Auto-tagging The Facebook

Jonathan Michelson and Jorge Ortiz
Stanford University 2006
E-mail: JonMich@Stanford.edu, jorge.ortiz@stanford.com

Introduction

For those not familiar, The Facebook is an extremely vast social networking web service for college students. Users can present contact information, retain lists of friends, and upload images to a personal online album using the service, and can even "tag" regions of those pictures where their friends appear, essentially putting names to all the faces. Many users have uploaded hundreds, sometimes even thousands, of fully tagged pictures of themselves and others. However, the tagging process is tedious and time-consuming, and as such our goal was to implement a machine learning algorithm for quickly and automatically tagging newly uploaded pictures.

To develop the automated tagging feature, our implementation was essentially divided into three parts:
1) Data Acquisition
2) Face Detection
3) Face Recognition

We will therefore address each part in turn. We note that, while our motivating example is The Facebook, the methods for detection and recognition used here could be applied to any area requiring object detection and recognition, which is the reason so much research is being conducted in this field.

Data Acquisition

Though there are many online face databases, most focus on specific features of face recognition (e.g. lighting, angle, etc.), and almost none contain the portion of the face we were interested in. Furthermore, these databases are usually produces from images taken in strict, controlled environments, and do not provide the range of faces we expect to encounter. Therefore, the decision was made to create our own Face Database, which we accomplished by extracting thousands of faces from images off the internet. A further benefit of this choice was that most pictures on the web exhibit a huge variety of poses, lighting conditions, and people in general, presenting both a greater challenge in terms of identification but also a greater variance with which to train data.

Face Detection

Method

It was decided from the outset that based on the variance and complexity of our dataset (every image is a 400-dimensional vector, and each pixel is a 256-dimensional multinomial), that a generative model would be more appropriate for the task of face detection. Previous research in the area supported this conclusion, and a Mixture of Factor Analyzers approach was taken. While a rigorous mathematical treatment of the subject can be found in [2], only an extremely simplified understanding is presented here.

Standard factor analysis basically proposes that a set of vectors can actually be reduced to a linear transformation of a much smaller random vector, plus some noise. The mixture of factor analyzers model, on the other hand, supposes that there are several of these latent vectors, each with a different mean, that contribute with certain

probabilities to the final images. While at first this may seem extremely complex computationally, an efficient EM algorithm exists for solving the problem [2].

Data Set

For our face detection algorithm we first manually cropped 850 images of faces from our Face Database, such that the smallest square containing both the eyes and mouth was used. We then took each face, and randomly rotated it by up to 15 degrees and zoomed in our out by a random factor of up to 20%. We did this five times per image, then mirrored the image and repeated, for a total of 10 images per face. Finally, we scaled the image down to a 20x20 image and loaded it into our Face Matrix. In total, we ended up with 8939 faces[1].

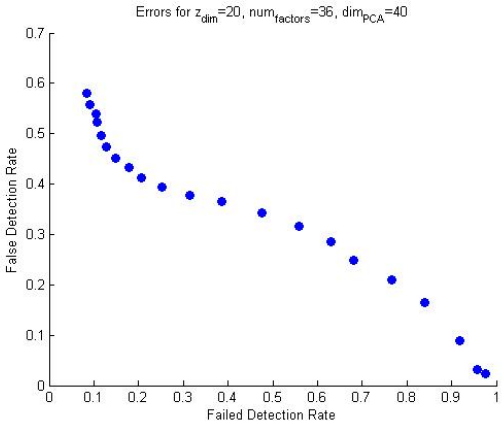
We then derived non-face images by taking 17 extremely large images of random objects, such as mountains, cars, maps, and horses, then taking random section of random sizes from those images and applying similar transformations to them as we did to our face images. In this way we generated a sample of 11922 non-face images.

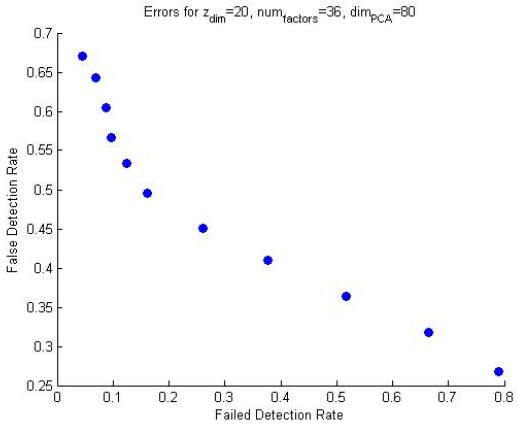
Finally, we broke our data up into training and testing samples for both faces and non-faces. The Training / Testing splits were 8420 / 519 and 3268 / 1000 for faces and non-faces, respectively[2].

Experimental Results

On our first attempt, we began our training with the same set of specifications as in [1]. First, we used PCA to reduce each image from 400 dimensions to 80. Following that, we tested the mixture of factor analyzer model with 36 factors and a latent dimension of 20. Finally, we tested this model on the training set of non-faces to seek a threshold value for the log-likelihood, such that any new vector tested could be classified based on whether it satisfied the threshold or not. This method failed.

Despite thorough testing with almost every combination of the parameters, no setting could be found that achieved acceptable results; we inevitably ended up with the same threshold trade-off curve, and though setting the PCA dimension reduction to 40 instead of 80 pushed the curve as far down as it would go, no threshold setting achieved an adequate success rate, as even trying to keep the false detection rate below 40% resulted in 20% missed faces, which would be unacceptable for our application. Our second approach proved far more successful.





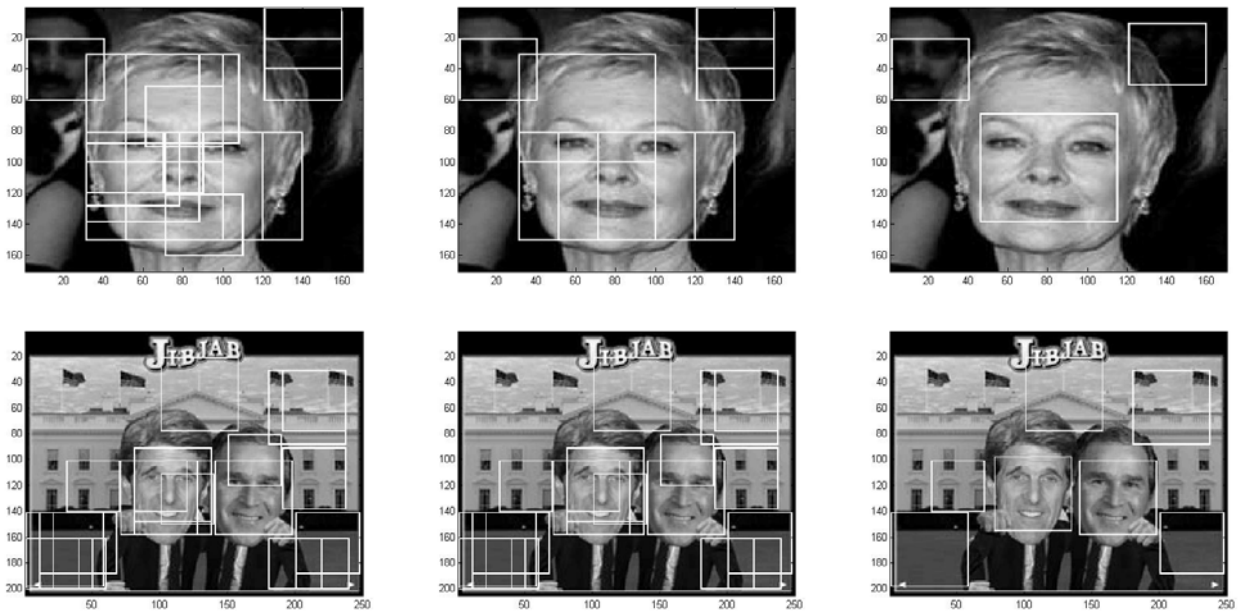
[1] In our post-processing step, images that were not square or could not be adjusted to 20x20 were discarded
[2] We randomly chose 1000 images from the remaining 8654 to examine

For our second attempt, an approach similar to that taken with Naïve Bayes was used, wherein we decided to fit a mixture of factor analyzers not only to the Faces set, but also to the Non-Faces set. Once given a new vector, we would ask both models what the likelihood of their generating that vector is, and classify based on which is higher. At its best, this method yielded a false detection rate of 8.9%, with a successful face detection rate of 88%. This was achieved with a PCA reduction to a 60-dimensional space, followed by a mixture of 36 Gaussians each with latent dimensionality 10.

Though initially it proved extremely slow to test the log-likelihoods of a new vector, it was discovered that the majority of the processing time went into matrix inversion and determinant computation – both operations that were independent of the vector being tested. Therefore those values were pre-computed, and calculating the log likelihood of a new vector was cut down to a 0.0019 second process[3].

Given the relatively high success rate, a new image was processed in the following way: A square of varying sizes (based on the input image size) scrolled[4] over the entire image, testing each sub-image it came across. Once completing the scan, we multiplied the square by 1.2 and repeated the process, continuing to do so until the square became over 5 times its original size. We note before proceeding that this is by far and away slower than the actual testing process, and for actual application a superior method of examining the photo would be needed to have auto-tagging work in an acceptable amount of time.

Once a scan completed and log likelihoods determined, sub-images that our algorithm classified as "faces" were clustered together by first conjoining any overlapping sub-images of different sizes, then clustering any overlapping images of like sizes. The results are best demonstrated with examples:



[3] Tested by dividing the time it took to calculate 10000 log likelihoods of vectors drawn from both our Face and non-Face databases

[4] The scrolling was done in a deterministic, iterative process, where each iteration involved shifting the square either ¼ of its size to the right or downward, eventually covering the entire image.

The face detection algorithm at work; for Judy Dench's image the algorithm seems to be successful, while for Bush and Kerry we note the damage that false positives could cause.

Comparison with previous work

Yang, Kriegman and Ahuja (2000) claimed to experience superior success with a similar algorithm, one that employed a threshold technique instead of a classifier comparison. However, it is important to note that both their training set and their testing set involved almost perfectly aligned faces, while our database, drawn from "real" images, involved faces with significantly greater variance. This distinction is further exaggerated by their results on 252 images drawn from the internet, where they experienced an 86.7% success rate with 45 false detections, which suggests great limitations on their input range. Finally, they began their set with 1618 manually cropped and perfectly aligned faces, while we began with roughly half that.

Face Recognition

Method

For the facial recognition aspect of the tagging process we were hopeful that a combination of principal component analysis and linear discriminant analysis could achieve satisfactory results. With the success of the face detection algorithm, we believed we could automatically locate the key eye/mouth portion of the face, which would in turn allow us to ignore much of the variance present in an image with a background.

Data Set

Two data sets were used. The first was a set of 900 images of nine different subjects, with each image being 170x170 pixels and having a roughly centered face. These images were not cropped.

The second data set contained 180 images of the same nine subjects, each manually cropped to contain only the eyes and mouth. They were then scaled to be 20x20, so as to simulate the results of a perfect face detector.

Experimental Results

Since similar experiments in the literature used only a small number of images per person to form a training set, we followed suit and selected between five and eight images per person. As in [4], PCA was run on the training set to reduce the images down to 8 dimensions (one fewer than the number of classes). Each image in the test set was then classified based on its norm-difference from the classes in our training set. We also tested an LDA algorithm (which first performed PCA before attempting to discriminate) on the same training data, and used a similar norm-difference approach for classification.

Percentage of correctly classified images (N=training set size)					
	Data Set 1				Data Set 2
	N = 5	N = 6	N = 7	N = 8	N = 5
PCA	22.4%	21.7%	24.5%	24.8	19.8%
LDA	18.6%	19.7%	20.5%	20.4	23.5%

Though performance was better than random guessing (~11%), the results were nevertheless quite disappointing. For the first data set, an explanation for the failure could be attributable to the extremely high variance of the images; faces were not necessarily centered, backgrounds alter, lighting conditions change. For the second

dataset, while variance was perhaps not as significant as in the first, we did have to deal with the added complication of different poses, which significantly skews PCA.

Comparison with previous work

Using a database of images taken under tightly controlled conditions, Turk and Pentland (1991) achieved accuracies of between 64% and 96%, depending on variations in lightning, orientation, and size. While such statistics seem phenomenal, it is important to note that we lacked such controls with our data; because of the need to classify people in vastly different poses and lighting conditions, a precise algorithm that required strict controls would not have been useful, regardless of the success rate.

Conclusion

Though at first it appeared tractable, the task of automatically tagging the Facebook now seems infeasible. While face detection using a mixture of factor analyzers model works, its false detection rate, even at 8.9%, is still too significant to make it functional. Perhaps with more training samples this factor could be reduced, but given that even the task of scanning a 170x170 image takes over 10 seconds, significant strides in image processing would be needed to make the task quick enough for online processing.

As for face recognition, the huge variance of poses, lighting, and time passing (hair changes especially) make the task extremely difficult, if not impossible. While attempts with non-linear kernel PCA or LDA could be tried, the acute degree of our failure to successfully match faces with only 9 classes suggests that obtaining satisfactory results with a network of hundreds of thousands of people seems infeasible.

Therefore, while the work produces a useful Database and a modestly accurate face detection program, the goal of automatically tagging the Facebook remains unattained.

Sources

[1] Face Detection Using Multimodal Density Models
Ming-Hsuan Yang, David Kriegman, and Narendra Ahuja
September 2000

[2] The EM Algorithm for Mixtures of Factor Analyzers
Zoubin Ghahramani, Geoffrey E. Hinton
May 1996

[3] Face Recognition Using Hybrid Feature
Hong-Tao Su, David-Dagan Feng, Xiu-Ying Wang, Rong-Chun Zhao
November 2003

[4] Using Graph Model for Face Analysis
Deng Cai, Xiaofei He, Jiawei Han
September 2005

[5] Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection
Peter N. Belhumeur, Joao P. Hespanha, David J. Kriegman

[6] Face Recognition Using Eigenfaces
Matthew A. Turk, Alex P. Pentland