{-8} and 0.001 respectively in L20DL. To ensure results reliable, we run the experiments 10 times and report the averages as final results. Following the standard experimental protocol in [26], for each run, we randomly selected 100 images from each category for training and use the remaining part for testing. We compare the result of our L20DL approach with those of LLC, SRC, KSVD, D-KSVD, LC-KSVD and RL21DL. As shown in Table 4, RL21DL outperformed all the other tested methods, while L20DL achieved the second best result with only 0.21% accuracy less than RL21DL. The confusion matrices for RL21DL and L20DL are shown in Figure 7. To demonstrate that our method can indeed obtain class-specific structured patterns, we show the coding result of L20DL and compare it with that of D-KSVD in Figure 6. It can be observed that the sparsity patterns obtained by L20DL are visually cleaner than those from D-KSVD.

Table 4: Scene classification results (%) of the compared methods applied on the Scene-15 dataset.

Method	Accuracy	Method	Accuracy
LLC [45]	89.20	SRC [46]	77.62
KSVD [1]	86.70	L0DL [4]	93.1
D-KSVD [56]	89.10	FDDL [50]	98.35
LC-KSVD [22]	92.90	RL21DL	98.62
MCDL [35]	97.35	L20DL	98.41

4.3.4 Action recognition on UCF Sports

The UCF Sports dataset [32] is used for evaluating the performance of our method in action recognition. The video



	bedroom	CALsuburb	industrial	kitchen	livingroom	MITcoast	MITforest	MIThighway	MITinsidecity	MITmountain	MITopencountry	MITstreet	MITtallbuilding	PARoffice	store
bedroom	98.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.5	0.2	1.0	0.0
CALsuburb	0.0	99.7	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.1	0.0	0.0	0.0
industrial	0.0	0.0	99.2	0.0	0.0	0.6	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0
kitchen	0.0	0.0	0.0	98.4	0.0	0.2	0.0	0.0	0.0	0.0	0.6	0.6	0.0	0.2	0.0
livingroom	0.0	0.2	0.0	0.0	99.3	0.3	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0
MITcoast	0.0	0.3	0.0	0.1	0.0	98.9	0.0	0.0	0.0	0.0	0.3	0.1	0.1	0.1	0.0
MITforest	0.0	0.1	0.0	0.0	0.0	0.0	99.2	0.0	0.6	0.0	0.0	0.0	0.0	0.0	0.0
MIThighway	0.0	0.0	0.0	0.0	0.0	0.0	0.0	99.4	0.0	0.0	0.0	0.6	0.0	0.0	0.0
MITinsidecity	0.0	0.2	0.0	0.0	0.0	0.0	0.1	0.0	98.7	0.0	0.0	0.9	0.0	0.1	0.0
MITmountain	0.0	0.2	1.1	0.0	0.0	0.0	1.0	0.0	1.6	94.3	0.1	0.6	0.0	0.2	0.9
MITopencountry	0.0	0.5	0.0	0.3	0.0	0.0	0.0	0.0	1.0	0.5	97.2	0.0	0.0	0.5	0.0
MITstreet	0.0	0.0	0.0	0.4	0.0	0.8	0.5	0.0	0.5	0.0	0.2	97.5	0.0	0.0	0.0
MITtallbuilding	0.0	0.0	0.0	0.1	0.1	0.6	1.9	0.0	0.1	0.0	0.0	0.0	97.3	0.0	0.0
PARoffice	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.6	0.0	0.2	0.0	0.0	99.0	0.0
store	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.4	99.2

(a) RL21DL

bedroom	CALsuburb	industrial	kitchen	livingroom	MITcoast	MITforest	MIThighway	MITinsidecity	MITmountain	MITopencountry	MITstreet	MITtallbuilding	PARoffice	store	
97.9	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.5	0.2	1.1	0.0	bedroom
0.0	99.4	0.0	0.0	0.0	0.0	0.1	0.0	0.1	0.1	0.1	0.1	0.0	0.0	0.1	CALsuburb
0.0	0.0	99.0	0.0	0.0	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.1	industrial
0.0	0.0	0.0	97.8	0.1	0.3	0.1	0.0	0.1	0.0	0.7	0.5	0.0	0.2	0.1	kitchen
0.0	0.2	0.0	0.0	99.1	0.3	0.0	0.0	0.1	0.0	0.0	0.2	0.0	0.0	0.0	livingroom
0.0	0.4	0.0	0.0	0.0	98.8	0.0	0.0	0.0	0.0	0.3	0.1	0.1	0.1	0.0	MITcoast
0.0	0.1	0.1	0.0	0.0	0.0	98.9	0.0	0.7	0.0	0.0	0.0	0.1	0.1	0.0	MITforest
0.0	0.0	0.0	0.0	0.0	0.0	0.0	99.4	0.0	0.0	0.0	0.6	0.0	0.0	0.0	MIThighway
0.0	0.2	0.1	0.0	0.0	0.0	0.1	0.0	98.4	0.0	0.0	1.0	0.0	0.1	0.0	MITinsidecity
0.0	0.2	1.5	0.0	0.0	0.0	1.0	0.0	1.8	93.6	0.1	0.6	0.0	0.3	1.0	MITmountain
0.0	0.6	0.0	0.2	0.0	0.0	0.0	0.0	1.3	0.5	96.8	0.0	0.0	0.6	0.0	MITopencountry
0.0	0.0	0.0	0.5	0.0	0.7	0.5	0.0	0.5	0.0	0.2	97.4	0.0	0.0	0.0	MITstreet
0.0	0.0	0.0	0.1	0.1	0.6	2.0	0.0	0.1	0.0	0.0	0.1	97.0	0.0	0.0	MITtallbuilding
0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.6	0.0	0.3	0.0	0.0	99.0	0.0	PARoffice
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.4	99.2	store

(b) L20DL

Fig. 7: Confusion matrices of the RL21DL and L20DL methods on the Scene-15 dataset.

sequences of the UCF sport dataset [32] is collected from various broadcast sport channels such as ESPN and BBC, which represent a natural pool of actions featured in a wide range of scenarios and viewpoints. These videos are categorized into 10 sports action classes: kicking, running, swinging (horizontal), swinging (vertical), golfing, diving, skateboarding, walking, lifting and horse riding, as shown in Figure 8. For evaluation, we extracted the action bank feature [40] from the original video sequences and reduced the dimensions of these features to 100 via PCA.

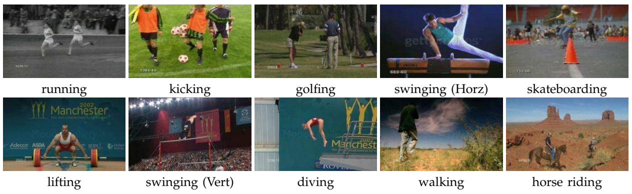


Fig. 8: Key frames of samples from the UCF sport dataset.

Table 5: Action recognition results (%) of the compared methods applied on the UCF Sport dataset.

Method	Accuracy	Method	Accuracy
SRC [46]	80.62	D-KSVD [56]	88.10
DLSI [38]	92.1	LC-KSVD [22]	91.20
COPAR [25]	90.7	LDL [52]	95.00
KSVD [1]	86.80	FDDL [50]	93.60
LLC [45]	87.50	RL21DL	94.29
MCDL [35]	91.65	L20DL	94.21

Following the common experimental settings, we adopt the five-fold cross validation scheme to evaluate the effectiveness of the proposed method. The default dictionary size is set as $m = 50$, and the parameters α and λ are set as 10^{-6} and 0.08 respectively in L20DL. For some compared class-specific dictionary learning methods including DLSI, COPAR, FDDL and LDL, the size of each sub-dictionary is set as the number of training samples from the corresponding class. The classification accuracies are listed in Table 5. It can be observed that L20DL outperformed all the tested methods except RL21DL and LDL. The RL21DL method has a classification accuracy only 0.08% better than that of our method. Compared with the LDL method, the proposed method has a relatively worse recognition accuracy. But note that our method uses a smaller-size dictionary. Moreover, the computational efficiency of our method is much better than LDL. The reason is that LDL incorporates the Fish discriminant into sparse coding which requires much time in solving the related optimization problem, while our method uses a very simple scheme to utilize supervised information and has an efficient numerical solver.

5 Summary

Structured sparsity is very helpful for classification. In this paper, we presented an effective method for structured sparse coding in the context of classification, which is capable of exploiting the class-specific structured sparsity patterns existing in data. The proposed method is constructed using the $\ell_{2,0}$ penalty on each class of data. The learned dictionaries are well adapted to the subspaces of samples from each category and thus can generate structured sparse codes with strong discriminability. The proposed method is equipped with an efficient numerical solver with proved global convergence property. We applied the proposed method to classifying both synthetic and real-world data. The experimental results demonstrated the competitive performance of the proposed method in comparison with recent sparse coding methods.

6 Acknowledgments

Yuping Sun would like to thank the support by Natural Science Foundation of Guangdong Province (Grand No. 2016A 030313516). Yuhui Quan would like to thank the support by National Natural Science Foundation of China (Grand No. 61602184). Jia Fu would like to thank the support by National Social Science Foundation of China (Grand No. 16BXW020), Fundamental Research Funds for the Central Universities (Grand No. 2014XM520), and Philosophy and Social Science Research of Guangdong Province (Grand No. GD14XXW03).

References

1. Aharon M, Elad M, Bruckstein A (2006) K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE T SIGNAL PROCES 54(11):4311–4322
2. Bach F, Jenatton R, Mairal J, Obozinski G, et al (2012) Structured sparsity through convex optimization. STAT SCI 27(4):450–468
3. Bao C, Ji H, Quan Y, Shen Z (2014) ℓ_0 norm based dictionary learning by proximal methods with global convergence. In: CVPR, pp 3858–3865
4. Bao C, Ji H, Quan Y, Shen Z (2016) Dictionary learning for sparse coding: Algorithms and convergence analysis. IEEE T PATTERN ANAL 38(7):1356–1369
5. Boureau YL, Bach F, LeCun Y, Ponce J (2010) Learning mid-level features for recognition. In: CVPR, IEEE, pp 2559–2566
6. Cai S, Zuo W, Zhang L, Feng X, Wang P (2014) Support vector guided dictionary learning. In: ECCV, Springer, pp 624–639
7. Cai X, Nie F, Cai W, Huang H (2013) New graph structured sparsity model for multi-label image annotations. In: ICCV, IEEE, pp 801–808
8. Cai X, Nie F, Huang H (2013) Exact top-k feature selection via $\ell_{2,0}$-norm constraint. In: IJCAI, AAAI Press, pp 1240–1246
9. Candes EJ, Romberg JK, Tao T (2006) Stable signal recovery from incomplete and inaccurate measurements. COMMUN PUR APPL MATH 59(8):1207–1223
10. Castrodad A, Sapiro G (2012) Sparse modeling of human actions from motion imagery. INT J COMPUT VISION 100(1):1–15
11. Chen YC, M Patel V, Shekhar S, Chellappa R, Phillips PJ (2013) Video-based face recognition via joint sparse representation. In: ICAFGR, IEEE, pp 1–8
12. Chi YT, Ali M, Rajwade A, Ho J (2013) Block and group regularized sparse modeling for dictionary learning. In: CVPR, IEEE, pp 377–382
13. Deng W, Yin W, Zhang Y (2013) Group sparse optimization by alternating direction method. Proc SPIE 8858
14. Donoho DL (2006) Compressed sensing. IEEE T INFORM THEORY 52(4):1289–1306
15. Elad M, Aharon M (2006) Image denoising via sparse and redundant representations over learned dictionaries. IEEE T IMAGE PROCESS 15(12):3736–3745
16. Elhamifar E, Vidal R (2012) Block-sparse recovery via convex optimization. IEEE T SIGNAL PROCES 60(8):4094–4107
17. Gao S, Tsang WH, Ma Y (2013) Learning category-specific dictionary and shared dictionary for fine-grained image categorization. IEEE T IMAGE PROCESS 23(2):623–634
18. Georghiades AS, Belhumeur PN, Kriegman D (2001) From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE T PATTERN ANAL 23(6):643–660
19. Huang J, Zhang T, Metaxas D (2011) Learning with structured sparsity. J MACH LEARN RES 12:3371–3412
20. Jacob L, Obozinski G, Vert JP (2009) Group lasso with overlap and graph lasso. In: ICML, ACM, pp 433–440
21. Jenatton R, Mairal J, Obozinski G, Bach F (2011) Proximal methods for hierarchical sparse coding. J MACH LEARN RES 12(7):2297–2334
22. Jiang Z, Lin Z, Davis L (2013) Label consistent K-SVD: Learning a discriminative dictionary for recognition. IEEE T PATTERN ANAL 35(11):2651–2664
23. Kavukcuoglu K, Ranzato M, Fergus R, LeCun Y (2009) Learning invariant features through topographic filter maps. In: CVPR, IEEE, pp 1605–1612
24. Kim S, Xing EP (2010) Tree-guided group lasso for multi-task regression with structured sparsity. In: ICML, pp 543–550
25. Kong S, Wang D (2012) A dictionary learning approach for classification: separating the particularity and the commonality. In: ECCV, Springer, pp 186–199
26. Lazebnik S, Schmid C, Ponce J (2006) Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR, IEEE, vol 2, pp 2169–2178
27. Lee H, Ekanadham C, Ng AY (2008) Sparse deep belief net model for visual area v2. In: NIPS, pp 873–880
28. Lian XC, Li Z, Lu BL, Zhang L (2010) Max-margin dictionary learning for multiclass image categorization. In: ECCV, Springer, pp 157–170
29. Mairal J, Bach F, Ponce J (2012) Task-driven dictionary learning. IEEE T PATTERN ANAL 34(4):791–804
30. Majumdar A, Ward RK (2009) Classification via group sparsity promoting regularization. In: ICASSP, IEEE, pp 861–864

31. Martinez AM (1998) The AR face database. CVC Technical Report 24
32. Mikel Rodriguez JA, Shah M (2008) Action mach: A spatio-temporal maximum average correlation height filter for action recognition. In: CVPR, IEEE, pp 1–8
33. Nie F, Huang H, Cai X, Ding CH (2010) Efficient and robust feature selection via joint l2,1-norms minimization. In: NIPS, pp 1813–1821
34. Pham DS, Venkatesh S (2008) Joint learning and dictionary construction for pattern recognition. In: CVPR, IEEE, pp 1–8
35. Quan Y, Xu Y, Sun Y, Huang Y (2016) Supervised dictionary learning with multiple classifier integration. PATTERN RECOGN 55:247–260
36. Quan Y, Xu Y, Sun Y, Huang Y, Ji H (2016) Sparse coding for classification via discrimination ensemble. In: CVPR
37. R Jenatton VMGOFB A Gramfort, Thirion B (2011) Multiscale mining of fmri data with hierarchical structured sparsity. SIAM J IMAGING SCI 5(3):835–856
38. Ramirez I, Sprechmann P, Sapiro G (2010) Classification and clustering via dictionary learning with structured incoherence and shared features. In: CVPR, IEEE, pp 3501–3508
39. Rao N, Nowak R, Cox C, Rogers T (2016) Classification with the sparse group lasso. IEEE T SIGNAL PROCES 64(2):448–463
40. Sadanand S, Corso JJ (2012) Action bank: A high-level representation of activity in video. In: CVPR, IEEE, pp 1234–1241
41. Shen Z, Toh KC, Yun S (2011) An accelerated proximal gradient algorithm for frame-based image restoration via the balanced approach. SIAM J IMAGING SCI 4(2):573–596
42. Sprechmann P, Ramirez I, Sapiro G, Eldar YC (2011) C-hilasso: A collaborative hierarchical sparse modeling framework. IEEE T SIGNAL PROCES 59(9):4183–4198
43. Sun F, Xu M, Hu X, Jiang X (2015) Graph regularized and sparse nonnegative matrix factorization with hard constraints for data representation. NEUROCOMPUTING 173:233–244
44. Szlam A, Gregor K, LeCun Y (2012) Fast approximations to structured sparse coding and applications to object classification. In: ECCV, Springer, pp 200–213
45. Wang J, Yang J, Yu K, Lv F, Huang T, Gong Y (2010) Locality-constrained linear coding for image classification. In: CVPR, IEEE, pp 3360–3367
46. Wright J, Yang AY, Ganesh A, Sastry SS, Ma Y (2009) Robust face recognition via sparse representation. IEEE T PATTERN ANAL 31(2):210–227
47. Xu Y, Sun Y, Quan Y, Luo Y (2015) Structured sparse coding for classification via reweighted\ ell _ {2, 1} minimization. In: CCCV, Springer, pp 189–199
48. Xu Y, Sun Y, Quan Y, Zheng B (2015) Discriminative structured dictionary learning with hierarchical group sparsity. COMPUT VIS IMAGE UND 136:59–68
49. Yang J, Yu K, Gong Y, Huang T (2009) Linear spatial pyramid matching using sparse coding for image classification. In: CVPR, IEEE, pp 1794–1801
50. Yang M, Zhang D, Feng X (2011) Fisher discrimination dictionary learning for sparse representation. In: ICCV, IEEE, pp 543–550
51. Yang M, Zhang D, Yang J (2011) Robust sparse coding for face recognition. In: CVPR, IEEE, pp 625–632
52. Yang M, Dai D, Shen L, Gool LV (2014) Latent dictionary learning for sparse representation based classification. In: CVPR, IEEE, pp 4138–4145
53. Yuan M, Lin Y (2006) Model selection and estimation in regression with grouped variables. J R STAT SOC B 68(1):49–67
54. Zelnik-Manor L, Rosenblum K, Eldar YC (2012) Dictionary optimization for block-sparse representations. IEEE T SIGNAL PROCES 60(5):2386–2395
55. Zhang D, Liu P, Zhang K, Zhang H, Wang Q, Jing X (2015) Class relatedness oriented-discriminative dictionary learning for multiclass image classification. PATTERN RECOGN 59:168–175
56. Zhang Q, Li B (2010) Discriminative K-SVD for dictionary learning in face recognition. In: CVPR, IEEE, pp 2691–2698
57. Zhang Y, Jiang Z, Davis LS (2013) Learning structured low-rank representations for image classification. In: CVPR, IEEE, pp 676–683
58. Zhou N, Shen Y, Peng J, Fan J (2012) Learning interrelated visual dictionary for object recognition. In: CVPR, IEEE, pp 3490–3497