Block and Group Regularized Sparse Modeling for Dictionary Learning

Yu-Tseh Chi†, Mohsen Ali†, Ajit Rajwade‡, Jeffrey Ho†
†University of Florida, Gainesville, FL, U. S. A.
‡Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, India
†{ychi, moali, jho}@cise.ufl.edu, ‡ajit_rajwade@daiict.ac.in

Abstract

This paper proposes a dictionary learning framework that combines the proposed block/group (BGSC) or reconstructed block/group (R-BGSC) sparse coding schemes with the novel Intra-block Coherence Suppression Dictionary Learning (ICS-DL) algorithm. An important and distinguishing feature of the proposed framework is that all dictionary blocks are trained simultaneously with respect to each data group while the intra-block coherence being explicitly minimized as an important objective. We provide both empirical evidence and heuristic support for this feature that can be considered as a direct consequence of incorporating both the group structure for the input data and the block structure for the dictionary in the learning process. The optimization problems for both the dictionary learning and sparse coding can be solved efficiently using block-gradient descent, and the details of the optimization algorithms are presented. We evaluate the proposed methods using well-known datasets, and favorable comparisons with state-of-the-art dictionary learning methods demonstrate the viability and validity of the proposed framework.

1. Introduction

Sparse modeling and dictionary learning have emerged recently as an effective and popular paradigm for solving many important learning problems in computer vision. Its appeal stems from its underlying simplicity: given a collection of data $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_l\} \in \mathbb{R}^n$, learning can be formulated using an objective function of the form:

$$\mathcal{Q}(\mathbf{D}, \mathbf{C}; \mathbf{X}) = \sum_g \|\mathbf{X}^{(g)} - \mathbf{D}\mathbf{C}^{(g)}\|_{\mathsf{F}}^2 + \lambda_{\mathbf{D}}\, \mathbf{\Psi}(\mathbf{D}) + \lambda_{\mathbf{C}}\, \mathbf{\Omega}(\mathbf{C}^{(g)}), \quad (1)$$

where the $\mathbf{X}^{(g)}$ are vectors/matrices generated from the data $\mathbf{X}$, and $\mathbf{\Psi}, \mathbf{\Omega}$ are regularizers on the learned dictionary $\mathbf{D}$ and sparse coefficients $\mathbf{C}^{(g)}$, respectively. In dictionary learning, $\mathbf{\Omega}(\mathbf{C})$ is usually based on various sparsity-promoting norms that depend on the extra structures placed on $\mathbf{D}$, and it is the regularizer $\mathbf{\Psi}(\mathbf{D})$ that largely determines the nature of the dictionary $\mathbf{D}$. It is surprising that such an innocuous formula template has generated an active and fertile research field.

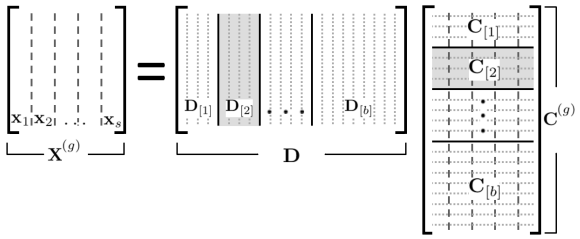


Figure 1: *Illustration of the proposed Block/Group Sparse Coding algorithm. A group of data $\mathbf{X}^{(g)}$ on the left is sparsely coded with respect to the dictionary $\mathbf{D}$ with block structure $\mathbf{D}_{[1]} \cdots \mathbf{D}_{[b]}$.*

If Eq. (1) provides the elegant theme, its variations are often composed of extra structures placed on the dictionary $\mathbf{D}$ ([15, 10, 7, 13]), and less frequently, different ways of generating sparsely-coded data $\mathbf{X}^{(g)}$ for training the dictionary. The former affects how the two regularizers $\mathbf{\Psi}, \mathbf{\Omega}$ should be defined, and the latter determines how the vectors/matrices $\mathbf{X}^{(g)}$ should be generated from $\mathbf{X}$. For classification, a block structure is often imposed on $\mathbf{D}$ and hierarchical structures could be further specified using these blocks ([16, 10]), with the aim of endowing the learned dictionary certain predictive power. To promote sparsity, the block structure on $\mathbf{D}$ is often accompanied by an appropriate block-based ℓ_2-norm (e.g., ℓ_1/ℓ_2-norm [18]) used in $\mathbf{\Omega}(\mathbf{C})$. On the other hand, for $\mathbf{X}^{(g)}$, a common approach is to generate a collection of groups of data vectors $\{\mathbf{x}_{g_1}, \cdots, \mathbf{x}_{g_k}\}$ from $\mathbf{X}$ and to simultaneously sparse code the data vectors in each data group $\mathbf{X}^{(g)}$ [1]. For classification problems, the idea is to generate data groups $\mathbf{X}^{(g)}$ with feature vectors $\mathbf{x}_{g_i}$ that should be similarly encoded, and such data groups $\mathbf{X}^{(g)}$ can be obtained using problem-specific information such as class labels, similarity values and other information sources (e.g., neighboring image patches).

In a noiseless setting, our proposed problem of encoding sparse representations for a group of data samples $\mathbf{X}$ using

the minimum number of blocks from $\mathbf{D}$ can be cast as the following optimization program:

$$\mathcal{P}_{\ell_{0,p}} : \min_{\mathbf{C}} \sum_i I(\|\mathbf{C}_{[i]}\|_p) \text{ s. t. } \mathbf{X} = \mathbf{DC}, \quad (2)$$

where $I(\cdot)$ is an indicator function , $p = 1, 2$ and $\mathbf{C}_{[i]}$ is the i-the block (sub-matrix) of $\mathbf{C}$ that corresponds to the i-th block of $\mathbf{D}$ as shown in Fig. 1. This combinatorial problem is known to be NP-hard, and the ℓ_1-relaxed version of the above program is:

$$\mathcal{P}_{\ell_{1,p}} : \min_{\mathbf{C}} \sum_i \|\mathbf{C}_{[i]}\|_p \text{ s. t. } \mathbf{X} = \mathbf{DC}. \quad (3)$$

We will call this program *Block/Group Sparse Coding* (BGSC) as it incorporates both the group structure in data and block structure in the dictionary.

In some applications of which the main concern is identifying contributing blocks rather than finding the sparse representation [7], the following optimization program is considered:

$$\mathcal{P}'_{\ell_{0,p}} : \min_{\mathbf{C}} \sum_i I(\|\mathbf{D}_{[i]}\mathbf{C}_{[i]}\|_p) \text{ s. t. } \mathbf{X} = \mathbf{DC}. \quad (4)$$

Again, this program is also NP- Hard and its ℓ_1 relaxation is:

$$\mathcal{P}'_{\ell_{1,p}} : \min_{\mathbf{C}} \sum_i \|\mathbf{D}_{[i]}\mathbf{C}_{[i]}\|_p \text{ s. t. } \mathbf{X} = \mathbf{DC}. \quad (5)$$

We will call the programs $\mathcal{P}'_{\ell_{0,p}}$ and $\mathcal{P}'_{\ell_{1,p}}$ *Reconstructed Block/Group Sparse Coding* (R-BGSC) as they minimize the norm of the reconstruction term ($\|\mathbf{D}_{[i]}\mathbf{C}_{[i]}\|$). The optimization algorithms for solving $\mathcal{P}_{\ell_{1,p}}$ and $\mathcal{P}'_{\ell_{1,p}}$ will be presented in Sec. 2.

Sharing of dictionary atoms for data in the same group had been shown to increase the discriminative power of the dictionary ([1]). With the block structure added to dictionary $\mathbf{D}$, our proposed BGSC and R-BGSC algorithms promote a group of data to share only few blocks of $\mathbf{D}$ for encoding. Therefore, incorporating these SC algorithms in a dictionary learning framework, which iteratively updates coefficients of data and updates atoms of $\mathbf{D}$, will result in training each block of $\mathbf{D}$ using only few groups of data. This means that, for example, a badly written digit '9', which looks like a '7', when grouped together with other normally written '9's, will be encoded using atoms these '9's used. The badly written '9' will, in turns, be used to train the atoms in $\mathbf{D}$ that represent '9's rather than those that represent '7's.

Another novelty of our framework is that we do not assign a class of signals to specific blocks of a dictionary, unlike other Sparse Representation based Classification (SRC) [6, 17] and [13]. This would allow some blocks to store shared features between some different classes. Ramirez et. al. [13] trained a single dictionary block for each group of data. This method increases the redundancy of the information encoded in the learned dictionary as the information common to two or more groups (a common scenario in many classification problems) will need to be stored separately within each block. Since one dictionary block is assigned to each class, the redundancy induced in the dictionary needs to be reduced for greater efficiency. This is done by removing dictionary elements whose mutual dot product has an absolute value greater than an arbitrary-chosen threshold (e.g. 0.95). Instead, we provide an objective function whose minimization naturally produces dictionaries that are less redundant. In particular, our proposal to encode data from a single class using multiple blocks obviates the need to even incorporate an explicit inter-block coherence minimization term unlike [13].

As proved in [5], the program $\mathcal{P}_{\ell_{1,p}}$ (Eq. 2) is equivalent to $\mathcal{P}_{\ell_{0,p}}$ (Eq. 3) [1] when

$$n_a(2k-1)\mu_B < 1 - (n_a - 1)\mu_S, \quad (6)$$

where n_a and k are the size and the rank of a block, respectively, and μ_B and μ_S are inter- and intra-block coherence defined in Section 2.4, respectively. In other words, the smaller μ_S is the more likely the two programs can be equivalent. A way to achieve minimum μ_S is to make atoms orthonormal within each block [11, 3]. However, such dictionaries (over-complete dictionary with union of orthonormal basis) do not perform as well as those with more flexible structure [14]. For example, in SRC-based face recognition, each block contains atoms representing faces of the same person. It does not make sense to impose strict orthogonality on each block. Therefore, rather than imposing strong orthogonality constraint on each block, we propose a dictionary learning algorithm that minimizes only intra-block coherence.

The proposed dictionary learning framework learns the dictionary $\mathbf{D}$ by minimizing the objective function given in Eq. (16), and the third term in Eq. (16) measures the mutual coherence within each block of $\mathbf{D}$. The corresponding sparse coding can be either BGSC or R-BGSC. Besides the novel sparse coding algorithms, BGSC and R-BGSC, there are three specific features that distinguish our dictionary framework from existing methods:

1. Instead of inter-block coherence, the proposed ICS-DL algorithm presented in Sec. 2.4 minimizes the intra-block coherence as one of its main objectives.
2. Our framework does not require to assign a class or a group of data to block(s) in the dictionary as in [13]. This allows some blocks of the dictionary to be shared by different classes.
3. The dictionary is trained simultaneously with respect to each group of training samples $\mathbf{X}^{(g)}$ using our proposed block/group regularized SC algorithm.

[1] They proved the case when $p = 2$ and $\mathbf{X}$ is a single vector. In Section 2.1, we will show that the condition still holds when $\mathbf{X}$ is a matrix.

2. Methods

In this section, we describe the algorithms in our proposed framework. We will start with sparse coding algorithms first and work our way towards the full dictionary learning algorithm. We denote scalars with lower-case letters, vectors with bold lower-case letters, matrices with upper-case letters, and the i-th *block* and *group* of a matrix (or vector) with $\mathbf{Z}_{[i]}$, and $\mathbf{Z}^{(i)}$, respectively.

2.1. Theoretical Guarantee

It is important to understand the conditions on $\mathbf{D}$ under which our convex relaxations (Eq. (3) and (5)) are equivalent to their original combinatorial (Eq. (2) and (4)) programs. In other words, we want to examine the conditions under which our proposed programs can indeed have exact recoveries as their corresponding combinatorial programs. The conditions when $\mathbf{X}^{(g)}$ is a single vector was proved in [7]. Using linear algebra, we can convert our programs, where $\mathbf{X}^{(g)}$ and $\mathbf{C}^{(g)}$ are matrices, into equivalent programs, where $\mathbf{X}^{(g)}$ and $\mathbf{C}^{(g)}$ are vectors. The conversion is straightforward and listed in the supplementary materials. We then prove the equivalence conditions of our programs in a similar way as given in [7].

2.2. Block/Group Sparse Coding

The program $\mathcal{P}_{\ell_{1,p}}$ in Eq. (3) can be cast as an optimization problem that minimizes the objective function:

$$\begin{aligned}\mathcal{Q}_c(\mathbf{C};\mathbf{X},\mathbf{D}) &= \sum_g \mathcal{Q}_c(\mathbf{C}^{(g)};\mathbf{X}^{(g)},\mathbf{D}) \\ &= \sum_g \left(\frac{1}{2}\|\mathbf{X}^{(g)} - \mathbf{D}\mathbf{C}^{(g)}\|_{\mathsf{F}}^2 + \lambda \sum_i \left\| \mathbf{C}_{[i]}^{(g)} \right\|_p \right). \quad (7)\end{aligned}$$

For clarity of presentation, we will present the optimization steps only for one specific group of data $\mathbf{X}$ and its corresponding sparse coefficients $\mathbf{C}$. Eq. (7) can be written as:

$$\begin{aligned}&\frac{1}{2}\|\mathbf{X} - \mathbf{D}\mathbf{C}\|_{\mathsf{F}}^2 + \lambda \sum_i \left\| \mathbf{C}_{[i]} \right\|_p \\ &= \frac{1}{2}\|\mathbf{X} - \sum_{i \neq r} \mathbf{D}_{[i]}\mathbf{C}_{[i]} - \mathbf{D}_{[r]}\mathbf{C}_{[r]}\|_{\mathsf{F}}^2 + \lambda \left\| \mathbf{C}_{[r]} \right\|_p + c, \quad (8)\end{aligned}$$

where c includes the terms that do not depend on $\mathbf{C}_{[r]}$. When $p = 1$, this objective function is separable. Iterates of elements in $\mathbf{C}_{[r]}$ can be solved using a method similar to [1]. When $p = 2$[2], it is only block-wise separable. Computing the gradient of Eq. (8) with respect to $\mathbf{C}_{[r]}$, we obtain the following sub-gradient condition:

$$-\mathbf{D}_{[r]}^{\intercal}\mathbf{X} + \mathbf{D}_{[r]}^{\intercal}\sum_{i \neq r} \mathbf{D}_{[i]}\mathbf{C}_{[i]} + \mathbf{D}_{[r]}^{\intercal}\mathbf{D}_{[r]}\mathbf{C}_{[r]} + \lambda\partial\|\mathbf{C}_{[r]}\|_{\mathsf{F}} \ni 0. \quad (9)$$

Assuming for now the optimal solution for $\mathbf{C}_{[r]}$ has a non-zero norm ($\|\mathbf{C}_{[r]}\|_{\mathsf{F}} > 0$). and denoting the first two terms by $-\mathbf{N}$, substituting the positive semi-definite matrix $\mathbf{D}_{[r]}^{\intercal}\mathbf{D}_{[r]}$ with its eigen-decomposition $\mathbf{U}\Sigma\mathbf{U}^{\intercal}$, multiplying both sides of the equation with $\mathbf{U}^{\intercal}$ and using the fact that $\partial\|\mathbf{C}_{[r]}\|_{\mathsf{F}} = \frac{\mathbf{C}_{[r]}}{\|\mathbf{C}_{[r]}\|_{\mathsf{F}}}$, we have

$$\begin{aligned}\mathbf{U}\Sigma\mathbf{U}^{\intercal}\mathbf{C}_{[r]} + \lambda\frac{\mathbf{C}_{[r]}}{\|\mathbf{C}_{[r]}\|_{\mathsf{F}}} &= \mathbf{N} \\ \Sigma\mathbf{U}^{\intercal}\mathbf{C}_{[r]} + \lambda\frac{\mathbf{U}^{\intercal}\mathbf{C}_{[r]}}{\|\mathbf{C}_{[r]}\|_{\mathsf{F}}} &= \mathbf{U}^{\intercal}\mathbf{N}. \quad (10)\end{aligned}$$

Changing the variables $\mathbf{Y} = \mathbf{U}^{\intercal}\mathbf{C}_{[r]}$ and using the fact that the Frobenius norm is invariant under orthogonal transformations, we have

$$\Sigma\mathbf{Y} + \lambda\frac{\mathbf{Y}}{\|\mathbf{Y}\|_{\mathsf{F}}} = \hat{\mathbf{N}}, \quad (11)$$

where $\hat{\mathbf{N}} = \mathbf{U}^{\intercal}\mathbf{N}$. Setting $\kappa = \|\mathbf{Y}\|_{\mathsf{F}}$ and $\hat{\mathbf{Y}} = \mathbf{Y}/\|\mathbf{Y}\|_{\mathsf{F}}$, we have

$$\hat{\mathbf{Y}} = (\kappa\Sigma + \lambda\mathbf{I})^{-1}\hat{\mathbf{N}}, \quad \text{s. t.} \quad \|\hat{\mathbf{Y}}\|_{\mathsf{F}} = 1. \quad (12)$$

Since Σ is a diagonal matrix, $(\kappa\Sigma + \lambda\mathbf{I})^{-1}$ is also a diagonal matrix with diagonal entries $1/(\kappa\sigma_i + \lambda)$, where σ_i is the i-th eigen-value in Σ. Therefore, the constraint $\|\hat{\mathbf{Y}}\|_{\mathsf{F}} = 1$ implies that

$$\sum_{i,j} \frac{\hat{\mathbf{N}}_{i,j}^2}{(\kappa\sigma_i + \lambda)^2} = 1, \quad (13)$$

where $\hat{\mathbf{N}}_{i,j}$ is (i,j)-th element of matrix $\hat{\mathbf{N}}$.

We solve for the root of the above one-variable equation w.r.t. κ using standard numerical methods such as Newton's method. Once κ is computed, we can obtain $\hat{\mathbf{Y}}$ and $\mathbf{Y}$ using Eqs. (12) and (11), respectively. The iterate of $\mathbf{C}_{[r]}$ can be computed by projecting $\mathbf{Y}$ back to the original domain i.e. $\mathbf{C}_{[r]} = \mathbf{U}\mathbf{Y}$.

When the solution of κ in Eq. (13) is not positive, there is no solution for Eq. (10) as it contradicts the assumption that $\kappa > 0$. In this case, the optimality happens at $\mathbf{C}_{[r]} = \mathbf{0}$ because the derivative of $\|\mathbf{C}_{[r]}\|_{\mathsf{F}}$ does not exist when $\|\mathbf{C}_{[r]}\|_{\mathsf{F}} = \mathbf{0}$ and our objective function, Eq. (8), is convex and bounded from below. The proof of this claim is straightforward: Let $f(x)$ be a continuous convex function which is bounded from below and differentiable everywhere except at $x = x_o$. We solve $\partial f(x) = 0$ for the minimum of $f(x)$. If the solution of $\partial f(x) = 0$ does not exist, the minimum of $f(x)$ must occur at $x = x_o$ for otherwise we would find x' such that $\partial f(x') = 0$.

As we can see from Eq. (13), the block sparsity of $\mathbf{C}$ depends on the value of λ. The larger λ is, the less likely there exists a feasible solution to κ in Eq. (13). On the other hand, when $\lambda = 0$, solution of κ will always be positive, and hence there is no non-zero $\mathbf{C}_{[r]}$s. This is analogous to the shrinkage mechanism in standard Lasso program ([4]). When $\mathbf{X}$ is a single vector, our BGSC is equivalent to the P_{ℓ_q/ℓ_1} program in [6]. When there is no block structure on $\mathbf{D}$, BGSC is equivalent to the group sparse coding (GSC) in [1].

[2]We use element-wise ℓ_2 norm here which is the Frobenius norm.

2.3. Reconstructed Block/Group Sparse Coding

For clarity of presentation, we will again derive the novel R-BGSC algorithm for one group of data in this section. $\mathcal{P}'_{\ell_{1,p}}$ in Eq. (5) can be cast as an optimization problem in terms of $\mathbf{C}_{[r]}$ that minimizes

$$\frac{1}{2}\|\mathbf{X}-\sum_{i\neq r}\mathbf{D}_{[i]}\mathbf{C}_{[i]}-\mathbf{D}_{[r]}\mathbf{C}_{[r]}\|_{\mathsf{F}}^2+\lambda\sum_i \left\|\mathbf{D}_{[i]}\mathbf{C}_{[i]}\right\|_p+c, \quad (14)$$

where c is a constant that includes the terms that do not depend on $\mathbf{C}_{[r]}$. The iterate of $\mathbf{C}_{[r]}$ can be derived in a similar fashion as the previous algorithm. We will leave the derivation to the supplementary material. Note that when $\mathbf{X}$ is a single vector, R-BGSC is equivalent to the P'_{ℓ_q/ℓ_1} program in [6].

2.4. Intra-Block Coherence Suppression Dictionary Learning

The intra-block coherence is defined as

$$\mu_S(\mathbf{D}) = \max_i \left(\max_{p,q\in\mathcal{I}(i),p\neq q} \frac{\left|\mathbf{d}_p^\intercal \cdot \mathbf{d}_q\right|}{\|\mathbf{d}_p\|\|\mathbf{d}_q\|} \right), \quad (15)$$

where $\mathcal{I}(i)$ is the index set of the atoms in block i. Inter-block coherence μ_B is defined as $\mu_B(\mathbf{D}) = \max_{i\neq j}\left((\frac{1}{n_a}\sigma_1(\mathbf{D}_{[i]}^\intercal\mathbf{D}_{[j]})\right)$, where σ_1 is the largest singular value and n_a is the size of block.

As mentioned in the Introduction, it is necessary to have a dictionary updating algorithm that minimizes the intra-block coherence. We therefore proposed the following objective function:

$$\mathcal{Q}_d(\mathbf{D};\mathbf{X},\mathbf{C}) = \sum_g \frac{1}{2}\|\mathbf{X}^{(g)}-\mathbf{D}\mathbf{C}^{(g)}\|_{\mathsf{F}}^2+\gamma\sum_{k=1}^{|\mathbf{D}|}\|\mathbf{d}_k\|_2+$$
$$\beta\sum_b\left(\sum_{p,q\in\mathcal{I}(b),p\neq q}\|\mathbf{d}_p^\intercal\mathbf{d}_q\|^2\right)+\lambda\,\Omega(\mathbf{C}), \quad (16)$$

where Ω is the regularizer term on $\mathbf{C}$ (See Eq.(14) and (8)), and the third term minimizes the intra-block coherence.

For the sake of clarity, we derive the update formula required in optimizing the objective function above for one group of data. Again, we first assume the optimal solution for $\mathbf{d}_k$ to have a non-zero norm. Computing the gradient with respect to $\mathbf{d}_r$, and equating it to zero, we have

$$-\mathbf{X}\mathbf{c}_r^\intercal+\sum_{k\neq r}\mathbf{d}_k\mathbf{c}_k\mathbf{c}_r^\intercal+\mathbf{d}_r\mathbf{c}_r\mathbf{c}_r^\intercal+\gamma\frac{\mathbf{d}_r}{\|\mathbf{d}_r\|_2}+$$
$$\beta\sum_{j\in\mathcal{I}(b),j\neq r}\mathbf{d}_j\mathbf{d}_j^\intercal\mathbf{d}_r=0, \quad (17)$$

where $\mathbf{c}_r$ is the r-th row of $\mathbf{C}$ and $\mathbf{d}_r$ is in block b.

Note that $\mathbf{c}_r\mathbf{c}_r^\intercal$ indicates the weight of how much the atom $\mathbf{d}_r$ is being used to encode $\mathbf{X}$. It is clear from the first three terms of the above equation why group-regularized SC algorithms tend to generate high intra-block coherence blocks. As we can see, the value of $\mathbf{d}_r$ depends not only on how much it is being used to encode $\mathbf{X}$ (1st and 3rd term) but also on how much other $\mathbf{d}_k$'s are being used to encode $\mathbf{X}$. Since, BGSC and R-BGSC minimize the number of blocks to be used for encoding $\mathbf{X}$, the atoms $\mathbf{d}_r$ are likely in the same block as $\mathbf{d}_k$. For example, if the coefficient $\mathbf{C}$ of $\mathbf{X}$ has only one non-zero block, then the atoms $\mathbf{d}_r$ and $\mathbf{d}_k$, which correspond to the non-zero rows of coefficients $\mathbf{c}$'s in the above equation, are all in the same block. Therefore, updating $\mathbf{d}_k$ using only the first three terms in the above equation will result in high intra-block coherence. This justifies putting the intra-block coherence suppressing regularizer term in Eq. (16).

To the best of our knowledge, there is no work discussing how to group the training samples. Intuitively, one would split a class of training data into multiple similar groups using techniques such as K-means. However, this might put all the *in-class outliers*, e.g. badly written '9's that look like a '7', into one group and hence allow them to act as one different class and to be used to train the dictionary blocks corresponding to the wrong classes. From our empirical observations, it is better to have a group of data that has similar variability as the whole class. This would force these *in-class outliers* to be regularized by inliers of the same class.

We will leave the rest of the derivation to the supplementary material as it is again similar to the derivation in the previous two sections. Note that it is not uncommon to add a post-processing step to make atoms in $\mathbf{D}$ unit vectors or requiring $\|\mathbf{d}_r\|_2 = 1$. This results in a more efficient algorithm (See Section 1.3 of the supplementary material for details).

3. Experiment on Hand-Written Digit Recognition

In this experiment, we used the USPS dataset [9], which contains a total of 9,298 16-by-16 images of hand-written digits[3]. We vectorized the images and normalized the vectors to have unit ℓ_2-norm. We collected 15 groups of data for each digit where each group contained 50 randomly chosen images from the same class.

The experiment was conducted as follows:

1. Generate a random dictionary $\mathbf{D}$ with n_b blocks and each block contains n_a columns (atoms) (a total of $n_b \times n_a$ columns).
2. Iteratively compute coefficients using BGSC and update the dictionary using ICS-DL algorithm.
3. Use the coefficients of the training data to train 10 one-vs-all linear SVMs[2].
4. Compute the sparse coefficients of the *test samples* using either BGSC or R-BGSC. Use the SVMs to classify the test samples using their coefficients.

[3]Matlab codes for this experiment and two other experimental results are provided in the supplementary materials.

Table 1 demonstrates the impact of the dictionary's block structure on the error rates. The parameters are $\beta = 200$, $\lambda_{\text{train}} = \sim 0.6$[4], and $\lambda_{\text{test}} = 0.2$. For the experiment in the last column of Table 1, we assign two blocks to each digit. The results show that the error rates are similar when the number of blocks (n_b) is greater than 10 even though the number of classes of this dataset is 10. The reason is that there exists some variability within each class and mutual similarity between images of different classes. In fact, as shown in Fig. 2(a), the sparse coefficients of most of the training data have 3 to 6 active blocks when $n_b = 20$.

The last column of Table 1 shows that the hard assignment of blocks to classes results in higher error rate even though the size of the dictionary is twice as large as those of the first three experiments in Table 1. As mentioned in the Introduction, we did not assign blocks to classes and prefer using more blocks for encoding data with larger variability. Moreover, we allow data from different class to share mutual blocks. Fig. 2(a) illustrates the coefficients of the training data. We can see that '7' and '9' share two blocks of dictionary due to their similarity. However, they each have an exclusive block with large coefficients (darker in color) to allow them to encode the difference.

Table 1: *Classification error(%) with different structure on* $\mathbf{D}$. n_b*: number of blocks in* $\mathbf{D}$. n_a*: number of atoms in each block.*

	(n_b, n_a)				
	(20,25)	(40,12)	(10,50)	(20,50)	(20,50)†
Error(%)	2.53	3.42	6.22	2.95	6.52

†: Assign each digit to two blocks of the dictionary.

Next we demonstrate the effect of the value β in ICS-DL on classification rates. The parameters are $\lambda_{\text{train}} = 0.4$ and $(n_b, n_a) = (20, 25)$. When $\beta = 0$, our ICS-DL algorithm does not suppress intra-block coherence and is hence equivalent to the dictionary learning algorithm in [1]. We used BGSC to compute the coefficients during training. During testing, we used either BGSC or R-BGSC to compute the coefficients of test samples. λ_{test} was varied between 0.15 and 0.35 and the best result was reported in Table 2. We stopped the training roughly after 200 iterations when the dictionary update did not change much. The results in Table 2 suggest that suppressing the intra-block coherence can indeed improve the performance. However, as β increases, the error rate increases. In the extreme case when imposing a strict orthogonality on the blocks using the UOB-DL, the error rate increases to 4.27(see Table 3). These results provide an empirical support for not using strict orthogonality constraint. Note that when $\beta = 0$, our result is very close to that of SISF-DL[13] (See Table 3). However, our ICS-DL algorithm does not impose any inter-block orthogonality constraint on the dictionary as SISF-DL does.

[4] λ_{train} varies slightly with respect to n_a.

Table 2: *Classification error(%) with different* β *in ICS-DL.*

	β						
	0	100	200	300	400	600	800
Error(%)	4.02	3.47	2.58	2.43	**2.26**	2.42	3.12

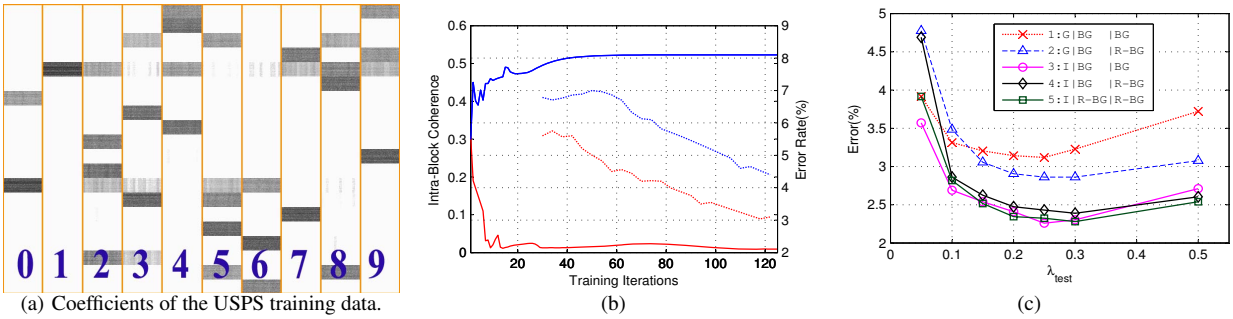
To further demonstrate the intra-block coherence suppressing property of our ICS-DL algorithm, we plot the intra-block coherence values of the dictionaries trained with $\beta = 0$ and $\beta = 200$, respectively, in Fig. 2(b). We also provided the error rates every 4 iterations from the 30-th iteration onward. Solid and dotted lines indicate the coherence and error, respectively. The red solid line demonstrates that our ICS-DL method can keep the intra-block coherence at a low value. On the contrary, without the intra-block coherence suppression term, the blue solid line shows that the coherence value becomes comparably large with increasing number of iterations. The blue dotted line shows that its associated error rate even increases between iterations 40 and 60 which implies that over-fitting occurs within some blocks.

Once we have a trained dictionary, we used the coefficients of training samples to train ten linear SVMs. We can use the already available coefficients computed during the training phase as they are computed as a group. Another way to obtain coefficients of training samples is to recompute them *individually*. Fig. 2(c) shows the error rates of *five* of the different scenarios. The dictionary was trained with $\beta = 400$, $\lambda_{\text{train}} = 0.4$, $(n_b, n_a) = (20, 25)$ and number of iteration is 150. The results in Fig. 2(c) shows that R-BGSC generally performed slightly better than BGSC especially in scenarios 1 and 2 in Fig. 2(c). However, the result from scenario 3 with $\lambda_{\text{test}} = 0.25$ achieves the best error rate at 2.26% (0.02% better than that of scenario 4 with $\lambda_{\text{test}} = 0.30$).

Finally, we compared our results with other state-of-the-art results using dictionary learning algorithms ([13, 12]) shown in Table 3. We also compare with the UOB-DL ([11]) which imposes strict orthogonality constraint on blocks. The results show that our algorithms outperform other dictionary learning methods, even the one specially tailored for hand-written digits recognition [8]. Although Table 2 suggests that suppressing intra-block coherence of $\mathbf{D}$ improves the classification performance, imposing a strict orthogonality on the blocks, however, does not result in any improvement.

Table 3: *Error rate(%) of the USPS and the MNIST datasets with recently published approaches. The results of SISF-DL, SDL-D L, TDK-SVM are taken from [13], [12], and [8], respectively.*

	BGSC	R-BGSC	SISF-DL	SDL-DL	UOB-DL	TDK
USPS	**2.26**	2.28	3.98	3.54	4.27	2.40
MNIST	2.32	–	1.26	1.05	–	–



(a) Coefficients of the USPS training data. (b) (c)

Figure 2: **(a)** *Visualization of the sparse coefficients of the training samples of digits. Each column contains 15 groups. Gray pixels correspond to non-zero coefficients.* **(b)** *Intra-block coherence (solid) and error rates (dotted) of two dictionaries (red for $\beta = 200$ and blue for $\beta = 0$). Error rates of the first 30 iterations are not shown.* **(c)** *Error rates (%) of the USPS dataset under five different scenarios. The scenarios differ in terms of how the training samples are organized for computing the coefficients and which proposed SC algorithms were used. First column in the legend (separated by '|') indicates how the coefficients of the training samples are computed, in groups (***G***) or individually (***I***) . The second column indicates which SC algorithm is used to compute the coefficients of the training samples. The third column indicates which SC algorithm is used to compute the coefficients of the test samples individually.*

We also apply our framework on the MNIST dataset. However, due to the amount and complexity of this dataset, we were not able to fully exploit different dictionary structures and parameters to obtain a reasonable result. The parameters to obtain the results in Table 3 are $(n_b, n_a) = (40, 40)$[5], $\beta = 500$, $\lambda_{\text{train}} = 1.20$, and $\lambda_{\text{test}} = 0.4$. 300 groups, each contains 100 data, were used.

4. Conclusion

We have proposed a novel dictionary learning framework that includes two novel block/group regularized sparse coding algorithms and one novel dictionary learning algorithm. Experimental comparisons with several state-of-the-art dictionary learning methods are favorable, and in particular, for hand-written digit recognition experiment, the proposed framework outperformed these state-of-the-art dictionary learning algorithms.

References

[1] S. Bengio, F. Pereira, Y. Singer, and D. Strelow. Group sparse coding. *Advances in NIPS*, 22:82–89, 2009.

[2] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2:27:1–27:27, 2011.

[3] A. Drémeau and C. Herzet. An em-algorithm approach for the design of orthonormal bases adapted to sparse representations. In *ICASSP, 2010*, pages 2046–2049. IEEE, 2010.

[4] M. Elad. *Sparse and Redundant Representations.* Springer Verlag, 2010.

[5] Y. Eldar, P. Kuppinger, and H. Bolcskei. Block-sparse signals: Uncertainty relations and efficient recovery. *Signal Process., IEEE Trans. on*, 58(6):3042–3054, 2010.

[6] E. Elhamifar and R. Vidal. Robust classification using structured sparse representation. In *CVPR, 2011 IEEE Conference on*, pages 1873–1879. IEEE, 2011.

[7] E. Elhamifar and R. Vidal. Block-sparse recovery via convex optimization. *Signal Process., IEEE Trans. on*, PP(99):1, 2012.

[8] B. Haasdonk and D. Keysers. Tangent distance kernels for support vector machines. In *ICPV, 2002*, volume 2, pages 864–868. IEEE, 2002.

[9] J. Hull. A database for handwritten text recognition research. *Pattern Anal. Mach. Intell., IEEE Trans. on*, 16(5):550–554, 1994.

[10] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for sparse hierarchical dictionary learning. In *International Conference on Machine Learning (ICML)*, 2010.

[11] S. Lesage, R. Gribonval, F. Bimbot, and L. Benaroya. Learning unions of orthonormal bases with thresholded svd. In *ICASSP'05.*, volume 5, pages v–293. IEEE, 2005.

[12] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. *Advances in NIPS*, 2008.

[13] I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *CVPR, 2010*, pages 3501–3508. IEEE, 2010.

[14] R. Rubinstein, A. Bruckstein, and M. Elad. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057, 2010.

[15] P. Sprechmann, I. Ramirez, G. Sapiro, and Y. Eldar. C-hilasso: A collaborative hierarchical sparse modeling. *Signal Process., IEEE Trans. on*, 59(9):4183–4198, 2011.

[16] Z. Szabo, B. Poczos, and A. Lorincz. Online group-structured dictionary learning. In *CVPR, 2011 IEEE Conference on*, pages 2865 –2872, june 2011.

[17] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *Pattern Anal. Mach. Intell., IEEE Trans. on*, 31(2):210–227, 2009.

[18] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, 68(1):49–67, 2006.

[5] Our dictionary size is 5 times smaller than what was used in SISF-DL.