Generalized Non-Linear Sparse Classifier

Angshul Majumdar and Rabab K. Ward

Department of Electrical and Computer Engineering, University of British Columbia

{angshulm, rababw}@ece.ubc.ca

Abstract

In a recent study a novel classification algorithm was proposed which assumed that a test sample can be approximately represented by a linear combination of the training samples belonging to that class. This assumption gives rise to a sparse optimization problem with perturbed linear constraints. Consequently the classification algorithm was called the Sparse Classifier (SC). Although the SC assumption is restrictive, good face recognition results were obtained by this method. This paper proposes major generalizations in the assumptions of the aforesaid work. The first generalization assumes that the test sample raised to some powers can be approximated by a linear combination of the training samples of that class raised to the same powers. This results in a group-sparse optimization problem with perturbed linear constraints. The second generalization is more general, it assumes that the test samples raised to certain powers can be approximately represented by a non-linear combination of the training samples raised to the same powers. This generalization leads to a group-sparse optimization problem with perturbed non-linear constraints. In principle the optimization problems can be solved by standard optimization techniques, but will be prohibitively slow. To overcome the limitations in speed we propose novel greedy algorithms to approximately solve the optimization problems with considerably lesser computational complexity. As a case study, we apply our classification method to the problem of single-image-per-person face recognition.

Index Terms

Non-linear optimization, Greedy Algorithms, Face Recognition.

1. INTRODUCTION

A recent work in face recognition [1] proposed a simple yet novel assumption: test samples of a particular class can be approximately expressed as a linear combination of the training samples belonging to that class. Their assumption led to the solution of a sparse optimization problem with perturbed linear constraints. With this simple assumption, very good recognition results were obtained on the Extended Yale and the AR face recognition databases [1]. This classification method is called the Sparse Classifier (SC). The SC is similar to the Nearest Neighbour in operation. It does not require a training phase. It just stores the training samples; all the computations are performed at run-time. Consequently this is a so called ‘Lazy Learning’ algorithm.

The classification assumption of [1] is restrictive. In this work we propose major generalization of the previous assumption in two steps. In the first step, we relax the condition that the test sample should be approximated by linear combination of training samples of that class. We assume that the test samples raised to certain powers can be approximated by a linear combination of the training samples raised to the same powers. This is a generalization of the previous assumption, where only a single power (unity) is considered. The first step of our generalization leads to a group sparse optimization problem with perturbed linear constraints. The second generalization step is more profound. It assumes that the test sample raised to certain powers can be expressed approximately as a non-linear

combination of the training samples raised to same powers. This leads to a group sparse optimization problem with perturbed non-linear constraints.

In terms of generalization of assumption, we come a long way from the original proposal in [1]. But such generalized assumptions throw new challenges for optimization. The sparse optimization problem formulated in the previous work is a well studied problem in Signal Processing and Machine Learning. The previous work was fortunate to be able to use already developed optimization tools. In our case the first generalization step leads to the formulation of a group sparse optimization problem. This is not well researched topic. The optimization tools developed for this problem, even though accurate are not fast enough to serve our purpose. As these classifiers follow the 'Lazy Learning' paradigm (all computations performed at run-time), the speed of optimization is of concern. Keeping the operational speed in mind, we propose fast greedy (sub-optimal) algorithms to approximate the group sparse optimization problem. The second generalization step leads to an optimization problem which is very non-standard. General optimization tools can be employed for this problem, but they are found to be prohibitively slow. This work proposes a new non-linear greedy algorithm for non-linear sparse optimization to overcome these speed limitations.

The classifiers proposed in this work can be used for general purpose classification. They are similar to Nearest Neighbour in operation. Nearest Neighbour (NN) is perhaps most often used for face recognition problems. In particular we are interested in the problem of face recognition when only a single training image of each person is available. More commonly it is referred to as the single-image-per-person recognition problem. A survey of this problem [2] shows that most of previous studies in this field employ the NN for classification. Studies like [3-9] differ from each other in their feature extraction method, but all use the same NN classification. Keeping the feature extraction phase same, but by changing the classification stage to SC and our generalizations we will show that significant improvements in recognition results can be obtained.

The rest of the paper will be organized into several sections. Section 2, describes the Sparse Classification method [1]. In section 3, the proposed generalizations and the optimization tools needed to implement them are discussed. A brief review of the single-image-per-person face recognition is in section 4. Section 5 tabulates the experimental results. Finally in section 6, discussions and future scope of work are discussed.

2. SPARSE CLASSIFIER

The sparse classifier is proposed in [1]. The classification algorithm is based on a novel assumption which departs from familiar traditional classification assumptions of Linear/Quadratic classifier, Support Vector Machine and Artificial Neural Networks. Since the work is, the machine learning community may not be familiar with this work. Therefore we are briefly reviewing the sparse classification method [1].

The Sparse Classifier assumes that the training samples of a particular class approximately form a linear basis for a new test sample belonging to the same class. The assumption can be expressed formally as:

$$v_{k,test} = \alpha_{k,1}v_{k,1} + \alpha_{k,2}v_{k,2} + ... + \alpha_{k,n_k}v_{k,n_k} + \varepsilon \quad (1)$$

where $v_{k,i}$ are the training samples and ε is the approximation error.

Equation (1) expresses the assumption in terms of the training samples of a single (correct) class. Alternately, it can be expressed in terms of all the training samples in the form:

$$v_{k,test} = V\alpha + \varepsilon \quad (2)$$

where $V = [v_{1,1} | ... | v_{n,1} | ... | v_{k,1} | ... | v_{k,n_k} | ... v_{C,1} | ... | v_{C,n_C}]$ and $\alpha = [\alpha_{1,1}...\alpha_{1,n_1}...\alpha_{k,1}...\alpha_{k,n_k}...\alpha_{C,1}...\alpha_{C,n_C}]'$.

According to the assumption, the solution to the inverse problem (2) should be sparse, i.e. only those coefficients in the vector α should be non-zeroes which correspond to the correct class of the test sample. The rest should all be zeroes. We will discuss later how a sparse solution to (2) is achieved. For the time being, we assume that the solution to the problem (2) has been obtained. Based on this solution, the following classification algorithm is proposed in [1]:

SC (Sparse Classifier) Algorithm

1. Find a sparse solution to inverse problem (2).
2. For each class i repeat the following two steps:
    a. Find a representative sample for each class by a linear combination of the training samples belonging to that class by the equation $v_{rep}(i) = \sum_{j=1}^{n_i} \alpha_{i,j} v_{i,j}$.
    b. Find the error between the reconstructed sample and the given test sample by $error(v_{test}, i) = \| v_{k,test} - v_{rep(i)} \|_2$.
3. Once the error for every class is obtained, choose the class having the minimum error as the class of the given test sample.

The aforesaid classification algorithm is based on a sparse solution to the inverse problem (2). Once the sparse solution is obtained the rest of the steps are straightforward to compute. Ideally, a sparse solution to the inverse problem (2) can be achieved by minimizing the l_0-norm of the coefficients:

$$\min \| \alpha \|_0 \text{ such that } \| v_{k,test} - V\alpha \| < \sigma \quad (3)$$

This is known to be an NP hard problem. Citing equivalence results from Compressed Sensing [10], the sparse classification algorithm replaces the NP hard l_0-norm minimization problem by a convex l_1-norm minimization problem. Fortunately for [1], l_1-norm minimization is a widely studied topic in current signal processing research and they had readymade solution to the optimization problem in 'l1-magic' [11] which is based on standard interior-point methods.

$$\min \| \alpha \|_1 \text{ such that } \| v_{k,test} - V\alpha \| < \sigma \quad (4)$$

Interior-point based solvers are too slow for any practical classification problem – this is because the optimization is taking place during testing. In this work we propose a fast algorithm for l_1-norm minimization based on reweighted least squares. Our method is a fast version of [12] which was proposed originally for sparse non-convex optimization.

L1-norm minimization by Reweighted Least Squares

The approach is to replace the l_1-norm problem by a weighted l_2-norm, i.e.

$$\min \sum_i w_i \alpha_i^2 \text{ subject to } \| v_{k,test} - V\alpha \|_2^2 \leq \eta \quad (5)$$

The above expression has a closed form solution of the form (Euler Lagrange solution)

$$\hat{\alpha} = QV^T (VQV^T)^{-1} v_{k,test} \quad (6)$$

where Q is a diagonal matrix consisting of elements $1/w_i$, *where* $w_i = (\hat{\alpha}_i^{\,2} + \varepsilon)^{-1/2}$, the term ε is a damping factor.

A minimum l_2-norm solution is the one where each coefficient value is small. This is exactly the opposite of what is desired: a sparse solution has very few coefficients but of high values. To achieve a sparse solution via iterated l_2-norm minimization, at each iteration the weights (w_i) are adjusted such that the coefficients α_i with low values are penalized and those with high values are favored.

Initialization – The initial solution for the coefficient is the least squares solution, i.e. $\hat{\alpha}^{(0)} = \min \| v_{k,test} - V\alpha \|_2^2$ which can be solved very fast using Conjugate Gradient method. The damping factor is initialized to ε = 1.

At Iteration t – Continue the following steps till solution is reached (i.e. till ε greater than a pre-defined value)

1. Find the weights $w_i = (\hat{\alpha}_i^{(t)2} + \varepsilon)^{-1/2}$.

At each iteration one can compute (6) to find the current solution and proceed to the next step. But computing explicit inverses of large matrices are never desirable (Sometimes it is not possible to compute the inverses

explicitly as the matrices are only obtained as operators). Therefore we formulate a solution which bypasses the necessity of computing explicit inverses. The steps are the following:

2. Form a diagonal matrix R with entries $1/\sqrt{w_i}$, so that $Q = RR$.

3. Form a new matrix $\Phi = VR$.

4. Compute the least squares solution $\hat{x} = \min \| v_{k,test} - \Phi x \|_2^2$ by Conjugate Gradient method.

5. Get the current estimate $\hat{\alpha}^{(t)} = Rx$.

6. Reduce the damping factor if the residual ($v_{k,test} - V\hat{\alpha}^{(t)}$) has reduced by a certain fraction.

Our solution to l_1-norm minimization is quite fast. In this work, we use this procedure to obtain the sparse estimate of the inverse problem (2) at step 1 of the classification algorithm.

3. GENERALIZATIONS

We mentioned earlier that this work generalizes the simple Sparse Classifier (SC) proposed in [1]. The full generalization is achieved in two steps. In the first step, it is assumed that the test sample raised to certain powers can be approximately represented by a linear combination of training samples of the correct class raised to the same power. This is discussed in sub-section 3.1. In the second step, it is assumed that the test sample raised to certain powers can be approximated by a non-linear combination of the training samples of the correct class raised to the same power. The second generalization is detailed in sub-section 3.2.

3.1 GENERALIZED LINEAR SPARSE CLASSIFIER

The simple assumption in (1) says that a test sample can be approximately expressed as a linear combination of the training samples from the correct class. This is a simplistic assumption. We argue that this approximation may hold for several powers (p_1, …p_M) such that

$$
\begin{aligned}
&{v^{p_1}}_{test} = \alpha_{p_1,k,1}{v^{p_1}}_{k,1} + \alpha_{p_1,k,2}{v^{p_1}}_{k,2} + \ldots + \alpha_{p_1,k,n_i}{v^{p_1}}_{k,n_k} + \varepsilon_{p_1} \\
&{v^{p_2}}_{test} = \alpha_{p_2,k,1}{v^{p_2}}_{k,1} + \alpha_{p_2,k,2}{v^{p_2}}_{k,2} + \ldots + \alpha_{p_2,k,n_i}{v^{p_2}}_{k,n_k} + \varepsilon_{p_2} \\
&\ldots \\
&{v^{p_M}}_{test} = \alpha_{p_M,k,1}{v^{p_M}}_{k,1} + \alpha_{p_M,k,2}{v^{p_M}}_{k,2} + \ldots + \alpha_{p_M,k,n_i}{v^{p_M}}_{k,n_k} + \varepsilon_{p_M}
\end{aligned}
\qquad (7)
$$

where v^p indicates that each coefficient of the sample v is raised to the power p.

We can write this expression (7) in terms of all the training samples of the class

$$v_{test}^{p_i} = V_{p_i}\alpha_{p_i} + \varepsilon_{p_i}, \forall i = 1:M \tag{8}$$

where V_{p_i} is a matrix formed by stacking the training samples raised to the power p_i column-wise and ε_{p_i} is the error.

Thus (8) is a generalization of (2), where the latter forms a special case of the former where only a single power (p = 1)is considered.

To design a classifier based on our generalized assumption, the first task will be to organize the system of equations (7) and (8). They can be expressed as a single system of equations of the form:

$$\begin{bmatrix} v_{k,test}^{p_1} \\ v_{k,test}^{p_2} \\ . \\ . \\ . \\ v_{k,test}^{p_M} \end{bmatrix} = \begin{bmatrix} V_{p_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & V_{p_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & . & 0 & 0 & 0 \\ 0 & 0 & 0 & . & 0 & 0 \\ 0 & 0 & 0 & 0 & . & 0 \\ 0 & 0 & 0 & 0 & 0 & V_{p_M} \end{bmatrix} \begin{bmatrix} \alpha_{p_1} \\ \alpha_{p_2} \\ . \\ . \\ . \\ \alpha_{p_M} \end{bmatrix} + \begin{bmatrix} \varepsilon_{p_1} \\ \varepsilon_{p_2} \\ . \\ . \\ . \\ \varepsilon_{p_M} \end{bmatrix} \tag{9}$$

This can be written as

$$v = V\alpha + \varepsilon \tag{10}$$

where $v_{k,test} = [v_{k,test}^{p_1}, v_{k,test}^{p_2}, ..., v_{k,test}^{p_M}]'$; $V = BlockDiag[V_{p_1}, V_{p_2}, ..., V_{p_M}]$ and $\alpha = [\alpha_{p_1}, \alpha_{p_2}, ..., \alpha_{p_M}]'$

By definition (7) the structure of the coefficient vector α demands group sparsity, i.e. the indices in each of the α_p's should be non-zeroes for the correct class of the test sample. The groups are formed by the class of indices, i.e.

$$\alpha = [\underbrace{\alpha_{p_1,1}, \alpha_{p_2,1} ... \alpha_{p_M,1}}_{\alpha_1} ... \underbrace{\alpha_{p_1,C}, \alpha_{p_2,C} ... \alpha_{p_M,C}}_{\alpha_C}]', where\ \alpha_{p_j,i} = [\alpha_{p_j,i,1} ... \alpha_{p_j,i,n_i}]'$$

With this notation, we frame a group sparsity promoting optimization problem

$$\min_{\alpha} \| \alpha \|_{2,0} \ \text{ such that } \| v_{test} - V\alpha \|_2 < \varepsilon \tag{11}$$

Where the mixed norm $\|\bullet\|_{2,0}$ is defined for α as $\| \alpha \|_{2,0} = \sum_{l=1}^{C} I(\| \alpha_l \|_2 > 0)$, where $I(\| \alpha_l \|_{2,0} > 0) = 1$ if $\| \alpha_l \|_{2,0} > 0$.

Solving the optimization problem (11) is NP hard. Group sparse optimization is not a well studied topic compared to sparse optimization. In all previous works [13-19] where the problem of group sparse optimization arose, the following convex relaxation to the above problem was employed:

$$\min_{\alpha} \| \alpha \|_{2,1} \ \text{ such that } \| v_{k,test} - V\alpha \|_2 < \varepsilon \tag{12}$$

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

Generally group-versions of the LASSO (Least Angle Shrinkage and Selection Operator) or the LARS (Least Angle Regression and Selection) are employed to solve (12) [16]. In this paper we will provide two alternative methods to solve the group-sparse optimization problem. The first one will be based on optimization of (12), which will employ the Reweighted Least Squares approach used in section 2. The second one will be a fast greedy algorithm for directly approximating the (11) called Stagewise Block Orthogonal Matching Pursuit (StBOMP).

L2,1-norm minimization by Reweighted Least Squares

The approach is to replace the $l_{2,1}$-norm problem by a weighted l_2-norm, i.e.

$$\min \sum_i w_i \sum_j \alpha_j^2 \text{ subject to } \| v_{k,test} - V\alpha \|_2^2 \le \eta \quad (13)$$

where i denotes the groups and j denotes the elements in groups

The weights (w_i) is the same for one group.

Initialization – The initial solution for the coefficient is the least squares solution, i.e. $\hat{\alpha}^{(0)} = \min \| v_{k,test} - V\alpha \|_2^2$ which can be solved very fast using Conjugate Gradient method. The damping factor is initialized to $\varepsilon = 1$.

At Iteration t – Continue the following steps till solution is reached (i.e. till ε greater than a pre-defined value)

1. Find the weights for each group $w_i = (\sum_j \hat{\alpha}_{i,j}^{(t)2} + \varepsilon)^{-1/2}$.

In the problem of group sparsity the each group has the same weight, therefore $w_{i,j} = w_i, \forall j$ in the group i

The rest of the steps arethe same as in the algorithm for l_1-norm minimization.

2. Form a diagonal matrix R with entries $1/\sqrt{w_{i,j}}$, so that $Q = RR$.

3. Form a new matrix $\Phi = VR$.

4. Compute the least squares solution $\hat{x} = \min \| v_{k,test} - \Phi x \|_2^2$ by Conjugate Gradient method.

5. Get the current estimate $\hat{\alpha}^{(t)} = Rx$.

6. Reduce the damping factor if the residual ($v_{k,test} - V\hat{\alpha}^{(t)}$) if the residual has reduced by a certain factor.

Our solution to $l_{2,1}$-norm minimization are faster than traditional group-sparse solutions based on LASSO, LARS or Non-negative Garrote.

We mentioned earlier that, all these classifiers follow the 'lazy learning' paradigm, i.e. all the computations are performed at run-time (testing). We would want our classifier to be as fast as possible, without sacrificing accuracy. All convex optimization algorithms have computational complexity of nearly the same order. There is a limit to the speed achievable by solving such optimization problems. In order to achieve faster execution times, we propose a greedy alternative (for solving (12)) to convex optimization. Our proposed fast greedy algorithm is called the Stagewise Block Orthogonal Matching Pursuit.

Stagewise Block Orthogonal Matching Pursuit

This is an iterative algorithm. At each iteration it first selects the groups that have non-zero coefficients. Once the indices of the selected groups are known, the coefficient values for those positions are estimated by least square method. At the final step of each iteration the residual is computed and the stopping criterion is checked. If the stopping criterion is not met the iteration continues.

Initialization – The sparse vector to be estimated is initialized to zero, $\alpha = 0$. The residual is initialized to the test sample, $r^{(0)} = v_{k,test}$. The set of chosen indices is empty $L^{(0)} = []$.

Iteration – Continue the following steps until the norm of the residual is less than a predefined value.

1. The first step computes the correlation between the current residual and the training matrix, i.e. $c(i) = \left\langle V(:i), r^{(t-1)} \right\rangle, \forall i$, where *V(:i)* denotes the i[th] column of the matrix *V*.

2. The groups having indices with correlation higher than a particular threshold are chosen, $l = \{group(i) : | c(i) | \geq k\sigma\}, 2 \leq k \leq 3$ and $\sigma = \| r^{(t)} \|_2 / \sqrt{length(\alpha)}$. The choice of this threshold is provided in [20].

3. The current set of indices is updated by adding the newly chosen indices $L^{(t)} = [L^{(t-1)} \; l]$.

4. The values of the signal at the chosen indices are computed by least squares $x = \min \| v_{k,test} - V(:L^{(t)})x \|_2$. This is computed using Conjugate Gradient method.

5. The coefficient vector and the residual are updated, $\alpha(L^{(t)}) = x$ and $r^{(t)} = v_{k,test} - V\alpha$.

The StBOMP is a very fast algorithm. Experimentally it has been found to terminate in three iterations.

So far we discussed how we can solve the group sparsity promoting optimization problem. This is the main step in the classification algorithm. Our classification algorithm is a slight modification of the one proposed in [1]:

GLSC (Generalized Linear Sparse Classifier) Algorithm

1. Solve the optimization problem expressed in (9) either by optimization or by greedy algorithm.
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. Obtain the representative a sample for each class by a linear combination of the training samples in that class via the equation $v_{rep}(i) = \sum_{j=1}^{n_i} \alpha_{i,j} v_{i,j}$.
   b. Find the error between the reconstructed sample and the given test sample by $error(v_{test}, i) = \| v_{k,test} - v_{rep(i)} \|_2$
4. Once the error for every class is obtained, choose the class having the minimum error as the class of the given test sample.

3.2 GENERALIZED NON-LINEAR SPARSE CLASSIFIER

In the first generalization step it was assumed that the test sample raised to certain powers can be approximately represented as linear combination of training samples belonging to the correct class raised to the same powers. Although this is a generalization from the original SC assumption [1], it is not a fully generalized model. In the second generalization step it is assumed that the test sample raised to certain powers can be approximately represented by a non-linear combination of the training samples raised to the same power, i.e. we are proposing an assumption of the form

$$v = f(V\alpha) + \varepsilon, \varepsilon \sim N(0, \sigma) \tag{13}$$

where $v_{k,test} = [v_{k,test}^{p_1}, v_{k,test}^{p_2}, ..., v_{k,test}^{p_M}]'$; $V = BlockDiag[V_{p_1}, V_{p_2}, ..., V_{p_M}]$ and $\alpha = [\alpha_{p_1}, \alpha_{p_2}, ..., \alpha_{p_M}]'$

This assumption opens a wide and powerful variety of possibilities in terms of modeling the classification problem since it breaks the restrictions imposed by linearity. The fully generalized model comes at the cost of computational complexity. The final form of the classification assumption leads to an optimization problem of the form:

$$\min \| \alpha \|_{2,0} \text{ such that } \|v_{test} - f(V\alpha)\|_2 < \eta \tag{14}$$

This is an NP hard problem to solve. Even the following convex relaxation is computationally intensive

$$\min \| \alpha \|_{2,1} \text{ such that } \|v_{test} - f(V\alpha)\|_2 < \eta \tag{15}$$

There are no specialized techniques to solve the sparse optimization problems with such perturbed non-linear constraints. General optimization tools need to be used in this case, which have been found to be slow. We overcome this problem by directly approximating (14) by a greedy (suboptimal) algorithm.

Recently a greedy algorithm for non-linear sparse system identification was proposed in [21]. It was meant for approximating sparse optimization problems of the form

$$\min \| \alpha \|_0 \text{ subject to } E(\alpha, v_{k,test}) \quad (16)$$

where $E(\alpha, v_{k,test})$ denotes an error measure not necessarily the l_2-norm (widely used for Gaussian Noise).

Our algorithm is based on ideas similar to [21]. It is tailored for solving (14).

Greedy Non Linear Sparse Solution

Initialization - The sparse vector to be estimated is initialized to zero, $\alpha = 0$. The residual is initialized to the test sample, $r^{(0)} = v_{k,test}$. The set of chosen indices is empty $L^{(0)} = []$.

Iteration – Continue the following steps until the norm of the residual is less than a predefined value.

1. The first step computes the gradient of the error at the current coefficient estimate, i.e. $g = \frac{d}{d\alpha} \| v_{k,test} - f(V\alpha) \|_2^2 \; at \; \alpha^{(t-1)}$. This is basically a generalization of the OMP algorithm [22] where the correlations are the negative gradient of error term $\| v_{k,test} - V\alpha \|_2^2$ evaluated at the current coefficient estimate.

2. The group having index with highest gradient magnitude is chosen, $l = \{group(i) : \max | g(i)) | \}$. This step is also similar to OMP, where the index of the highest correlation is chosen.

3. The current set of indices is updated by adding the newly chosen indices $L^{(t)} = [L^{(t-1)} \; l]$.

4. The values of the signal at the chosen indices are computed by least squares optimization $x = \min \| v_{k,test} - f(V(:L^{(t)})x) \|_2$. This is a problem of non-linear least squares and does not have a closed form solution and needs to be solved iteratively.

5. The coefficient vector and the residual are updated, $\alpha(L^{(t)}) = x$ and $r^{(t)} = v_{k,test} - f(V\alpha)$.

This algorithm can be applied for a wide class of functions, the only restriction being $f(V\alpha) = 0, at \; \alpha = 0$.

The non-linear sparse estimation is the core behind the classification algorithm. Based on this estimation we propose the classification algorithm as follows:

GNSC (Generalized Non-linear Sparse Classifier) Algorithm

1. Solve the optimization problem expressed in (14) by the greedy algorithm.
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. Obtain the representative a sample for each class by a linear combination of the training samples in that class via the equation $v_{rep}(i) = f(\sum_{j=1}^{n_i} \alpha_{i,j} v_{i,j})$.

b. Find the error between the reconstructed sample and the given test sample by $error(v_{test}, i) = \| v_{k,test} - v_{rep(i)} \|_2$

4. Once the error for every class is obtained, choose the class having the minimum error as the class of the given test sample.

4. BRIEF REVIEW OF SINGLE IMAGE RECOGNITION PROBLEM

The challenges of the single image per person face recognition problem and the various approaches for tackling it are thoroughly discussed in [2]. There are three broad approaches to tackle the problem – i) Holistic ii) Local and iii) Hybrid. In the first approach, features are extracted from the whole image and are used for classification. In local methods, the face image is divided into patches and features are extracted from the individual patches. These local features are finally used for classification. The hybrid approach is a mix of the above two.

In the following subsections we discuss the implementation of the various single-image-per-person face recognition schemes that we use in this work. All of them have a separate feature extraction stage followed by NN classification. In all these methods, we keep the feature extraction stage intact but replace the NN classification by our sparse classifiers for comparison.

4.1 EXTENSIONS OF PCA

In this method, the standard Eigenface technique is extended to capture as much information as possible from the only face image, examples are $(PC)^2A$ [4], SPCA [5] and Eigenface Selection [6]. These implementations are discussed in the following sub sections.

4.1.1 $(PC)^2A$

The vertical and horizontal projections of an image *I(x,y)* are defined as $V(x) = \sum_y I(x,y)$ and $H(y) = \sum_x I(x,y)$ respectively. The projection map $M(x,y) = \frac{V(x)H(y)}{\sum_x \sum_y I(x,y)}$ is combined with the image $I_\alpha(x,y) = \frac{I(x,y) + \alpha M(x,y)}{1+\alpha}$, where α is the combination parameter fixed at 0.25.

The combined image $I_\alpha(x,y)$ has more discriminating information compared to the *I(x,y)* [4].The Eigenface technique is applied on the projection combined image for recognition.

4.1.2 SPCA

The original intensity image *I* undergoes Singular Value Decomposition (SVD) to yield $I = U\Sigma V^T$. The algorithm reconstructs a new image $P = U\Sigma^n V^T$, the value of n being fixed at 3/2. A new combined image $I_\alpha = \frac{I + \alpha P}{1+\alpha}$, where α is the combination parameter fixed at 0.25, is used. In [5], it is suggested that the value of n be kept at 3/2 and an Eigenface recognition is used on the combined image.

4.1.3 Eigenface Selection

This algorithm requires a generic dataset having multiple images per person. PCA is applied to the generic gallery and the PCs are sorted in a descending order of the eigenvalues $A = [a_1, a_2, ..., a_M]$. Then A_m the subset of *A* with cardinality $m < M$ is found from the complete eigenface set *A* by optimizing $J = \frac{Var_{\text{inter}}(A_m)}{Var_{\text{intra}}(A_m)}$.

For the single training image per person, the intra-subject variance is estimated from the generic dataset [6]. And the inter-subject variation is estimated from a combination of the generic dataset and training set. So the optimization criterion is $J = \frac{\eta Var_{\text{inter:generic}}(A_m) + (1-\eta) Var_{\text{inter:train}}(A_m)}{Var_{\text{intra:generic}}(A_m)}$.

A Sequential Forward Selection algorithm is used to find the *m* most discriminating PCs. Once the discriminating Eigenfaces are obtained, NN is used for classification.

4.2 SYNTHESIZING NEW IMAGES

This method synthesizes new images from the single available image and then standard multi-sample classification methods are used. The SPCA method [5] described above can also be used for synthesizing new images. This synthesis method is called SPCA+. Other synthesis methods include Sampling [8], Non-linear approximations [9] etc. These methods are discussed next.

4.2.1 SPCA+

In the SPCA+ scheme, the SPCA combined image $I_\alpha = \frac{I + \alpha P}{1+\alpha}$, where $P = U\Sigma^n V^T$ is first obtained and the value of n is fixed at 5/4. Instead of using only the synthesized image, both the original image (*I*) and the combined image (I_α) are used, so that there are two samples for each person. The original dataset is augmented in this fashion. Eigenface recognition is employed on the augmented dataset for recognition.

4.2.2 Sampled FLDA

In the sampled Fisher Linear Discriminant Analysis (FLDA) method, the original image is sub-sampled to create several smaller images [8]. The following diagram makes the notion clear.

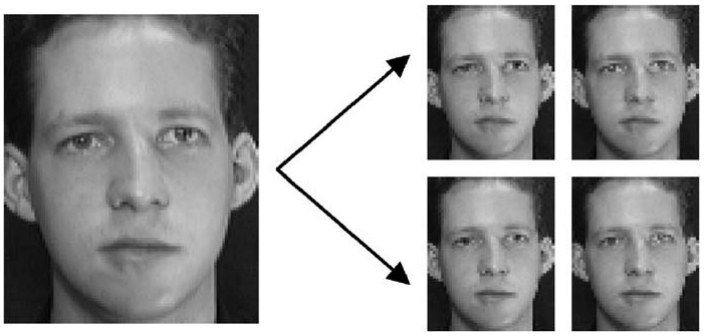

Fig. 1 The 2X2 sub-sampling of the original image [8]

The original image is partitioned into blocks of 2X2 pixel. The top-right image is formed by the top-right pixel of the 2X2 pixel blocks. The rest of the sub-sampled images (on right) are formed from the corresponding pixels of the 2X2 blocks.

The original training images are not used. Instead their sampled versions form the training set. The test image is also sampled. The recognition is carried out using Fisherfaces. Each sub-sampled test image is separately classified using KNN. The results from the 4 sub-samples are combined using majority voting.

4.2.3 Non-linear Approximation

This method uses curvelet transform to generate images having different edge content [9] using the curvelet transform. The analysis and the synthesis equations for the curvelet transform is given by

$x = CI$ (analysis equation)

$I = C^T x$ (synthesis equation)

where C is the curvelet transform matrix, I is the image and the vector x are its curvelet coefficients.

The curvelet coefficients contain the edge information of the image. If some of the curvelet coefficients are deleted, the edge information is changed. The deletion of curvelet coefficients is controlled via the following optimization

$$\min \| x_0 \|_1 \; subject\ to \; \| I - C^T x_0 \| \leq \varepsilon$$

Once some curvelet coefficients are deleted, the image approximation is generated by the synthesis equation ($I = C^T x_0$).

In [9], the error is controlled to be ε (10^{-2}, 10^{-1}, 1, 10, 20 and 50) and six synthesized versions of the original image are obtained. There are 7 (original + 6 approximations) databases for each version of the image. Eigenface recognition is used on the 7 versions, and the final class is decided using majority voting.

5. RESULTS

The proposed classification algorithms are applied on the problem of recognizing faces from a single training image of each person. We follow a similar experimental evaluation methodology as in [6]. Our evaluation is performed over the FERET database which is considered as the de facto standard for current face recognition experiments. The FERET database has 14501 images of 1209 subjects. We only use the 3817 images (of 1200 subjects) that have the

eye-position available, as we are interested only in face recognition and not face detection. The eye positions are required a priori for carrying forth the standard preprocessing steps from the FERET protocol.

Of the 1200 subjects, 226 subjects have 3 images per subject. In [6], it is suggested that this set be used as the generic gallery. These 678 images are also used for tuning our classifier. The training and testing datasets are formed from 1703 images which consist of at least 4 images per subject for another 256 persons. The training dataset is formed by randomly selecting 256 images (one image for each person) for the 256 people; the remaining 1447 images form the testing set.

Tables 1-4 show the recognition accuracy results for single image per person face recognition. In all the tables the top three algorithms in the first column are 'Extensions of the PCA' technique, while the bottom three are of the 'Synthesizing New Images' type. Table 1, shows the results of the original algorithms which uses Nearest Neighbour classification. From table 2 onwards we apply the new classifiers in place of NN. Table 2 shows the recognition accuracy results of the sparse classifier [1]. Table 3 and 4, tabulates the results for the Generalized Linear Sparse classifier. Table 5, tabulates the results for our Generalized Non-linear Sparse classifier.

The NN and the SC are non-parametric classifiers. But certain parameters need to be decided for our proposed classifiers. For the Generalized Linear Sparse Classifiers (tables 3 and 4), we found that the values of index between 0.1 and 2 give good recognition accuracy. In this work we considered the values of p – 0.125, 0.25, 0.5, 1 and 2. We tried sampling the range uniformly (0.1 to 2 in steps of 0.1) but saw that there was no gain in recognition accuracy with such fine sampling.

The Generalized Non-linear Sparse Classifier offers a wide range of modeling functions to be used for classification. It is not possible to test all the different functional forms and decide the best one for our problem. The main idea of this work is to show that simple non-linear models can provide significant improvements in recognition accuracy. In this work, we tested the following functions:

$$f_1(A,x) = (Ax)^2 + Ax$$
$$f_2(A,x) = (Ax)^{1/2} + (Ax)^2 + Ax$$
$$f_3(A,x) = (Ax)^{1/2} + Ax$$

Of these we found that the third function gives the best recognition results. The results are shown in table 5.

Table 1. Recognition Accuracy using Nearest Neighbour (Original Algorithms)

Name of the Feature Extraction Algorithm	Number of Eigenfaces/Fisherfaces		
	20	40	80
$(PC)^2A$	0.42	0.48	0.5
SPCA	0.44	0.51	0.55
Eigenface Selection	0.46	**0.54**	**0.57**
SPCA+	0.44	0.52	0.55
Sampled FLDA	0.42	0.5	0.51
Non-linear Approximation	**0.47**	**0.54**	0.56

Table 2. Recognition Accuracy using Sparse Classifier

Name of the Feature Extraction Algorithm	Number of Eigenfaces/Fisherfaces		
	20	40	80
$(PC)^2A$	0.45	0.5	0.53
SPCA	0.47	0.52	0.55
Eigenface Selection	0.48	0.55	**0.6**
SPCA+	0.48	**0.56**	**0.6**
Sampled FLDA	0.46	0.51	0.55
Non-linear Approximation	**0.5**	**0.56**	**0.6**

Table 3. Recognition Accuracy using Generalized Linear Sparse Classifier (Optimization)

Name of the Feature Extraction Algorithm	Number of Eigenfaces/Fisherfaces		
	20	40	80
$(PC)^2A$	0.49	0.54	0.56
SPCA	0.5	0.56	0.57
Eigenface Selection	0.51	0.58	**0.62**
SPCA+	**0.52**	**0.59**	**0.62**
Sampled FLDA	0.48	0.56	0.58
Non-linear Approximation	0.5	**0.59**	**0.62**

Table 4. Recognition Accuracy using Generalized Linear Sparse Classifier (Greedy Algorithm)

Name of the Feature Extraction Algorithm	Number of Eigenfaces/Fisherfaces		
	20	40	80
$(PC)^2A$	0.49	0.54	0.56
SPCA	0.5	0.56	0.57
Eigenface Selection	0.5	0.57	**0.62**
SPCA+	**0.51**	**0.58**	**0.62**
Sampled FLDA	0.46	0.54	0.58
Non-linear Approximation	0.49	0.60	**0.62**

Table 5. Recognition Accuracy using Generalized Non-linear Sparse Classifier

Name of the Feature Extraction Algorithm	Number of Eigenfaces/Fisherfaces		
	20	40	80
$(PC)^2A$	0.52	0.58	0.62
SPCA	0.54	0.59	0.63
Eigenface Selection	0.57	0.64	**0.69**
SPCA+	0.55	0.64	**0.69**
Sampled FLDA	0.52	0.60	0.65
Non-linear Approximation	**0.57**	**0.65**	**0.69**

There are a few interesting observations from the above tables:

- The Sparse Classifier (SC) is always better than the Nearest Neighbour (2-3% improvement).
- The Generalized Linear Sparse Classifier (GLSC) gives better results than the NN and the SC. It shows about 2-3% improvement over the SC. The GLSC based on optimization gives slightly better results than the one based on the group sparsity promoting greedy algorithm.
- The Generalized Non-linear Sparse Classifier gives significantly better results compared to the others. It shows about 6-7% improvement in recognition accuracy over the nearest competitor the GLSC.

The Support Vector Machine (SVM) is a very well known classifier, and has been widely used for general purpose classification tasks. SVM however, is not a good choice for a classifier when the number of training samples is few, as in face recognition. In [1] it was shown that the SC outperforms SVM in terms of recognition accuracy. Consequently, we did not test SVM for our experiments. Our proposed classifiers significantly outperform the SC and would obviously be better than SVM.

6. DISCUSSION AND FUTURE SCOPE

This work proposes major generalization of the sparse classification framework. The proposed classifiers were tested on the real-life problem of identifying faces of people from a single training image. The results show major improvement over previous Nearest Neighbour based methods.

This paper is exploratory in nature. The classification algorithms are highly generalized and flexible. But in order to make good use of these classifiers several questions must be answered – The first being the choice on the values of p for other classification problems (not necessarily face recognition). In our case, we found the values manually. The second question is even more important – how to choose the non-linear classification model. Again in this case, we tried several simple models and found the one that suits us the best.

References

[1] Yang, Y., Wright, J., Ma, Y., Sastry, S. S., (to appear). Feature Selection in Face Recognition: A Sparse Representation Perspective. IEEE Transactions on Pattern Analysis and Machine Intelligence.
[2] Tan, X., Chen, S., Zhou, Z. H, Zhang, F., 2006. Face recognition from a single image per person: A survey. Pattern Recognition 39 (9), 1725-1745.
[3] Wu, J., Zhou, Z. H., 2002. Face Recognition with one training image per person. Pattern Recognition Letters, 23(14), 1711-1719.
[4] Chen, S. C., Zhang D.Q., Zhou, Z. H., 2004. Enhanced (PC)2A for face recognition with one training image per person. Pattern Recognition Letters, 25(10), 1173-1181.
[5] Zhang D.Q., Chen S.C., Zhou, Z. H., 2005. A new face recognition method based on SVD perturbation for single example image per person. Applied Mathematics and Computation 163(2), 895-907.
[6] Wang J., Plataniotis K.N., Venetsanopoulos A. N., 2005. Selecting discriminant eigenfaces for face recognition. Pattern Recognition Letters 26(10), 1470-1482.
[7] Jung, H. C., Hwang, B. W., Lee, S. W., 2004. Authenticating Corrupted Face Image Based on Noise Model. International Conference on Automatic Face and Gesture Recognition, 272-277.
[8] Yin, H., Fu, P., Meng, S., 2006. Sampled FLDA for face recognition with single training image per person. Neurocomputing 69, 2443-2445.
[9] Majumdar, A., Ward, R. K., 2008. Single Image per Person Face Recognition with Images Synthesized by Non-Linear Approximation. International Conference on Image Processing, 2740-2743.

[10] D. Donoho, “For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution,” Comm. on Pure and Applied Math, Vol. 59 (6), pp. 797–829, 2006.
[11] http://www.acm.caltech.edu/l1magic/
[12] Rick Chartrand and Wotao Yin, "Iteratively reweighted algorithms for compressive sensing", in 33rd International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2008.
[13] Y. Kim, J. Kim, and Y. Kim, “Blockwise sparse regression”, Statistica Sinica, 16, pp. 375-390, 2006.
[14] L. Meier, S. van de Geer, and P. Buhlmann, The group lasso for logistic regression, J. R. Statist. Soc. B, 70, pp. 53-71, 2008.
[15] B. Turlach, W. Venables, and S. Wright, “Simultaneous Variable Selection, Technometrics, 47, pp. 349-363, 2005.
[16] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables”, J. R. Statist. Soc. B, 68, pp. 49-67, 2006.
[17] J. Huang and T. Zhang, “The Benefit of Group Sparsity” arXiv:0901.2962v1.
[18] M. Stojnic, F. Parvaresh and B. Hassibi, “On the reconstruction of block-sparse signals with an optimal number of measurements”, arXiv:0804.0041v1
[19] Y. C. Eldar and M. Mishali, “Robust recovery of signals from a union of subspaces,” IEEE Trans. Inf. Theory, 2008, submitted.
[20] D.L. Donoho, Y. Tsaig, I. Drori, J.-L. Starck, “Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit”, preprint http://www-stat.stanford.edu/~idrori/StOMP.pdf
[21] T. Blumensath and M. E. Davies; "Gradient Pursuit for Non-Linear Sparse Signal Modelling", European Signal Processing Conference (EUSIPCO), Lausanne, Switzerland, April 2008.
[22] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via Orthogonal Matching Pursuit”, IEEE Trans. Info. Theory, vol. 53, num. 12, pp. 4655-4666, 2007.