Discriminative Training of Structured Dictionaries via Block Orthogonal Matching Pursuit

Wenling Shang* Kihyuk Sohn† Honglak Lee, Anna Gilbert‡

Abstract

It is well established that high-level representations learned via sparse coding are effective for many machine learning applications such as denoising and classification. In addition to being reconstructive, sparse representations that are discriminative and invariant can further help with such applications. In order to achieve these desired properties, this paper proposes a new framework that discriminatively trains structured dictionaries via block orthogonal matching pursuit. Specifically, the dictionary atoms are assumed to be organized into blocks. Distinct classes correspond to distinct blocks of dictionary atoms; however, our algorithm can handle the case where multiple classes share blocks. We provide theoretical justification and empirical evaluation of our method.

1 Introduction

Sparse coding is the task of representing the input data as a linear combination of a few atoms from an overcomplete dictionary [1]. In recent years, sparse coding has been extensively applied in many applications of machine learning to efficiently represent the input signal (e.g., image or audio denoising tasks [2, 3, 4]) or to construct abstract features for high-level tasks, such as classification [5, 6, 7, 8, 9].

Standard sparse coding, however, does not enforce any further constraints besides reconstruction. Recent works have shown that dictionaries trained with additional discriminative objectives are beneficial to classification tasks [10, 11]. Other works have shown that enforcing structured sparsity during training results in more invariant thus robust features [12, 13]. In our work, combining these two criteria in addition to reconstruction, we propose to learn a structured discriminative dictionary via block orthogonal matching pursuit (B-OMP), a greedy approach that can be more computationally efficient than the unconstrained convex relaxation solvers adopted by most dictionary learning algorithms. The learned dictionary can produce more robust and discriminative features, compared to unsupervised OMP-based algorithms without block-sparsity.

1.1 Motivations and Contributions.

We hypothesize that two properties can potentially enhance the robustness of sparse representations in a supervised setting: (1) related atoms are grouped in the same block, and (2) the blocks are discriminative to one another.

To achieve the first property, we enforce a block structure on the dictionary to group "similar" atoms together. In standard sparse coding without structures [1, 14] (Figure 1(a)), any combination of atoms can be used to represent an input signal. As a result, the input signals that are semantically similar (e.g., with same class labels) can end up activating quite different atoms. This is problematic in tasks such as classification since it causes significant within-class variation in the feature space. Meanwhile, when there is a well-designed block structure in the dictionary such that a block will be activated if and only if a significant portion of the atoms from the block can be activated together, such within-class variation will then be alleviated.

To achieve the second property, during dictionary learning, we leverage label information to craft label-driven blocks: each block is associated with one or multiple classes and a training example is encouraged to activate blocks associated with its own class.

Our sparse coding algorithm is based on a greedy pursuit method, namely block orthogonal matching pursuit (B-OMP). It not only enables us to assign each block primarily to a specific class, but also to manipulate the degree of block sharing between different classes. Furthermore, we demonstrate that B-OMP significantly speeds up the sparse coefficients update compared with l_1/l_2 norm constrained convex relaxation solver such as block/group sparse coding (BGSC) [15].

In summary, our contributions are as follows: (1) we propose a fast dictionary learning algorithm that produces discriminative dictionaries with structures based on block sparse coding, (2) we provide a theoretical analysis of the dictionaries that optimize our proposed objective function, (3) we evaluate our algorithm on an

*University of Michigan, Ann Arbor, MI. shangw@umich.edu
†NEC Labs America, Cupertino, CA. ksohn@nec-labs.com
‡University of Michigan, Ann Arbor, MI. honglak@eecs.umich.edu, annacg@umich.edu

occlusion denoising task with USPS handwritten digit dataset [16] and a facial expression recognition task with Toronto Face Database (TFD) [17] to empirically demonstrate that integrating the two properties above indeed results in robust sparse representations that are suitable for discriminative tasks.

1.2 Related Work. In recent years, much work has been done in sparse coding and dictionary learning. In [15], the idea of block structure is explored: their algorithm learns structured dictionaries and utilizes an intra-block coherence penalty term (see Section 4.4 for definition). However, [15] does not incorporate the label information. The idea of leveraging label information to discriminatively train dictionaries appears in both [10] and [11]. The major difference that distinguishes our work from theirs is that we do not add an additional classification loss penalty to the objective function. In addition, neither [10] nor [11] learns structured dictionaries.

In [18], both "structured" and "discriminative" properties are considered. However, for certain datasets, we find that the penalty term proposed in [18], the inter-block coherence (definition see Section 3) penalty, does not necessarily improve the classification performance in practice. In our work, we further explore our observations and reach a potential theoretical justification for why this penalty term is unnecessary in some cases. In addition, there is no sharing of dictionaries between different classes during learning in [18].

Lastly, [19] is related to our work in the sense that a greedy pursuit method is used in dictionary learning. However, in [19], there is no label information involved. We use gradient descent to update the dictionary atoms, which provides more flexibility in the objective function, unlike the K-SVD based approach utilized in [19].

1.3 Notation. We consider that the (labeled) data matrix is given as $Y^c \in \mathbb{R}^{d\times n_c}$ for each class $c \in \{1,\cdots,C\}$. Here, d denotes the dimension of a data point; n_c is the number of data points in class c, and n denotes the total number of examples, i.e., $n = \sum_{c=1}^{C} n_c$. The dictionary matrix is given as $D \in \mathbb{R}^{d\times(K\cdot s)}$, which is composed of K non-overlapping sub-matrices $D[k] \in \mathbb{R}^{d\times s}$, $k \in \{1,\cdots,K\}$, i.e., $D = [D[1];\cdots;D[K]]$, where s is the number of atoms in each $D[k]$. We assume throughout the paper that all data points and dictionary atoms are normalized, i.e., $\|Y_j^c\|_2 = 1$ and $\|D[k]_l\|_2 = 1$, where Y_j^c denotes the j-th column of Y^c, and $D[k]_l$ denotes the l-th column of $D[k]$. The sparse coefficient, i.e. sparse representation, of Y_j^c is denoted by $S_j^c \in \mathbb{R}^{K\cdot s}$ and S (or S^c) denote the coefficient sub-matrices of Y (or Y^c). In addition,

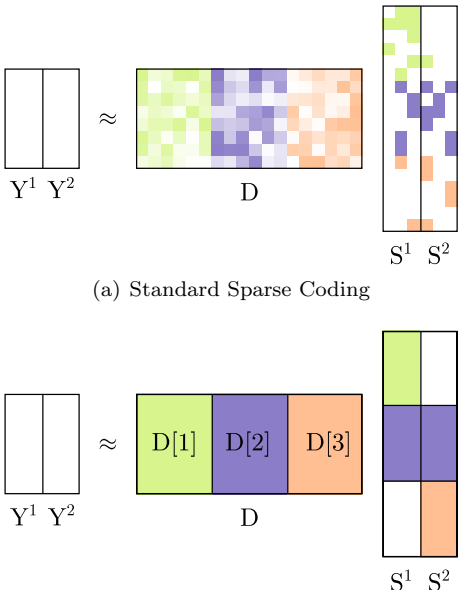

Figure 1: In standard sparse coding, suppose examples from class 1 tend to activate subsets from the first two thirds atoms in D, and rarely the last one third; suppose those from class 2 tend to activate subsets from the last two thirds but rarely the top one third. For discriminative and structured sparse coding, we can group the first one third atoms into a block colored green, second one third to a purple block and last ones to an orange block. A block is activated if and only if a significant number of atoms in the block are activated. In this example, $W_1^1 = W_2^1 = W_2^2 = W_3^2 = 1$ and the other entries of W are zeros. Best viewed in color.

$S[k] \in \mathbb{R}^{s\times n}$ (or $S^c[k] \in \mathbb{R}^{s\times n_c}$) denotes a coefficient matrix of data in Y (or Y^c) corresponding to the atoms in $D[k]$. Finally, for clarity, we introduce binary variable W_k^c to indicate whether the atoms in $D[k]$ are used to represent the data in class c. In other words, if $W_k^c = 1$, we may have non-zero elements in $S^c[k]$, but when $W_k^c = 0$, all elements in $S^c[k]$ are zeros. We describe the usage of W_k^c with a toy example in Figure 1(b). The block sparsity of a sparse coefficient vector $S_j^c \in \mathbb{R}^{K\cdot s}$ is defined as the number of active blocks, which can be defined using an indicator function $\mathbf{1}(\cdot)$ as follows:

$$\|S_j^c\|_{2,0} = \sum_{k=1}^{K} \mathbf{1}(\|S_j^c[k]\|_2 > 0).$$

2 Objective Function and Theoretical Analysis

2.1 The Proposed Objective Function. Our goal is to construct a dictionary such that: (1) the dictionary

is structured, (2) activation is block-sparse, (3) each class is associated with some blocks and a limited amount of block sharing is allowed among different classes, (4) an input data only activates the blocks associated with its class, and (5) the block structure is both highly representative and discriminative–an input data's reconstruction error is minimized only when activating the blocks associated with its class. To construct such a dictionary, we propose the following objective function:

$$\min_{D,S,W} \sum_{c=1}^{C}\sum_{j=1}^{n_c} \frac{1}{2}\left\|Y_j^c - \sum_{k=1}^{K} W_k^c D[k] S_j^c[k]\right\|_2^2 \tag{2.1}$$

$$\text{subject to } \forall c, \forall j, \|S_j^c\|_{2,0} \leq \lambda$$

$$\forall c_1, \forall c_2, c_1 \neq c_2, \sum_{k=1}^{K} W_k^{c_1} \cdot W_k^{c_2} \leq \rho, \rho \in \mathbb{Z}^+ \cup \{0\}$$

The proposed objective function captures most of the desired properties. However, in order to avoid over-complication in training, it does not directly enforce part of property (5) mentioned above, i.e., the discriminative power of the block structure. Instead, we mathematically explore conditions under which the learned blocks have strong discriminative power even without explicitly taking property (5) into consideration during training in the next section.

2.2 Theoretical Analysis. Recent theoretical works on structured sparse coding [20] imply that low inter-block coherence positively impacts the discriminative power of the blocks. Intuitively, even if a block produces small reconstruction errors for the examples from its associated class, there may exist other candidates that can also achieve small reconstruction errors but associated with a different class. If additionally the inter-block coherence between blocks associated with distinct classes is low, then more likely an example from a certain class will correctly select blocks associated with its own class during sparse approximation. As a consequence, the resulting sparse coefficients for an example tend to select the blocks associated with its class and not to activate irrelevant blocks.

Thus we would like to show an upper bound of the inter-block coherence when the blocks are optimal for the proposed objective function. In particular, we measure the inter-block coherence between two blocks using the dot products (in absolute value) between atoms from the two blocks.

For the theoretical analysis, we introduce further assumptions and some definitions:

1. We consider two classes only and each class only has one block associated with it; thus there are two blocks in total. In practice, we use multiple blocks for each class because the examples from the same class can be further divided into smaller sub-classes (e.g., very different writing styles for the same digit class).

2. Let $Y_1^1, Y_2^1 \cdots Y_{n_1}^1 \in \mathbb{R}^d$ be examples from class 1, and $Y_1^2, Y_2^2 \cdots Y_{n_2}^2 \in \mathbb{R}^d$ be examples from class 2 (note that all input vectors are unit norm, as described previously). We assume $\forall Y_i^1, Y_j^1 \in Y^1$, $\langle Y_i^1, Y_j^1 \rangle \geq \delta_1 > 0$, i.e. examples from class 1 are sufficiently close to one another. We denote class-specific mean vectors as

$$\bar{Y}^1 = \frac{\sum_{i=1}^{n_1} Y_i^1}{\|\sum_{i=1}^{n_1} Y_i^1\|_2}, \bar{Y}^2 = \frac{\sum_{i=1}^{n_2} Y_i^2}{\|\sum_{i=1}^{n_2} Y_i^2\|_2},$$

and the coherence between the class-specific mean vectors is defined as

$$|\langle \bar{Y}^1, \bar{Y}^2 \rangle| = \delta_{12}. \tag{2.2}$$

3. We do not include any overlapping between classes, i.e., $\rho = 0$. Therefore, we do not need to learn the W_k^c in our case: $W_1^1 = W_2^2 = 1$ and $W_1^2 = W_2^1 = 0$. In this special case, the objective functions can be simplified into two independent objective functions: For class 1,

$$\min_{D[1],S^1} \frac{1}{2}\sum_{i=1}^{n_1} \left\|Y_i^1 - D[1]S_i^1\right\|^2. \tag{2.3}$$

For class 2,

$$\min_{D[2],S^2} \frac{1}{2}\sum_{j=1}^{n_2} \left\|Y_j^2 - D[2]S_j^2\right\|^2. \tag{2.4}$$

4. We assume that $\|S_i^c\|_2$'s are bounded by some large positive number $\alpha > 0$, so that the optimization problem is defined on a compact domain. And since the objective function is continuous in both D and S, there exists a global optimum. We further assume that the globally optimal dictionaries for (2.3) and (2.4) are obtained and denote them by $D[1]$ and $D[2]$ respectively.

5. Finally, we enforce one more assumption on class 1. Let

$$p_1 = \sup_{\dim(W)=s-1} \frac{\sum_{j=1}^{n_1} \left\|\text{Proj}_W Y_j^1\right\|_2}{n_1},$$

denote the maximal average projection of the data onto an $(s-1)$-dimensional subspace $W \subset \mathbb{R}^d$, and

similarly let

$$q_1 = \sup_{\dim(V)=s} \frac{\sum_{j=1}^{n_1} \left\| \mathrm{Proj}_V Y_j^1 \right\|_2}{n_1}.$$

We assume that $q_1 - p_1 > \delta_{12} + \sqrt{1-\delta_1^2}$.

We are ready to introduce the main theorem. In the theorem, we show that under the above assumptions, if an atom, d_2, in $D[2]$ represents class 2 well, then it must be far away from atoms in $D[1]$.

THEOREM 2.1. *With notation and assumptions in (1) to (5) stated above, let d_2 be a column vector in $D[2]$ such that*

$$|\langle d_2, \bar{Y}^2 \rangle| = \gamma_2. \tag{2.5}$$

then, for any column vectors d_1 in $D[1]$, we have

$$|\langle d_1, d_2 \rangle| \leq \delta_d,$$

where

$$\delta_d \gamma_2 - \left((1-\delta_d^2)(1-\gamma_2^2)\right)^{\frac{1}{2}} > \left(1 - (q_1 - p_1 - \delta_{12} - \left(1-\delta_1^2\right))^2\right)^{\frac{1}{2}}.$$

The proof to the theorem and associated lemmas are presented in the Appendix. The theorem provides an upper bound to the inter-block coherence under certain assumptions. As illustrated at the beginning of this section, low inter-block coherence leads to more classifiable sparse representations. In other words, a dictionary satisfying our objective function 2.1 naturally possesses strong discriminative blocks.

Finally, we would like to generalize Theorem 2.1 heuristically to the block-sharing case. When $\rho > 0$, the two classes share ρ blocks. If we subtract the contributions of the shared blocks from both class 1 and class 2, then the resulting residuals are likely to become incoherent. Then, the non-shared blocks are responsible to represent the residuals. Thanks to the incoherence between the residuals, Theorem 2.1 implies that blocks only associated with class 1 and blocks only associated with class 2 are incoherent. Again, as a potential consequence, for an example from class 1, its sparse coefficients corresponding to the blocks only associated with class 1 are non-zero but those corresponding to the blocks only associated with class 2 are zeros. Therefore the sparse representations are still highly discriminative and can lead to a better classification performance.

3 Block Orthogonal Matching Pursuit

Algorithm 1 Orthogonal Matching Pursuit

1: Initialize the activation set to be empty.
2: **for** iter $= 1 : \lambda$, **do**
3: Select the atom with maximum (in absolute value) inner product with the residual.
4: Add the atom to the activation set.
5: Update all sparse coefficients by computing the orthogonal projection of the input signal onto the set of currently activated atoms.
6: **end for**

Algorithm 2 (Modified) Block Orthogonal Mathing Pursuit

1: Initialize the activation set to be empty.
2: **for** iter $= 1 : \lambda$, **do**
3: Select the block hosting maximum (in l^2 norm) orthogonal projection of the residual.
4: Add the block to the activation set.
5: Update all sparse coefficients by computing the orthogonal projection of the input signal onto the set of atoms from the set of currently activated blocks.
6: **end for**

In this section, we describe a Block Orthogonal Matching Pursuit, which we use for the sparse approximation step during dictionary learning.

Orthogonal Matching Pursuit (OMP) [21] is a greedy algorithm to perform sparse approximation. It chooses the atom from D that most correlates (largest dot product in magnitude) with the residual, then subtracts the contribution from all selected atoms from the original input signal by computing its orthogonal projection onto these atoms to update the residual and repeats till λ atoms are selected (Algorithm 1). Block OMP (B-OMP) [22] is an extension of OMP. It greedily selects λ optimal dictionary blocks instead of λ atoms to approximate input signals. The original B-OMP algorithm selects the block with the largest (in absolute value) sum of dot products with the residual. Such selection criterion performs well under certain assumptions such as linear independence of the atoms in the same block–so that the sum of dot products is an approximation to the orthogonal projection, which do not carry over to our classification setting. In our setting, empirical experiments show that the atoms from the same block are highly correlated, hence such selection criterion can unfairly ignore the input examples that are very well represented, yet by only a few atoms in the block. Therefore, we modify the original B-OMP to fit our task: during each iteration, we select the block whose atoms span the subspace that hosts the largest

Algorithm 3 Dictionary Learning

1: $D \leftarrow$ initial random weight.
2: $W \leftarrow$ initial assignment (Sec. 4.1).
3: **for** $i = 1 : \text{maxiter}$ **do**
4: **for** $c = 1 : C$ **do**
5: Update S^c and W^c (Sec. 4.2).
6: Update D^c (Sec. 4.3).
7: **end for**
8: **end for**

(in l^2 norm) orthogonal projection of the residual at this iteration (Algorithm 2). If not mentioned otherwise, B-OMP refers to our modified version throughout the rest of the paper.

In addition, we extend B-OMP to Simultaneous B-OMP (SB-OMP) similarly as the extension from OMP to Simultaneous OMP in [23]. SB-OMP selects the optimal blocks over all input signals *simultaneously* (i.e. over multiple examples) instead of individually (i.e. over each example) in order to be more robust to noise in the input signals.

4 Dictionary Learning Algorithm

Now, we introduce the dictionary learning algorithm with our proposed objective function. As is usual for many dictionary learning algorithms of sparse coding [14], we alternately update the coefficient matrix S, the assignment matrix W, and the dictionary matrix D. In our case, we perform such an alternating update for each class separately. An overview of our dictionary learning algorithm is given by Algorithm 3.

4.1 Initial assignment of W. Assuming that the data is balanced in terms of class labels, we assign the same number of dictionary blocks, each of which contains the same number of dictionary atoms, to each class. To assign equal number of blocks to each class without overlapping, we enforce the total number of blocks to be a multiple of the number of classes. For example, when there are 20 blocks of dictionary with 10 class labels, we assign first two blocks to class 1 (i.e., $W_1^1 = W_2^1 = 1$), next two blocks to class 2 (i.e., $W_3^2 = W_4^2 = 1$), and so on, while setting all the other entries of W to zero, where the superscripts denote class label and the subscripts denote the block index. During dictionary learning, the initial assignment, referred as class-specific blocks, won't be changed, but we assign additional blocks to each class that are not initially assigned to each class at each iteration; this assignment can vary iteration from iteration. More details about updating W will be discussed in the following section.

4.2 Updating S^c and W^c. If there is no sharing allowed (i.e., $\rho = 0$), we can simply use the B-OMP algorithm to compute the sparse coefficients S^c using at most λ blocks out of $D[k]$'s whose $W_k^c = 1$. When dictionary blocks are allowed to be shared across different classes, we need to jointly estimate the sparse coefficient matrix S^c while assigning additional blocks that are not initially assigned to that class. Finding an exact solution for S^c and W^c is difficult, thus we propose a heuristic algorithm that goes around such a difficulty. We describe the details of our proposed optimization steps of S^c and W^c in Algorithm 4.

In a high-level viewpoint, our proposed algorithm can be interpreted as a sequential procedure that first determines W^c for the blocks that are not initially assigned to class c, referred as non-class specific blocks, and then fit the data using the selected dictionary blocks both from class-specific and non class-specific sets to obtain S^c. In order to determine W^c, we first fit the data using only $\lambda - \rho$ blocks (ρ blocks are reserved for the next step) among class-specific blocks using B-OMP (line 2) and compute the residual (line 3). Then, we greedily select the ρ blocks among those whose were non-specific to class c that best represent the residuals (line 4). Since we select one block at a time that is going to be shared by all residuals of the training examples from class c, we use SB-OMP instead of B-OMP. Once we select the blocks, we use the B-OMP algorithm to compute S^c (line 5) and finally, we revert W^c back to initial assignment matrix. Empirical experiments (see Section 5.1) show that such approximated updates for S^c and W^c work well in practice.

4.3 Updating D^c. After updating the sparse coefficient matrix S^c of data Y^c,[1] we update the atoms in each block. For the dictionary block $D[k]$ with $W_k^c = 1$, let $\widetilde{Y}^c = Y^c - \sum_{l \neq k} D[l]S^c[l]$ be the residual of the data using all the dictionary blocks (and corresponding coefficients) other than kth block. Then, the objective function for dictionary learning can be simplified as follows:

$$\min_{D[k]} \frac{1}{2}\left\|\widetilde{Y}^c - D[k]S^c[k]\right\|_2^2, \quad \text{s.t. } \|D[k]_i\|_2 = 1, i = 1, \ldots, s. \tag{4.11}$$

We update the dictionary atoms in $D[k]_i$ one by one, i.e., update one column while fixing other columns and the coefficient matrix. Each dictionary element update involves two steps: firstly, we analytically solve for the optimal $D[k]_i$ while ignoring the unit norm constraint,

[1] Note that W^c is reverted back to the initial assignment matrix after updating the sparse coefficient matrix (line 6 of Algorithm 4).

Algorithm 4 Optimization algorithm of W^c and S^c for class c.

1: Given: an input data matrix $Y^c \in \mathbb{R}^{d\times n_c}$, initial assignment matrix W^c, block sparsity level λ, and number of shared dictionary blocks ρ whose initial assignments are zero (i.e., $W_k^c = 0$).

2: Use the B-OMP to solve the following optimization problem for all $j = 1, \ldots, n_c$:

$$\min_{S_j^c} \frac{1}{2}\left\|Y_j^c - \sum_{k:W_k^c=1} D[k]S_j^c[k]\right\|_2^2, \text{ s.t. } \|S_j^c\|_{2,0} \le \lambda - \rho \tag{4.6}$$

3: Compute the residual matrix of class c:

$$\widetilde{Y}^c = Y^c - \sum_k D[k]S^c[k]. \tag{4.7}$$

4: Use the SB-OMP to solve the following optimization problem:

$$\begin{aligned} &\min_{S^c} \frac{1}{2}\left\|\widetilde{Y}^c - \sum_{k:W_k^c=0} D[k]S^c[k]\right\|_2^2, \\ &\text{s.t.} \sum_{k:W_k^c=0} \mathbf{1}\left(\|S^c[k]\|_{2,0} > 0\right) \le \rho \end{aligned} \tag{4.8}$$

and update $W_k^c = 1$ if $\|S^c[k]\|_{2,0} > 0$.

5: Use the B-OMP to solve the following optimization problem for all $j = 1, \ldots, n_c$ to obtain the final S^c:

$$\begin{aligned} &\min_{S_j^c} \frac{1}{2}\left\|Y_j^c - \sum_{k:W_k^c=1} D[k]S_j^c[k]\right\|_2^2, \\ &\text{s.t. } \|S_j^c\|_{2,0} \le \lambda \end{aligned} \tag{4.9}$$

6: Revert W^c back to the initial assignment matrix.

and secondly, we normalize it to have unit norm. More details in dictionary learning algorithm are provided in Algorithm 5.

4.4 Implementation Details: Intra-Block Coherence Penalty. We can add an intra-block coherence penalty term to the objective function so that

$$\begin{aligned} &\min_{D[k]} \frac{1}{2}\left\|\widetilde{Y}^c - D[k]S^c[k]\right\|_2^2 + \beta\frac{1}{2}\left\|D[k]^\top D[k]\right\|_F^2, \\ &\text{s.t. } \|D[k]_i\|_2 = 1, i = 1, \ldots, s. \end{aligned} \tag{4.12}$$

atoms in each block can learn more diverse patterns while avoiding redundancy between elements [15]. Adding such a diversity constraint between atoms in the same block is especially important since we use greedy pursuit method B-OMP for sparse approximation [24]. We can still obtain a closed-form solution without the unit norm constraint for atom; for more details, please see [15] (Supplementary Materials, 1.2).

Algorithm 5 Dictionary learning algorithm of $D[k]$'s with $W_k^c = 1$.

1: **for** k such that $W_k^c = 1$, **do**

2: $\widetilde{Y}^c = Y^c - \sum_{l\neq k} D[l]S^c[l]$.

3: **for** $i = 1 : s$, **do**

4: Update $D[k]_i$ using the following closed-form solution and normalize:

$$D[k]_i \leftarrow \frac{\left(\widetilde{Y}^c S^c[k]_i^\top - \sum_{j\neq i} D[k]_j S^c[k]_j S^c[k]_i^\top\right)}{\left(S^c[k]_i S^c[k]_i^\top\right)} \tag{4.10}$$

$$D[k]_i \leftarrow D[k]_i / \|D[k]_i\|_2$$

5: **end for**

6: **end for**

5 Experiments

We evaluate our proposed framework on computer vision datasets, namely USPS handwritten digit dataset [16] for image denoising and Toronto Face Database (TFD) [17] for facial expression recognition.

5.1 Hand-Written Digit Denoising: USPS. We first test our algorithm on the USPS dataset [16] for denoising occlusion noise. Based on Theorem 2.1, our proposed algorithm will learn dictionaries with low inter-block coherence. In other words, the blocks are not only capable of representing the digits from the classes they are associated with but also highly differentiated from the blocks associated with other classes. Hence, an occluded digit still is likely to activate the correct blocks, i.e., the blocks associated with its class. In turn, the subspace spanned by the atoms from the correctly activated blocks can fill up the information that is missing due to the occlusion and thus achieve a realistic reconstruction.

The handwritten digit images are of size 16×16 range from 0 to 9. The dataset contains 7921 training and 2007 testing examples. The only preprocessing is to normalize each example to have l^2 norm 1. We randomly corrupt the testing examples with 8×8 zeroed patches and our goal is to estimate the original digits from the occluded ones. During dictionary learning, as described in Section 4, we alternately update the coefficient S along with the assignment W and the dictionary

D, with uncorrupted training examples. Specifically, we apply SB-OMP to find ρ non-class specific blocks and compute S using λ different dictionary blocks in total. Since label is unknown during testing time, we compute the sparse coefficients S for the corrupted testing examples using block sparse coding algorithms, BGSC, as proposed in [15]. Finally, we combine the sparse coefficients S and the learned dictionary D to reconstruct the ground truth with no occlusions. The final result is normalized to have l^2 norm 1.

In this experiment, we use K=20 blocks of dictionaries (which assigns 2 blocks per category for the initial assignment), each of which contains $s = 25$ dictionary elements, which are consistent with the corresponding hyperparameters from [15]. For the other parameters during dictionary learning, we set $\lambda = 5$, $\rho = 3$, and $\beta = 50$. During sparse approximation using BGSC, we set the sparsity parameter $\lambda_s = 0.25$. These parameters, namely λ, ρ, β, and λ_s, are chosen by cross validation. We compare our proposed algorithm with the standard OMP [25] which is neither structured nor discriminative and BGSC [15] which is structured but not discriminative. For OMP baseline, to be consistent with our experiments using the proposed algorithm, we set the number of atoms in D to be 500 and allow 50 of them to be activated during both dictionary learning and sparse approximation. For BGSC, we follow the hyperparameters from [15].

The denoising quality is measured quantitatively by PSNR[2] in Table 1 and qualitatively by visualization of reconstructed digits in Figure 2. Our proposed algorithm notably outperforms the standard OMP both quantitatively and qualitatively. Thanks to its structured and discriminative nature, dictionaries learned through our proposed algorithm manage to fill in the occluded regions based on the available information from the corrupted testing examples. After filling in the occluded regions, we may further filter the remaining noise by applying existing general denoising tools such as [26].

Indeed, the performance of BGSC is within a small margin from the proposed method, though our algorithm gives better results both quantitatively and perceptibly. In addition, our proposed algorithm is very easy to implement, converges with significantly fewer number of iterations, and runs much faster than the unconstrained sparse coding algorithms used in BGSC. For example, it took about 1.41×10^3 seconds (150 iterations) for our proposed algorithm to converge during dictionary learning, whereas the BGSC took about 1.54×10^4 seconds (300 iterations) to converge. Both are evaluated on a single CPU of Intel i7 processor with 16G RAM. Such speedup is mainly due to the greedy pursuit algorithm that we use to update S, which is arguably faster than the standard sparse coding algorithm.

[2]The PSNR values are negative because we assume that maximum possible pixel value is 1.

Algorithm	OMP [25]	BGSC [15]	Proposed
PSNR	−45.9	−39.4	−37.1

Table 1: Denoising PSNR on USPS dataset.

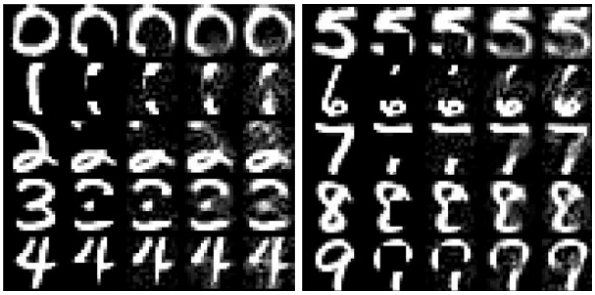

Figure 2: **Visualization of digit reconstructions**. From left to right: ground truth, corrupted testing examples with occlusions, reconstructions from OMP, BGSC and our proposed algorithm. The block structure manages to fill up the missing region.

5.2 Facial Expression Recognition: Toronto Face Database. The Toronto Face Database (TFD) is a collection of aligned and rescaled (48×48 pixels) faces, among which 4,178 are expression-labeled and 112,234 are unlabeled. There are 7 expressions and our task is to recognize these expressions. Additionally, the dataset comes with a pre-defined 5-fold cross validation. We evaluate our algorithm on all 5 training/testing splits and take the average.

We build a two-layer architecture for TFD facial expression recognition. The first layer is unsupervised, following the same procedures as in [27]. The resulting features after the first layer are of dimension 4-by-4-by-200. For the second layer, we first extract patches of 10 different spatial locations in the following ways: (1) 3-by-3 patches with stride 1, (2) 2-by-4 patches with stride 1 and (3) 4-by-2 patches with stride 1. We use our proposed algorithm to learn 10 $(4 + 3 + 3)$ independent dictionaries for the 10 types of patches. For each dictionary, we set $K = 7$, thus each expression is associated with 1 block, and $s = 300$. We do not consider block sharing, i.e $\rho = 0$. The intra-block coherence parameter is set to $\beta = 5$. We run our proposed dictionary learning algorithm for 25 iterations and then finetune the dictionary with OMP-1 [28] for another 25 iterations.

Furthermore, we use threshold encoding (threshold parameter $\lambda_t = 0.5$ for the top two 3-by-3 patches and

$\lambda_t = 0.3$ for the rest) [28] to extract sparse features at testing time. Although we are not directly performing block-sparse encoding at testing time, Theorem 2.1 implies low coherence between individual atoms from different blocks and hence an example would still favor atoms from its own class. Next the extracted sparse features are fed into 10 linear SVMs ($C = 0.5$), and the 10 classifiers vote together to reach a final decision.

Among the baseline algorithms for USPS dataset evaluation, BGSC is too slow for TFD adaptation, thus we only compare our algorithm with OMP baseline. For this baseline, we apply OMP-1 [28] to update the dictionary for 100 iterations during dictionary learning step. There are 2100 atoms in the dictionary, which is consistent with the number of total atoms used in our proposed algorithm for TFD. We also compare our algorithm with other existing two-layer facial expression recognition pipelines. Our algorithm achieves a classification accuracy after averaging across the 5 cross validation sets that is comparable with those from much more complicated architectures.

Proposed	Baseline	DBN[29]	CDA[29]	disBM[27]
85.54%	84.86%	82.40%	85.00%	85.43%

Table 2: Facial expression recognition accuracy on TFD.

6 Conclusion

We have proposed a novel discriminative structured dictionary learning framework that is fast and more scalable than many other dictionary learning algorithms. We also have theoretically justified the inter-block incoherence properties for the dictionaries optimizing our proposed objective function. Our proposed framework performs favorably on the USPS dataset for occlusion denoising and TFD dataset for facial expression recognition. We hope our proposed method becomes a useful tool or inspires new ideas for learning discriminative, invariant features through sparse coding.

Acknowledgments We thank Erik Brinkman, Yuting Zhang, Jimei Yang, Scott Reed, Matus Telgarsky, Ross Kravitz, Chris Fraser and Andrew Zimmer for helpful comments and discussions.

References

[1] B. Olshausen and D. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[2] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006.

[3] M. Lewicki. Efficient coding of natural sounds. *Nature neuroscience*, 5(4):356–363, 2002.

[4] M. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. Davies. Sparse representations in audio and music: from coding to source separation. *Proceedings of the IEEE*, 98(6):995–1005, 2010.

[5] R. Raina, A. Battle, H. Lee, B. Packer, and A. Ng. Self-taught learning: transfer learning from unlabeled data. In *ICML*, 2007.

[6] R Grosse, R Raina, H Kwong, and AY Ng. Shift-invariant sparse coding for audio classification. In *UAI*, 2007.

[7] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[8] K. Kavukcuoglu, M. Ranzato, and Y. LeCun. Fast inference in sparse coding algorithms with applications to object recognition. *arXiv*, 2010.

[9] Y. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *CVPR*, 2010.

[10] Z. Jiang, Z. Lin, and L. Davis. Learning a discriminative dictionary for sparse coding via label consistent K-SVD. In *CVPR*, 2011.

[11] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. Bach. Supervised dictionary learning. In *NIPS*, 2009.

[12] J. Huang. *Structured Sparsity: Theorems, Algorithms and Applications*. PhD thesis, 2011.

[13] K. Kavukcuoglu, M Ranzato, R. Fergus, and Y. LeCun. Learning invariant features through topographic filter maps. In *CVPR*, 2009.

[14] H. Lee, A. Battle, R. Raina, and A. Ng. Efficient sparse coding algorithms. In *NIPS*, 2006.

[15] Y. Chi, M. Ali, A. Rajwade, and J. Ho. Block and group regularized sparse modeling for dictionary learning. In *CVPR*, 2013.

[16] J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994.

[17] J. Susskind, A. Anderson, and G. Hinton. The Toronto Face Database. Technical report, 2010.

[18] I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *CVPR*, 2010.

[19] L. Zelnik-Manor, K. Rosenblum, and Y. Eldar. Dictionary optimization for block-sparse representations. *IEEE Transactions on Signal Processing*, 60(5):2386–2395, 2012.

[20] S. Arora, R. Ge, and A. Moitra. New algorithms for learning incoherent and overcomplete dictionaries. In *ICML*, 2014.

[21] J. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.

[22] Y. Eldar, P. Kuppinger, and H. Bölcskei. Block-sparse signals: Uncertainty relations and efficient recovery. *IEEE Transactions on Signal Processing*, 58(6):3042–3054, 2010.

[23] J. Tropp, A. Gilbert, and M. Strauss. Algorithms for simultaneous sparse approximation. part I: Greedy pursuit. *Signal Processing*, 86(3):572–588, 2006.

[24] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

[25] R. Rubinstein, M. Zibulevsky, and M. Elad. Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit. *CS Technion*, 40(8):1–15, 2008.

[26] F. Agostinelli, M. Anderson, and H. Lee. Adaptive multi-column deep neural networks with application to robust image denoising. In *NIPS*. 2013.

[27] S. Reed, K. Sohn, Y. Zhang, and H. Lee. Learning to disentangle factors of variation with manifold interaction. In *ICML*, 2014.

[28] A. Coates and A. Ng. The importance of encoding versus training with sparse coding and vector quantization. In *ICML*, 2011.

[29] S. Rifai, Y. Bengio, A. Courville, P. Vincent, and M. Mirza. Disentangling factors of variation for facial expression recognition. In *ECCV*. 2012.