Face Recognition Using Dimensionality Reduction

Yamin Tun **Name**
Department of Computer Science Affiliation
ytun@umass.edu *email*

1 Introduction

Face recognition has been an interesting research area that can be useful in security, customer electronics and social media. Before, face recognition was used only for military purpose such as security and surveillance applications. However, for the last few decades, face recognition has gained a lot of interest especially after the emergence of a number of novel computer vision applications such as robotics, augmented reality and personal devices such as computational glasses. However, there are a number of challenges in face recognition problem due to its high dimensionality, occlusion, lighting, etc.

In this report, three linear dimensionality reduction techniques, NMF, ICA and PCA, were used for face recognition problem using Label Faces in the Wild (LFW) dataset [1]. The experimental results show that PCA and ICA perform at reasonably high accuracy and similar training and testing time while the SVM with NMF takes about three times as long to train as the SVMs with PCA and ICA despite its short duration to decompose. PCA achieves 85% precision while ICA achieves 84% precision NMF performs the worst out of all models in terms of accuracy with the precision of 65%.

2 Related Work

Within the last decades, several algorithms were proposed to tackle face recognition problems.

In 1997, Dr. Belhumeur, Dr. Hespanha, and Dr. Kriegman proposed FisherFace algorithm. [2][3] FisherFace is often compared with another well-known algorithm named eigenface. Eigenface technique focuses on obtaining least square error for representing the data in a lower dimension while FisherFace's goal is more towards the classification aspect. FisherFace algorithm places the sample vectors of the same class in a single spot of feature representation as far away from other different classes as possible using discriminant analysis (DA). [2] This gives the advantage of invariance in lighting and facial expression while Eigenface might suffer from classifying images in the same class with different lightings as different classes. Using the observation that the face is not a Lambertian surface, therefore causing unexpected shadows and specularties and facial expressions, the face data were projected linearly from high dimension to low dimension subspace that is insensitive to such changes. The assumption that the data in each class is normally distributed is made. [2][3]

Ensemble classifiers such as random forests were also used in face recognition. The paper proposed Random forest is composed of multiple decision trees each of which vote on the predicted class. The forest selects the class with the most number of votes. This method emphasize the importance of automated robust face recognition. The results demonstrate that the proposed random-forest approach performs better than the SVM as well as bagging support, AdaBoost decision classifiers. [4]

Another method of face recognition is learning a dictionary for sparse representation of face. The benefit of this method is that it is robust towards the occlusion since it detects separate parts of the face instead of generalizing the entire face as other methods like fisherface and eigenface do. The author uses sparse representation theory to choose the training images to maximize robustness and occlusion. [5] Similar to such dimensionality reduction approach, the proposed solution in this report is three other approaches to dimensionality reduction for face recognition.

3 Proposed Solution

The challenge with image classification problems in general is its high dimensionality. The concept of Dimensionality Reduction is to project high dimensional data to a lower dimensional space by maintaining most of its information. [6] The coding was completed using scikit-learn package and Viola-Jones matlab vision toolbox. [7][8] The details of the pipeline for face recognition are described below. Three selected techniques of linear dimensionality reduction are Principle Component Analysis (PCA), Non-negative matrix factorization (NMF) and Independent Component Analysis (ICA). The first step is to project data to lower dimension using these three approaches and then feed in the resulting lower-dimensional features into Support Vector Classifier. The choice of whether to use linear kernel or rbf kernel is mentioned in section 5.1.

3.1 Pipeline for Face Recognition Using Dimensionality Reduction

1. Create a face-aligned dataset
   a. Load Labeled Faces in the Wild dataset
   b. Slice the centered face area in each image and resize as a 50x37 image
2. Hyper-parameter Selection
   a. Split training and testing sets using 4-fold stratified cross-validation (test ratio=0.25)
   b. Select hyper-parameter of SVM using 3-fold cross-validation
3. Dimensionality Reduction (Details in section 3.2)
4. Training SVM using optimal parameters obtained from cross-validation
5. Testing using aligned-face dataset
6. Performance Evaluation in terms of precision, recall, f1-score, training/testing time
7. Testing model in the wild automatically
   a. Select 3 random images of George W. Bush from the web [9][10][11]
   b. Detect face in each image using Viola-Jones and crop face area
   c. Predict the names of the person in the face image using each model

The purpose of the last step of testing in the wild is to see how the whole face recognition pipeline can be automated from scratch, as well as to test for cases when the faces might not centered well with some disturbance of background as shown in the second and third face images of figure 1b.

3.2 Details of Linear Dimensionality Reduction Models

The problem of linear dimensionality reduction is to compute basis B that is composed of lower-dimensional components and weights of basis Z for a given dataset X such that sum of squared errors is minimized.

$$\arg\min_{\mathbf{Z},\mathbf{B}} ||\mathbf{X} - \mathbf{ZB}||_F \quad \text{Eq 1 [6]}$$

3.2.1 PCA

The concept of PCA in face recognition is that the variance information of faces in data could be encoded in k eigen face components where k<n, assuming that faces are more or less aligned. First, mean image and covariance matrix S was computed as shown in figure 1b (Eq 2, Eq 3). Eigen values and eigen vectors of S were calculated. After eigen vectors were sorted according to eigen values, top k eigen vectors were chosen. The k principle components of a given image x can be computed using Eq 4.[6]

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \text{Eq2,} \qquad S = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T \quad \text{Eq3,} \qquad y = W^T(x - \mu) \quad \text{Eq4[6]}$$

where x=image vector, μ=mean image, λ =eigen values, v=eigen vectors, W=(v1, v2, …, vk)

3.2.2 Non-Negative Matrix Factorization (NMF)

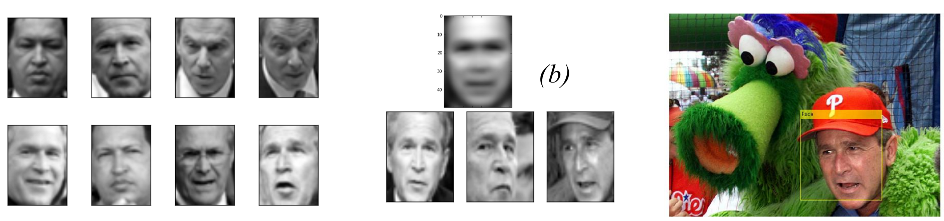
NMF is a type of linear dimensionality reduction where basis B and weights of basis Z in equation 1 are constrained to be non-negative. Intuitively, NMF components might feature non-negative face parts, with each of them being weighted positive to result in a complex image. [2]

3.2.2 Independent Component Analysis (ICA)

ICA is another linear dimensionality reduction technique where random variables (components of Z in equation 1) that represent a given vector must be independent of each other. [6]

4 Data Set

For training and testing models, a subset of Label Faces in the Wild (LFW) dataset with minimum of 70 data cases was used [1]. The dataset consists of 1,288 grayscale face images of 7 public figures with the resolution of 50x37. Each entry in feature vector of 1850 before dimensionality reduction represents each pixel value in a given image ranging from 0 to 1. The target names are Ariel Sharon, Colin Powell, Donald Rumsfeld, George W Bush, Gerhard Schroeder, Hugo Chavez and Tony Blair.



(a) (c) (d)

Figure 1. (a) Sample images from LFW Dataset (b) Mean Image of dataset (c) Test image in the wild with the detected face area cropped (d) Face detected in bounding box using Viola- Jones

5 Experiments and Results

The experimental results of NMF, PCA and ICA are summarized below.

5.1 Experiment 1: Cross-validation

Cross-validation was performed not only for gamma and C of SVC to find optimal parameter values, but also for both linear and rbf kernels. Although linear kernel gives reasonable precision at about 71% on average of 3 models for 5 different runs of validation set, rbf kernel outperforms the linear SVM by approximately 5% on average. Considering that precision is more important than training time, rbf kernel was chosen to use for all three techniques. The best number of basis component k, 150, was picked by cross-validation after trying a number of values ranging from 50 to 200 with the interval of 10 for all 3 models. For fairness in comparing running time, the same k value that gives best performance for all models on average is used for all 3 models.

5.2 Experiment 2: Comparison of Different Models

It was observed that eigen faces get noiser as their eigen values get lower. As expected, the eigenfaces give a smooth-face like image with different highlights and contours in different parts of the face.

NMF components are similar to eigenfaces but more bold with strong edges. The first NMF component is very close to the mean image shown in figure 1b while the other face images look like different lighting results of facial features. From these NMF components, a given image can be constructed starting with the mean face of 7 subject data. Different area of the face is added more weights by adding the rest of the component. The IC images are more toned-down with attenuated facial features compared to eigenfaces and NMF components.

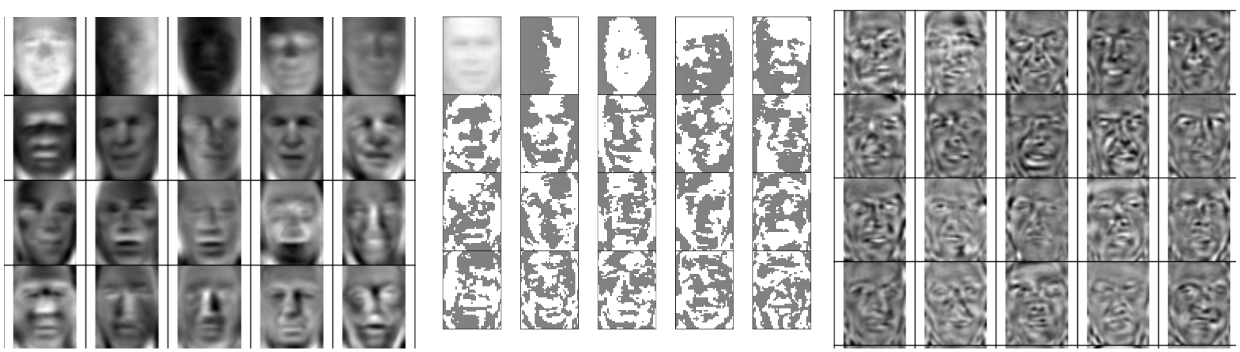


Figure 2. First 20 of decomposed components of LFW face dataset (a) PCA (Eigen faces), (b) NMF, (c) ICA [left to right]

Three models were compared in terms of precision, recall, F1-score, decomposing time, training time, testing time. Decomposing time is only the duration it takes to compute the lower-dimensional components of an image. Training time is the total time including decomposing, cross-validation, projecting and fitting. In terms of training and testing times, ICA, NMF and PCA perform close to each other except that NMF takes about three times as long as ICA and PCA for training as shown in figure 3. In terms of just the duration to compute the decomposed representations of the image, ICA takes the longest time, about 17 times as long as NMF and PCA. For decomposing, NMF takes slightly shorter time than PCA. (Note: Running time graph is in log space.) However, in terms of classification, NMF performs the worst with about 65% precision and about 62% recall and 63% F1-measure while PCA performs at 85% precision, 84% recall and 83% F1-measure, and ICA performs at 84% precision, 83% recall and 83% F1-measure. The precision, recall and F1 score are close to each other for all models. This means that all models makes relatively good number of correct classifications when, at the same time, there are few number of times when the model classifies an image in a wrong category.

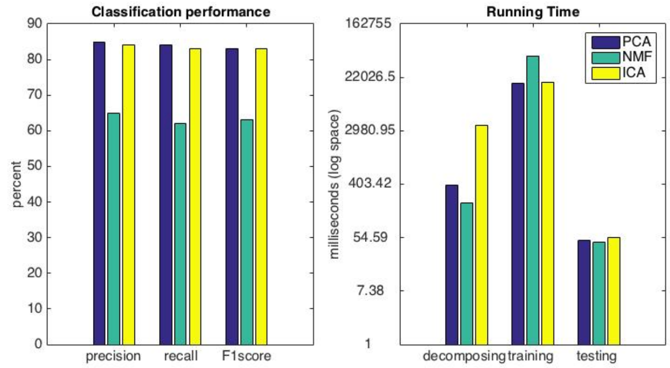


Figure 3. Performance Comparison of PCA, NMF and ICA in terms of accuracy and running time

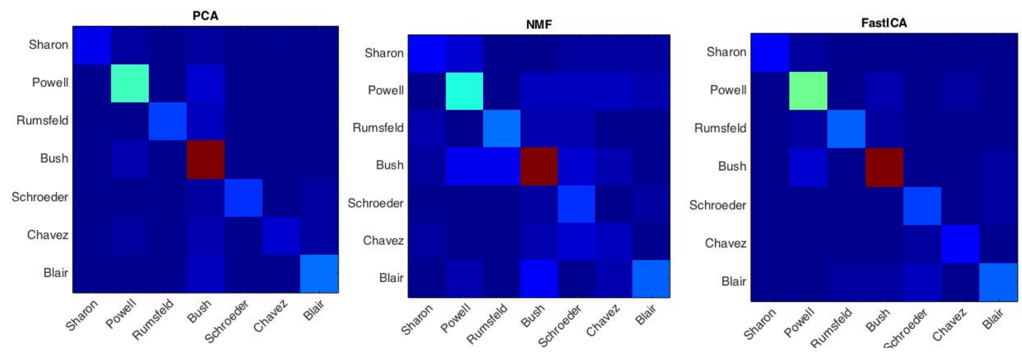


Figure 4. Confusion matrices of PCA, NMF and ICA [left to right]

The confusion matrix shows that George Bush and Collin Powell are mistakened frequently for PCA and ICA. For NFM, there were a lot of false positive and false negatives for almost all subjects.

5.2 Experiment 3: Testing Model in the Wild

ICA model classifies all 3 faces wrong while both NMF and PCA models classify all 3 faces correctly. One reason might be that the centerness of face and the rough alignment of facial features in the image are important for ICA. However, from such small number of test set, we cannot make any strong conclusion about each technique.

6 Discussion and Conclusions

Decomposing for ICA might take the longest than others possibly due to its strict constraints of being all decomposition vectors independent. However, the training time in total takes the longest for NMF.

In terms of accuracy, NMF is the lowest. There was some peculiarity in NMF components. Due to the non-negativity constraints that NMF components, it was expected to see each NMF component to be focused on individual facial feature. However, the NMF components are more face-like features with strong edges, which might not scale. This might be due to the misalignment and different rotation of faces since the dataset is from the wild. The fact that the components are not part-wise separable, might make it difficult to combine with non-negative weights to compose a given image. The testing set could be from the wild with varying rotations though the training set could be more standardized. Further improvement might be possible by normalizing the color values and rotating faces in the training set for better alignment before training process.

Out of 3 SVM models with 3 different linear dimensionality reduction techniques, both PCA and ICA perform similarly in terms of accuracy and running time, while NMF performs the least favorably in both accuracy and overall speed.

References

1. LFW Face Database : Main. (unknown date) Labeled Faces in the Wild Home. University of Massachusetts Amherst. Access on April 20, 2015
2. Aleix Martinez (2011), Scholarpedia. FisherFace. Scholarpedia, School of Computer Science, The University of Manchester, U.K.
3. Belhumeur, Peter N, Hespanha, Joao, P & Kriegman, David J., (1997) Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. IEEE on Pattern Analysis and Machine Intelligence, Volume 19 Issue 7, July 1997
4. Kouzani, A.Z. ; Deakin Univ., Geelong ; Nahavandi, S. ; Khoshmanesh, K., (2007)Face classification by a random forest, TENCON 2007 - 2007 IEEE Region 10 Conference, Oct. 30 2007-Nov. 2 2007
5. Yang, A.Y. ; Ganesh, A. ; Sastry, S.S. ; Yi Ma, (2008) Robust Face Recognition via Sparse Representation. Journal of Pattern Analysis and Machine . Volume:31 Issue:2
6. OpenCV: Face Recognition Tutorial (unknown date) Access on April 20, 2015
7. Scikit-Learn: Decomposition (unknown date). Access on April 20, 2015
8. Mathworks: CascadeObjectDector Toolbox (unknown datea). Access on April 20, 2015
9. Wikimedia. (Unknown author, date) *Image Citation,* Access on April 20, 2015
10. Business Insider (Unknown author, date) *Image Citation,* Access on April 20, 2015
11. Gstatic.com (Unknown author, date) *Image Citation,* Access on April 20, 2015