Robust Face Recognition via Sparse Representation

Panqu Wang
Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, CA 92092
pawang@ucsd.edu

Can Xu
Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, CA 92092
canxu@ucsd.edu

Abstract

In this project, we implement a robust face recognition system via sparse representation and convex optimization. We treat each test sample as sparse linear combination of training samples, and get the sparse solution via L_1-minimization. We also explore the group sparseness (L_2-norm) as well as normal L_1-norm regularization.We discuss the role of feature extraction and classification robustness to occlusion or pixel corruption of face recognition system. The experiments demonstrate the choice of features is no longer critical once the sparseness is properly harnessed. We also verify that the proposed algorithm outperforms other methods.

1 Introduction

Face recognition has been put into extensive research effort. Notwithstanding considerable complicated models have been proposed to deal with this problem, computational parsimony is still a principle and supported by low-level and mid-level human vision [1]. J. Wright et. al. [2] introduces a novel approach using sparse representation to deal with the face recognition problem. When the optimal representation for the test face is sparse enough, the problem can be solved by convex optimization efficiently [4]. In this project, we will discuss the relevant theory and perform experiments with our own implementation of the framework.

The sparse representation means among all the coefficients of base vectors, only a small fraction on the entries are nonzero. This representation is discriminative naturally, as it could select the subset of base vectors which express the input signal most concentrated and automatically reject other less concentrated representations. Therefore, we can exploit sparse representation to perform classification task. The test examples are represented by considerably large number of training examples from various classes. The sparsest solution we get therefore discriminate the object class and other classes automatically. This linear sparse representation can be recovered efficiently via L_1-minimization.

Section 2 will give a brief theoretical overview for sparse representation and propose the Sparse Representation-based Classification (SRC) algorithm and Group Sparseness Representation (GSRC). In section 3, we will discuss the role of feature extraction and the improvement based on group sparseness. We will also explore the classification robustness to occlusion or pixel corruption of face recognition system. Section 4 will present the experiment result under various conditions. Section 5 will be the conclude the paper.

2 Sparse Representation-based Classification (SRC)

2.1 Basic Problem For Single Test Example

According to J. Wright et. al., we arrange the nth training sample (image with size $w \times h$ in this case) from the ith class as one column of a matrix $A_i = [v_{1i}, ..., v_{ni}]$ where $v \in \mathbb{R}^m (m = w \times h)$. For the entire training set, if there are k classes in total, we form a matrix $A = [A_1, ..., A_k] = [v_{1i}, ..., v_{nk}] \in \mathbb{R}^{m \times nk}$ to store all training examples. Then, a linear representation of a test image y can be written in terms of all training examples as

$$y = Ax_0 \qquad \in \mathbb{R}^m, \tag{1}$$

where $x_0 \in \mathbb{R}^n$ is a coefficient vector with most entries have a zero element.

We use L_0 minimization to get the sparse solution of (1), which can be rewritten as:

$$\hat{x}_0 = \text{argmin} \parallel x \parallel_0 \quad \text{subject to } Ax = y. \tag{2}$$

However, this problem for underdetermined system of linear equations is NP-hard [6]. Recent theory in compress sensing shows that if the solution of $\hat{x}_0$ is sufficiently sparse, solving the L_0-minimization problem (3) is equal to solve the following *basic L_1-minimization* problem:

$$\hat{x}_1 = \text{argmin} \parallel x \parallel_1 \quad \text{subject to } Ax = y. \tag{3}$$

This problem should be solved in polynomial time by standard linear programming methods [10].

To deal with small dense noise, the model (2) could be modified using $y = Ax_0 + z$ where z is a noise term. Now the optimization problem turns to:

$$\hat{x}_1 = \text{argmin} \parallel x \parallel_1 \quad \text{subject to } \|Ax - y\| \leq \epsilon \tag{4}$$

where ϵ is a bounded term for z. This convex optimization problem can be efficiently solved using second-order cone programming.

However, l_1-norm does not yield sparsity at the group level. A more general sparse group lasso criterion will include l_2-norm[5]. We consider the convex optimization problem

$$\min_{x \in R^k} (||x||_1 + \lambda \sum_{i=1}^{k} ||x_i||_2) \qquad \text{subject to } Ax = y \tag{5}$$

where $\lambda$is the tuning factor which describes the contribution of group sparseness to the cost function.

2.2 Classification Based on Sparse Representation

After we get the solution $\hat{x}_1$ for (3), we classify the test example y based on how well the sparse coefficients associated with each object class that reproduce y. Suppose $\gamma_i : \mathbb{R}^n \to \mathbb{R}^n$ is the function that selects the coefficients within the ith class. For each $x \in \mathbb{R}^n$, $\gamma_i(x)$ returns the vector of length n whose only non-zero entries are within the range of class i. We can approximate the test image $\hat{y_i} = A\gamma_i(\hat{x})$. We then classify y by assigning it to object i that minimize the residual between real y and the approximations:

$$\min_i \; r_i(y) \doteq \parallel y - A\gamma_i(\hat{x}_1) \parallel_2 \tag{6}$$

In some circumstances, the quality of test images are not good, some of the images may not contain faces at all. In this sense, we will perform a validation step to reject the invalid test images. We will use the proposed Sparsity Concentration Index (SCI):

$$\text{SCI}(x) = \frac{k \cdot \max_i \parallel \gamma_i(x) \parallel_1 / \parallel x \parallel_1 - 1}{k-1} \tag{7}$$

to validate the test images. k is the total number of classes in the test set. If $\text{SCI}(x) = 1$, the test image is only represented using images from single class. If $\text{SCI}(x) = 0$, the sparse coefficients are distributed uniformly across all classes. More concentrated representation indicates a better solution. We can set a threshold $\tau \in (0, 1)$ to determine whether to accept the test image.

3 The Effect of Feature Extraction and Pixel Corruption

In this section, we will extend the basic SRC model to apply to a wider range of problems in face recognition. We will study: 1) the effect of feature transformation, and 2) the robustness to pixel corruption, i.e., noise and occlusion.

3.1 The Role of Feature Extraction

Numerous feature extraction methods have been proposed in computer vision literature. For face recognition problems, features are extracted either holistically or partially. Traditionally, feature extraction is always followed by some simple classifiers, like Nearest Neighbors, to perform the recognition task. In this section, we will examine the effect of feature extraction within the framework of face recognition.

Most feature extraction processes only involves linear transformation and project the original image to feature space. Suppose $R \in \mathbb{R}^{d \times m}$ is the transformation matrix with $d < m$. Applying R to both sides of (1) yields

$$\tilde{y} \doteq Ry = RAx_0 \qquad \in \mathbb{R}^d \tag{8}$$

We can still hope the desired solution x_0 is sparse, so we can solve the following *reduced* L_1*minimization problem*:

$$\hat{x}_1 = \text{argmin} \parallel x \parallel_1 \quad \text{subject to } RAx = y. \tag{9}$$

3.2 Robustness to Noise and Occlusion

In many practical face recognition situation, the test image y might be corrupted or occluded. At this time the robustness of the classifier is important. The model is modified as

$$y = y_0 + e_0 = Ax_0 + e_0 \tag{10}$$

where $e_0 \in \mathbb{R}^m$ is the error vector. Since the error (occlusion or corruption) appears randomly in the test image, some entries in e_0 has non-zero values with different magnitudes. The location of the error can vary from different test images and are unknown to the algorithm.

Since redundancy plays an important role in object recognition as the number of pixels in an image is much more than the number of subjects, people can recognize the subjects despite of noise corruption or occlusion. In order to keep the most redundancy, the test image should be processed with the highest possible resolution, as within the ability of processors.

Let us assume that the corruption happened in a small fraction of pixels in a test image, thus the the vector e_0 has sparse non-zero entries. The equation 1 can be rewrote as:

$$y = [A,\ I][\ \begin{matrix} x_0 \\ e_0 \end{matrix}\ ] = Bw_0 \tag{11}$$

where $B = [A\, I] \in \mathbb{R}^{m \times (n+m)}$. The task now is to solve w_0 as the sparsest solution to the system $y = Bw$. The first n coefficients of w_0 are the original sparse coefficients for x_0, the following

m coefficients are the coefficients for the errors. As before, we attempt to get the sparsest w_0 by solving the following *extended L_1minimization* problem:

$$\hat{w}_1 = \text{argmin} \parallel w \parallel_1 \quad \text{subject to } Bw = y. \tag{12}$$

Once we get the sparse solution $\hat{w}_1 = [\hat{x}_1; \hat{e}_1]$, the clean version of the image can be recovered as $y_c = y - \hat{e}_1$, where $\hat{e}_1$ is the compensated image for occlusion or corruption. To perform the classification task, we modify the residual $r_i(y)$ slightly:

$$r_i(y) = \parallel y_c - A\delta_i(\hat{x}_1) \parallel_2 = \parallel y - \hat{e}_1 - A\delta_i(\hat{x}_1) \parallel_2 \tag{13}$$

4 Experiments

In this section, we discuss the experiment on extended Yale B database for face recognition, which validates the efficacy of the proposed classification method based on sparse representation. We will first examine the role of feature extraction within the sparse representation framework by comparing the performance on various global features and feature dimensions. Then we will show that the proposed method outperforms some popular classifiers. The GSRC is also performed to compare with the SRC. In the second and third experiment, we will test the performance of SRC on noisy images and occluded images respectively, and then compare it to other methods.

4.1 Feature Extraction and Classification Methods

We test the proposed method with some global features, such as down-sampled images, Eigenfaces [7], Laplacianfaces [8] and compare their performances under sparse representation. We also compare the algorithm with two popular classifiers, namely linear Support Vector Machine (SVM) and Nearest Neighbors (NN). All the experiments are implemented in Matlab on a typical 2.1-GHz PC.

We perform the experiments on the Extended Yale B database [9], which contains 2,414 images of 38 individuals. The cropped 192×168 frontal images are taken under different lighting conditions. For each individual , we randomly pick half of the images for training and the other half for testing.

We compute the recognition rate with feature dimensions 36, 56, 132, 504, which correspond to down-sampling ratio 1/32,1/24,1/16,1/8 respectively. Figure 1 shows the recognition rate using sparse representation on various feature space. The recognition rate increases as the dimensionality of feature space grows. This is in consistent to our intuition. In addition, the three features being used achieve close performances. The unconventional down-sampled images work even slightly better than Laplacianfaces and Eiegenfaces. The reason is that for the computation of Eigenfaces and Laplacianfaces, we have to down sample the images first because of the Matlab memory limit. The three features achieve recognition rates between 92.1 percent and 94.9 percent on 132D. The maximum recognition rate is 96.1 percent achieved on 504D down-sampled images. In general, we can conclude that for SRC method, the choice of features is no longer critical once the sparsity of the recognition problem is properly harnessed.

The rest plots in Figure 1 show the performance of the three features on various classifiers respectively. Sparse representation work beyond linear SVM and nearest neighbor method on all three features and four feature dimensions used in the experiment.

4.2 Incorporating Group Sparseness

In section 2 we introduced group sparseness in the cost function (5). We choose the tuning factor λ by 5-fold cross validation on the training set. Table 1 compares the recognition rates of SRC and GSRC using downsampled images as features. The results show that GSRC performs slightly better than SRC. Figure 2 shows a sample misclassified by SRC and correctly classified by GSRC. The testing input belongs to the 9th class but there is a fake cluster of peaks in the 32th class, thus make the residual of the 32th class smaller and leads to false classification. It can be observed that GSRC almost eliminates the fake peaks for the same testing input. Although the solution(coefficients) appears much noisy, its sparseness is harnessed on the group level.

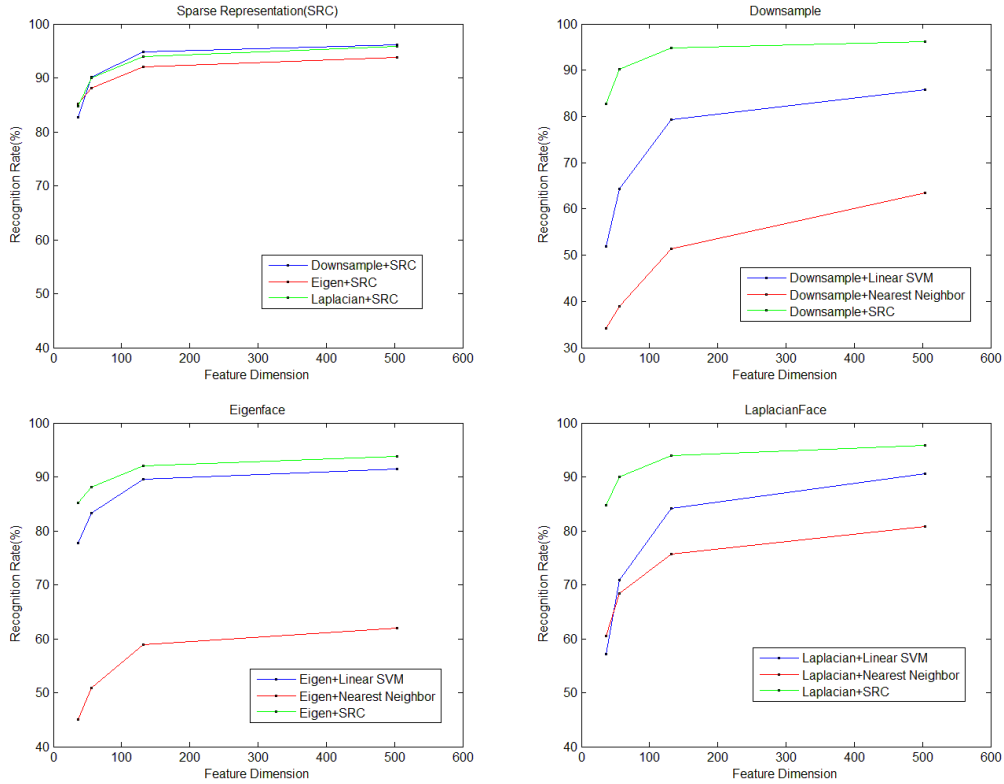

Figure 1: **Recognition rates on Extended Yale B database**, under different feature transformations and classifiers. Upper left: SRC for various features. The recognition rate does not vary much along the choice of features. Upper right: down-sampled images. Lower left: eigenfaces. Lower right: Laplacian faces. The SRC outperforms other techniques.

Feature Dimension	36	56	132	504
SRC	84.1%	90.1%	94.2%	96.1%
GSRC	86.1%	90.6 %	94.8 %	96.3%

Table 1: Recognition rates Comparison between SRC and GSRC using downsampled images

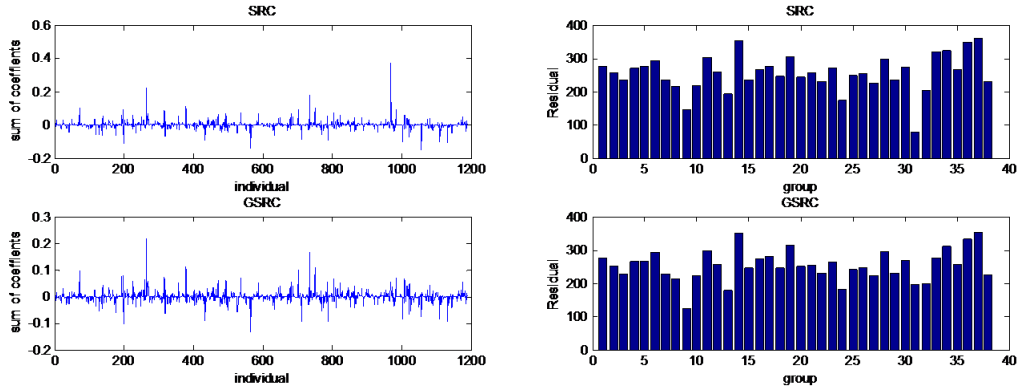

Figure 2: Coefficients and Residuals of a testing sample which is misclassified by SRC and correctly classified by GSRC. Left: Coefficients(Solution); Right: Residuals of each class.

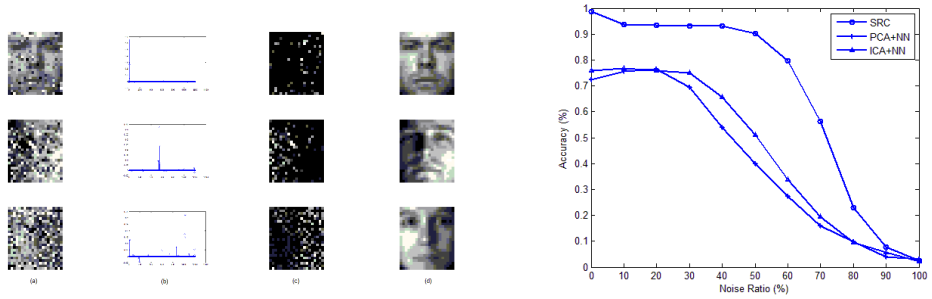

Figure 3: **Face recognition under pixel corruption.** The left figure shows some examples. Column (a): test images from the dataset with random corruption. Top row: 20% pixels corrupted. Middle row: 40% pixels corrupted. Bottom row: 60% pixels corrupted. Column (b): estimated sparse coefficients $\hat{x}$. Column (c): estimated errors $\hat{e}$. Column (d): Reconstructed images $\hat{y}$. The right figure shows the recognition rate across the whole range of noise ratio for different algorithms (SRC, PCA and ICA). The SRC result dramatically outperforms others.

4.3 Recognition despite noise

For this experiment, we test the performance of sparse representation-based classification using noisy testing images. We solve this problem using the *extended* L_1*minimization* method, as shown in equation (11). The dataset we use is the still the Extended Yale B Face Database. Same as the feature extraction experiment, we use half of the dataset (1205 images) for training, and the other half (1209 images) for testing. For SRC method, the images are resized to 24×21 pixels in order to be able to run within the memory size of Matlab.

To implement the added noise, we corrupt a percentage of the pixel value of the test images. We replace the pixel value with independent and identically distributed samples from a uniform distribution whose lower bound and upper bound are the minimum and maximum pixel value of the image, respectively. The location of the corrupted pixels are selected randomly for each test image and is unknown to the SRC algorithm. We vary the percentage of corruption from 0 percent to 100 percent. Figure 3 column (a), (b), (c) and (d) shows some sample test images and their result. For human eye, if the corruption is beyond 50 percent, we can hardly recognize them as face images. Nevertheless, the SRC algorithm still performs well under this extreme condition.

We compare the SRC method to other two popular algorithms for face recognition. The Principal Component Analysis (PCA) [3] first compute the covariance matrix for the feature variables (here are pixel values), and then find the principal components by ranking the eigenvalues of the covariance matrix. After this, we reduce the dimensionality by taking the first several eigenvectors and project the test data onto this new space. Finally we will implement the simple 1-Nearest Neighbor classifier (1-NN) to perform the classification. The other algorithm is Independent Component Analysis (ICA) [10]. ICA attempts to separate the training set as a linear combination of statistically independent base images. Here we use the FastICA algorithm [11] to obtain the ICA basis and project the test image on these bases to perform classification.

The right plot in Figure 3 plots the recognition rates for the three algorithms. We can see that the SRC performs significantly better than its competitors. From 0 percent to 50 percent corruption, SRC keeps the accuracy always higher than 90 percent, while the performance of PCA and ICA has dropped to around 50 percent at 50% corruption rate. We can also observe the power of SRC from the sample results: Even at the 60% corruption rate, the coefficients still concentrate within the desired class. We can also recover the original image quite well (see column (d)) from the noise.

4.4 Recognition despite occlusion

For this experiment, we add various levels of occlusion range from 10 percent to 50 percent for each test image. If the occlusion is greater than 50 percent, the occluding part will dominate the test image and make people wonder whether the task is to recognize the foreground or background. The

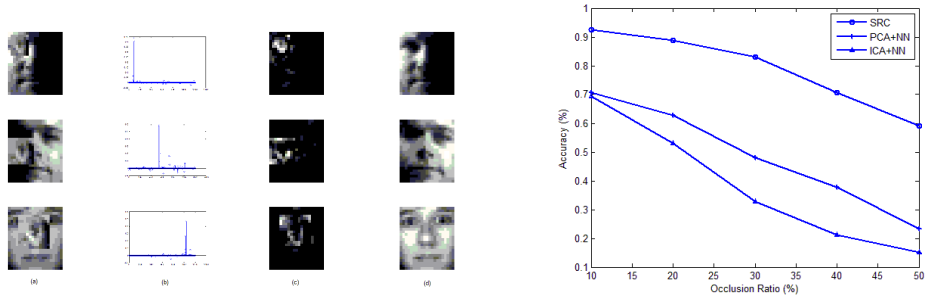

Figure 4: **Face recognition under occlusion.** The left figure shows some examples. Column (a): test images from the dataset with random occlusion. Top row: 10% pixels occluded. Middle row: 20% pixels occluded. Bottom row: 30% pixels occluded. Column (b): estimated sparse coefficients $\hat{x}$. Column (c): estimated errors $\hat{e}$. Column (d): Reconstructed images $\hat{y}$. The right figure shows the recognition rate across the whole range of occlusion ratio for different algorithms (SRC, PCA and ICA). The SRC result dramatically outperforms others.

formation for training set and test set is the same as the previous experiment. We solve the problem using equation (11) as well. To perform the occlusion, we replace a random located block of the test image with an totally unrelated image. The location is unknown to the algorithm. Figure 4 column (a), (b), (c) and (d) shows some sample test images and their result. In the third row, the occlusion ratio is 30 percent and the center of the face is occluded; this is a difficult recognition problem even for humans. We can see the SRC still deal with the problem very well.

We compare SRC with PCA and ICA again, as in experiment two. SRC again outperforms its competitors significantly. For 10 percent occlusion rate, the recognition rate for SRC is 92.56%, while PCA and ICA are both just 70%. The gap is even bigger when the occlusion rate grows. For 40 percent occlusion rate, SRC's accuracy is 70.72%, while both PCA and ICA's accuracy are below 40%. Finally, when comparing the occlusion with noise corruption, we can find the occlusion is a worse type of error for all algorithms.

5 Conclusion

In this paper, we introduce the theory of sparse representation and its application onto face recognition. We verify that the feature extraction is no longer critical to recognition once the sparsity of the problem is properly harnessed. We improve the sparse description by incorporating group sparseness. We also test the SRC algorithm under noisy and occluded images. The experiment results show that the SRC outperforms other techniques under all circumstances.

References

[1] B. Olshausen and D. Field, "Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1?", Vision Research, vol. 37, pp. 3311-3325, 1997.

[2] J. Wright., A. Y. Yang; Ganesh, A., S. S. Sastry, and Y. Ma, "Robust Face Recognition via Sparse Representation", Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.31, no.2, pp.210-227, Feb. 2009 doi: 10.1109/TPAMI.2008.79

[3] P. Pearson, "On Lines and Planes of Closest Fit to Systems of Points in Space", Philosophical Magazine 2 (6): 559-572 (1901).

[4] D. Donoho, "For Most Large Underdetermined Systems of Linear Equations the Minimal l1-Norm Solution Is Also the Sparsest Solution", Comm. Pure and Applied Math., vol. 59, no. 6, pp. 797- 829, 2006.

[5] J. Friedman, T. Hastie, R. Tibshirani, "A note on the group lasso and a sparse group lasso", Statistics Theory, Feb. 2010

[6] E. Amaldi and V. Kann, "On the Approximability of Minimizing Nonzero Variables or Unsatisfied Relations in Linear Systems", Theoretical Computer Science, vol. 209, pp. 237-260, 1998.

[7] M. Turk and A. Pentland, "Face recognition using eigenfaces", Proc. IEEE Conference on Computer Vision and Pattern Recognition. pp. 586-591,1991.

[8] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang, "Face Recognition Using Laplacianfaces", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 3, march 2005

[9] A. Georghiades, P. Belhumeur, and D. Kriegman, "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23, no. 6, pp. 643-660, June 2001.

[10] P. Comon, "Independent Component Analysis: a new concept?", Signal Processing, 36(3):287-314 (1994).

[11] Hyvarinen, A., Oja, E. "Independent Component Analysis: Algorithms and Application", Neural Networks, 13(4-5):411-430 (2000).