

# COS429 Term Project: Discriminative Face Recognition Via Regression

Yongxin Xi  
Princeton University

## Abstract

*For machine learning problems with high dimensional data such as human faces, dimension reduction is usually applied first to enhance the generalization capacity as well as decrease the computational cost. In this project, the LARS-EN model—a regularized regression method is used to identify sparse and meaningful features in the original pixel domain or after certain transforms in face recognition. Also, motivated by the fact that the regular simplex vertices are separate points with highest degree of symmetry, we choose such vertices as the targets for the distinct faces. During recognition, a new face can be identified by mapping it into this face subspace and comparing its distance to all individual targets.*

*The uniqueness and strength of this method is that during regression, it can automatically choose the most interesting regressors from a large pool of candidate regressors. Under the assumption that faces can be sparsely represented under certain transformation, LARS following different transforms will offer a more flexible mechanism of regression and a boosted performance of recognition, beating other recognition methods, such as Eigenfaces and Fisherfaces.*

## 1. Introduction

Face recognition has attracted tremendous attention in the computer vision community over the past few decades and many new techniques have been developed. Among them the appearance-based method is one of the most successful and well-studied techniques. In appearance-based methods, the image is represented by a high dimensional vector of pixels. To overcome the difficulty incurred by high dimensionality, a lot of subspace methods, such as the Eigenfaces, and, Fisherfaces have been developed. Eigenfaces applies Principle Component Analysis (PCA) to project the original  $n$ -dimensional data onto a low dimensional subspace which preserves the maximum variance. Fisherfaces uses Linear Discriminant Analysis (LDA) to find the most discriminant eigenvectors which maximizes the ratio of between-class and within-class variances. LDA usually outperforms PCA in face recognition problems.

Regression is another way to conduct face recognition. For linear regression, we consider the class of a face as a linear function of the face features, for example, the gray scale values of all pixels. Every training face is vectorized into a row then put in a big data matrix, where each column is considered a regressor. However, feature representation in pixel domain is not quite concise and sparse. Other face features can be obtained via for instance, certain linear transformation (DFT, PCA, etc) of the face data. Once we have other ways of representing faces in new bases, new regressors can be introduced for the regression problem.

If there are totally  $m$  classes, each class can be represented as a vertex of a regular  $m$ -simplex, whose vertices are geometrically the most balanced and symmetric separate points in the  $(m-1)$  subspace. The cluster centers of individual faces after regression then have the highest degree of symmetry and balance, i.e. they have equal pairwise distances. Then we apply least angle regression—elastic net (LARS-EN) method for the regression problem. LARS-EN was introduced by Zou et. al in 2005. It regularizes the ordinary least squares (OLS) solution with both the Ridge regression and Lasso constraints, so that sparse solutions can be provided. In the algorithm, it adds a new regressor into the model during each iteration, loosening the regularization with the Lasso constraint. The ridge constraints make it possible for the algorithm to include more regressors than the number of observations.

In this paper, a regression approach to increase the face recognition rate is proposed. It looks at possible transformations (either discriminative or not) and automatically seeks a combination of interesting regressors obtained from them. Compared to methods with the same reduced dimension permitted, it performs a lot better in testing phase. In this sense, it can be viewed as a boosting method of recognition.

The rest of the paper is organized as follows: In section two, a review of the standard face recognition techniques such as Eigenfaces and Fisherfaces is presented. Section three describes our approach of recognition. First we introduce the regression problem. Then LARS algorithm is explained, and a way of generating regular  $m$ -simplex will be provided. Next we will show that Elastic Net regression can be transformed to a regression problem with Lasso

constraint only, so that LARS algorithm can be applied to solve the problem. At the end a flow chart of LARS-EN in face recognition is attached. In section four, we describe the two databases and preprocessing details, then compare the recognition results of LARS series methods with other methods. Finally, section five concludes the experiments and discusses some future potential of this approach. Reference and supplementary materials (main codes) will be attached at the end.

## 2. Background

In many areas of machine learning, we are often confronted with intrinsically low-dimensional data lying in a very high-dimensional space. Consider the identity of a human face for instance. Each such image would typically be represented by a big matrix of pixels in gray value. If there are  $n^2$  pixels in all, then each image yields a data point in  $\mathbf{R}^{n^2}$ . It gets very large when  $n$  is large. Besides, the high dimensionality introduces noisiness in the data as well. Dimension reduction techniques, on the other hand, tend to pick out fewer but meaningful variables from the original data, to decrease the noisiness and usually enhance the generalization performance for new data. Among methods in face recognition, Eigenfaces via PCA and Fisherfaces via LDA are two popular dimension reduction and recognition techniques.

### 2.1. PCA

Principle Component Analysis (PCA) and Fisher's Discriminant Analysis (FDA) following PCA both apply linear projection to the original image space to achieve dimensionality reduction. The projected space can be seen as a feature space where each component is seen as a feature. Given an  $p$ -dimensional vector representation of each face in a training set of images, Principal Component Analysis (PCA) tends to find a  $t$ -dimensional subspace whose basis vectors correspond to the maximum variance direction in the original image space. This new subspace is normally lower dimensional ( $t \ll p$ ). Put in maths, let matrix  $X$  be matrix of row vectors of the training images, and  $W$  represent the projection matrix. Then the projected data is  $Y = XW$ . The projection is chosen to maximize the variance of  $Y$ :

$$W_{opt} = \operatorname{argmax} |W^T \bar{X}^T \bar{X} W|$$

where  $\bar{X}$  is the data matrix with the mean subtracted.  $W_{opt}$  can be obtained from solving the SVD problem of  $\bar{X}^T$ . Let

$$\bar{X}^T = U \Sigma V$$

So,

$$\begin{aligned} \bar{X}^T \bar{X} &= (U \Sigma V)(U \Sigma V)^T \\ &= U \Sigma V V^T \Sigma U \\ &= U \Sigma^2 U \end{aligned}$$

Therefore  $W_{opt}$  should be the  $t$  eigenvector columns associated with the largest  $t$  eigenvalues chosen from  $U$ , i.e., the first  $t$  columns of  $U$ .

### 2.2. FDA

FDA is based on Linear Discriminant Analysis (LDA), which finds the vectors in the underlying space that best discriminate among classes. For all samples of all classes the between-class scatter matrix  $S_B$  and the within-class scatter matrix  $S_W$  are defined. The goal is to maximize  $S_B$  while minimizing  $S_W$ , in other words, maximize the ratio of the two. Put in maths, the definitions of the scatter matrices are:

$$\begin{aligned} S_B &= \sum_c N_C (\mu_C - \bar{x})(\mu_C - \bar{x})^T \\ S_W &= \sum_C \sum_i N_{i \in c} (x_i - \mu_C)(x_i - \mu_C)^T \end{aligned}$$

where,

$$\begin{aligned} \mu_C &= \frac{1}{N_C} \sum_{i \in C} x_i \\ \bar{x} &= \frac{1}{N} \sum_i x_i = \frac{1}{N} \sum_C N_C \mu_C \end{aligned}$$

and  $N_C$  is the number of training examples in class  $C$ . So the optimization problem is:

$$\operatorname{Max}_W J(W) = \frac{W^T S_B W}{W^T S_W W}$$

which is equivalent to solving:

$$\begin{aligned} \operatorname{Min}_W &(-W^T S_B W) \\ \text{S.t.} &W^T S_W W = 1 \end{aligned}$$

Corresponding to the lagrangian,

$$L(W, \lambda) = -W^T S_B W + \lambda(W^T S_W W - 1)$$

According to the KKT condition,

$$\nabla L(W) = -2S_B W + 2\lambda S_W W = 0$$

and it brings down to the generalized eigenvalue problem

$$S_B W = \lambda S_W W$$

For a  $p \times t$  ( $t < p$ ) linear transformation  $W$ , select the  $t$  eigenvectors associated with the largest  $t$  eigenvalues from the generalized eigenvalue problem.

In case the number of training samples is much smaller than the number of pixels of an image,  $S_W$  will be singular so that the solution is no longer reliable. So we apply PCA before LDA to reduce the dimensionality to below the number of training samples.

To sum up these two recognition methods via dimension reduction, PCA seeks a projection upon which the total variance of the data will be maximized, so it does not take label information into account. FDA, on the other hand, is a supervised learning method, which provides a linear transformation of the data such that the classes are "best" separated.

### 3. Approach

The regression approach gains in discrimination among faces by forcing all the training images from each individual, after linear transformation  $B$ , to locate near one of the vertices of a regular simplex. The regression problem, due to more variables than the number of observations, is regularized by both  $L1$  norm (known as Lasso constraint) and  $L2$  norm (known as Ridge regression if applied alone). This problem can be turned into an equivalent regression problem with Lasso constraint only. Then the Least Angle Regression (LARS) algorithm is applied to solve for the transformation  $B$ .

#### 3.1. Regression

Linear regression with regularization is a classical statistical problem that aims to find a linear function that models the dependencies between variables  $x_{i=1}^n$  in  $\mathbf{R}^p$  and response variables (observations)  $y_{i=1}^n$  in  $\mathbf{R}$ . The classical way is the ordinary least square (OLS) method which minimizes the squared loss:

$$\sum_i (y_i - w^T x_i)^2$$

Due to limited training samples, the solution is not unique so the variance of the estimate  $w$  by OLS may be large so the estimate is not reliable. An effective way to overcome this problem is to penalize the norm of  $w$ , for instance, the  $L1$  norm and/or  $L2$  norm. Ridge regression would penalize  $L2$  norm alone. Solving this problem is easy, yet the solution is not sparse at all. In the situation that  $p \gg n$ , we sometimes hope for a sparser solution which can explain what are the most significant variables for the problem. Regression with Lasso constraint would provide sparse solutions via the  $L1$  norm penalization. Elastic net regression takes both penalization norms into the optimization objective, and the advantage of it is that the number

of nonzero coefficients of the solution could exceed  $n$ , the number of training examples.

The naive elastic net estimator is defined as:

$$\hat{\beta} = \operatorname{argmin}_{\beta} \|y - X\beta\|_2^2 + \lambda(\sigma\|\beta\|_1 + (1 - \sigma)\|\beta\|_2^2)$$

where  $\|\beta\|_1 = \sum_{i=1}^p |\beta_i|$  is the sum of the absolute value of all coefficients, and  $\|\beta\|_2^2 = \sum_{i=1}^p \beta_i^2$ .  $\lambda$  controls the overall weight of penalization, and  $\sigma$  adjusts the weight between  $L1$  and  $L2$  norms.

#### 3.2. Least Angle Regression(LARS)

The least angle regression selection algorithm proposed by Efron et. al is aiming at selecting a parsimonious set for the efficient prediction of a response variable. It is a useful and less greedy version of the traditional forward selection methods, and much computationally easier than the forward stagewise method. We will explain two easy but important methods, i.e., the forward selection regression and the forward stagewise regression, before introducing the LARS idea and algorithm.

In a typical Lasso regression problem:

$$\begin{aligned} & \operatorname{Min}_{\beta} \{ \|y - X\beta\| \} \\ & \text{S.t. } \|\beta\|_1 < t \end{aligned}$$

Each row of  $X$  stands for an example, and we call columns of  $X$  the regressors, or predictors, i.e.,  $X_{n \times p} = (x_1, x_2, \dots, x_p)$ . If it is required that the solution can at most have  $m$  nonzero coefficients, then only  $m$  regressors can be chosen to fit  $X\beta$  to  $y$ . In the traditional forward selection regression, given a collection of possible predictors, we select the one having largest absolute correlation with the response  $y$ , say  $x_j$ , and perform simple linear regression of  $y$  on  $x_j$ . This leaves a residual vector orthogonal to  $x_j$ , now considered to be the response. After  $k$  steps this results in a set of  $k$  predictors that are then used in the usual way to construct a  $k$ -parameter linear model. Forward selection is an aggressive fitting technique that can be overly greedy, possibly eliminating at the second step useful predictors that happen to correlate with  $x_j$ .

The Forward stagewise regression, on the other hand, is a much more cautious version, which takes thousands of tiny steps as it moves forward a final model. In one iteration, it computes the correlation between the residual vector (response  $y$  minus the current estimate) and all the regressors, and the largest one corresponds to the current target regressor. But what makes it different from traditional forward selection is that forward stagewise would only take a tiny step forward along the target regressor, so the estimate only changes a little bit before next iteration. Despite time consumption, this method works better the greedy forward selection method.

The LARS algorithm is a stylized version of the Stage-wise procedure that uses a simple mathematical formula to accelerate the computation. It only requires the same number of steps as the Forward selection method, yet produces almost the same solutions as the Stagewise method. The LARS procedure works roughly as follows. First preprocess the data. By location and scale transformations we can always assume that the covariates have been standardized to have mean 0 and unit length (by subtracting the mean of the training samples and scale each regressor), and so the response also has mean 0, i.e.,

$$\sum_{i=1}^n y_i = 0, \sum_{i=1}^n x_{ij} = 0$$

$$\sum_{i=1}^n x_{ij}^2 = 1, \text{ for } j = 1, 2, \dots, m.$$

Let the current vector of regression coefficients be  $\hat{\beta} = (\hat{\beta}_1, \hat{\beta}_2, \dots, \hat{\beta}_p)^T$ , and it gives the current estimate of  $y$ ,

$$\hat{y} = X\hat{\beta} = \sum_{j=1}^p x_j \hat{\beta}_j$$

and the correlation between regressor  $x_i$  and the residual  $y - \hat{y}$  is  $c_i$ :

$$\hat{c}_i = x_i^T (y - \hat{y})$$

We start with all coefficients equal to zero, and find the regressor most correlated with the residual (which is  $y$  for the time being), say  $x_{j_1}$ . Then we take the largest step possible in this direction until some other regressor, say  $x_{j_2}$ , has as much correlation with the current residual as  $x_{j_1}$ . So then  $x_{j_2}$  is included into the set of active regressors. Instead of continuing along  $x_{j_1}$ , LARS proceeds in a direction equiangular between the two regressors until a third regressor  $x_{j_3}$  has the same correlation with the residual as the previous two. LARS then proceeds equiangularly between  $x_{j_1}, x_{j_2}, x_{j_3}$ , i.e., along the "least angle direction", and so on so forth.

The LARS algorithm is described in detail as follows:

Notation:

$\mu$ —current least angle direction (proceeding direction)

$\hat{y}$ —current estimate of  $y$

$\hat{c}$ —current correlation between regressors and the residual

$\hat{r}$ —the current proceeding step size

$\hat{y} = 0$ . For  $k=1:m$  (look for  $m$  best regressors):

- 1)  $\hat{c} = X^T(y - \hat{y}) = (\hat{c}_1, \hat{c}_2, \dots, \hat{c}_p)^T$
- 2)  $A = \{j : |\hat{c}_j| = \max(\hat{c}) = C\}$
- 3)  $X_A = [\dots x_j \dots, j \in A]$
- 4)  $\mu(k) = X_A w_A$   
 $(w_A = a_A G_A^{-1} 1_A)$   
 $a_A = (1_A^T G_A^{-1} 1_A)^{-\frac{1}{2}}$   
 $G_A = X_A^T X_A, 1_A = (1, \dots, 1)^T$
- 5)  $a = X^T \mu = (a_1, a_2, \dots, a_p)$
- 6)  $\hat{r} = \min_{j \in A^c}^+ \left\{ \frac{C - \hat{c}_j}{a_A - a_j}, \frac{C + \hat{c}_j}{a_A + a_j} \right\}$
- 7)  $\hat{y}(k) = \hat{y}(k-1) + \hat{r} \mu(k)$

Proof for the equiangular vector  $\mu$  (step 4):

Proof: In order that  $\mu$  is the equiangular vector of the active regressor set  $X_A$ ,  $X_A \mu$  should a vector of equal numbers, say  $a_A 1_A$ .

$$\begin{aligned} X_A^T \mu &= X_A^T X_A w_A \\ &= G_A w_A \\ &= a_A 1_A \end{aligned}$$

then,  $w_A = a_A G_A^{-1} 1_A$ .

What left now is to show that  $a_A = (1_A^T G_A^{-1} 1_A)^{-\frac{1}{2}}$ . Since  $\mu$  is required to be unit norm,

$$\begin{aligned} \mu^T \mu &= w_A^T X_A^T X_A w_A \\ &= (a_A G_A^{-1} 1_A)^T G_A a_A G_A^{-1} 1_A \\ &= a_A^2 1_A^T G_A^{-1} 1_A \\ &= 1 \end{aligned}$$

Therefore,  $a_A = (1_A^T G_A^{-1} 1_A)^{-\frac{1}{2}}$ . (end of proof)

Proof for the proceeding step size  $\hat{r}$  (step 6):

Proof: During the  $k$ th iteration, we get the equiangular vector  $\mu(k)$ , and let

$$\hat{y}(k+1, r) = \hat{y}(k) + r \mu(k)$$

And we want to solve for  $\hat{r}$ .

The corresponding correlation of regressor  $x_j$  and the residual  $(y - \hat{y}(k+1, r))$  is:

$$\begin{aligned} c_j(r) &= x_j^T (y - \hat{y}(k+1, r)) \\ &= x_j^T (y - \hat{y}(k)) - x_j^T (r \cdot \mu(k)) \\ &= \hat{c}_j - r \cdot a_j \end{aligned}$$

When  $r = 0$ , for  $j \in A$ ,  $c_j(0) = C$  would give the largest correlation. As  $r$  increases, for  $j \in A$ ,  $c_j(r) = C - r \cdot a_A$ ,

a regular 4-simplex

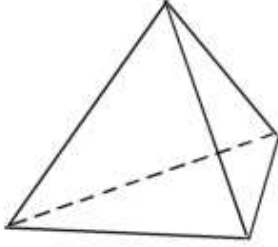


Figure 1. Illustration of a regular 4-simplex.

and for  $j \notin A$ ,  $c_j(r) = \hat{c}_j - r \cdot a_j$ . According to the algorithm, the estimate proceeds in a the equiangular direction among the current active regressors until a new regressor has the same correlation with the residual as the active regressors. So  $\hat{r}$  shall be found from the  $r$ 's such that  $C - r \cdot a_A = \hat{c}_j - r \cdot a_j$ , satisfying  $\hat{r}$  is the smallest positive among them. If we take the negative direction of all the regressors into account,  $r$ 's can also be found from such that  $C - r \cdot a_A = -\hat{c}_j + r \cdot a_j$ . Therefore,  $\hat{r} = \min_{j \in A^c}^+ \left\{ \frac{C - \hat{c}_j}{a_A - a_j}, \frac{C + \hat{c}_j}{a_A + a_j} \right\}$ . (end of proof)

### 3.3. Regular simplex

In this section we generate the regular  $m$ -simplex as the response variables for  $m$  classes of human faces. A regular  $m$ -simplex, is a geometrical entry sitting in the  $(m - 1)$  space, who has  $m$  vertices and the distance between any two vertices is 1. For instance, the 3-simplex is an equilateral triangle with unit edge in  $\mathbf{R}^2$ , and a 4-simplex will be a tetrahedron in  $\mathbf{R}^3$  such as Fig.1.

The following algorithm produces the vertices of one regular  $m$ -simplex:

1) If  $m < 2$ , no such simplex exists.

2) Else if  $m = 2$ ,  $v_1 = (1)$ ,  $v_2 = (0)$ , and  $V = \begin{pmatrix} v_1^T \\ \vdots \\ v_m^T \end{pmatrix}$

3) Else for  $K = 3 : m$

Increase the dimension of all vertices by 1 via adding a column of all zeros to the right of vertex matrix  $V$ . Compute the mean of the rows of  $V$ , and denote it as  $v_c$ . Let  $v_p$  be the vector perpendicular to all the row vectors in  $V$ , i.e.,  $v_p = (0, \dots, 0, 1)$ . The  $k$ th new vertex for the regular  $K$ -simplex will be  $v(k) = v_c + \sqrt{\frac{K}{2(K-1)}} v_p$ .

It is easy to prove that the  $m$  vertices generated in this manner will satisfy the properties of the regular  $m$ -simplex.

### 3.4. Elastic Net as a Lasso regression problem

By simple linear algebra we can show that the Elastic Net regression problem:

$$\begin{aligned} \hat{\beta} &= \operatorname{argmin}_{\beta} \{ \|y - X\beta\|_2^2 + \lambda(\sigma\|\beta\|_1 + (1 - \sigma)\|\beta\|_2^2) \} \\ &= \operatorname{argmin}_{\beta} \{ \|y - X\beta\|_2^2 + \lambda_1\|\beta\|_1 + \lambda_2\|\beta\|_2^2 \} \end{aligned}$$

is equivalent to the Lasso regression problem:

$$\beta^* = \operatorname{argmin}_{\beta} \{ \|\tilde{y} - \tilde{X}\beta\|^2 + \lambda_1^*\|\beta\|_1 \}$$

Where,

$$\begin{aligned} \tilde{y} &= \begin{pmatrix} y \\ O_p \end{pmatrix} \\ \tilde{X} &= (1 + \lambda_2)^{-1/2} \begin{pmatrix} X \\ \sqrt{\lambda_2} I_p \end{pmatrix} \\ \lambda_1^* &= \frac{\lambda_1}{\sqrt{1 + \lambda_2}} \\ \beta^* &= \sqrt{1 + \lambda_2} \hat{\beta}. \end{aligned}$$

After the transform, LARS algorithm can be used to efficiently solve for the Elastic Net regression problem, and thus is named as LARS-EN.

### 3.5. Flow chart of LARS-EN in face recognition

The overall algorithm of LARS-EN regression method for face recognition is summarized in the flow chart in Fig.2.

## 4. Experiments

We conduct the experiments on the following two databases: 1. The Yale Face Database B, from which we select 230 face images of 10 human subjects (For each subject, 8 faces for training, 4 faces for validation and 11 for recognition test). 2. Manually collected Princetonian Album from the internet, consisting of 50 face images of 10 human subjects (no validation images, leave-one-out method applied).

### 4.1. Preprocessing

All test image data used in the experiments will be manually aligned, cropped, and then re-sized to appropriate size. We use three different preprocessing methods,

1) horizontally adjust two eyes then crop out the face area then re-size to the same size(zero borders will be

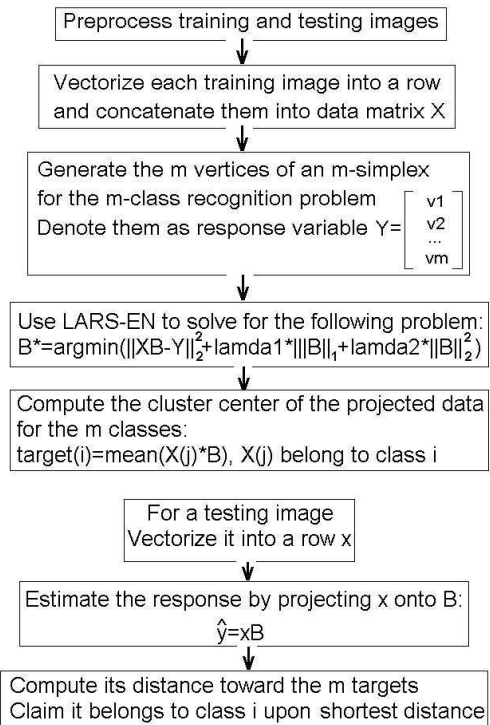


Figure 2. Flow chart of the LARS-EN algorithm in face recognition. The first part is the training phase, and the second part is the recognition phase. For PCA/Lars, Fisher/Lars, etc. methods which will be discussed later, simply transform the data matrix  $X$  by PCA and/or FDA after the second step in the flow chart. During recognition, apply the same transformation to the new data before projection.

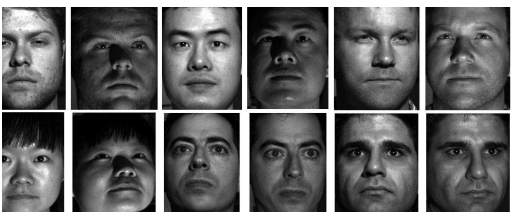


Figure 3. Examples of faces in Yale Face Database B, in different front poses and light conditions, and same facial expression

padding due to different shapes of faces), shown in Fig.5;

2) locate the two eyes in each image then scale all the image such that the location of two eyes are the same for all, and select same number of pixels for each image; Some result images are shown in Fig.6.



Figure 4. Examples of faces in Princetonian Database. Different facial expressions and various environments included, such as indoors, outdoors, day-time, night-time.



Figure 5. Illustration of preprocess method 1 (first training face for each individual). It includes more information of face borders than method 2. However, facial features are not well aligned with each other. Each image has dimension 60 by 40.



Figure 6. Illustration of preprocess method 2, i.e., with two eyes aligned (first training face for each individual). Each image has dimension 35 by 29.



Figure 7. Illustration of preprocess method 3, i.e., histogram equalization followed (first training face for each individual). Each image has dimension 35 by 29.

3) apply histogram equalization after method 2) to reduce the variance of the data introduced by light condition, shown in Fig.7.

## 4.2. Tested methods

For the two databases and the three preprocessing methods, we test the performance of LARS-EN and compare it with techniques such as PCA and FDA, for different dimensions. Since LARS-EN has an underlying assumption that a sparse set of covariates is significant for the response variable (here it is the class of a face), it would be interesting then to look for a set of pixels most important for the recognition problem. If it works for a certain database, then we can evaluate the performance versus the number of pixels allowed (we expect an increase in recognition performance as the number goes up).

Other experiments include applying LARS-EN to the linearly transformed data (such as PCA, FDA, etc.) As we can see for Princetonian Database, LARS-EN directly on pixel domain does not do a good job, because we believe that the face data is not represented sparsely enough in that database. If a certain transformation can render more sparsity to the data for the regression problem, then LARS-EN should work even better. So the problem turns to how to find a sparse representation of the data for the particular regression problem. In this paper we only consider linear transformation of the data, and divide them into two categories, data-dependent transforms, such as PCA and/or FDA, and data-independent transforms, such as discrete Fourier Transform (DFT). From the results of experiments LARS-EN works gracefully by automatically picking out the "best" regressors from the associated transform vectors. The amazing ability of this method is that it can offer a perspective of the good projections with respect to certain discrimination problem, and can freely choose interesting regressors from different transforms to better predict the response. The coefficients will then have weights for these regressors, higher meaning more important. The estimate  $\hat{y}$ , then, is a linear combination of those "interesting" regressors produced from various discriminative projections. This property could be associated with the boosting idea in machine learning.

## 4.3. Parameter selection

For the LARS-EN algorithm, two parameters  $\lambda$  and  $\sigma$  need to be addressed via validation. We use the validation set from the Yale Face Database B to run LARS-EN on pixel domain and select a good set of parameters ( $\lambda = 0.08, \sigma = 0.7$ ). Then we apply this parameter set to all the recognition via regression problems.

## 4.4. Princetonian Database

Compared with the Yale Face Database B, this manually collected database has a lot fewer total images for each

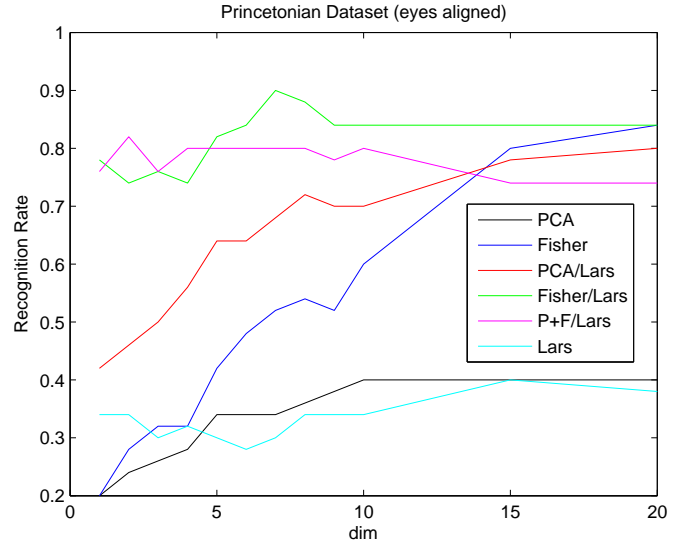


Figure 8. Recognition performance of 6 methods as dimension increases (applying the second preprocessing method). Lars directly in pixel domain does not perform well in this dataset, presumably the representation of face in pixel domain is not sparse enough for the recognition problem. Due to various expressions, the meaning of a certain pixel is unstable, thus bringing difficulty to a sparse representation. However, Lars following linear transforms such as PCA, Fisher, or both do a better job, especially around dimension 7 for Fisher/Lars and P+F/Lars. A good result at merely dimension 1 for them is rather surprising. The later-on decrease of performance might be due to overfitting. The performance of PCA/Lars is improving quite steadily as dimension grows. At dimension  $k$ , there are exactly  $k$  eigenvectors chosen to project the data before running the regression. Lastly, Fisher does a significantly better job than PCA, as expected.

person (i.e., 5 images /person) Therefore, we use "Leave-one-out" method to conduct training and testing. To reduce the unnecessary variance of the data, all the images are pre-processed by fixing the locations of two eyes (preprocess method 2).

Six experiments, which are PCA, multi-class FDA, LARS directly on pixel domain, LARS following PCA (denoted as PCA/Lars), LARS following FDA (denoted as FDA/Lars), and LARS following a combination of PCA and FDA (denoted as P+F/Lars), are conducted in the "Leave-one-out" manner. In the P+F/Lars method, it first includes regressors obtained both from PCA transform and FDA transform, then let Lars choose the "best" regressors from the pool of regressors. The result of the experiments is shown as Fig.8.

Another set of experiments is conducted after histogram equalization of all images (preprocess method 3). The comparison of two preprocessing results is briefly shown in

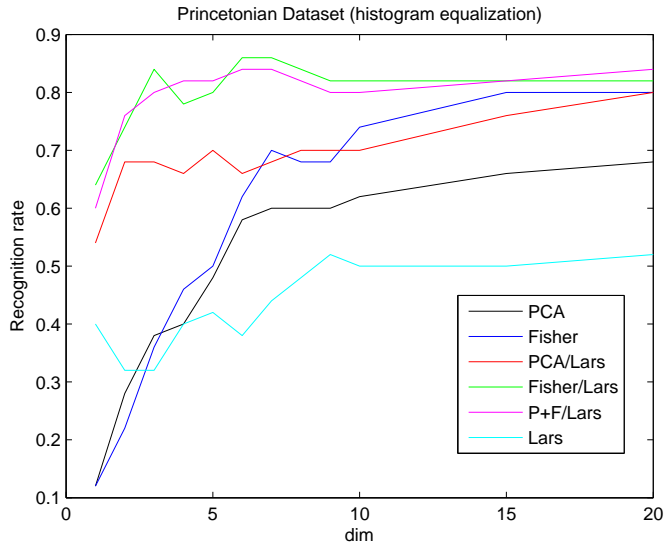


Figure 9. Recognition performance of 6 methods as dimension increases (applying the third preprocessing method). The overall performance is better than preprocess method 2, especially the PCA method. It is reasonable because PCA seeks for largest variance projection, and histogram equalization would decrease the unnecessary variance brought by different lighting conditions. Again, Lars regressions following transformations produce good results.



Figure 10. Illustration of preprocess method 2 versus 3 on Princetonian Dataset. Histogram equalization significantly helps reduce the large light condition variances in the original images. Each image has dimension 30 by 25.

Fig.10, and the result of experiments is shown in Fig.9.

#### 4.5. Yale Face Database B

The original database contains 5760 single light source images of 10 subjects each seen under 576 viewing conditions (9 poses x 64 illumination conditions). For the course project we choose 230 images of the 10 subjects under 23 viewing conditions (4 poses x 6 illumination conditions, 1 image gets deleted due to total blackness probably because of bad shot). Some image examples are shown in Fig.4. For each subject, 8 images are used for training, 4 for validation, and the rest 11 for testing. Validation images are used in determining the two weight parameters of the

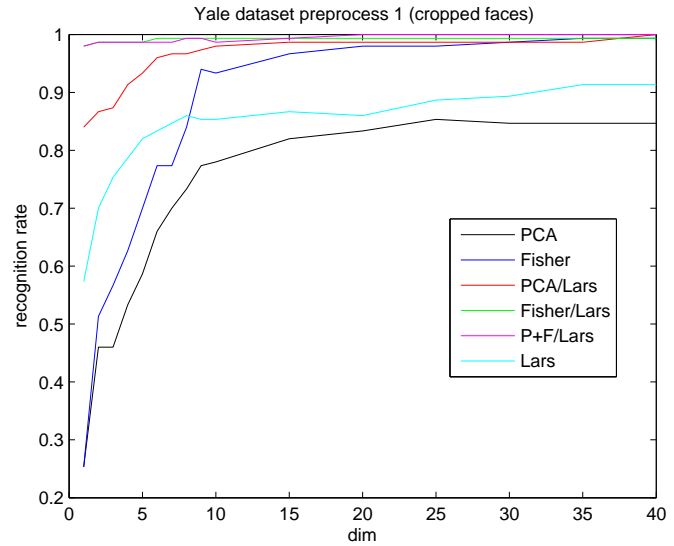


Figure 11. Recognition performance of 6 methods as dimension increases (applying the first preprocessing method). Lars algorithm directly in pixel domain performs nicely, although not excellently, for all three preprocessing cases. The accuracy exceeds 90% for each case at dimension 40. Fisher/Lars and P+F/Lars beat other methods in this case.

regression problem. Once the two weights are picked, we fix them for all the regression experiments conducted.

The six experiments are again applied to differently pre-processed Yale Face Database. The results for reduced dimension  $dim = 1 : 40$  are shown in Fig.11, Fig.12, and Fig.13.

### 5. Conclusion and Analysis

In Section 4, the performances of the proposed LARS-EN, PCA/Lars, Fisher/Lars and P+F/Lars are evaluated with the other two popular face recognition techniques (Eigenfaces and Fisherfaces) in the two face databases. The Fisher/Lars, P+F/Lars and PCA/Lars work well for all the experiments. Lars-EN conducted directly on the pixel domain, on the other hand, is not satisfactory for all datasets. It might work well when images in the dataset have less irrelevant variance so that a stable sparse set of pixels can be relied on for regression. For example, the Yale Face Database B with the third preprocess method is a successful application of Lars-EN in pixel domain. Now let us illustrate the selected pixels via the algorithm and see what they might convey.

From Fig.14, the selected pixels appear to be distributed near two eyes and the mouth. It makes sense because these areas contain important facial characteristics of individuals.

Another strength of Lars-EN following linear trans-

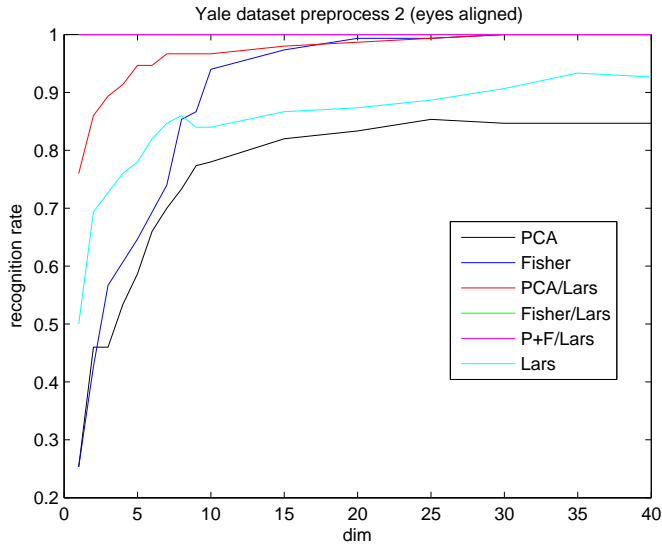


Figure 12. Recognition performance of 6 methods as dimension increases (applying the second preprocessing method). Again Fisher/Lars and P+F/Lars beat other methods in this case. The recognition accuracy for these two methods is 100% from dimension 1. PCA/Lars reaches rate 100% slightly before dimension 30.

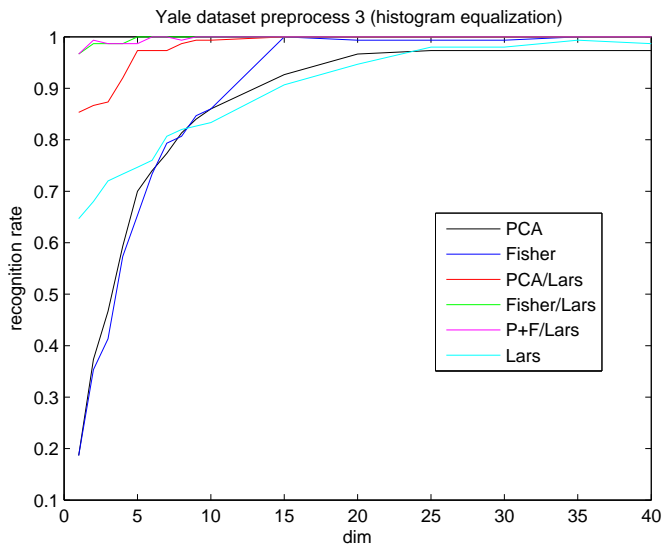


Figure 13. Recognition performance of 6 methods as dimension increases (applying the third preprocessing method). Again we experience a boosted recognition ability of PCA after histogram equalization. Also, several methods reach 100% correctness at an earlier stage (around dimension 13).

forms, is that it can serve as an automatic mechanism of selecting best projecting vectors from the different linear transforms. For instance, in PCA/Lars, we find that the first several projecting vectors are not necessarily the eigenvectors associated with the largest eigenvalues

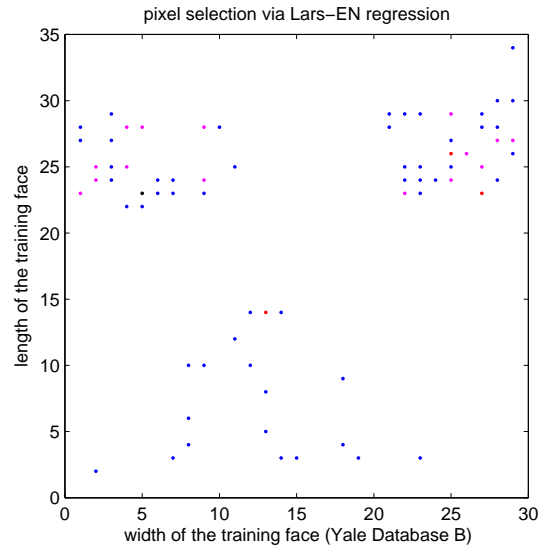


Figure 14. The pixels in the face image that get selected no less than 2 times. Black-5 times; Red-4 times; Magenta-3 times; Blue-2 times. Total 72 pixels shown. Clearly, the eye area and mouth area are considered "important" features in Lars-EN algorithm.

(For details of PCA, please refer to Section 2.1), which means the projection directions which will provide largest variance of the data are not the guaranteed to be the best discriminative directions. Since PCA/Lars automatically selects the directions for us, we have tried using these directions instead in the PCA algorithm. And the recognition performance does get enhanced.

Thirdly, the gracefulness of Lars-EN by automatically picking out the "best" regressors from the associated transform vectors inspires us to include different kinds of discriminative transformations into the same training phase, and let Lars-EN to choose interesting regressors caused by certain transforms, such as the P+F/Lars method in this paper. The coefficients will then have weights for transform vectors, higher meaning more important. The estimate  $\hat{y}$ , then, is a linear combination of those "interesting" regressors produced from various discriminative projections. Usually the recognition accuracy gets boosted in this way. This property could be associated with the boosting idea in machine learning.

Future research of face recognition with this regression technique involves 1) looking for and testing other data-independent (e.g. DFT, Wavelets, etc.) transforms that help provide sparseness for regression; 2) better preprocessing methods to the images (Histogram equalization in our experiments is easy but not perfect, because it introduces noises. c.f. Fig.10 and Fig.7); 3) Vary the number of training examples and see how the recognition performance

of Lars series methods evolves with respect to that.

In conclusion, the Lars-EN series regression methods for face recognition is innovative and useful. The idea of Lars following transforms of data to increase the sparseness of the dataset is for the first time introduced and tested. It works very well in the two face databases, and has the potential of performing even better with new transforms found and added. This will be one of the focuses in future research. Also, this idea could also be naturally applied toward other recognition problems.

## 6. References

1. Efron, B., Hastie, T., Johnstone, I.M. and Tibshirani, R.: Least Angle Regression. 2002
2. Clemmensen H., Gomez D., Ersboll K.: Individual discriminative face recognition models based on subsets of features. SCIA 2007, LNCS 4522, pp. 61-71, 2007.
3. Senjian An, Wanquan Liu, Svetha Venkatesh: Face recognition using kernel ridge regression. CVPR 2007.
4. Max Wemng, University of Toronto, a note on Fisher Linear Discriminant Analysis.
5. S. W. Joo, Face recognition using PCA and FDA with intensity normalization.
6. F. Lazebnik, On a Regular Simplex in  $\mathbf{R}^n$ , 2006
7. P. Bellhumeur, et. al., Eigenfaces vs. Fisher faces: Recognition using class specific linear projection, IEEE Transactions on PAMI, 1997
8. H. Zou and T. Hastie, Regularization and variable selection via the elastic net. J. R. Statist. Soc. B 67(Part 2) (2005) 301-320