4

# COMPRESSIVE CLASSIFICATION FOR FACE RECOGNITION

Angshul Majumdar and Rabab K. Ward

## 1. INTRODUCTION

Face images (with column/row concatenation) form very high dimensional vectors, e.g. a standard webcam takes images of size 320x240 pixels, which leads to a vector of length 76,800. The computational complexity of most classifiers is dependent on the dimensionality of the input features, therefore if all the pixel values of the face image are used as features for classification the time required to finish the task will be excessively large. This prohibits direct usage of pixel values as features for face recognition.

To overcome this problem, different dimensionality reduction techniques has been proposed over the last two decades – starting from Principal Component Analysis and Fisher Linear Discriminant. Such dimensionality reduction techniques have a basic problem – they are data-dependent adaptive techniques, i.e. the projection function from the higher to lower dimension cannot be computed unless all the training samples are available. Thus the system cannot be updated efficiently when new data needs to be added.

Data dependency is the major computational bottleneck of such adaptive dimensionality reduction methods. Consider a situation where a bank intends to authenticate a person at the ATM, based on face recognition. So, when a new client is added to its customer base, a training image of the person is acquired. When that person goes to an ATM, another image is acquired by a camera at the ATM and the new image is compared against the old one for identification. Suppose that at a certain time the bank has 200 customers, and is employing a data-dependent dimensionality reduction method. At that point of time it has computed the projection function from higher to lower dimension for the current set of images. Assume that at a later time, the bank has 10 more clients, then with the data-dependent dimensionality reduction technique, the projection function for all the 210 samples must be recomputed from scratch; in general there is no way the previous projection function can be updated with results of the 10 new samples only. This is a major computational bottleneck for the practical application of current face recognition research.

For an organization such as a bank, where new customers are added regularly, it means that the projection function from higher to lower dimension will have to be updated regularly. The cost of computing the projection function is intensive and is dependent on the number of samples. As the number of samples keeps on increasing, the computational cost keeps on increasing as well (as every time new customers are added to the training dataset, the projection function has to be recalculated from scratch). This becomes a major issue for any practical face recognition system.

One way to work around this problem is to skip the dimensionality reduction step. But as mentioned earlier this increases the classification time. With the ATM scenario there is another problem as well. This is from the perspective of communication cost. There are two possible scenarios in terms of transmission of information – 1) the ATM sends the image to some central station where dimensionality reduction and classification are carried out or 2) the dimensionality reduction is carried out at the ATM so that the dimensionality reduced feature vector is sent instead. The latter reduces the volume of data to be sent over the internet but requires that the dimensionality reduction function is available at the ATM. With the first scenario, the communication cost arises from sending the whole image over the communication channel. In the second scenario, the dimensionality reduction function is available at the ATM. As this function is data-dependent it needs to be updated every time new samples are added. Periodically updating the function increases the communication cost as well.

In this work we propose a dimensionality reduction method that is independent of the data. Practically this implies that the dimensionality reduction function is computed once and for all and is available at all the ATMs. There is no need to update it, and the ATM can send the dimensionality reduced features of the image. Thus both the computational cost of calculating the projection function and the communication cost of updating it are reduced simultaneously.

Our dimensionality reduction is based on Random Projection (RP). Dimensionality reduction by random projection is not a well researched topic. Of the known classifiers only the K Nearest Neighbor (KNN) is robust to such dimensionality reduction [1]. By robust, it is meant that the classification accuracy does not vary much when the RP dimensionality reduced samples are used in classification instead of the original samples (without dimensionality reduction). Although the KNN is robust, its recognition accuracy is not high. This shortcoming has motivated researchers in recent times to look for more sophisticated classification algorithms that will be robust to RP dimensionality reduction [2, 3].

In this chapter we will review the different compressive classification algorithms that are robust to RP dimensionality reduction. However, it should be remembered that these classifiers can also be used with standard dimensionality reduction techniques like Principal Component Analysis.

In signal processing literature random projection of data are called 'Compressive Samples'. Therefore the classifiers which can classify such RP dimensionality reduced data are called 'Compressive Classifiers'. In this chapter we will theoretically prove the robustness of compressive classifiers to RP dimensionality reduction. The theoretical proofs will be validated by thorough experimentation. Rest of the chapter will be segregated into several sections. In section 2, the different compressive classification algorithms will be discussed. The theoretical proofs regarding their robustness will be provided in section 3. The experimental evaluation will be carried out in section 4. Finally in section 5, conclusions of this work will be discussed.

## 2. CLASSIFICATION ALGORITHMS

The classification problem is that of finding the identity of an unknown test sample given a set of training samples and their class labels. Compressive Classification addresses the case where compressive samples (random projections) of the original signals are available instead of the signal itself.

If the original high dimensional signal is 'x', then its dimensionality is reduced by

$$y = Ax$$

where A is a random projection matrix formed by normalizing the columns of an i.i.d. Gaussian matrix and y is the dimensionality reduced compressive sample. The compressive classifier has access to the compressive samples and must decide the class based on them.
Compressive Classifiers have two challenges to meet:
The classification accuracy of CC on the original signals should be at par with classification accuracy from traditional classifiers (SVM or ANN or KNN).
The classification accuracy from CC should not degrade much when compressed samples are used instead of the original signals.
Recently some classifiers have been proposed which can be employed as compressive classifiers. We discuss those classification algorithms in this section.

### 2.1 The Sparse Classifier

The Sparse Classifier (SC) is proposed in [2]. It is based on the assumption that the training samples of a particular class approximately form a linear basis for a new test sample belonging to the same class. If $\boldsymbol{v}_{k,test}$ is the test sample belonging to the $k^{th}$ class then,

$$v_{k,test} = \alpha_{k,1}v_{k,1} + \alpha_{k,2}v_{k,2} + \ldots + \alpha_{k,n_k}v_{k,n_k} + \varepsilon_k = \sum_{i=1}^{n_k}\alpha_{k,i}v_{k,i} + \varepsilon_k \tag{1}$$

where $\boldsymbol{v}_{k,i}$'s are the training samples of the $k^{th}$ class and $\varepsilon_k$ is the approximation error (assumed to be Normally distributed).
Equation (1) expresses the assumption in terms of the training samples of a single class. Alternatively, it can be expressed in terms of all the training samples such that

$$\begin{aligned} v_{k,test} &= \alpha_{1,1} + \ldots\alpha_{k,1}v_{k,1} + \ldots + \alpha_{k,n_k}v_{k,n_k} + \ldots + \alpha_{C,n_C}v_{C,n_C} + \varepsilon \\ &= \sum_{i=1}^{n_1}\alpha_{1,i}v_{1,i} + \ldots\sum_{i=k}^{n_k}\alpha_{k,i}v_{k,i} + \ldots + \sum_{i=1}^{n_C}\alpha_{C,i}v_{C,i} + \varepsilon \end{aligned} \tag{2}$$

where C is the total number of classes.
In matrix vector notation, equation (2) can be expressed as

$$v_{k,test} = V\alpha + \varepsilon \tag{3}$$

where $V = [v_{1,1} | \ldots | v_{k,1} | \ldots | v_{k,n_k} | \ldots | v_{C,n_C}]$ and $\alpha = [\alpha_{1,1}\ldots\alpha_{k,1}\ldots\alpha_{k,n_k}\ldots\alpha_{C,n_C}]'$.
The linearity assumption in [2] coupled with the formulation (3) implies that the coefficients vector α should be non-zero only when they correspond to the correct class of the test sample.

Based on this assumption the following sparse optimization problem was proposed in [2]

$$\min \| \alpha \|_0 \text{ subject to } \| v_{k,test} - V\alpha \|_2 \leq \eta, \ \eta \text{ is related to } \varepsilon \tag{4}$$

As it has already been mentioned, (4) is an NP hard problem. Consequently in [2] a convex relaxation to the NP hard problem was made and the following problem was solved instead

$$\min \| \alpha \|_1 \quad \text{subject to} \quad \| v_{k,test} - V\alpha \|_2 \leq \eta \tag{5}$$

The formulation of the sparse optimization problem as in (5) is not ideal for this scenario as it does not impose sparsity on the entire class as the assumption implies. The proponents of Sparse Classifier [2] 'hope' that the l1-norm minimization will find the correct solution even though it is not imposed in the optimization problem explicitly. We will speak more about group sparse classification later.
The sparse classification (SC) algorithm proposed in [2] is the following:

Sparse Classifier Algorithm
1. Solve the optimization problem expressed in (5).
2. For each class (i) repeat the following two steps:
3. Reconstruct a sample for each class by a linear combination of the training samples

$$v_{recon}(i) = \sum_{j=1}^{n_i} \alpha_{i,j} v_{i,j}$$

belonging to that class using.

4. Find the error between the reconstructed sample and the given test sample by $error(v_{test}, i) = \| v_{k,test} - v_{recon(i)} \|_2$.
5. Once the error for every class is obtained, choose the class having the minimum error as the class of the given test sample.
The main workhorse behind the SC algorithm is the optimization problem (5). The rest of the steps are straightforward. We give a very simple algorithm to solve this optimization problem.

IRLS algorithm for l1 minimization

Initialization – set δ(0) = 0 and find the initial $\hat{x}(0) = \min \| y - Ax \|_2^2$ by conjugate gradient method.
At iteration t – continue the following steps till convergence (i.e. either δ is less than $10^{-6}$ or the number of iterations has reached maximum limit)
1. Find the current weight matricex as $W_m(t) = diag(2 \mid x(t-1) + \delta(t) \mid^{-1/2})$
2. Form a new matrix, $L = AW_m$.
3. Solve $\hat{u}(t) = \min \| y - Lu \|_2^2$ by conjugate gradient method.
4. Find x by rescaling u, $x(t) = W_m u(t)$.
5. Reduce δ by a factor of 10 if ||y-Ax|| q has reduced.

This algorithm is called the Iterated Reweighted Least Squares (IRLS) algorithm [4] and falls under the general category of FOCUSS algorithms [5].

### 2.2 Fast Sparse Classifiers

The above sparse classification (SC) algorithm yields good classification results, but it is slow. This is because of the convex optimization (l1 minimization). It is possible to create faster versions of the SC by replacing the optimization step (step 1 of the above algorithm) by a fast greedy (suboptimal) alternative that approximates the original l0 minimization problem (4). Such greedy algorithms serve as a fast alternative to convex-optimization for sparse signal estimation problems. In this work, we apply these algorithms in a new perspective (classification).

We will discuss a basic greedy algorithm that can be employed to speed-up the SC [2]. The greedy algorithm is called the Orthogonal Matching Pursuit (OMP) [6]. We repeat the OMP algorithms here for the sake of completeness. This algorithm approximates the NP hard problem, $\min \| x \|_0 \text{ subject to } \| y - Ax \|_2 \leq \eta$.

OMP Algorithm

Inputs: measurement vector y (mX1), measurement matrix A (mXn) and error tolerance η.

Output: estimated sparse signal x.

Initialize: residual $r_0=y$, the index set $\Lambda_0=\varnothing$, the matrix of chosen atoms $\Phi_0=\varnothing$, and the iteration counter t = 1.

1. At the iteration = t, find $\lambda_t = \arg\max_{j=1\ldots n} |< r_{t-1}, \varphi_j >|$

2. Augment the index set $\Lambda_t = \Lambda_{t-1} \cup \lambda_t$ and the matrix of chosen atoms $\Phi_t = \left[ \Phi_{t-1} A_{\lambda_t} \right]$.

3. Get the new signal estimate $\min_x \| x_t - \Phi_t y \|_2^2$.

4. Calculate the new approximation and the residual $a_t = \Phi_t x_t$ and $r_t = y - a_t$.

Increment t and return to step 1 if $\| r_t \| \geq \varepsilon$.

The problem is to estimate the sparse signal. Initially the residual is initialized to the measurement vector. The index set and the matrix of chosen atoms (columns from the measurement matrix) are empty. The first step of each iteration is to select a non-zero index of the sparse signal. In OMP, the current residual is correlated with the measurement matrix and the index of the highest correlation is selected. In the second step of the iteration, the selected index is added to the set of current index and the set of selected atoms (columns from the measurement matrix) is also updated from the current index set. In the third step the estimates of the signal at the given indices are obtained via least squares. In step 4, the residual is updated. Once all the steps are performed for the iteration, a check is done to see if the norm of the residual falls below the error estimate. If it does, the algorithm terminates otherwise it repeats steps 2 to 4.

The Fast Sparse Classification algorithm differs from the Sparse Classification algorithm only in step 1. Instead of solving the *l1 minimization* problem, FSC uses OMP for a greedy approximation of the original *l0 minimization* problem.

### 2.3 Group Sparse Classifier

As mentioned in subsection 2.1, the optimization algorithm formulated in [2] does not exactly address the desired aim. A sparse optimization problem was formulated in the hope of selecting training samples of a particular (correct) class. It has been shown in [7] that *l1 minimization* cannot select a sparse group of correlated samples (in the limiting case it selects only a single sample from all the correlated samples). In classification problems, the training samples from each class are highly correlated, therefore *l1 minimization* is not an ideal choice for ensuring selection of all the training samples from a group. To overcome this problem of [2] the Group Sparse Classifier was proposed in [3]. It has the same basic assumption as [2] but the optimization criterion is formulated so that it promotes selection of the entire class of training samples.

The basic assumption of expressing the test sample as a linear combination of training samples is formulated in (3) as $v_{k,test} = V\alpha + \varepsilon$

where $V = [v_{1,1} | ... | v_{1,n_1} | ... | v_{k,1} | ... | v_{k,n_k} | ...v_{C,1} | ... | v_{C,n_C}]$ and

$$\alpha = [\underbrace{\alpha_{1,1},...,\alpha_{1,n_1}}_{\alpha_1},\underbrace{\alpha_{2,1},...,\alpha_{2,n_2}}_{\alpha_2},...\underbrace{\alpha_{C,1},...,\alpha_{C,n_C}}_{\alpha_C}]^T .$$

The above formulation demand that α should be 'group sparse' - meaning that the solution of the inverse problem (3) should have non-zero coefficients corresponding to a particular group of training samples and zero elsewhere (i.e. $\alpha_i \neq 0$ for only one of the αi's, i=1,…,C). This requires the solution of

$$\min_{\alpha} \| \alpha \|_{2,0} \text{ such that } \|v_{test} - V\alpha\|_2 < \varepsilon \quad (6)$$

The mixed norm $\|\bullet\|_{2,0}$ is defined for $\alpha = [\underbrace{\alpha_{1,1},...,\alpha_{1,n_1}}_{\alpha_1},\underbrace{\alpha_{2,1},...,\alpha_{2,n_2}}_{\alpha_2},...\underbrace{\alpha_{k,1},...,\alpha_{k,n_k}}_{\alpha_k}]^T$ as

$\| \alpha \|_{2,0} = \sum_{l=1}^{k} I(\| \alpha_l \|_2 > 0)$, where $I(\| \alpha_l \|_2 > 0) = 1$ if $\| \alpha_l \|_2 > 0$.

Solving the *l2,0 minimization* problem is NP hard. We proposed a convex relaxation in [3], so that the optimization takes the form

$$\min_{\alpha} \| \alpha \|_{2,1} \text{ such that } \|v_{test} - V\alpha\|_2 < \varepsilon \quad (7)$$

where $\| \alpha \|_{2,1} = \| \alpha_1 \|_2 + \| \alpha_2 \|_2 + ... + \| \alpha_k \|_2$.

Solving the *l2,1 minimization* problem is the core behind the GSC. Once the optimization problem (7) is solved, the classification algorithm is straight forward.

Group Sparse Classification Algorithm

1. Solve the optimization problem expressed in (13).
2. Find those i's for which $||\alpha i||2 > 0$.
3. For those classes (i) satisfying the condition in step 2, repeat the following two steps:
   a. Reconstruct a sample for each class by a linear combination of the training samples
   $$v_{recon}(i) = \sum_{j=1}^{n_i} \alpha_{i,j} v_{i,j}$$
   in that class via the equation .
   b. Find the error between the reconstructed sample and the given test sample by $error(v_{test}, i) = || v_{k,test} - v_{recon(i)} ||_2$ .
4. Once the error for every class is obtained, choose the class having the minimum error as the class of the given test sample.

As said earlier the work horse behind the GSC is the optimization problem (7). We propose a solution to this problem via an IRLS method.

IRLS algorithm for *l2,1 minimization*

Initialization – set $\delta(0) = 0$ and find the initial $\hat{x}(0) = \min || y - Ax ||_2^2$ by conjugate gradient method.

At iteration t – continue the following steps till convergence (i.e. either $\delta$ is less than $10^{-6}$ or the number of iterations has reached maximum limit)

1. Find the weights for each group (i) $w_i = (|| x_i^{(k-1)} ||_2^2 + \delta(t))^{-1/2}$ .
2. Form a diagonal weight matrix $W_m$ having weights $w_i$ corresponding to each coefficient of the group $x_i$.
3. Form a new matrix, $L = AW_m$ .
4. Solve $\hat{u}(t) = \min || y - Lu ||_2^2$ .
5. Find x by rescaling u, $x(t) = W_m u(t)$ .
6. Reduce $\delta$ by a factor of 10 if $||y-Ax||q$ has reduced.

This algorithm is similar to the one in section 2.1 used for solving the sparse optimization problem except that the weight matrix is different.

### 2.4 Fast Group Sparse Classification

The Group Sparse Classifier [3] gives better results than the Sparse Classifier [2] but is slower. In a very recent work [8] we proposed alternate greedy algorithms for group sparse classification and were able to increase the operating speed by two orders of magnitude. These classifiers were named Fast Group Sparse Classifiers (FGSC).

FSC is built upon greedy approximation algorithms of the NP hard sparse optimization problem (10). Such greedy algorithms form a well studied topic in signal processing. Therefore it was straightforward to apply known greedy algorithms (such as OMP) to the sparse classification problem. Group sparsity promoting optimization however is not a vastly researched topic like sparse optimization. As previous work in group sparsity solely

rely on convex optimization. We had to develop a number of greedy algorithms as (fast and accurate) alternatives to convex group sparse optimization [8].

All greedy group sparse algorithms approximate the problem $\min \| x \|_{2,0}$ subject to $\| y - Ax \|_2 \leq \eta$. They work in a very intuitive way – first they try to identify the group which has non-zero coefficients. Once the group is identified, the coefficients for the group indices are estimated by some simple means. There are several ways to approximate the NP hard problem. It is not possible to discuss all of them in this chapter. We discuss the Group Orthogonal Matching Pursuit (GOMP) algorithm. The interested reader can peruse [8] for other methods to solve this problem.

GOMP Algorithm

Inputs: the measurement vector y (mX1), the measurement matrix A (mXn), the group labels and the error tolerance η.
Output: the estimated sparse signal x.

Initialize: the residual $r_0 = y$, the index set $\Lambda_0 = \varnothing$, the matrix of chosen atoms $\Phi_0 = \varnothing$, and the iteration counter t = 1.

1. At iteration t, compute $\lambda(j) = |< r_{t-1}, \varphi_j >|, \forall j = 1 \ldots n$

2. Group selection – select the class with the maximum average correlation $\tau_t = \arg\max_{i=1 \ldots C} (\frac{1}{n_i} \sum_{j=1}^{n_i} \lambda(j))$, denote it by $class(\tau_t)$.

3. Augment the index set $\Lambda_t = \Lambda_{t-1} \cup class(\tau_t)$ and the matrix of the chosen atoms $\Phi_t = [\Phi_{t-1} \; A_{class(\tau_t)}]$.

4. Get the new signal estimate using $\min_x \| x_t - \Phi_t y \|_2^2$.

5. Calculate the new approximation and the residual $a_t = \Phi_t x_t$ and $r_t = y - a_t$.

Increment t and return to step 1 if $\| r_t \| \geq \varepsilon$.

The classification method for the GSC and the FGSC are the same. Only the convex optimization of step of the former is replaced by a greedy algorithm in the latter.

### 2.5 Nearest Subspace Classifier

The Nearest Subspace Classifier (NSC) [9] makes a novel classification assumption – samples from each class lie on a hyper-plane specific to that class. According to this assumption, the training samples of a particular class span a subspace. Thus the problem of classification is to find the correct hyperplane for the test sample. According to this assumption, any new test sample belonging to that class can thus be represented as a linear combination of the test samples, i.e.

$$v_{k,test} = \sum_{i=1}^{n_k} \alpha_{k,i} \cdot v_{k,i} + \varepsilon_k \tag{8}$$

where $v_{k,test}$ is the test sample (i.e. the vector of features) assumed to belong to the k$^{th}$ class, $v_{k,i}$ is the i$^{th}$ training sample of the k$^{th}$ class, and $\varepsilon_k$ is the approximation error for the k$^{th}$ class.

Owing to the error term in equation (8), the relation holds for all the classes $k=1\ldots C$. In such a situation, it is reasonable to assume that for the correct class the test sample has the minimum error $\varepsilon_k$.

To find the class that has the minimum error in equation (8), the coefficients $\alpha_{k,i}$ $k=1\ldots C$ must be estimated first. This can be performed by rewriting (8) in matrix-vector notation

$$v_{k,test} = V_k \alpha_k + \varepsilon_k \tag{9}$$

where $V_k = [v_{k,1} \,|\, v_{k,2} \,|\, \ldots \,|\, v_{k,n_k}]$ and $\alpha_k = [\alpha_{k,1}, \alpha_{k,2} \ldots \alpha_{k,n_k}]^T$.

The solution to (9) can be obtained by minimizing

$$\hat{\alpha}_k = \arg\min_{\alpha} \| v_{k,test} - V_k \alpha \|_2^2 \tag{10}$$

The previous work on NSC [9] directly solves (10). However, the matrix $V_k$ may be under-determined, i.e. the number the number of samples may be greater than the dimensionality of the inputs. In such a case, instead of solving (10), Tikhonov regularization is employed so that the following is minimized

$$\hat{\alpha}_k = \arg\min_{\alpha} \| v_{k,test} - V_k \alpha \|_2^2 + \lambda \| \alpha \|_2^2 \tag{11}$$

The analytical solution of (11) is

$$\hat{\alpha}_k = (V_k^T V_k + \lambda I)^{-1} V_k^T v_{k,test} \tag{12}$$

Plugging this expression in (9), and solving for the error term, we get

$$\varepsilon_k = (V_k (V_k^T V_k + \lambda I)^{-1} V_k^T - I) v_{k,test} \tag{13}$$

Based on equations (9-13) the Nearest Subspace Classifier algorithm has the following steps.

NSC Algorithm

Training

1. For each class 'k', by computing the orthoprojector (the term in brackets in equation (13)).

Testing

2. Calculate the error for each class 'k' by computing the matrix vector product between the orthoprojector and $v_{k,test}$.

3. Classify the test sample as the class having the minimum error ($\| \varepsilon_k \|$).

## 3. CLASSIFICATION ROBUSTNESS TO DATA ACQUIRED BY CS

The idea of using random projection for dimensionality reduction of face images was proposed in [1, 2]. It was experimentally shown that the Nearest Neighbor (NN) and the Sparse Classifier (SC) are robust to such dimensionality reduction. However the theoretical understanding behind the robustness to such dimensionality reduction was lacking there in. In this section, we will prove why all classifiers discussed in the previous section can be categorized as Compressive Classifiers. The two conditions that guarantee the robustness of CC under random projection are the following:

Restricted Isometric Property (RIP) [10] – The l2-norm of a sparse vector is approximately preserved under a random lower dimensional projection, i.e. when a sparse vector x is projected by a random projection matrix A, then $(1-\delta)\,\|x\|_2 \leq \|Ax\|_2 \leq (1+\delta)\,\|x\|_2$. The constant δ is a RIP constant whose value depends on the type of the matrix A and the number of rows and columns of A and the nature of x. An approximate form (without upper and lower bounds) of RIP states $\|Ax\|_2 \approx \|x\|_2$.

Generalized Restricted Isometric Property (GRIP) [11] – For a matrix A which satisfies RIP for inputs $x_i$, the inner product of two vectors ($< w, v >= \|w\|_2 \cdot \|v\|_2 \cos\theta$) is approximately maintained under the random projection A, i.e. for two vectors x1 and x2 (which satisfies RIP with matrix A), the following inequality is satisfied:

$$(1-\delta)\,\|x_1\|_2 \cdot \|x_2\|_2 \cos[(1+\sqrt{3}\delta_m)\theta] \leq \langle Ax_1, Ax_2 \rangle \leq (1+\delta)\,\|x_1\|_2 \cdot \|x_2\|_2 \cos[(1-\sqrt{3}\delta_m)\theta]$$

The constants δ and δm depend on the dimensionality and the type of matrix A and also on the nature of the vectors. Even though the expression seems overwhelming, it can be simply stated as: the angle between two sparse vectors (θ) is approximately preserved under random projections. An approximate form of GRIP is $\langle Ax_1, Ax_2 \rangle \approx \langle x_1, x_2 \rangle$.

RIP and the GRIP were originally proven for sparse vectors, but natural images are in general dense. We will show why these two properties are satisfied by natural images as well. Images are sparse in several orthogonal transform domains like DCT and wavelets. If I is the image and x is the transform domain representation, then

$I = \Phi^T x$ synthesis equation

$x = \Phi I$ analysis equation

where Φ is the sparsifying transform and x is sparse.

Now if the sparse vector x is randomly projected by a Gaussian matrix A following RIP, then

$\|Ax\|_2 \approx \|x\|_2$

$\Rightarrow \|A\Phi I\|_2 \approx \|\Phi I\|_2$ (by anaslysis equation)

$\Rightarrow \|A\Phi I\|_2 \approx \|I\|_2$ ($\because$ Φ is orthogonal)

$\Rightarrow \|BI\|_2 \approx \|I\|_2$, B = AΦ

Since Φ is an orthogonal matrix, the matrix AΦ (=B) is also Gaussian, being formed by a linear combination of i.i.d. Gaussian columns. Thus it is seen how the RIP condition holds

for dense natural images. This fact is the main cornerstone of all compressed sensing imaging applications. In a similar manner it can be also shown that the GRIP is satisfied by natural images as well.

### 3.1 The Nearest Neighbor Classifier

The Nearest Neighbor (NN) is a compressive classifier. It was used for classification under RP dimensionality reduction in [1]. The criterion for NN classification depends on the magnitude of the distance between the test sample and each training sample. There are two popular distance measures –

Euclidean distance ($\| v_{test} - v_{i,j} \|_2, i = 1...C \text{ and } j = 1...n_i$)

Cosine distance ($\left\langle v_{test}, v_{i,j} \right\rangle, i = 1...C \text{ and } j = 1...n_i$)

It is easy to show that both these distance measures are approximately preserved under random dimensionality reduction, assuming that the random dimensionality reduction matrix A follows RIP with the samples v. Then following the RIP approximation, the Euclidean distance between samples is approximately preserved, i.e.

$$\| Av_{test} - Av_{i,j} \|_2 = \| A(v_{test} - v_{i,j}) \|_2 \approx \| (v_{test} - v_{i,j}) \|_2$$

The fact that the Cosine distance is approximately preserved follows directly from the GRIP assumption

$$\left\langle Av_{test}, Av_{i,j} \right\rangle \approx \left\langle v_{test}, v_{i,j} \right\rangle .$$

### 3.2 The Sparse and the Group Sparse Classifier

In this subsection it will be shown why the Sparse Classifier and the Group Sparse Classifier can act as compressive classifiers. At the core of SC and GSC classifiers are the l1 minimization and the l2,1 minimization optimization problems respectively

$$\begin{aligned} &\text{SC-min} \| \alpha \|_1 \ \text{ subject to } \ \| v_{k,test} - V\alpha \|_2 \le \eta \\ &\text{GSC-min} \| \alpha \|_{2,1} \ \text{ subject to } \ \| v_{k,test} - V\alpha \|_2 \le \eta \end{aligned} \tag{14}$$

In compressive classification, all the samples are projected from a higher to a lower dimension by a random matrix A. Therefore the optimization is the following:

$$\begin{aligned} &\text{SC-min} \| \beta \|_1 \ \text{ subject to } \ \| Av_{k,test} - AV\beta \|_2 \le \eta \\ &\text{GSC-min} \| \beta \|_{2,1} \ \text{ subject to } \ \| Av_{k,test} - AV\beta \|_2 \le \eta \end{aligned} \tag{15}$$

The objective function does not change before and after projection, but the constraints do. We will show that the constraints of (14) and (15) are approximately the same; therefore the optimization problems are the same as well. The constraint in (15) can be represented as:

$$\begin{aligned} &\| Av_{k,test} - AV\beta \|_2 \le \eta \\ &= \| A(v_{k,test} - V\beta) \|_2 \le \eta \\ &\approx \| (v_{k,test} - V\beta) \|_2 \le \eta, \text{ following RIP} \end{aligned}$$

Since the constraints are approximately preserved and the objective function remains the same, the solution to the two optimization problems (14) and (15) will be approximately the same, i.e. $\beta \approx \alpha$ .

In the classification algorithm for SC and GSC (this is also true for both the FSC, FGSC and NSC), the deciding factor behind the class of the test sample is the class-wise error

$$error(v_{test}, i) = \| v_{k,test} - \sum_{j=1}^{n_i} \alpha_{i,j} v_{i,j} \|_2, i = 1...C$$

.

We show why the class-wise error is approximately preserved after random projection.

$$\begin{aligned} error(Av_{test}, i) &= \| Av_{k,test} - A\sum_{j=1}^{n_i} \alpha_{i,j} v_{i,j} \|_2 \\ &= \| A(v_{k,test} - \sum_{j=1}^{n_i} \alpha_{i,j} v_{i,j}) \|_2 \\ &\approx \| (v_{k,test} - \sum_{j=1}^{n_i} \alpha_{i,j} v_{i,j}) \|_2, \text{ due to RIP} \end{aligned}$$

As the class-wise error is approximately preserved under random projections, the recognition results too will be approximately the same.

Fast Sparse and Fast Group Sparse Classifiers

In the FSC and the FGSC classifiers, the NP hard optimization problem (14) is solved greedily.

$$\begin{aligned} &\text{SC-min} \| \alpha \|_0 \text{ subject to } \| v_{k,test} - V\alpha \|_2 \le \eta \\ &\text{GSC-min} \| \alpha \|_{2,0} \text{ subject to } \| v_{k,test} - V\alpha \|_2 \le \eta \end{aligned}$$

The problem (14) pertains to the case of original data. When the samples are randomly projected, the problem has the following form:

$$\begin{aligned} &\text{SC-min} \| \beta \|_0 \text{ subject to } \| Av_{k,test} - AV\beta \|_2 \le \eta \\ &\text{GSC-min} \| \beta \|_{2,0} \text{ subject to } \| Av_{k,test} - AV\beta \|_2 \le \eta \end{aligned} \quad (16)$$

We need to show that the results of greedy approximation to the above problems yields $\beta \approx \alpha$ .

There are two main computational steps in the OMP/GOMP algorithms – i) the selection step, i.e the criterion for choosing the indices, and ii) the least squares signal estimation step. In order to prove the robustness of the OMP/GOMP algorithm to random projection, it is sufficient to show that the results from the aforesaid steps are approximately preserved.

In OMP/GOMP, the selection is based on the correlation between the measurement matrix Φ and the observations y, i.e. $\Phi^T y$ . If we have $\Phi_{m \times n}$ and $y_{m \times 1}$ , then the correlation can be written as inner products between the columns of Φ and the vector y i.e. $\langle \phi_i, y \rangle, i = 1...n$ . After random projection, both columns of Φ and the measurement y are randomly sub-

sampled by a random projection matrix A. The correlation can be calculated as $\langle A\phi_i, Ay \rangle, i = 1...n$, which by GRIP can be approximated as $\langle \phi_i, y \rangle, i = 1...n$ .n Since the correlations are approximately preserved before and after the random projection, the OMP/GOMP selection is also robust under such random sub-sampling.
The signal estimation step is also robust to random projection. The least squares estimation is performed as:

$$\min \| y - \Phi x \|_2 \tag{17}$$

The problem is to estimate the signal x, from measurements y given the matrix Φ.
Both y and Φ are randomly sub-sampled by a random projection matrix A which satisfies RIP. Therefore, the least squares problem in the sub-sampled case takes the form

$$\begin{aligned} &\min \| Ay - A\Phi x \|_2 \\ &= \min \| A(y - \Phi x) \|_2 \\ &\approx \min \| y - \Phi x \|_2, \text{ since RIP holds} \end{aligned}$$

Thus the signal estimate x, obtained by solving the original least squares problem (22) and the randomly sub-sampled problem are approximately the same.
The main criterion of the FSC and the FGSC classification algorithms is the class-wise error. It has already been shown that the class-wise error is approximately preserved after random projection. Therefore the classification results before and after projection will remain approximately the same.
3.3 Nearest Subspace Classifier
The classification criterion for the NSC is the norm of the class-wise error expressed as

$$\| \varepsilon_k \|_2 = \| (V_k (V_k^T V_k + \lambda I)^{-1} V_k^T - I) v_{k,test} \|_2$$

We need to show that the class-wise error is approximately preserved after a random dimensionality reduction. When both the training and the test samples are randomly projected by a matrix A, the class-wise error takes the form

$$\begin{aligned} &\| (AV_k ((AV_k)^T (AV_k) + \lambda I)^{-1} (AV_k)^T - I) A v_{k,test} \|_2 \\ &= \| AV_k ((AV_k)^T (AV_k) + \lambda I)^{-1} (AV_k)^T A v_{k,test} - A v_{k,test} \|_2 \\ &\approx \| AV_k (V_k{}^T V_k + \lambda I)^{-1} V_k{}^T v_{k,test} - A v_{k,test} \|_2, \text{ since GRIP holds} \\ &= \| A(V_k (V_k{}^T V_k + \lambda I)^{-1} V_k{}^T v_{k,test} - v_{k,test}) \|_2 \\ &\approx \| V_k (V_k{}^T V_k + \lambda I)^{-1} V_k{}^T v_{k,test} - v_{k,test} \|_2, \text{ since RIP holds} \end{aligned}$$

Since the norm of the class-wise error is approximately preserved under random dimensionality reduction, the classification results will also remain approximately the same.

## 4. EXPERIMENTAL RESULTS

As mentioned in section 2, compressive classifiers should meet two challenges. First and foremost it should have classification accuracy comparable to traditional classifiers. Experiments for general purpose classification are carried out on some benchmark databases from the University of California Irvine Machine Learning (UCI ML) repository [12] to compare the new classifiers (SC, FSC, GSC, FGSC and NSC) with the well known NN. We chose those databases that do not have missing values in feature vectors or unlabeled training data. The results are tabulated in Table 1. The results show that the classification accuracy from the new classifiers are better than NN.

| Dataset | SC | FSC | GSC | FGSC | NSC | NN-Euclid | NN-Cosine |
|---|---|---|---|---|---|---|---|
| Page Block | 94.78 | 94.64 | 95.66 | 95.66 | 95.01 | 93.34 | 93.27 |
| Abalone | 27.17 | 27.29 | 27.17 | 26.98 | 27.05 | 26.67 | 25.99 |
| Segmentation | 96.31 | 96.10 | 94.09 | 94.09 | 94.85 | 96.31 | 95.58 |
| Yeast | 57.75 | 57.54 | 58.94 | 58.36 | 59.57 | 57.71 | 57.54 |
| German Credit | 69.30 | 70.00 | 74.50 | 74.50 | 72.6 | 74.50 | 74.50 |
| Tic-Tac-Toe | 78.89 | 78.28 | 84.41 | 84.41 | 81.00 | 83.28 | 82.98 |
| Vehicle | 65.58 | 66.49 | 73.86 | 71.98 | 74.84 | 73.86 | 71.98 |
| Australian Cr. | 85.94 | 85.94 | 86.66 | 86.66 | 86.66 | 86.66 | 86.66 |
| Balance Scale | 93.33 | 93.33 | 95.08 | 95.08 | 95.08 | 93.33 | 93.33 |
| Ionosphere | 86.94 | 86.94 | 90.32 | 90.32 | 90.32 | 90.32 | 90.32 |
| Liver | 66.68 | 65.79 | 70.21 | 70.21 | 70.21 | 69.04 | 69.04 |
| Ecoli | 81.53 | 81.53 | 82.88 | 82.88 | 82.88 | 80.98 | 81.54 |
| Glass | 68.43 | 69.62 | 70.19 | 71.02 | 69.62 | 68.43 | 69.62 |
| Wine | 85.62 | 85.62 | 85.62 | 85.95 | 82.58 | 82.21 | 82.21 |
| Iris | 96.00 | 96.00 | 96.00 | 96.00 | 96.00 | 96.00 | 96.00 |
| Lymphography | 85.81 | 85.81 | 86.42 | 86.42 | 86.42 | 85.32 | 85.81 |
| Hayes Roth | 40.23 | 43.12 | 41.01 | 43.12 | 43.12 | 33.33 | 33.33 |
| Satellite | 80.30 | 80.30 | 82.37 | 82.37 | 80.30 | 77.00 | 77.08 |
| Haberman | 40.52 | 40.85 | 43.28 | 43.28 | 46.07 | 57.40 | 56.20 |

Table 1. Recognition Accuracy (%age)

The second challenge the Compressive Classifiers should meet is that their classification accuracy should approximately be the same, when sparsifiable data is randomly sub-sampled by RIP matrices. In section 3 we have already proved the robustness of these classifiers. The experimental verification of this claim is shown in table 2. It has already been mentioned (section 3) that images follow RIP with random matrices having i.i.d Gaussian columns normalized to unity.

The face recognition experiments were carried out on the Yale B face database. The images are stored as 192X168 pixel grayscale images. We followed the same methodology as in [2]. Only the frontal faces were chosen for recognition. Half of the images (for each individual) were selected for training and the other half for testing. The experiments were repeated 5 times with 5 sets of random splits. The average results of 5 sets of experiments are shown in

table 2. The first column of the following table indicates the number of lower dimensional projections (1/32, 1/24, 1/16 and 1/8 of original dimension).

| Dimensionality | SC | FSC | GSC | FGSC | NSC | NN-Euclid | NN-Cosine |
|---|---|---|---|---|---|---|---|
| 30 | 82.73 | 82.08 | 85.57 | 83.18 | 87.68 | 70.39 | 70.16 |
| 56 | 92.60 | 92.34 | 92.60 | 91.83 | 91.83 | 75.45 | 75.09 |
| 120 | 95.29 | 95.04 | 95.68 | 95.06 | 93.74 | 78.62 | 78.37 |
| 504 | 98.09 | 97.57 | 98.09 | 97.21 | 94.42 | 79.13 | 78.51 |
| Full | 98.09 | 98.09 | 98.09 | 98.09 | 95.05 | 82.08 | 82.08 |

Table 2. Recognition Results (%) on Yale B (RP)

Table 2 shows that the new compressive classifiers are way better than the NN classifiers in terms of recognition accuracy. The Group Sparse Classifier gives by far the best results. All the classifiers are relatively robust to random sub-sampling. The results are at par with the ones obtained from the previous study on Sparse Classification [2].
The compressive classifiers have the special advantage of being robust to dimensionality reduction via random projection. However, they can be used for any other dimensionality reduction as well. In Table 3, the results of compressive classification on PCA dimensionality reduced data is shown for the Yale B database.

| Dimensionality | SC | FSC | GSC | FGSC | NSC | NN-Euclid | NN-Cosine |
|---|---|---|---|---|---|---|---|
| 30 | 83.10 | 82.87 | 86.61 | 84.10 | 88.92 | 72.50 | 71.79 |
| 56 | 92.83 | 92.55 | 93.40 | 92.57 | 92.74 | 78.82 | 77.40 |
| 120 | 95.92 | 95.60 | 96.15 | 95.81 | 94.98 | 84.67 | 82.35 |
| 504 | 98.09 | 97.33 | 98.09 | 98.09 | 95.66 | 88.95 | 86.08 |
| Full | 98.09 | 98.09 | 98.09 | 98.09 | 96.28 | 89.50 | 88.00 |

Table 3. Recognition Results (%) on Yale B (PCA)

Experimental results corroborate our claim regarding the efficacy of compressive classifiers. Results for Table 1 indicate that they can be used for general purpose classification. Table 2 successfully verifies the main claim of this chapter, i.e. the compressive classifiers are robust to dimensionality reduction via random projection. In Table 3, we show that the compressive classifiers are also applicable to data whose dimensionality has been reduced by standard techniques like PCA.

## 5. CONCLUSION

This chapter reviews an alternate face recognition method than those provided by traditional machine learning tools. Conventional machine learning solutions to dimensionality reduction and classification require all the data to be present beforehand, i.e. whenever new data is added, the system cannot be updated in online fashion, rather all the calculations need to be re-done from scratch. This creates a computational bottleneck for large scale implementation of face recognition systems.

The face recognition community has started to appreciate this problem in the recent past and there have been some studies that modified the existing dimensionality reduction methods for online training [13, 14]. The classifier employed along with such online dimensionality reduction methods has been the traditional Nearest Neighbour.
This work addresses the aforesaid problem from a completely different perspective. It is based on recent theoretical breakthroughs in signal processing [15, 16]. It advocates applying random projection for dimensionality reduction. Such dimensionality reduction necessitates new classification algorithms. This chapter assimilates some recent studies in classification within the unifying framework of compressive classification. The Sparse Classifier [2] is the first of these. The latter ones like the Group Sparse Classifier [3], Fast Group Sparse Classifier [8] and Nearest Subspace Classifier [9] were proposed by us. The Fast Sparse Classifier has been proposed for the first time in this chapter.
For each of the classifiers, their classification algorithms have been written concisely in the corresponding sub-sections. Solutions to different optimization problems required by the classifiers are presented in a fashion that can be implemented by non-experts. Moreover the theoretical understanding behind the different classifiers is also provided in this chapter. These theoretical proofs are thoroughly validated by experimental results.
It should be remembered that the classifiers discussed in this chapter can be used with other dimensionality reduction techniques as well such as – Principal Component Analysis, Linear Discriminant Analysis and the likes. In principle the compressive classifiers can be employed in any classification task as better substitutes for the Nearest Neighbour classifier.

## 6. REFERENCES

Goel, N., Bebis, G.m and Nefian, A. V., 2005. Face recognition experiments with random projections. SPIE Conference on Biometric Technology for Human Identification, 426-437.

Y. Yang, J. Wright, Y. Ma and S. S. Sastry, “Feature Selection in Face Recognition: A Sparse Representation Perspective”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 1 (2), pp. 210-227, 2009.

A. Majumdar and R. K. Ward, “Classification via Group Sparsity Promoting Regularization”, IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 873-876, 2009.

R. Chartrand, and W. Yin, "Iteratively reweighted algorithms for compressive sensing," ICASSP 2008. pp. 3869-3872, 2008.

B. D. Rao and K. Kreutz-Delgado, “An affine scaling methodology for best basis selection”, IEEE Transactions on Signal Processing, Vol. 47 (1), pp. 187-200, 1999.

Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. “Orthogonal Matching Pursuit: Recursive function approximation with applications to wavelet decomposition”, Asilomar Conf. Sig., Sys., and Comp., Nov. 1993.

H. Zou and T. Hastie, “Regularization and variable selection via the elastic net”, Journal of Royal Statistical Society B., Vol. 67 (2), pp. 301-320.

A. Majumdar and R. K. Ward, “Fast Group Sparse Classification”, IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, B.C., Canada, August 2009.

A. Majumdar and R. K. Ward, “Nearest Subspace Classifier” submitted to International Conference on Image Processing (ICIP09).

E. J. Cand`es and T. Tao, “Near optimal signal recovery from random projections: Universal encoding strategies?”, IEEE Trans. Info. Theory, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

Haupt, J.; Nowak, R., "Compressive Sampling for Signal Detection," Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on, vol. 3, no., pp. III-1509-III-1512, 15-20 April 2007.

http://archive.ics.uci.edu/ml/

T. J. Chin and D. Suter, “Incremental Kernel Principal Component Analysis”, IEEE Transactions on Image Processing, Vol. 16, (6), pp. 1662-1674, 2007.

H. Zhao and P. C. Yuen, “Incremental Linear Discriminant Analysis for Face Recognition”, IEEE Trans. on Systems, Man, And Cybernetics—Part B: Cybernetics, Vol. 38 (1), pp. 210-221, 2008.

D. L. Donoho, “Compressed sensing,” IEEE Transactions on Information Theory, Vol. 52 (4), pp. 1289–1306, 2006.

E. J. Cand`es and T. Tao, “Near optimal signal recovery from random projections: Universal encoding strategies?”, IEEE Transactions on Information Theory, Vol. 52 (12), pp. 5406–5425, 2006.