Robust Classification using Structured Sparse Representation

Ehsan Elhamifar   René Vidal
Center for Imaging Science, Johns Hopkins University, Baltimore MD 21218, USA

Abstract

In many problems in computer vision, data in multiple classes lie in multiple low-dimensional subspaces of a high-dimensional ambient space. However, most of the existing classification methods do not explicitly take this structure into account. In this paper, we consider the problem of classification in the multi-subspace setting using sparse representation techniques. We exploit the fact that the dictionary of all the training data has a block structure where the training data in each class form few blocks of the dictionary. We cast the classification as a structured sparse recovery problem where our goal is to find a representation of a test example that uses the minimum number of blocks from the dictionary. We formulate this problem using two different classes of non-convex optimization programs. We propose convex relaxations for these two non-convex programs and study conditions under which the relaxations are equivalent to the original problems. In addition, we show that the proposed optimization programs can be modified properly to also deal with corrupted data. To evaluate the proposed algorithms, we consider the problem of automatic face recognition. We show that casting the face recognition problem as a structured sparse recovery problem can improve the results of the state-of-the-art face recognition algorithms, especially when we have relatively small number of training data for each class. In particular, we show that the new class of convex programs can improve the state-of-the-art face recognition results by 10% *with only* 25% *of the training data. In addition, we show that the algorithms are robust to occlusion, corruption, and disguise.*

1. Introduction

Classification is one of the most fundamental problems in machine learning and has numerous applications in different areas including computer vision. Given training data from multiple classes, the task is to find the class to which a test example belongs.

Recently, there has been an increasing interest in classification problems where the data across multiple classes come from a collection of low-dimensional linear subspaces. In fact, for many important problems in computer vision such as face recognition [13], motion segmentation [5], and activity recognition [14], the data lie in multiple low-dimensional subspaces of a high dimensional ambient space. However, most existing classification methods do not explicitly take into account the multi-subspace structure of the data.

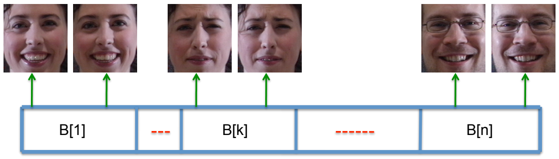

Figure 1. In face recognition, the dictionary has a block structure where the training images of each subject form a few blocks of the dictionary.

An important class of methods that deals with data on multiple subspaces relies on the notion of sparsity. Specifically, the sparse representation-based classification (SRC) method [13] looks for the sparsest representation of a test example in a dictionary composed of all training data across all classes. More formally, given a dictionary $\boldsymbol{B}$ and a test example $\boldsymbol{y}$, it solves the following non-convex program

$$P_{\ell_0}: \quad \min \|\boldsymbol{c}\|_0 \quad \text{s.t.} \quad \boldsymbol{y} = \boldsymbol{B}\boldsymbol{c},$$

where $\|\boldsymbol{c}\|_0$ denotes the number of nonzero elements of $\boldsymbol{c}$. Assuming that the underlying subspace for each class is low-dimensional, the sparsest representation of a test example ideally corresponds to the training data from the same class. When it comes to the problem of robust classification, the SRC method offers a great advantage over many classification methods since it can effectively deal with corrupted data within the same sparse representation framework.

Challenges. While sparse representation-based methods have been shown to be effective for classification, there still remain questions about classification in the multi-subspace setting using sparse representation which have not been sufficiently explored or have not been answered yet.

C1– The SRC method looks for the sparsest representation of a test example with the hope that such a representation selects few training data from the correct class. However, as shown in Figure 1, the dictionary of the training data has a structure in which data from each class form few blocks of the dictionary. *Is there a way to direct the SRC method to take into account the dictionary structure, e.g., by finding a representation of a test example that involves only a few blocks of the dictionary corresponding to the training data from a single class.* If so, *what would be the behavior of the new algorithms in dealing with corrupted data?*

C2– When it comes to the problem of classification on multiple subspaces, there is a fundamental gap between the theory of sparse recovery and the practice of machine learning.

C2a– When the number of training data in each class is large, we can better capture the underlying distribution of data and the classification performance increases. Nonetheless, existing sparse recovery algorithms do not have theoretical guarantees when it comes to highly redundant dictionaries and the conditions for their success almost never hold. *Can we fill the gap between the current sparse representation theory and the classification practice?*

C2b– When the number of training data in each class is small, sparse recovery methods have good theoretical guarantees. However, classification algorithms do not perform well. *Can we have alternative methods based on sparse representation that can lead to better classification results when the number of training data in each class is small?*

Paper contributions. The goal of this paper is to address the aforementioned challenges. We show that instead of looking for the sparsest representation of a test example $\boldsymbol{y}$ in the dictionary of all the training data $\boldsymbol{B}$, a better criterion for classification is to look for a representation of the test example that involves the minimum number of blocks from the dictionary. We formulate this problem using the following non-convex optimization programs

$$P_{\ell_q/\ell_0} : \ \min \sum_{i=1}^{n} I(\|\boldsymbol{c}[i]\|_q > 0) \quad \text{s.t.} \quad \boldsymbol{y} = \boldsymbol{B}\boldsymbol{c}, \quad (1)$$

and

$$P'_{\ell_q/\ell_0} : \ \min \sum_{i=1}^{n} I(\|\boldsymbol{B}[i]\boldsymbol{c}[i]\|_q > 0) \text{ s.t.} \quad \boldsymbol{y} = \boldsymbol{B}\boldsymbol{c}, \quad (2)$$

where $I(\cdot)$ is the indicator function, $q \geq 1$, and $\boldsymbol{c}[i] \in \mathbb{R}^{m_i}$ are the entries of $\boldsymbol{c}$ corresponding to the i-th block of the dictionary, $\boldsymbol{B}[i] \in \mathbb{R}^{D \times m_i}$, as shown in Figure 1. We also show that both optimization programs can be properly modified to deal with corrupted data.

In order to solve these problems efficiently, we propose convex relaxations for the two classes of non-convex programs and study conditions under which each class of convex programs is equivalent to the original non-convex formulation, hence can be used for classification. The state-of-the-art structured sparse recovery literature [4, 3, 7] considers the case where $q = 2$ and the training data in each block are linearly independent. We consider an arbitrary $q \geq 1$ and, motivated by practical problems such as face recognition, we allow for arbitrary number of data in each block.

To evaluate the classification performance of the two classes of convex programs, we consider the problem of automatic face recognition. By extensive experiments, we show that the methods based on structured sparse representation improve the state-of-the-art face recognition results for classifying both uncorrupted and corrupted data. More specifically, we show that the proposed convex programs improve the face recognition results by 10% when the number of training data in each class is as small as the dimension of the face subspace [2]. In addition, we show that the algorithms can efficiently handle corruption and occlusion.

Paper organization. In Section 2, we review the sparse representation-based classification (SRC) method. In Section 3, we formulate the classification problem as a structured sparse recovery problem using two different non-convex optimization programs and propose convex relaxations. In Section 4, we derive conditions under which the convex programs are equivalent to the original non-convex formulations. In Section 5, we evaluate the performance of the proposed algorithms on the problem of automatic face recognition. Section 6 concludes the paper.

2. Classification via Sparse Representation

In this section, we review the problem of classification of data in multiple subspaces using sparse representation. Assume we have n classes and we are given m_i training data $\{\boldsymbol{b}_{ij} \in \mathbb{R}^D\}_{j=1}^{m_i}$ for each class i. We denote by $\boldsymbol{B}[i] \in \mathbb{R}^{D \times m_i}$ the collection of training data in the i-th class

$$\boldsymbol{B}[i] \triangleq \begin{bmatrix} \boldsymbol{b}_{i1} & \boldsymbol{b}_{i2} & \cdots & \boldsymbol{b}_{im_i} \end{bmatrix} \in \mathbb{R}^{D \times m_i}, \qquad (3)$$

and denote by $\boldsymbol{B}$ the collection of all training data across all classes

$$\boldsymbol{B} \triangleq \begin{bmatrix} \boldsymbol{B}[1] & \boldsymbol{B}[2] & \cdots & \boldsymbol{B}[n] \end{bmatrix}. \qquad (4)$$

Given a test example $\boldsymbol{y} \in \mathbb{R}^D$, which belongs to one of the n classes, our goal is to find the class to which the test example belongs.

In this paper, we assume that the data in each class live in a low-dimensional linear subspace of $\mathbb{R}^D$. More precisely, we assume that the data in the i-th class live in a subspace S_i of dimension d_i, where $d_i \ll D$. Thus, the training data live in multiple low-dimensional subspaces of a high-dimensional space. In fact, in several important problems in computer vision such as face recognition [13], motion segmentation [5], and activity recognition [14] the data can be well approximated by a union of subspaces. The SRC method [13] is based on the idea that in such cases, a test example has a sparse representation in the dictionary of all the training data across different classes. More precisely, since a test example belonging to one of the classes lives in a low-dimensional subspace, its sparsest representation is a linear combination of a few training data from the correct class. Thus, in principle, we are interested in solving the following optimization problem

$$P_{\ell_0}: \quad \min \|\boldsymbol{c}\|_0 \quad \text{s.t.} \quad \boldsymbol{y} = \boldsymbol{B}\boldsymbol{c}, \tag{5}$$

where $\|\cdot\|_0$ denotes the ℓ_0 semi-norm and indicates the number of nonzero elements of the given vector. Since the P_{ℓ_0} optimization program is NP-hard, a convex relaxation of it is obtained by replacing the ℓ_0 with the ℓ_1 norm and solving the following convex program

$$P_{\ell_1}: \quad \min \|\boldsymbol{c}\|_1 \quad \text{s.t.} \quad \boldsymbol{y} = \boldsymbol{B}\boldsymbol{c}. \tag{6}$$

An important advantage of classification methods based on sparse representation is their ability to deal with corrupted data within the same framework. To see this, let $\boldsymbol{y}_0$ be a test example corrupted with an error $\boldsymbol{e}$ that has a few nonzero entries, *i.e.*, $\boldsymbol{y} = \boldsymbol{y}_0 + \boldsymbol{e}$. Note that $\boldsymbol{y}_0$ has a sparse representation in the dictionary of the training data $\boldsymbol{B}$ and the error has a sparse representation in the standard basis $\boldsymbol{I}$ (the identity matrix in $\mathbb{R}^D$). Thus, in a new dictionary formed by concatenating the training data and the standard basis, $\boldsymbol{y}$ has a sparse representation that can be recovered from

$$\bar{P}_{\ell_0}: \quad \min \left\|\begin{bmatrix}\boldsymbol{c}\\ \boldsymbol{e}\end{bmatrix}\right\|_0 \quad \text{s.t.} \quad \boldsymbol{y} = \begin{bmatrix}\boldsymbol{B} & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{c}\\ \boldsymbol{e}\end{bmatrix}. \tag{7}$$

To solve this problem efficiently, we can use an ℓ_1 relaxation and instead solve the following convex program

$$\bar{P}_{\ell_1}: \quad \min \left\|\begin{bmatrix}\boldsymbol{c}\\ \boldsymbol{e}\end{bmatrix}\right\|_1 \quad \text{s.t.} \quad \boldsymbol{y} = \begin{bmatrix}\boldsymbol{B} & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{c}\\ \boldsymbol{e}\end{bmatrix}. \tag{8}$$

We can then find the class of a given test example as the class that best represents the test example using its training data. More precisely, for a given test example $\boldsymbol{y}$, if we denote by $\boldsymbol{c}^{*\top} = \begin{bmatrix}\boldsymbol{c}^{*\top}[1] & \cdots & \boldsymbol{c}^{*\top}[n]\end{bmatrix}$ the optimal solution of P_{ℓ_1}, the class of $\boldsymbol{y}$ can be obtained by[1]

$$\text{class}(\boldsymbol{y}) = \arg\min_i \|\boldsymbol{y} - \boldsymbol{B}[i]\boldsymbol{c}^*[i]\|_2. \tag{9}$$

3. Classification via Structured Sparsity

As discussed in the previous section, when the training data in each class live in a low-dimensional subspace of a high-dimensional ambient space, the classification problem can be cast as the problem of finding the sparsest representation of a test data in the dictionary of all the training data.

In this section, we argue that looking for the sparsest representation of a test example might not be the best criterion for classification. In order to see this, we consider the example in Figure 2 (left) where we have 3 classes whose data live in three subspaces; S_1 being a 2-dimensional subspace, S_2 and S_3 being 1-dimensional subspaces. The test example $\boldsymbol{y}$, which belongs to class 1, can be written as a linear combination of any two data points from class 1, while it can also be written as a linear combination of one data point from class 2 and one from class 3. Thus, from the sparsest representation perspective, there is no difference between the two representations as they both have two nonzero elements, while obviously from a classification perspective, the first one is the desired solution. Now, if instead of looking for the sparsest representation we look for a representation that uses the minimum number of blocks, we obtain the desired solution for perfect classification.

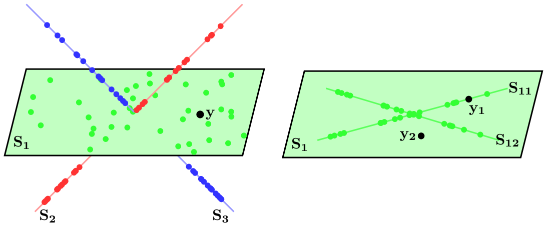

Figure 2. Left: sparsest representation of a test example does not necessarily come from the correct class. $\boldsymbol{y}$ can be written as a linear combination of one data point from S_2 and one from S_3 as well as a linear combination of two data points from S_1. Right: training data in a class might be separated into several blocks. Thus, a test example can be written as a linear combination of a few blocks in each class.

In a general classification task, the dictionary of the training data has a block structure where a few blocks of the dictionary correspond to the training data in each class. Thus, a test example can be represented as a linear combination of training data from a few blocks of the dictionary corresponding to its class. For example, in Figure 2 (right), the test example $\boldsymbol{y}_1$ can be written as a linear combination of 1 block while $\boldsymbol{y}_2$ can be written as a linear combination of two blocks of the underlying class.[2] As another example, in the face recognition problem, each class consists of images of a single subject that can be separated into multiple blocks based on different expressions as shown in Figure 1.

Structured sparse representation via P_{ℓ_q/ℓ_0}. Based on what we have discussed so far, a better objective for classification is to solve

$$P_{\ell_q/\ell_0}: \; \min \sum_{i=1}^{n} I(\|\boldsymbol{c}[i]\|_q > 0) \quad \text{s.t.} \quad \boldsymbol{y} = \boldsymbol{B}\boldsymbol{c}, \tag{10}$$

where $I(\cdot)$ is the indicator function and $q \geq 1$. While in principle we could have chosen any value of $q > 0$, we choose $q \geq 1$ for reasons that will become clear shortly.

[1] For data corrupted with sparse outliers, we use the modified criterion $\text{class}(\boldsymbol{y}) = \arg\min_i \|\boldsymbol{y} - \boldsymbol{e}^* - \boldsymbol{B}[i]\boldsymbol{c}^*[i]\|_2$.

[2] When each class i consists of several blocks indexed by C_i, the class of a test example $\boldsymbol{y}$ is given by $\arg\min_i \|\boldsymbol{y} - \sum_{j \in C_i} \boldsymbol{B}[j]\boldsymbol{c}^*[j]\|_2$.

This optimization problem seeks the minimum number of nonzero coefficient blocks that reconstruct the test example. Note that the optimization program P_{ℓ_q/ℓ_0} is NP-hard since it requires searching exhaustively over all possible few blocks of $\boldsymbol{B}$ and checking whether they span the given $\boldsymbol{y}$. An ℓ_1 relaxation of this program is given by

$$P_{\ell_q/\ell_1}: \quad \min \sum_{i=1}^{n} \|\boldsymbol{c}[i]\|_q \quad \text{s.t.} \quad \boldsymbol{y} = \boldsymbol{B}\boldsymbol{c}, \qquad (11)$$

which is a convex program when $q \geq 1$.

For $q = 1$, while the non-convex programs P_{ℓ_1/ℓ_0} and P_{ℓ_0} are different, their convex relaxations P_{ℓ_1/ℓ_1} and P_{ℓ_1} are the same. Thus, P_{ℓ_1} can also be thought of as a structured sparse recovery method that under appropriate conditions, as will be discussed in the next section, finds a representation of the test example with the minimum number of nonzero blocks.

Structured sparse representation via P'_{ℓ_q/ℓ_0}. We will also consider an alternative optimization program for the classification problem, which can be formulated as

$$P'_{\ell_q/\ell_0}: \min \sum_{i=1}^{n} I(\|\boldsymbol{B}[i]\boldsymbol{c}[i]\|_q > 0) \text{ s.t.} \quad \boldsymbol{y} = \boldsymbol{B}\boldsymbol{c}, \quad (12)$$

whose ℓ_1 relaxation for $q \geq 1$ gives the following convex optimization program

$$P'_{\ell_q/\ell_1}: \min \sum_{i=1}^{n} \|\boldsymbol{B}[i]\boldsymbol{c}[i]\|_q \quad \text{s.t.} \quad \boldsymbol{y} = \boldsymbol{B}\boldsymbol{c}. \qquad (13)$$

Unlike P_{ℓ_q/ℓ_0} that minimizes the number of nonzero coefficient blocks $\boldsymbol{c}[i]$, the optimization program P'_{ℓ_q/ℓ_0} minimizes the number of nonzero reconstructed vectors $\boldsymbol{B}[i]\boldsymbol{c}[i]$. When the blocks consist of linearly independent data, the solution of P'_{ℓ_q/ℓ_0} has also the minimum number of nonzero coefficient blocks, because $\|\boldsymbol{B}[i]\boldsymbol{c}[i]\|_q > 0$ if and only if $\|\boldsymbol{c}[i]\|_q > 0$. However, this does not necessarily hold when the blocks consist of linearly dependent data. To see this, consider a simple example where the data in each class form a single block of the dictionary. Let $\boldsymbol{y}$ be a test example belonging to class l. Thus, it can be written as a linear combination of the training data in the l-th class. Since the vectors in each block are linearly dependent, for every $i \neq l$, we can choose a nonzero $\boldsymbol{c}[i]$ in the null space of $\boldsymbol{B}[i]$, *i.e.*, $\|\boldsymbol{B}[i]\boldsymbol{c}[i]\|_q = 0$, while $\|\boldsymbol{c}[i]\|_q > 0$. Obviously, this does not affect either the value of the cost function or the equality constraint in P'_{ℓ_q/ℓ_0}.

Despite the above argument, we will use P'_{ℓ_q/ℓ_0} for classification in dictionaries whose blocks have linearly independent or linearly dependent data because it still gives the correct classification as per (9). To see this, let us assume for simplicity that each class consists of training data in a single block. If for a test example that belongs to the l-th class, we denote the optimal solution of P'_{ℓ_q/ℓ_0} by $\boldsymbol{c}^*$, then for the l-th block we have $\boldsymbol{B}[l]\boldsymbol{c}^*[l] = \boldsymbol{y}$. For other blocks $i \neq l$, while $\boldsymbol{c}^*[i]$ might be nonzero, we always have $\boldsymbol{B}[i]\boldsymbol{c}^*[i] = 0$. Thus, the l-th class would minimize the classification objective function in (9), $\|\boldsymbol{y} - \boldsymbol{B}[i]\boldsymbol{c}^*[i]\|_2$, resulting in correct classification.

Dealing with corrupted data. We will now show that the proposed structured sparse recovery methods can also deal with corrupted data within the same framework. Let $\boldsymbol{y}_0$ be a test example corrupted by a sparse error $\boldsymbol{e}$, *i.e.*, $\boldsymbol{y} = \boldsymbol{y}_0 + \boldsymbol{e}$. The uncorrupted data $\boldsymbol{y}_0$ can be written as a linear combination of a few blocks of the training data dictionary. Also, $\boldsymbol{e}$ can be written as a linear combination of a few blocks of the standard basis $\boldsymbol{I}$, where we treat each column of $\boldsymbol{I}$ as a block of length 1. Thus, the corrupted test example, $\boldsymbol{y}$, can be written as a linear combination of few blocks of a new dictionary composed of the training data and the standard basis. This structured sparse representation can be recovered, after convex relaxation, by

$$\bar{P}_{\ell_q/\ell_1}: \min \sum_{i=1}^{n} \|\boldsymbol{c}[i]\|_q + \|\boldsymbol{e}\|_1 \text{ s.t. } \boldsymbol{y} = \begin{bmatrix}\boldsymbol{B} & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{c}\\ \boldsymbol{e}\end{bmatrix}, \quad (14)$$

where we used the fact that the blocks of $\boldsymbol{I}$ have length 1, *i.e.*, $\mathbf{e}[i] \in \mathbb{R}$. Thus, $\sum_{i=1}^{D} \|\boldsymbol{e}[i]\|_q = \|\boldsymbol{e}\|_1$. Similarly, P'_{ℓ_q/ℓ_0} can also deal with corrupted data, in which case we have to solve the following convex program

$$\bar{P}_{\ell_q/\ell_1}: \min \sum_{i=1}^{n} \|\boldsymbol{B}[i]\boldsymbol{c}[i]\|_q + \|\boldsymbol{e}\|_1 \text{ s.t. } \boldsymbol{y} = \begin{bmatrix}\boldsymbol{B} & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{c}\\ \boldsymbol{e}\end{bmatrix}. \quad (15)$$

4. Theoretical Results

In the previous section, we showed that when data in multiple classes live in multiple low-dimensional subspaces, the classification problem can be cast as a structured sparse recovery problem where we are interested in solving the non-convex optimization programs P_{ℓ_q/ℓ_0} and P'_{ℓ_q/ℓ_0}.

In this section, we study conditions under which the convex relaxations P_{ℓ_q/ℓ_1} and P'_{ℓ_q/ℓ_1} are equivalent to the original non-convex programs. Unlike the state-of-the-art structured sparse recovery literature [4, 3, 7] that only consider the case where $q = 2$ and the data in each block are linearly independent, our theoretical analysis allows for arbitrary $q \geq 1$. Also, motivated by practical problems such as classification, we allow for arbitrary number of data in each block. In addition, our theoretical results can be simply generalized to the convex programs $\bar{P}_{\ell_q/\ell_1}$ and $\bar{P}'_{\ell_q/\ell_1}$ that deal with corrupted data.

Recall that a dictionary $\boldsymbol{B}$ consists of the training data from n blocks $\boldsymbol{B}[i] \in \mathbb{R}^{D \times m_i}$, where the data in each block live in a low-dimensional subspace S_i of dimension $d_i \ll D$. We can characterize a dictionary from two different perspectives. On the one hand, we can capture the relations between different blocks (interblock relation) and on the other hand, we can capture the relations among data in each block (intrablock relation). First, we characterize the relations between different blocks of a dictionary by assuming that the collection of subspaces is disjoint.

Definition 1 *A collection of subspaces $\{S_i\}_{i=1}^n$ is called* disjoint *if each pair of subspaces intersect only at the origin. For a collection of disjoint subspaces, define the mutual subspace coherence as*

$$\mu_S = \max_{i \neq j} \max_{\boldsymbol{x} \in S_i, \boldsymbol{z} \in S_j} \frac{\boldsymbol{x}^\top \boldsymbol{z}}{\|\boldsymbol{x}\|_2 \|\boldsymbol{z}\|_2}. \tag{16}$$

Note that the mutual subspace coherence is equal to the cosine of the smallest principal angle among all pairs of different subspaces. Next, we characterize the relation among the data in the blocks of the dictionary.

Definition 2 *For a dictionary $\boldsymbol{B}$, define ϵ_q as the smallest constant such that for every i there exists a full rank submatrix of $\boldsymbol{B}[i]$, denoted by $\bar{\boldsymbol{B}}[i] \in \mathbb{R}^{D \times d_i}$, such that for every $\bar{\boldsymbol{c}}[i]$ we have*

$$(1-\epsilon_q)\|\bar{\boldsymbol{c}}[i]\|_q^2 \leq \|\bar{\boldsymbol{B}}[i]\bar{\boldsymbol{c}}[i]\|_2^2 \leq (1+\epsilon_q)\|\bar{\boldsymbol{c}}[i]\|_q^2. \tag{17}$$

Also, define σ_q as the smallest constant such that for every i and $\boldsymbol{c}[i]$ we have

$$\|\boldsymbol{B}[i]\boldsymbol{c}[i]\|_q^2 \leq \sigma_q \|\boldsymbol{c}[i]\|_q^2. \tag{18}$$

When $q = 2$ and the blocks are full rank, *i.e.*, $\boldsymbol{B}[i] = \bar{\boldsymbol{B}}[i]$, ϵ_2 coincides with the 1-block restricted isometry constant defined in [4]. Thus, ϵ_q can be thought of as a more general notion that allows for any $q \geq 1$ and arbitrary number of data in each block.

Definition 3 *For a dictionary $\boldsymbol{B}$, define ϵ'_q as the smallest constant such that for every i and $\boldsymbol{c}[i]$ we have*

$$(1-\epsilon'_q)\|\boldsymbol{B}[i]\boldsymbol{c}[i]\|_q^2 \leq \|\boldsymbol{B}[i]\boldsymbol{c}[i]\|_2^2 \leq (1+\epsilon'_q)\|\boldsymbol{B}[i]\boldsymbol{c}[i]\|_q^2. \tag{19}$$

ϵ'_q characterizes the relation between the ℓ_q and ℓ_2 norms of vectors in $\mathbb{R}^D$ and does not depend on the number of the data in each block. Note that for $q = 2$, we have $\epsilon'_2 = 0$.

The following results establish conditions under which the convex programs P_{ℓ_q/ℓ_1} and P'_{ℓ_q/ℓ_1} are equivalent to the original non-convex programs. The proofs of the results can be found in [6].

Theorem 1 *For a vector that can be written as a linear combination of k blocks of the dictionary, the optimization program P_{ℓ_q/ℓ_1} is equivalent to P_{ℓ_q/ℓ_0} if*

$$\left(k\sqrt{\frac{\sigma_q}{1+\epsilon_q}} + k - 1\right)\mu_S < \frac{1-\epsilon_q}{1+\epsilon_q}. \tag{20}$$

Theorem 2 *For a vector that can be written as a linear combination of k blocks of the dictionary, the optimization program P'_{ℓ_q/ℓ_1} is equivalent to P'_{ℓ_q/ℓ_0} if*

$$(2k-1)\,\mu_S < \frac{1-\epsilon'_q}{1+\epsilon'_q}. \tag{21}$$

5. Experiments

In this section, we evaluate the performance of the two classes of convex programs on the problem of automatic face recognition. We also investigate the robustness of the proposed algorithms in dealing with corrupted data.

5.1. Classifying Uncorrupted Images

In this part, we evaluate the performance of the proposed methods as well as the state-of-the-art face recognition algorithms for classifying uncorrupted data on the Extended Yale B database [10]. The Extended Yale B database consists of $2,414$ cropped frontal face images of $n = 38$ individuals. For each subject, there are approximately 64 face images of size $192 \times 168 = 32,256$, which are captured under various laboratory-controlled lighting conditions. Since the dimension of the original face vectors is large, we reduce the dimension of the data using the following methods.
–We down-sample the images by a factor r such that the dimension of the down-sampled face vectors is D.
–We use the eigenfaces approach [12] by projecting the face vectors to the first D principal components of the training data covariance matrix.
–We multiply the face vectors by a random projection matrix $\Phi \in \mathbb{R}^{D \times 32,256}$ which has i.i.d. entries drawn from a zero mean Gaussian distribution with variance $\frac{1}{D}$ [1] .

In the experiments, we set $D = 132$. For simplicity of the analysis, we assume that all classes have the same number of training data, $m_i = m$. To investigate the effect of the number of training data in the classification performance, we randomly select $m \in \{9, 18, 25, 32\}$ training images for each subject, forming blocks $\boldsymbol{B}[i] \in \mathbb{R}^{D \times m}$, and use the remaining images for testing. For every test image, we solve the convex programs P_{ℓ_q/ℓ_1} and P'_{ℓ_q/ℓ_1} for $q \in \{1, 2\}$ and determine the identity of the test image using (9).[3] We compute the classification rate as the average number of correctly classified test images for which the recovered identity matches the ground-truth. We repeat this experiment 20 times for random choices of m training data for each subject and compute the mean classification rate among all the trials.

Note that P_{ℓ_1/ℓ_1}, which is equivalent to P_{ℓ_1}, corresponds to the SRC method [13] that has previously reported the best

[3] Because of modeling error and noise in the real data, we use the modified convex programs whose equality constraints are replaced with $\|\boldsymbol{y} - \boldsymbol{B}\boldsymbol{c}\|_2 \leq \epsilon$. In our experiments, $\epsilon = 0.05$.

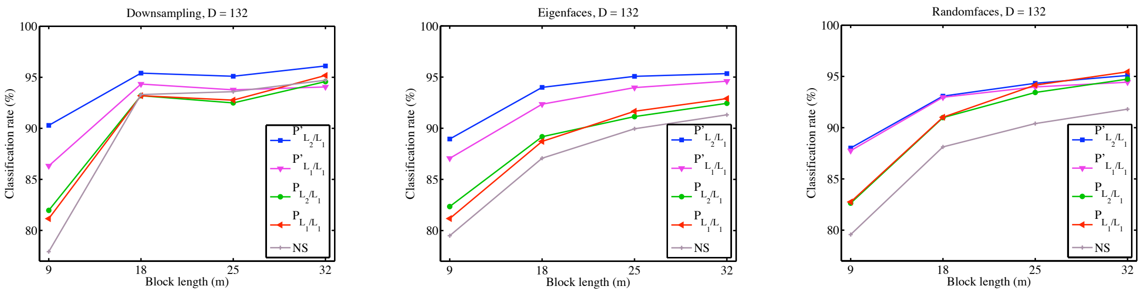

Figure 3. Recognition results on the Extended Yale B database as a function of the number of training data in each class.

results on the database. Since the dataset contains a single expression for each person, the training data for each subject form one block of the dictionary. Hence, one can think of a classification method that looks for a subject whose underlying subspace is closest to the given test image. This method is known in the literature as the nearest subspace (NS) method [8] and we use it as a baseline for comparison.

The recognition results are shown in Figure 3. As the results show, the NS method has, in general, lower performance than methods based on sparse representation. Moreover, for a fixed value of q, the convex program P'_{ℓ_q/ℓ_1} almost always outperforms P_{ℓ_q/ℓ_1}.

While the performances of different methods are close for large number of training data in each class, the difference in their performances becomes evident when the number of data in each class decreases. Note that while the performance of all the algorithms degrade by decreasing the number of data in each class, the convex programs P'_{ℓ_q/ℓ_1} are more robust to decreasing the number of training data. Specifically, when the number of training data in each class is as small as the dimension of the face subspace [2], *i.e.*, $m = d = 9$, P'_{ℓ_2/ℓ_1} has almost 10% higher recognition rate than the SRC method. It is also important to note that our results are independent of the choice of features, *i.e.*, the results follow the same pattern for the three types of features as shown in Figure 3. In all of them P'_{ℓ_2/ℓ_1} and P'_{ℓ_1/ℓ_1} achieve the best recognition results.

5.2. Robustness to Random Corruption

In this section, we test the robust versions of the structured sparsity-based algorithms in dealing with random pixel corruption. To that end, we choose images in subset 1 (and 2) of the Extended Yale B database for training and choose images in subset 3 for testing. We downsample the images so that $D = 1,400$. Without corrupting the images, this is not a hard problem and this choice is to isolate the effect of random corruption. Next, we corrupt ρ percentage of randomly chosen pixels in each test image. We replace the values of the chosen pixels by i.i.d. values drawn from a uniform distribution in the range of the image pixel values. We change ρ from 0 to 90 percent and compute the recognition rate. We compare the results of the robust structured sparsity-based classification algorithms $\bar{P}_{\ell_2/\ell_1}$ and $\bar{P}'_{\ell_2/\ell_1}$ with three other methods. First, we use the robust version of the SRC method, $\bar{P}_{\ell_1}$. Next, we use the basic PCA to project the data into lower dimensions and use the NN classifier. Third, we use the Independent Component Analysis (ICA) architecture I [9] with a NN classifier.[4]

For $m \in \{7, 19\}$ training data in each class, the recognition rates as a function of the percentage of corrupted pixels are shown in Figure 4. For both cases, $\bar{P}_{\ell_2/\ell_1}$ and $\bar{P}'_{\ell_2/\ell_1}$ as well as $\bar{P}_{\ell_1}$ achieve almost 100% recognition rate with up to 50% corruption, while the recognition rates of the two other methods drop quickly to less than 30% when we have 50% corrupted pixels. Note that $\bar{P}_{\ell_2/\ell_1}$ and $\bar{P}'_{\ell_2/\ell_1}$ obtain better classification results than $\bar{P}_{\ell_1}$ when the number of training data in each class is small ($m = 7$). However, for $m = 19$, the performances of $\bar{P}_{\ell_2/\ell_1}$ and $\bar{P}_{\ell_1}$ are similar.

5.3. Robustness to Random Block Occlusion

In this section, we test the performance of the structured sparsity-based classification methods in dealing with corrupted data, where corruption appears in a block of a face image instead of being distributed across all image pixels.

We use images in subset 1 (and 2) of the Extended Yale B database for training and use images in subset 3 for testing. We down-sample the images so that $D = 1,400$. In order to examine the robustness of the methods to occlusions we replace a randomly chosen square block of each test image with an unrelated image and change the percentage of occlusion from 0 to 50 percent. Similar to the previous section, we compare the performance of $\bar{P}_{\ell_2/\ell_1}$ and $\bar{P}'_{\ell_2/\ell_1}$ against the SRC method, PCA+NN and ICA+NN. For $m \in \{7, 19\}$ training data in each class, the results are shown in Figure 5. Note that the sparse representation based methods achieve almost 100% recognition rate up to 20% occlusion, while the recognition rates of PCA+NN and ICA+NN quickly drop as we increase the the percentage of occlusion. In addition, for $m = 7$, both $\bar{P}_{\ell_2/\ell_1}$ and $\bar{P}'_{\ell_2/\ell_1}$

[4] For PCA and ICA, we choose the number of basis components over the range $\{100, 200, 300, 400\}$ to give the best test performance.

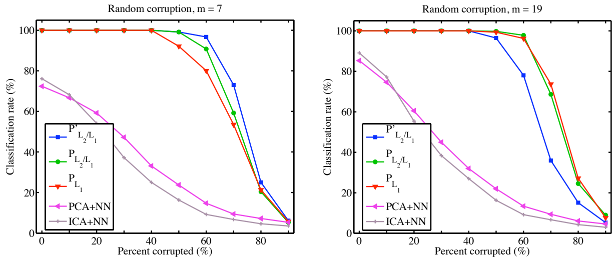

Figure 4. Recognition results on the Extended Yale B database as a function of the percentage of corruption.

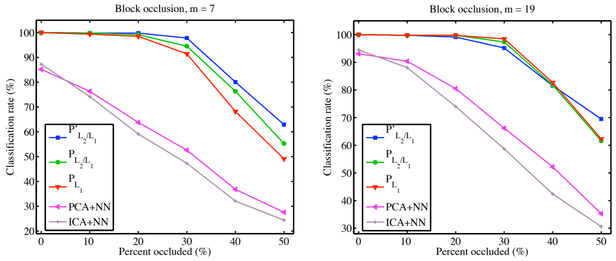

Figure 5. Recognition results on the Extended Yale B database as a function of the percentage of block occlusion.

Table 1. Recognition rates on the AR database for robustness to disguise.

Algorithms	P'_{ℓ_2/ℓ_1}	P_{ℓ_2/ℓ_1}	P_{ℓ_1}	PCA+NN	ICA+NN
Sunglasses	66.5%	80.5%	**84.3%**	57.5%	51.7%
Scarves	41.7%	**59.8%**	35.2%	10.5%	9.2%
All	54.1%	**70.2%**	59.8%	34.0%	30.5%

obtain better recognition rates than $\bar{P}_{\ell_1}$ for all percentages of occlusion.

5.4. Robustness to Disguise

In this part, we examine the robustness of the proposed algorithms to real malicious occlusions in images. We use the AR database [11] which consists of face images of $n = 100$ individuals acquired under the same pose with varying illuminations and expressions. Out of the 26 images for each subject, in 6 images the subject is wearing sunglasses, roughly occluding 20% of the image, and in 6 images, the subject is wearing a scarf, occluding nearly 40% of the image. We down-sample the images so that $D = 1,400$. We randomly select $m = 9$ images for each subject as the training data and use the images with sunglasses and scarves as test examples. We evaluate the recognition rates of the structured-sparsity based algorithms as well as the SRC method and the two other algorithms we used in the previous experiments: PCA+NN and ICA+NN. The results are shown in Table 1. While $\bar{P}_{\ell_1}$ obtains slightly better recognition rate than $\bar{P}_{\ell_2/\ell_1}$ for images with sunglasses, $\bar{P}_{\ell_2/\ell_1}$ obtains about 25% higher recognition rate than $\bar{P}_{\ell_1}$ for images with scarves.

6. Conclusions

We formulated the problem of classification as a structured sparse recovery problem using two non-convex optimization programs P_{ℓ_q/ℓ_0} and P'_{ℓ_q/ℓ_0}. To solve them efficiently, we proposed convex relaxations for the two non-convex programs and studied conditions under which they are equivalent to the original non-convex formulations. We showed that the proposed algorithms can be modified to also deal with corrupted data. Our experiments on the face recognition problem showed that the proposed classification methods lead to better recognition results especially when the number of training data in each class is relatively small.

Acknowledgment

This work was partially supported by grants NSF CNS-0931805, NSF CNS-0941463.

References

[1] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 2008.

[2] R. Basri and D. Jacobs. Lambertian reflection and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(3):218–233, 2003.

[3] Y. C. Eldar, P. Kuppinger, and H. Bolcskei. Compressed sensing of block-sparse signals: Uncertainty relations and efficient recovery. *IEEE Trans. Signal Processing*, 58(6):3042–3054, June 2010.

[4] Y. C. Eldar and M. Mishali. Robust recovery of signals from a structured union of subspaces. *IEEE Trans. Inform. Theory*, 55(11):5302–5316, 2009.

[5] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[6] E. Elhamifar and R. Vidal. Structured sparse recovery via convex optimization. *IEEE Trans. Signal Process.*, submitted. Available: http://arxiv.org/abs/1104.0654.

[7] A. Ganesh, Z. Zhou, and Y. Ma. Separation of a subspace-sparse signal: Algorithms and conditions. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2009.

[8] J. Ho, M. H. Yang, J. Lim, K. Lee, and D. Kriegman. Clustering appearances of objects under varying illumination conditions. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[9] J. Y. J. Kim, J. Choi and M. Turk. Effective representation using ica for face recognition robust to local distortion and partial occlusion. *PAMI*, 27(12):1977–1981, 2005.

[10] K.-C. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):684–698, 2005.

[11] A. Martinez and R. Benavente. The ar face database. *CVC Technical Report 24*, 1998.

[12] M. Turk and A. Pentland. Face recognition using eigenfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[13] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, Feb. 2009.

[14] A. Yang, R. Jafari, P. Kuryloski, S. Iyengar, S. Sastry, and R. Bajcsy. Distributed segmentation and classification of human actions using a wearable sensor network. *CVPR Workshop on Human Communicative Behavior Analysis*, 2008.