THESIS

THE BAYESIAN INTRAPERSONAL/EXTRAPERSONAL CLASSIFIER

Submitted by

Marcio Luis Teixeira

Computer Science Departament

In partial fulfillment of the requirements

for the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Summer 2003

COLORADO STATE UNIVERSITY

July 10, 2003

WE HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER OUR SUPERVISION BY MARCIO LUIS TEIXEIRA ENTITLED THE BAYESIAN INTRAPERSONAL/EXTRAPERSONAL CLASSIFIER BE ACCEPTED AS FULFILLING IN PART REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE.

Committee on Graduate Work

Committee Member

Committee Member

Adviser

Department Head

ABSTRACT OF THESIS

THE BAYESIAN INTRAPERSONAL/EXTRAPERSONAL CLASSIFIER

This thesis documents work that was performed using the Bayesian Interpersonal/Extrapersonal Classifier (BIC). We examine the implementation of the algorithm and address several numerical stability issues that were identified with the original design of the classifier. We also examine the performance of the algorithm on standard FERET data sets and explore a hybrid classifier which combines features of the BIC with a standard nearest-neighbor classifier.

Marcio Luis Teixeira
Computer Science Departament
Colorado State University
Fort Collins, Colorado 80523
Summer 2003

TABLE OF CONTENTS

Chapter 1

Introduction

1.1 Overview

This thesis provides an overview of the probabilistic matching techniques proposed by Moghaddam and Pentland for face recognition. Chapter 2 present the theory behind the Bayesian Interpersonal/Extrapersonal Classifier, taking care to examine the many assumptions which went into the derivation of the classifier equations, and chapter 3 explores a numerical stability problem that had to be resolved before we were able to produce a working system. This work shows that the troublesome computation can be avoided entirely by using a simplification of the scoring equation, which, in the context of a rank classifier, provides identical performance to the original equations. The simplified scores for both the *maximum a posteriori* (MAP) and *maximum likelihood* (ML) classifiers are derived. In chapter 4 the performance of the MAP and ML classifiers are compared on recognition tasks using the FERET image data. The simpler MAP formulation, surprisingly, seems slightly better. Both are shown to modestly out-perform a baseline Eigenfaces classifier.

In the course of our research, we have evaluated the probabilistic matching techniques proposed by Moghaddam and Pentland for face recognition and found it to perform well. However, the direct application of the algorithm is not generally possible since for each probe image it computes the difference images between the probe image and all the gallery images. For very large galleries, doing so is not computationally feasible. More conventional subspace methods, such as nearest-neighbor matching using principle components analysis, avoid this problem by doing classification in a compressed subspace. In chapter 4, we examine a two-stage system which combines traditional PCA subspace matching with a bayesian intrapersonal classifier. We demonstrate that we are able to acheive a system which maintains much of the efficiency of the PCA nearest-neighbor classifier while providing an improvement in accuracy over a baseline nearest-neighbor algorithm.

Chapter 5 explores the question of whether it is possible to predict, using statistical measures

on the images, the failure or success of two common face recognition algorithms in classifying a pair of images of the same person as being the same person (a success) or a different person (a failure). We devised a set of 52 image statistics over 12 different image regions corresponding to major facial features. For each pair of images, we computed the statistics for both the images in the pair, and also the difference in these two values. These 156 unique values were then evaluated as to their ability to predict the success or failure of the algorithms.

1.2 Prior Work

1.2.1 Nearest-Neighbor Classifiers

Principle components analysis has been investigated extensively as a method for matching faces. The original FERET evaluation compared various subspace algorithms [1]. Of these, PCA is surprisingly effective and has become a baseline against which other methods are judged. Other subspace methods, such as linear discriminate analysis (LDA) or independent component analysis (ICA), have also been studied.

What these algorithms share in common is that the gallery images and the probe images are projected into eigenspace and the nearest gallery image is chosen as the best match. In [2] Yabor, Draper and Beveridge evaluated four traditional distance metrics (city block, euclidean, mahalanobis angle and mahalanobis) and compared their performance when matching images in the FERET database. In that report, it was determined that mahanobis angle provided the best overall performance.

1.2.2 Bayesian Difference Image Classifiers

A significant departure from previous work is research done by Moghaddan and Pentland [3]. Their probabilistic classifiers differs from traditional algorithms in two important ways. First, their system uses a Bayesian classifier rather than the a nearest-neighbor classifier. Second, they cast the multi-class problem of distinguising among images of different subjects into the problem of distinguishing between intrapersonal and extrapersonal difference images.

In nearest-neighbor classifiers, the face images are projected directly into the compressed space and it is expected that images from the same person will map to points that are relatively close to one another. Conversely, it is expected that images from different people will be widely separated. Certain subspace methods, such as LDA, make this objective explicit by purposefully trying to choose a projection that minimizes the interclass separation while maximizing the in-between class separation [5]. Such methods, however, rest on the assumption that an optimal projection exists and that such a projection would be able to map the facial images into a subspace with the special

property of having distinct non-overlapping regions corresponding to each subject (nearest-neighbor classifiers, for instance, depend heavily on this assumption). Moghaddan and Pentland address this problem by introducing the notion of intrapersonal and extrapersonal subspaces.

The key to the classification of the difference images as intrapersonal or extrapersonal is to regard each difference image as a point in a high-dimensional space. The space of all possible difference images is vast and very sparsely populated. Each difference image occupies a point in this space. As in the real universe, a great majority of the space is unoccupied – the majority of vacant points correspond to difference images that never arise in practice. Difference images which arise in practice will form clusters, analogous to galaxies in the real universe, and on the whole, these "galaxies" will take only an infinitesimally small portion of the entire space. The key assumption in Moghaddam and Pentland's work is that the particular difference images belonging to the *intrapersonal* and *extrapersonal* difference images come from two such "galaxies" – that is, they originate from distinct and localized Gaussian distributions within the space of all possible images.

Chapter 2

Theoretical Foundation

2.1 Intrapersonal and Extrapersonal Subspaces

In traditional classifiers, face images are projected directly into a compressed subspace, under the assumption that images of a single person will map to a tight cluster of points. Conversely, it is expected that the projections of images of different subjects will be widely separated. Certain subspace methods, such as LDA, make this objective explicit by purposefully trying to choose a projection that minimizes the interclass separation while maximizing the in-between class separation [8]. Such methods, however, assume that projections exist that map facial images onto non-overlapping regions for each subject (nearest-neighbor classifiers, for instance, depend heavily on this).

Moghaddam and Pentland propose an alternative. Their classifier defines the subspace in a different way: rather than treating face images as points in a face subspace, they instead look at the space spanned by the *difference* between two face images. The difference image for two face images is the signed arithmetic difference between respective pixels in the source images. Such difference images fall into two distinct classes: *intrapersonal* difference images are those derived from two images of the same subject, while *extrapersonal* difference images are derived from two images of different subjects. Moghaddam and Pentland suggest that intrapersonal and intrapersonal difference images form distributions that are approximately Gaussian [3]. Their classifier matches probe images to stored images by computing the likelihood that the corresponding difference images came from the subspace of interpersonal rather than extrapersonal images.

2.2 Probability distributions for interpersonal and extrapersonal images

In their paper, Moghaddan and Pentland show that these two classes of difference images span subspaces that can be approximated by Gaussian distributions [3]. Later in this chapter we derive

the equations for these distributions. For any difference image Δ. the probability of that image belonging to class Ω is computed as follows:

$$\widehat{P}(\Delta|\Omega) \equiv \frac{\exp\left(-\frac{1}{2}\sum_{i=1}^{M}\frac{y_i^2}{\lambda_i}\right)}{(2\pi)^{M/2}\prod_{i=1}^{M}\lambda_i^2} \cdot \frac{\exp\left(-\frac{1}{2\rho}\epsilon^2(\Delta)\right)}{(2\pi\rho)^{(N-M)/2}} \tag{2.1}$$

This equation is written in terms of the eigenvalues λ and truncated eigenvector matrix Φ_M that are obtained by running principle components on a training set of difference images belonging to class Ω.

While in practice direct computation of equation 2.5 is numerically unstable, chapter 3 will be demonstrate that for the purposes of ranking, a simpler score may be defined for the maximum likelihood (ML) version of the bayesian intrapersonal classifier:

$$S_{ML}(\Delta) \equiv \frac{\epsilon^2(\Delta)}{\rho} + \sum_{i=1}^{M}\frac{y_i^2}{\lambda_i} \tag{2.2}$$

The actual parameters for such distributions are not known, so the first stage of the probabilistic classifiers estimates the parameters that define the Gaussian distributions corresponding to the *intrapersonal* and *extrapersonal* difference images. This training stage, called *density estimation*, is accomplished using Principle Components Analysis (PCA). This stage estimates the statistical properties of two subspaces: one for difference images that belong to the *intrapersonal* class and another for difference images that belong to the *extrapersonal* class. These classes are denoted Ω_I and Ω_E, respectively.

During the testing phase, the classifier takes a difference image Δ of unknown class membership and uses the estimates of the the probability distributions $P(\Delta|\Omega_I)$ and $P(\Delta|\Omega_E)$ as a means of identification. The *maximum a posteriori* (MAP) classifier uses Bayes rule to estimate the *a posteriori* probability of Δ belonging to the intrapersonal class and equates a similarity measure S_{MAP} with this probability:

$$S_{MAP} \equiv \widehat{P}(\Omega_I|\Delta) = \frac{\widehat{P}(\Delta|\Omega_I)\cdot\widehat{P}(\Omega_I)}{\widehat{P}(\Delta|\Omega_I)\cdot\widehat{P}(\Omega_I) + \widehat{P}(\Delta|\Omega_E)\cdot\widehat{P}(\Omega_E)} \tag{2.3}$$

When comparing a novel probe image to a set of n known gallery images, n difference images are created by subtracting the probe image from the n gallery images. These difference images are then ranked and the subject associated with the gallery image yielding the highest similarity score is taken to be the person in the probe image. This operation is essentially identical to that carried out by other nearest neighbor classifiers, with the caveat that difference images are used, one difference image per gallery image.

A simpler maximum likelihood formulation uses only the intrapersonal class probability, and ignores the extrapersonal class information. Moghaddam and Pentland propose this *maximum likelihood* (ML) classifier as an computationally more expedient substitute for the MAP classifier. The similarity score for the ML classifier is:

$$S_{ML} \equiv \widehat{P}(\Delta|\Omega_I) \tag{2.4}$$

The difference between these two formulations in terms of efficiency of implementation as well as recognition performance is of critical interest. In chapter 3 we will examine how each may be computed in a stable fashion, as well as if and when the more complete MAP formulation yields superior recognition performance.

However simple these classifiers may appear, implementation presents several mathematical issues which were not fully addressed in Moghaddam and Pentland's original papers. Chapter 3 explores the numerical stability problems inherent in estimating $P(\Delta|\Omega_I)$ and $P(\Delta|\Omega_E)$ and derives well-behaved formulations for both S_{MAP} and S_{ML}.

2.3 Distribution Estimation Through Prinicipal Components Analysis

PCA is run twice to train the MAP classifier. Once for a set of *intrapersonal* difference images and again for a set of *extrapersonal* difference images. PCA is only run once for the ML classifier; once for the *intrapersonal* difference images. In both cases, the training parameters are N, M and T:

- N is the dimensionality of the original data.
- T is the number of training difference images.
- M is the number of dimensions we keep

The number of pixels in the difference images, N, depends of course on the image data being used. For FERET images preprocessed in the standard fashion, images are 150 by 130 pixels, and hence $N = 19,500$. The number of training images T establishes an upper bound on the intrinsic dimensionality of the difference image subspace, or equivalently the number of non-zero dimensions we will obtain after running PCA on the sample covariance matrix associated with the training images. Typically T ranges between perhaps 100 to $1,000$. Primarily for reasons of computational expedience, it is common to further truncate the PCA subspace, and thus only retain the M most

significant dimensions. In our system, the number of dimensions we discard is $T-M$ and is controlled indirectly by the *cutoff* parameter to our system.

After training, for each subspace we have:

- A projection matrix Φ_M and a vector of eigenvalues λ.
- A parameter ρ that is the average of the eigenvalues between M and T.
- The projection, $y = \left[\begin{array}{cccc} y_1 & y_2 & \ldots & y_M \end{array} \right]^T$, of each difference image Δ into the PCA subspace.

2.3.1 The Simpler Case with no Subspace Truncation

In general form, these distributions are complex and the variances along the dimensions are highly correlated. PCA simplifies our analysis by accomplishing two things. First, it provides us with a linear transformation matrix Φ that rotates the training data in such a way that the variances along each dimension are uncorrelated. Second, it provides estimates for the variances along those dimensions. In short, PCA provides us with all the parameters that we need to estimate the probability function for a Gaussian distribution given a set of samples:

$$\widehat{P}\left(\Delta|\Omega\right) \equiv \frac{\exp\left(-\frac{1}{2}\sum_{i=1}^{N}\frac{y_i^2}{\sigma_i^2}\right)}{\left(2\pi\right)^{N/2}\prod_{i=1}N_i\sigma_i} \tag{2.5}$$

This formulation uses all N possible dimensions, and thus the vector y is an N-dimensional vector which is the result of applying the rotation[1] matrix Φ to a mean subtracted difference image Δ:

$$y = \Phi^T \cdot \Delta$$

The square of the parameters σ are the variances, which are obtained indirectly from PCA through their relationship to the eigenvalues:

$$\sigma^2 = \lambda \tag{2.6}$$

Substitution of equation 2.6 into 2.5 yields:

$$\widehat{P}\left(\Delta|\Omega\right) \equiv \frac{\exp\left(-\frac{1}{2}\sum_{i=1}^{N}\frac{y_i^2}{\lambda_i}\right)}{\left(2\pi\right)^{N/2}\prod_{i=1}^{N}\lambda_i^{1/2}} \tag{2.7}$$

[1]The term rotation is used loosely here. The matrix Φ is ortho-normal and it is helpful to think of it in terms of rotation. However, it may include reflections about some axes and is thus not strictly speaking a rotation.

In practice, however, computing Φ (an N-by-N matrix) and the N-ary vector y is impractical since N is very large. Rather than performing this computation, which is termed the *direct method*, one often performs what is called the *snapshot method* [6]. The snapshot method is preferred because the computation is limited by the number of training images, denoted T, which is typically much smaller than N.

At the end of the computation, the snapshot method yields a T-by-T projection matrix Φ_T which maps difference images into a vector y having only T elements. The snapshot method is related to the direct method in that it computes only the T non-zero eigenvalues and corresponding eigenvectors. The zero valued eigenvalues and corresponding eigenvectors are not computed and do not need to appear in the probability equation. The probability equation is virtually unchanged from 2.7, but is evaluated over fewer elements:

$$\widehat{P}\left(\Delta|\Omega\right) \equiv \frac{\exp\left(-\frac{1}{2}\sum_{i=1}^{T}\frac{y_i^2}{\lambda_i}\right)}{\left(2\pi\right)^{T/2}\prod_{i=1}^{T}\lambda_i^{1/2}} \tag{2.8}$$

The snapshot method is an example of a *dimensionality-reducing transformation* that takes data from a high-dimensional space and maps it onto a lower-dimensional space. In this particular transformation, there is no loss of information since the reduction is achieved through the elimination of redundant data.

2.3.2 Basis Vector Truncation

Although PCA is inherently a lossless transformation, it has an interesting property that allows it to be applied in a lossy manner. In PCA, the columns of the transformation matrix Φ_T are ranked according to the total variance they capture. By discarding the columns corresponding to the smallest eigenvalues, one is often able to discard noise in the data while preserving the meaningful information. In typical systems, this truncation is done by choosing a cutoff parameter M and constructing a truncated projection matrix Φ_M that only contains the first M columns of Φ_T corresponding to the largest eigenvalues.

While truncation is useful, it presents a special problem when evaluating the probability equation 2.8. The truncation of Φ_T into Φ_M means that we are no longer computing y_i for $M \leq i < T$ and cannot therefore properly evaluate the true estimated probability. To see this more clearly, we write equation 2.8 as two parts, one of which is computable and the other which is not:

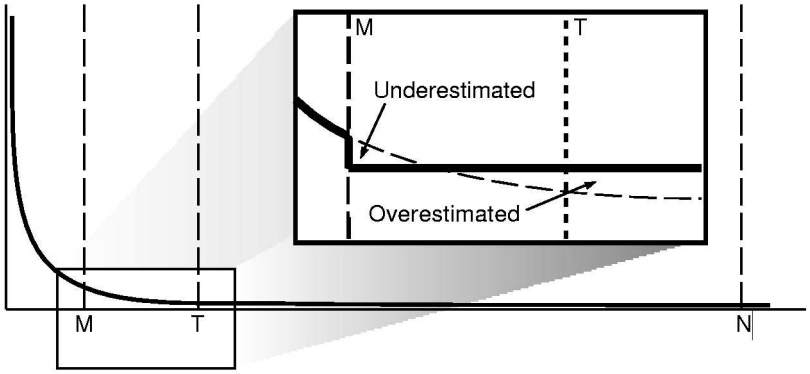


Figure 2.1: A representative distribution of λ_i and the effects of ρ.

$$\widehat{P}(\Delta|\Omega) \equiv \overbrace{\frac{\exp\left(-\frac{1}{2}\sum_{i=1}^{M}\frac{y_i^2}{\lambda_i}\right)}{(2\pi)^{M/2}\prod_{i=1}^{M}\lambda_i^{1/2}}}^{y_i\ known} \cdot \overbrace{\frac{\exp\left(-\frac{1}{2}\sum_{i=M+1}^{T}\frac{y_i^2}{\lambda_i}\right)}{(2\pi)^{(T-M)/2}\prod_{i=M+1}^{T}\lambda_i^{1/2}}}^{y_i\ unknown} \tag{2.9}$$

Equation 2.9 presents the first major issue in the derivation of the probabilistic classifier. In the present form, the rightmost term the equation cannot be evaluated since the values of y_i for $M \leq i < T$ are not known because the PCA space has been truncated – the columns which would have allowed us to compute λ_i for $M \leq i < T$ were precisely those that were removed from Φ_T in order to make Φ_M. One approach would be to disregard that part of the equation altogether. This solution is not acceptable, as it fails to account for the out-of-space variance. Indeed, Moghaddam and Pentland stress the importance of accounting for the out-of-space variance.

To get around this quandary, Moghaddam and Pentland assume the values of λ_i are constant within the range $M \leq i < N$. Figure 2.1 shows a graph of typical values of λ_i. The horizontal axis shows the index i and the vertical axis shows the magnitude of the corresponding λ_i. Representative values of M, T and N are indicated by the dotted lines.

Moghaddam and Pentland compute the average of the values for λ_i for the range $M \leq i < T$ (which are available from PCA) and designate this ρ:

$$\rho = \frac{1}{T-M}\Sigma_{i=M+1}^{T}\lambda_i \tag{2.10}$$

Then they use this value as a surrogate for the actual λ_i in the entire (much longer) range $M \leq i < N$. The cutout in figure 2.1 shows the effect of this assumption. The dotted line represents the true values of λ_i and the solid line represents the assumption made by Moghaddam and Pentland. The

arrows point to the regions under which the true values of λ_i are either under or overestimated. The cutout also shows the discontinuity that occurs at $i = M$.

The assumption made by Moghaddam and Pentland is essentially that there is a certain amount of variance in the dimensions beyond T. Due to the limited sample size, PCA cannot estimate these values and they do not appear in equation 2.9. By assuming a constant value ρ for these variances, however, Moghaddam and Pentland extend the terms of the probability equation 2.9 over the entire range from 1 to N. In the following equation, notice the change of superscripts on the sums from T to N and the substitution of ρ for λ_i:

$$\widehat{P}(\Delta|\Omega) \equiv \overbrace{\frac{\exp\left(-\frac{1}{2}\sum_{i=1}^{M}\frac{y_i^2}{\lambda_i}\right)}{(2\pi)^{M/2}\prod_{i=1}^{M}\lambda_i^{1/2}}}^{y_i\ known} \cdot \overbrace{\frac{\exp\left(-\frac{1}{2}\sum_{i=M+1}^{N}\frac{y_i^2}{\rho}\right)}{(2\pi)^{(N-M)/2}\prod_{i=M+1}^{N}\rho^{1/2}}}^{y_i\ unknown} \tag{2.11}$$

The assumption that ρ is constant and independent of i allows us to move it outside the summation:

$$\widehat{P}(\Delta|\Omega) \equiv \overbrace{\frac{\exp\left(-\frac{1}{2}\sum_{i=1}^{M}\frac{y_i^2}{\lambda_i}\right)}{(2\pi)^{M/2}\prod_{i=1}^{M}\lambda_i^{2}}}^{y_i\ known} \cdot \overbrace{\frac{\exp\left(-\frac{1}{2\rho}\sum_{i=M+1}^{N}y_i^2\right)}{(2\pi)^{(N-M)/2}\rho^{(N-M)/2}}}^{y_i\ unknown} \tag{2.12}$$

Further simplification leads us to the equation that appears in [4]:

$$\widehat{P}(\Delta|\Omega) \equiv \overbrace{\frac{\exp\left(-\frac{1}{2}\sum_{i=1}^{M}\frac{y_i^2}{\lambda_i}\right)}{(2\pi)^{M/2}\prod_{i=1}^{M}\lambda_i^{2}}}^{y_i\ known} \cdot \overbrace{\frac{\exp\left(-\frac{1}{2\rho}\sum_{i=M+1}^{N}y_i^2\right)}{(2\pi\rho)^{(N-M)/2}}}^{y_i\ unknown} \tag{2.13}$$

At this point, we are able to resolve the problem which was presented earlier. Although we do not know the values of y_i on the right side of the equation, we are able to make use of the following identity:

$$\sum_{i=M+1}^{N} y_i^2 = \sum_{i=1}^{N} y_i^2 - \sum_{i=1}^{M} y_i^2 = \left\|\widetilde{\Delta}\right\|^2 - \sum_{i=1}^{M} y_i^2 \tag{2.14}$$

We call this result $\epsilon^2(\Delta)$:

$$\epsilon^2(\Delta) \equiv \left\|\widetilde{\Delta}\right\|^2 - \sum_{i=1}^{M} y_i^2$$

Notice that $\epsilon^2(\Delta)$ can be easily computed. The value $\left\|\widetilde{\Delta}\right\|^2$ is the square of the vector length of the difference image Δ *before* projection by Φ_M, while the sum $\sum_{i=1}^{M} y_i^2$ is the vector length *after* projection by Φ_M. The difference of these quantities, $\epsilon^2(\Delta)$, captures the change in vector length as the difference image is projected into the subspace. Recall that if Φ_M had not been truncated, the

length would not have changed (it is a general property of an unaltered PCA rotation matrix Φ that vector lengths are preserved). Hence, $\epsilon^2(\Delta)$ is said to be the measure of the *residual reconstruction error* in the partial KL expansion. The residual reconstruction error is in fact the distance-from-face-space (DFFS) which is introduced in [4]. With $\epsilon^2(\Delta)$, we are able to write 2.13 as follows:

$$\widehat{P}(\Delta|\Omega) \equiv \frac{\exp\left(-\frac{1}{2}\sum_{i=1}^{M}\frac{y_i^2}{\lambda_i}\right)}{(2\pi)^{M/2}\prod_{i=1}^{M}\lambda_i^2} \cdot \frac{\exp\left(-\frac{1}{2\rho}\epsilon^2(\Delta)\right)}{(2\pi\rho)^{(N-M)/2}} \tag{2.15}$$

To close this section, we reiterate the meanings of the variables used in this equation:

- Δ is a vector of length N containing a mean-normalized difference image.
- y is the vector of length M of the coefficients resulting from the projection of Δ into the the compressed subspace corresponding to Ω.
- λ is the vector of length M of the eigenvalues corresponding to Φ_M.
- i ranges from 1 and M and is used to index both λ and M.
- The variables N and M are scalars: N is the dimensionality of the original images, M is the number of basis vectors which are kept in Φ_M.
- T is the number of training images that were used to estimate $P(\Delta|\Omega)$. Generally $T \ll N$.
- $\epsilon^2(\Delta)$ is the residual reconstruction error in the partial KL expansion:

$$\epsilon^2(\Delta) \equiv \left\|\widetilde{\Delta}\right\|^2 - \sum_{i=1}^{M} y_i^2 \tag{2.16}$$

- The parameter ρ is a mean of variances, computed by the following equation[2]:

$$\rho = \frac{1}{T-M}\Sigma_{i=M+1}^{T}\lambda_i \tag{2.17}$$

[2] Whether Moghaddam and Pentland used N or T (which they call N_T) for their estimate of ρ is unclear to us.

Chapter 3

Numerical Issues

3.1 The Incomputability of the Probability Equation

At first glance, it would seem as if the direct computation of $P(\Delta|\Omega_I)$ through equation 2.15 is necessary for the ML classifier defined by equation 2.4. Likewise, it would seem both $\widehat{P}(\Delta|\Omega_I)$ and $\widehat{P}(\Delta|\Omega_E)$ must be computed for the MAP classifier defined by equation 2.3. However, computing these probabilities direct is, in practice, very difficult. To start off with, in many common cases the probabilities are small enough to be well beyond the minimum value that is representable by standard floating point representations. To illustrate this, we provide empirically determined values for ρ_I, N_I and M_I gathered from a run on a subset from the FERET database. For a set of thirty individuals, we generated four 130×150 pixel interpersonal difference images, ran PCA and kept thirteen basis vectors. The values of the relevant parameters are the following:

$$\begin{aligned} N_I &= 19,500 \\ M_I &= 13 \\ \rho_I &\cong 4,300 \end{aligned}$$

To estimate the magnitude of equation 2.15 we assume the exponentials on the numerator are 1 and the λ's on the numerator are all 1. Even with these conservative choices, we expect that the value will be in the order of $10^{-30,000}$ which is more than typical machine precision[1] for floating point numbers and hence not computable:

[1]The IEEE double precision floating point standard reserves eleven bits for the exponent. This means that to a rough approximation the smallest representable number is in the order of 2^{-1024} and the largest is of the order 2^{1024} (in fact, it is less since certain exponent values have special meanings). Clearly, the expected value for the probability, $10^{-30,000}$ is much smaller than what could be represented by a double precision number.

$$\widehat{P}(\Delta|\Omega_I) \cong \left[\frac{1}{(2\pi)^{13/2}}\right] \cdot \left[\frac{1}{(2\pi \cdot 4,300)^{(19,500-13)/2}}\right] \cong 10^{-30,000} \tag{3.1}$$

A similar analysis of the extrapersonal subspace yields a value of $N_E = 19,500$, $M_I = 17$ and $\rho_E = 2,500$, which can also be shown to lead to an incomputable number.

For the MAP classifier, the computation of $\widehat{P}(\Omega_I|\Delta)$ is possible because equation 2.3 is a ratio. In [7], the terms of the equation are carefully regrouped to yield a numerically stable expression. Unfortunately, this technique cannot be applied to the ML classifier, as the maximum likelihood equation (2.4) does not involve a ratio of probabilities.

3.2 A Solution to the Problem

Knowing that we cannot use equation 2.15 directly to compute $P(\Delta|\Omega_I)$, the only alternative is to show a way around it. To do so, we observe that we are dealing with a classifier that ranks probabilities. We do not necessarily need to know the probabilities *per se*, but rather we want to ask which probability is the greatest. We can express this question as a boolean function B for two difference images Δ_1 and Δ_2:

$$B(\Delta_1, \Delta_2) \equiv \widehat{P}(\Delta_1|\Omega) < \widehat{P}(\Delta_2|\Omega) \tag{3.2}$$

The key to simplifying this problem is to notice that the result of B is unaffected if we scale both sides of the inequality by a positive constant k:

$$B(\Delta_1, \Delta_2) = k \cdot \widehat{P}(\Delta_1|\Omega) < k \cdot \widehat{P}(\Delta_2|\Omega) \tag{3.3}$$

This motivates us to factor $\widehat{P}(\Delta|\Omega)$ into two parts, one part which is a function of Ω and the other which is a function of Ω and Δ:

$$\widehat{P}(\Delta|\Omega) = f(\Omega) \cdot g(\Omega, \Delta) \tag{3.4}$$

If we choose a specific factorization such that $f(\Omega) \geq 0$, we can eliminate it from the inequality by choosing $k = \frac{1}{f(\Omega)}$ and simplifying equation 3.3:

$$\begin{aligned} B(\Delta_1, \Delta_2) &= k \cdot \widehat{P}(\Delta_1|\Omega) < k \cdot \widehat{P}(\Delta_2|\Omega) && (3.5)\\ &= k \cdot f(\Omega) \cdot g(\Omega, \Delta_1) < k \cdot f(\Omega) \cdot g(\Omega, \Delta_2) && (3.6)\\ &= \frac{1}{f(\Omega)} \cdot f(\Omega) \cdot g(\Omega, \Delta_1) < \frac{1}{f(\Omega)} \cdot f(\Omega) \cdot g(\Omega, \Delta_2) && (3.7) \end{aligned}$$

$$= \quad g(\Omega, \Delta_1) < g(\Omega, \Delta_2) \tag{3.8}$$

As a last step, we take the logarithm of both sides:

$$B(\Delta_1, \Delta_2) = \ln(g(\Omega, \Delta_1)) < \ln(g(\Omega, \Delta_2)) \tag{3.9}$$

In the subsequent section, we will see how equation 3.9 can be applied to the ML classifier.

3.2.1 Factorization of the Probability Equations for the ML classifier

The factorization of the ML probability equation (eq. 2.15) into $f(\Omega)$ and $g(\Omega, \Delta)$ is accomplished by separating the parts which depend on Δ from those that do not. It turns out that this is easy to do: only y and $\epsilon^2(\Delta)$ depend on Δ. On the other hand, ρ, N, M and λ are properties of the subspace and are constant with respect to Δ. Thus we obtain the factorization:

$$\widehat{P}(\Delta|\Omega_I) = f(\Omega_I) \cdot g(\Omega_I, \Delta) \tag{3.10}$$

By choosing:

$$f(\Omega_I) = \frac{1}{(2\pi)^{M_I/2} \prod_{i=1}^{M_I} \lambda_{I,i}^{1/2}} \cdot \frac{1}{(2\pi\rho_I)^{(N_I - M_I)/2}} \tag{3.11}$$

$$g(\Omega_I, \Delta) = \exp\left(-\frac{1}{2}\sum_{i=1}^{M_I} \frac{y_{I,i}^2}{\lambda_{I,i}}\right) \cdot \exp\left(-\frac{1}{2}\frac{\epsilon^2(\Delta)}{\rho_I}\right) \tag{3.12}$$

$$= \exp\left(-\frac{1}{2}\left[\frac{\epsilon^2(\Delta)}{\rho_I} + \sum_{i=1}^{M_I} \frac{y_{I,i}^2}{\lambda_{I,i}}\right]\right) \tag{3.13}$$

Since λ_I and ρ_I are positive, we have no trouble showing that $f(\Omega_I) \geq 0$ and we can eliminate it entirely using the simplification steps described in equations 3.5 through 3.8. Substituting equation 3.13 into equation 3.9 yields:

$$\begin{aligned} B(\Delta_1, \Delta_2) &= -\frac{1}{2}\left[\frac{\epsilon^2(\Delta_1)}{\rho_I} + \sum_{i=1}^{M_I} \frac{y_{I,i}^2}{\lambda_{I,i}}\right] < -\frac{1}{2}\left[\frac{\epsilon^2(\Delta_2)}{\rho_I} + \sum_{i=1}^{M_I} \frac{y_{I,i}^2}{\lambda_{I,i}}\right] \\ &= \frac{1}{2}S_{ML}(\Delta_1) < \frac{1}{2}S_{ML}(\Delta_2) \end{aligned}$$

At this point, it should be apparent that we have resolved our numerical problems through the elimination of $f(\Omega)$. The troublesome powers are no longer present. As a further refinement, we eliminate the factor of $\frac{1}{2}$ and define the score for the ML classifier as:

$$S_{ML}(\Delta) \equiv -\frac{\epsilon^2(\Delta)}{\rho_I} - \sum_{i=1}^{M_I} \frac{y_{I,i}^2}{\lambda_{I,i}} \tag{3.14}$$

It is interesting to notice that the resulting score consists of two parts. If we relate this to the terminology introduced in [4], we see the summation on the right is equivalent to a Mahalanobis distance and captures the distance-in-face-space (DIFS), while the term on the left captures the distance-from-face-space (DFFS).

3.2.2 An Indirect Method for the MAP Classifier

Though it may not be initially apparent, the very same strategy that we used for the ML classifier may be applied for the MAP classifier. We begin with equation 2.3:

$$\widehat{P}(\Omega_I|\Delta) = \frac{\widehat{P}(\Delta|\Omega_I) \cdot \widehat{P}(\Omega_I)}{\widehat{P}(\Delta|\Omega_I) \cdot \widehat{P}(\Omega_I) + \widehat{P}(\Delta|\Omega_E) \cdot \widehat{P}(\Omega_E)} \tag{3.15}$$

Dividing the numerator and denominator by $\widehat{P}(\Delta|\Omega_I) \cdot \widehat{P}(\Omega_I)$ forms a ratio of probabilities:

$$\widehat{P}(\Omega_I|\Delta) = \frac{1}{1 + \frac{\widehat{P}(\Delta|\Omega_E) \cdot \widehat{P}(\Omega_E)}{\widehat{P}(\Delta|\Omega_I) \cdot \widehat{P}(\Omega_I)}} \tag{3.16}$$

At this point, we decompose $\widehat{P}(\Delta|\Omega_E)$ and $\widehat{P}(\Delta|\Omega_I)$ into parts:

$$\widehat{P}(\Omega_I|\Delta) = \frac{1}{1 + \frac{f(\Omega_E) \cdot g(\Omega_E, \Delta)}{f(\Omega_I) \cdot g(\Omega_I, \Delta)} \cdot \frac{\widehat{P}(\Omega_E)}{\widehat{P}(\Omega_I)}} \tag{3.17}$$

Observe that we do not need to compute the probability $P(\Omega_I|\Delta)$ *per se*, but rather than we eventually wish to evaluate the inequality Q for Δ_1 and Δ_2:

$$B(\Delta_1, \Delta_2) \equiv \widehat{P}(\Omega_I|\Delta_1) < \widehat{P}(\Omega_I|\Delta_2) \tag{3.18}$$

We substitute 3.17 into 3.18 to obtain:

$$B(\Delta_1, \Delta_2) = \frac{1}{1 + \frac{f(\Omega_E) \cdot g(\Omega_E, \Delta_1)}{f(\Omega_I) \cdot g(\Omega_I, \Delta_1)} \cdot \frac{\widehat{P}(\Omega_E)}{\widehat{P}(\Omega_I)}} < \frac{1}{1 + \frac{f(\Omega_E) \cdot g(\Omega_E, \Delta_2)}{f(\Omega_I) \cdot g(\Omega_I, \Delta_2)} \cdot \frac{\widehat{P}(\Omega_E)}{\widehat{P}(\Omega_I)}} \tag{3.19}$$

We take the reciprocal of each side, being careful to reverse the inequality:

$$B(\Delta_1, \Delta_2) = 1 + \frac{f(\Omega_E) \cdot g(\Omega_E, \Delta_1)}{f(\Omega_I) \cdot g(\Omega_I, \Delta_1)} \cdot \frac{\widehat{P}(\Omega_E)}{\widehat{P}(\Omega_I)} > 1 + \frac{f(\Omega_E) \cdot g(\Omega_E, \Delta_2)}{f(\Omega_I) \cdot g(\Omega_I, \Delta_2)} \cdot \frac{\widehat{P}(\Omega_E)}{\widehat{P}(\Omega_I)} \tag{3.20}$$

We eliminate the ones:

$$B(\Delta_1, \Delta_2) = \frac{f(\Omega_E) \cdot g(\Omega_E, \Delta_1)}{f(\Omega_I) \cdot g(\Omega_I, \Delta_1)} \cdot \frac{\widehat{P}(\Omega_E)}{\widehat{P}(\Omega_I)} > \frac{f(\Omega_E) \cdot g(\Omega_E, \Delta_2)}{f(\Omega_I) \cdot g(\Omega_I, \Delta_2)} \cdot \frac{\widehat{P}(\Omega_E)}{\widehat{P}(\Omega_I)} \tag{3.21}$$

And regroup:

$$B(\Delta_1, \Delta_2) = \frac{f(\Omega_E)}{f(\Omega_I)} \cdot \frac{g(\Omega_E, \Delta_1)}{g(\Omega_I, \Delta_1)} \cdot \frac{\widehat{P}(\Omega_E)}{\widehat{P}(\Omega_I)} > \frac{f(\Omega_E)}{f(\Omega_I)} \cdot \frac{g(\Omega_E, \Delta_2)}{g(\Omega_I, \Delta_2)} \cdot \frac{\widehat{P}(\Omega_E)}{\widehat{P}(\Omega_I)} \tag{3.22}$$

The ratio $\frac{f(\Omega_E)}{f(\Omega_I)}$ appears on both sides of the inequality. Since $f(\Omega_I) \geq 0$ and $f(\Omega_E) \geq 0$, it can be eliminated:

$$B(\Delta_1, \Delta_2) = \frac{g(\Omega_E, \Delta_1)}{g(\Omega_I, \Delta_1)} \cdot \frac{\widehat{P}(\Omega_E)}{\widehat{P}(\Omega_I)} > \frac{g(\Omega_E, \Delta_2)}{g(\Omega_I, \Delta_2)} \cdot \frac{\widehat{P}(\Omega_E)}{\widehat{P}(\Omega_I)} \tag{3.23}$$

The ratio of priors $\frac{\widehat{P}(\Omega_E)}{\widehat{P}(\Omega_I)}$ can also be eliminated since it is always a positive number:

$$B(\Delta_1, \Delta_2) = \frac{g(\Omega_E, \Delta_1)}{g(\Omega_I, \Delta_1)} > \frac{g(\Omega_E, \Delta_2)}{g(\Omega_I, \Delta_2)} \tag{3.24}$$

We apply a logarithm to both sides:

$$B(\Delta_1, \Delta_2) = \ln\left(\frac{g(\Omega_E, \Delta_1)}{g(\Omega_I, \Delta_1)}\right) > \ln\left(\frac{g(\Omega_E, \Delta_2)}{g(\Omega_I, \Delta_2)}\right) \tag{3.25}$$

To reverse the inequality and make it consistent with our classifier's ranking (which treats the score as a similarity measure, that is, smaller numbers are treated as a match), we multiply both sides by -1:

$$B(\Delta_1, \Delta_2) = -\ln\left(\frac{g(\Omega_E, \Delta_1)}{g(\Omega_I, \Delta_1)}\right) < -\ln\left(\frac{g(\Omega_E, \Delta_2)}{g(\Omega_I, \Delta_2)}\right) \tag{3.26}$$

Recall that:

$$g(\Omega_E, \Delta) = \exp\left(-\frac{1}{2}\left[\frac{\epsilon^2(\Delta)}{\rho_E} + \sum_{i=1}^{M_E} \frac{y_{E,i}^2}{\lambda_{E,i}}\right]\right) \tag{3.27}$$

$$g(\Omega_I, \Delta) = \exp\left(-\frac{1}{2}\left[\frac{\epsilon^2(\Delta)}{\rho_I} + \sum_{i=1}^{M_I} \frac{y_{I,i}^2}{\lambda_{I,i}}\right]\right) \tag{3.28}$$

After substitution of 3.27 and 3.28 into 3.26, equation 3.26 becomes the following (for brevity, we shorten the terms on the right side of the inequality):

$$\frac{1}{2}\left[\frac{\epsilon^2(\Delta_1)}{\rho_E} + \sum_{i=1}^{M_E} \frac{y_{E,i}^2}{\lambda_{E,i}}\right] - \frac{1}{2}\left[\frac{\epsilon^2(\Delta_1)}{\rho_I} + \sum_{i=1}^{M_I} \frac{y_{I,i}^2}{\lambda_{I,i}}\right] < \frac{1}{2}[\cdots] - \frac{1}{2}[\cdots] \tag{3.29}$$

We drop the $\frac{1}{2}$ and take the left hard side of the inequality as a simplified score for the MAP classifier:

$$S_{MAP}(\Delta) \equiv \left[\frac{\epsilon^2(\Delta)}{\rho_E} + \sum_{i=1}^{M_E} \frac{y_{E,i}^2}{\lambda_{E,i}}\right] - \left[\frac{\epsilon^2(\Delta)}{\rho_I} + \sum_{i=1}^{M_I} \frac{y_{I,i}^2}{\lambda_{I,i}}\right] \tag{3.30}$$

As in equation 3.14, the resulting score is a combination of distance-from-face-space (DFFS) and distance-in-face-space (DIFS) terms. Also note that the simplified similarity score for the MAP classifier contains within it the simplified score for the ML classifier:

$$S_{MAP}(\Delta) \equiv S_{ML}(\Delta) + \left[\frac{\epsilon^2(\Delta)}{\rho_E} + \sum_{i=1}^{M_E} \frac{y_{E,i}^2}{\lambda_{E,i}}\right] \tag{3.31}$$

3.3 The Irrelevance of the Intrapersonal and Extrapersonal Priors

A significant consequence of the derivation presented in the previous section is that the prior probabilities for the two classes, $\hat{P}(\Omega_I)$ and $\hat{P}(\Omega_E)$, do not come into play when computing MAP similarity. This is interesting given that Moghaddam and Pentland paper [4] specifically mention setting these priors to $\frac{1}{2}$ and make no mention of the priors not altering the rank choices made by a nearest neighbor classifier using the MAP similarity measure. One must assume from their paper that they were not aware, or at least did not mention, that the priors are irrelevant. It is fortunate that the priors do not matter, since were they to matter, their selection would be problematic.

3.4 A Simpler View of ML and MAP Similarity Scores

Observe that S_{ML} and S_{MAP} are very similar in form. If we define a general function $S(\Delta, \Omega)$:

$$S(\Delta, \Omega) \equiv \frac{\epsilon^2(\Delta)}{\rho} + \sum_{i=1}^{M} \frac{y_i^2}{\lambda_i} \tag{3.32}$$

We can now write both S_{ML} and S_{MAP} in terms of this new function:

$$S_{ML}(\Delta) = -S(\Delta, \Omega_I) \tag{3.33}$$

$$S_{MAP}(\Delta) = S(\Delta, \Omega_I) - S(\Delta, \Omega_E) \tag{3.34}$$

$$= -S_{ML}(\Delta) - S(\Delta, \Omega_E) \tag{3.35}$$

Equation 3.34 reveals a rather interesting property of the MAP classifier: it is simply $S(\Delta, \Omega)$ evaluated for Ω_I and minus $S(\Delta, \Omega)$ evaluated for Ω_E. Equation 3.35 shows that the relationship of the

MAP classifier and the ML classifier is much simpler than one may have guessed by looking at equations 2.3 and 2.4 directly. Equation 3.32 leads to an extremely compact and efficient implementation of the probabilistic classifiers.

Chapter 4

Performance of the Bayesian Interpersonal/Extrapersonal Classifier

4.1 Algorithm Training and Tuning

Before the algorithm can actually be used, parameters for the intrapersonal and extrapersonal densities must be estimated using training data. Let us assume a set of training images with R images of S subjects. Typical numbers using the FERET data might be $R = 4$ and $S = 100$. For a standard PCA image classifier, a sample covariance matrix would be constructed for the $S\,R$ images: 400 for the case just described. However, for the Bayesian Classifier, both intrapersonal and extrapersonal difference training images must be created. The total number of possible intrapersonal difference images, T_I, and extrapersonal difference images, T_E is

$$T_I = \binom{R}{2} S \qquad T_E = (R\,S)^2 - T_I$$

The possible number of extrapersonal difference images, T_E, grows quadratically in the number of subjects. Hence, for the example above, $T_I = 600$ and $T_E = 159,400$.

There are a variety of problems with the obvious choice of using all the potential training data, not the least of which is the impracticality of computing the PCA decomposition for such a large number of extrapersonal images. The Moghaddam and Pentland paper [4] did not address this issue in any detail. For our current work, only a small set of the possible extrapersonal images will be used for training, with the guiding rule being to choose a number of extrapersonal training images comparable to the number of intrapersonal training images. The exact images used for training in order to compare the Bayesian algorithm to a standard PCA algorithm are described in the following section.

	Fb	Fc	dup1	dup2
Bayesian ML	0.82	0.38	0.53	0.32
Bayesian MAP	0.82	0.37	0.52	0.32
EBGM Standard	0.88	0.40	0.44	0.22
LDA Euclidean	0.61	0.19	0.38	0.14
LDA ldaSoft	0.61	0.19	0.37	0.14
PCA Euclidean	0.74	0.05	0.34	0.14
PCA mahCosine	0.85	0.65	0.44	0.22

Table 1: Rank one performance for the various classifiers.

4.2 Classifier Performance on FERET Data

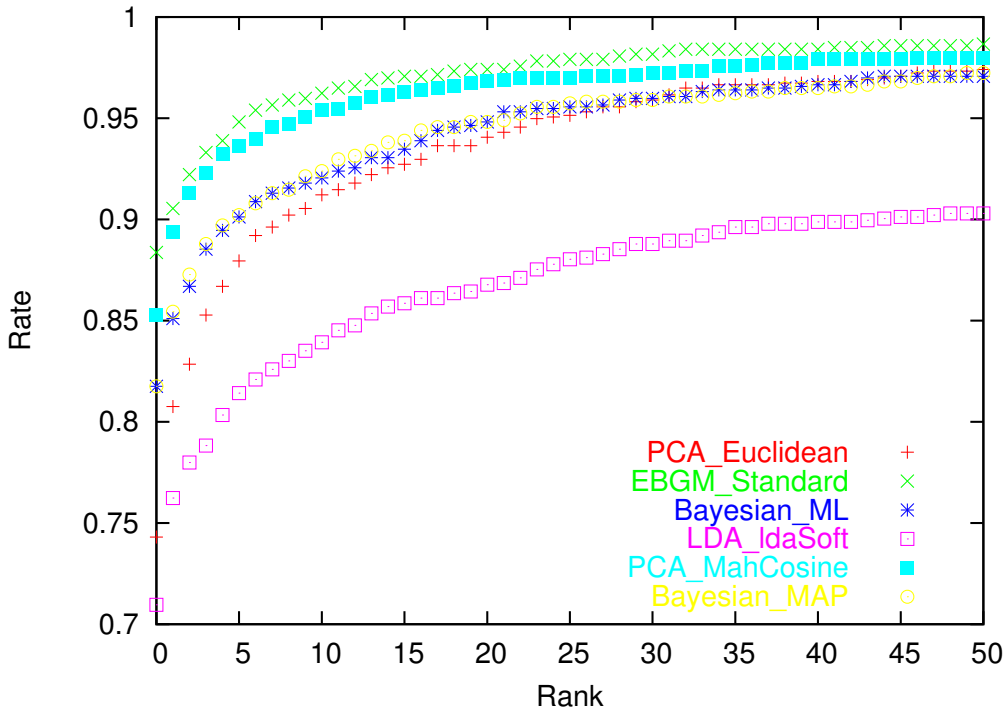
Figures 4.1 and 4.2 compare the other three face recognition control algorithms developed by CSU with the probabilistic ML and MAP classifiers. These algorithms include the elastic bunch graph matching algorithm, the LDA algorithm with two different distances metrics, and the nearest-neighbor PCA classifier using two different distance metrics. The results show that the ML classifier outperforms the more complex Bayesian MAP classifier. For clarity, the data points for rank one are summarized in table 1.

From the figures and table, we see evidence that the Bayesian ML algorithms performs better than all the other algorithms on the dup1 and dup2 data sets. For the fb dataset, the elastic bunch graph classifier performs the best. For the fc dataset, the nearest neighbor classifier with the mahalanobis cosine distance metric performs the best.

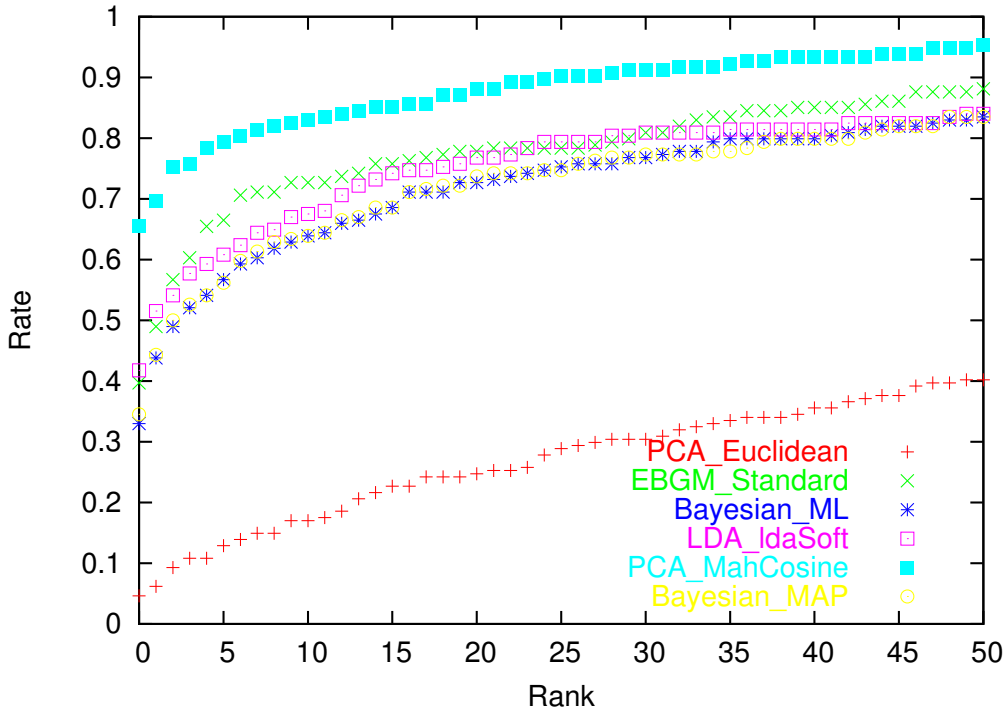
4.3 Effect of using a Hybrid Classifier

In the previous section, we evaluated the probabilistic matching techniques proposed by Moghaddam and Pentland for face recognition and found it to be comparable to nearest-neighbor classifiers. However, the direct application of the algorithm is not generally possible since for each probe image it computes the difference images between the probe image and all the gallery images. For very large galleries, doing so is not computationally feasible. More conventional subspace methods, such as nearest-neighbor matching using principle components analysis, avoid this problem by doing classification in a compressed subspace. In addition to the standard classifier, we examined a two-stage system which combines traditional PCA subspace matching with a bayesian intrapersonal classifier.

To evaluate whether a face matching system could benefit from a hybrid approach, a system was build that uses both eigenspace matching and difference image matching. In the preprocessing stage, the system uses traditional nearest neighbor matching to select a small set of candidate

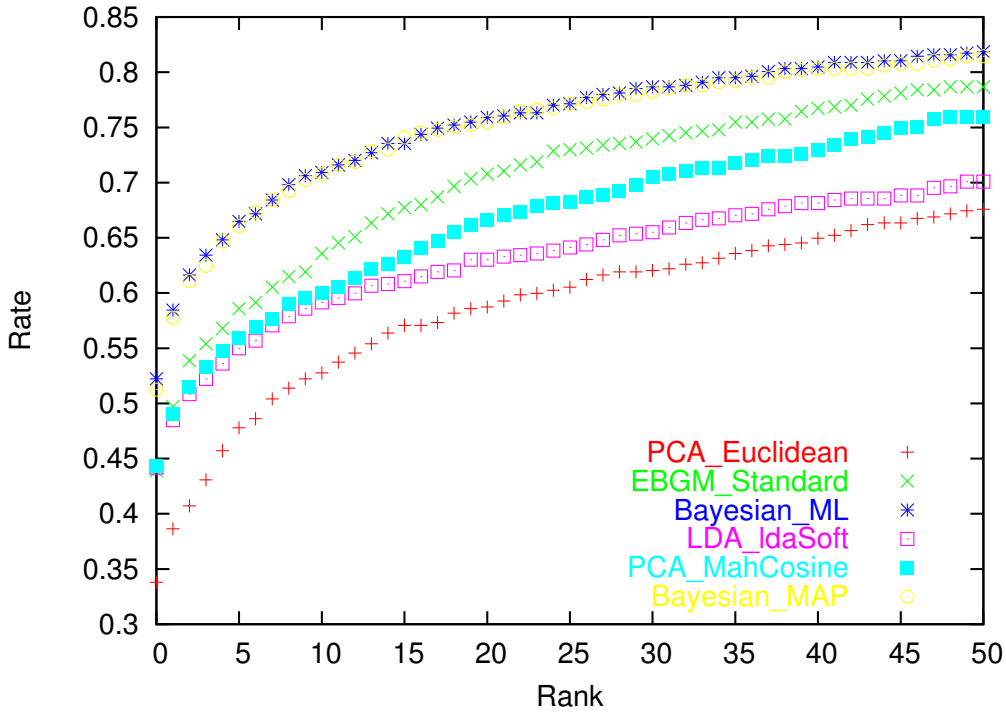


Probe Set Fb

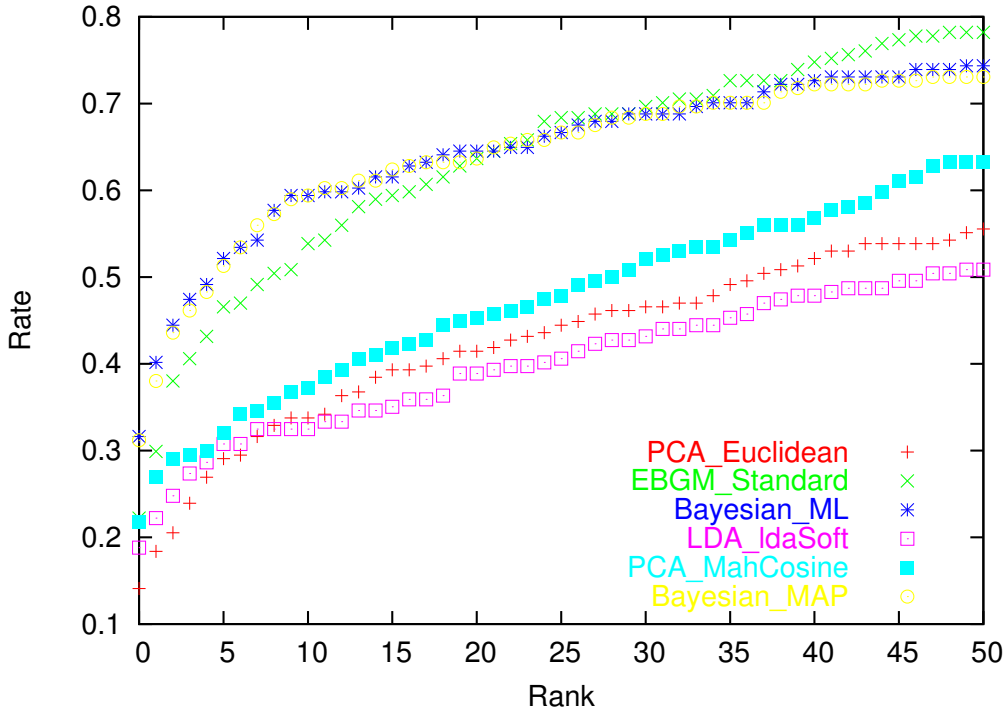


Probe Set Fc

Figure 4.1: Comparison of the four CSU FERET algorithms on FERET probes sets Fa and Fc.



Probe Set Dup 1



Probe Set Dup 2

Figure 4.2: Comparison of the four CSU FERET algorithms on FERET probes sets Dup1 and Dup2

	fafb	fafc	dup1	dup2
PCA mahAngle	0.81	0.37	0.43	0.20
Bayesian ML	0.80	0.35	0.51	0.29
Hybrid (n=5)	0.82	0.40	0.46	0.26

Table 1: Comparison of the rank one performance for the individual and combined algorithm.

images but forgoes making the final choice among those candidates. For each probe, only the set of the n top-ranked images are considered for matching in the post processing stage. The size of this set is a tunable parameter from one to the number of images in the gallery. The extreme cases are degenerate: when it is one, the system behaves as a nearest-neighbor classifier and the postprocessing stage is kept from making any choices. When it is set to the entire collection of gallery images, the ranking done in the first stage becomes irrelevant and the classification is done entirely by the bayesian intrapersonal classifier.

The second stage takes place once the candiate images have been chosen for a probe image. The system computes the signed difference between the probe image and each candidate from the gallery. These images are then projected into the intrapersonal subspace by the bayesian intrapersonal classifier. The classifier uses equation 2.2 to generate a score for that difference image. The difference images corresponding to each candidate are sorted by their score and the one having the maximal score is chosen as the match.

4.3.1 Results with Hybrid System

Figure 4.3 presents our results for the four FERET data sets. We used images that were scaled down to 26 by 30 pixels in our testing. The nearest-neighbor classifier was trained on the 501 images in the standard FERET training set. The bayesian interpersonal classifier was trained on the subset of the training data that had two replicates. There are 72 such subjects, from which we were able to generate 144 intrapersonal difference images. Since we were using the maximum likelihood classifier, we did not need to train on any extrapersonal images.

The horizontal axis of the plots show different experimental trials in which the size of the candidate set was varied from one to one hundred. The vertical axis of the graph represents the rank-one classification accuracy, expressed as a ratio of the number of correctly identified images to number of images presented to the system. For the purposes of establishing an upper and a lower bound on performance, we show the rank one performance of the nearest-neighbor classifier and the bayesian interpersonal classifier without preprocessing.

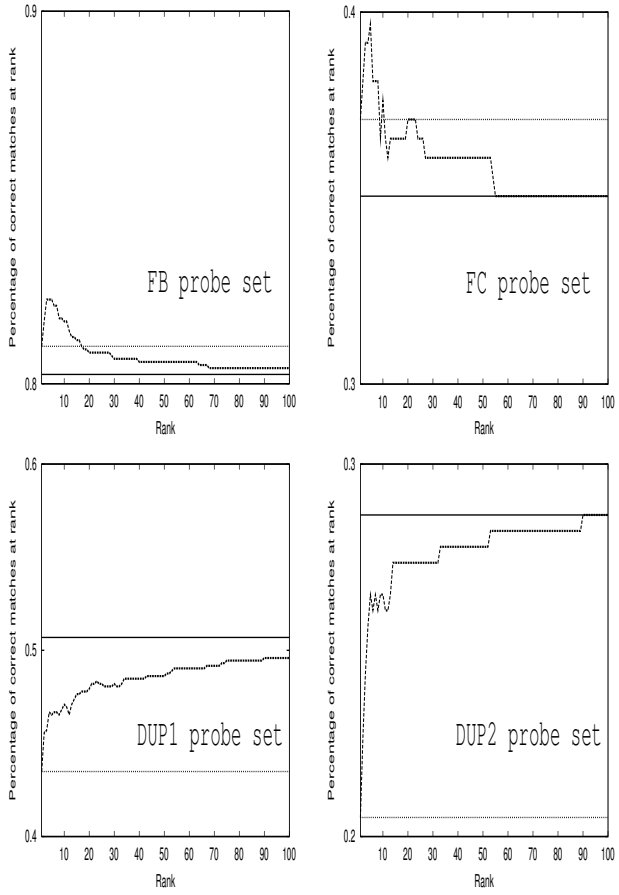


Figure 4.3: Comparison of rank one performance of the individual algorithms and combined algorithm on the various FERET probe sets

Training Set	Eigvecs Kept	Metric	Hybrid	Mean %	Mode %	Lower %	Upper %
16 subjects	90%	ML	No	5.9	4.4	2.0	13.2
16 subjects	90%	ML	Yes	51.7	51.5	46.1	57.4
16 subjects	90%	MAP	No	7.6	5.9	2.5	16.2
16 subjects	90%	MAP	Yes	51.5	51.5	46.1	56.9
16 subjects	60%	ML	No	37.3	36.3	23.0	51.0
16 subjects	60%	ML	Yes	53.0	52.5	47.5	58.3
16 subjects	60%	MAP	No	39.0	34.3	25.5	52.9
16 subjects	60%	MAP	Yes	52.9	52.5	47.5	58.3
64 subjects	90%	ML	No	64.6	64.2	59.3	70.1
64 subjects	90%	ML	Yes	61.6	61.3	56.4	66.7
64 subjects	90%	MAP	No	64.6	64.2	59.3	70.1
64 subjects	90%	MAP	Yes	61.6	61.3	56.4	66.7
64 subjects	60%	ML	No	71.3	71.6	66.2	76.0
64 subjects	60%	ML	Yes	62.7	62.7	57.4	67.6
64 subjects	60%	MAP	No	70.4	69.6	65.2	75.5
64 subjects	60%	MAP	Yes	62.1	62.3	56.9	67.2

Table 4.1: Performance of algorithm under different training configurations

4.4 Effect of Training on Algorithm Performance

Table 4.1 shows the distribution of rank one performance over randomly chosen galleries of the BIC classifier as training parameters are varied. The table shows the mean, mode, lower and upper bounds of the distribution of the classifier performance score on 10,000 trials in which the gallery and probe sets were selected randomly. The study was run on 204 subjects with 4 replicates per subject, for a total of 816 images. Three out of four images are used for algorithm training. There are two different training sets, one with 16 subjects and another with 64 subjects. These training sets contain all three of the replicates for each subject, for a total of 48 and 192 images respectively. The following parameters were varied, training set size (16 images vs. 64 images), the PCA cutoff (none vs. 60% of total eigenvectors), the distance metric (either ML or MAP) and whether the algorithm was straight BIC or the hybrid system.

From table 4.1 we see that increasing the training set size from 16 to 64 improves performance on all cases. We also see that ML performs very similar to MAP. We observe that keeping more PCA eigenvectors hurts performance. The effect of the hybrid system is much less clear cut. In the cases in which we train on 16 subjects, it seems to increase performance. Otherwise, for the cases in which we have 64 training images, it hurts performance. This seems to indicate that the hybrid approach works well in cases in which the algorithm is starved for training data.

Chapter 5

Image Covariates for the Combined Classifier

The FERET Subject Covariate Study[10] focused primarily on qualitative factors such as race, gender, age and face expression. Although that study was able to draw several conclusions about the effect of those factors on performance, it relied on hand-labeling of the factors and did not provide predictors suitable for an automated system. In this chapter, we explore the question of whether it is possible to extract quantitative statistics from the actual image data which can then be used by the system to provide an estimate of classification uncertainty, or adaptively select among differently-abled algorithms to improve overall system performance. We devised a set of 52 image statistics over 12 different image regions corresponding to major facial features. The study looked at 1072 pair of images corresponding to 1072 distinct FERET subjects. For each pair of images, we computed the statistics for both the images in the pair, and also the difference in these two values. These 156 unique values were then evaluated as to their ability to predict the success or failure of the algorithms. The study aimed to ask the question of what particular traits would make an image (or a pair of images) harder or easier to correctly match.

In the FERET Subject Covariate Study a distinction was made between *subject covariates* and *image covariates*. The subjects covariates were factors which described things such as gender, race or age group – things that weren't expected to change from photo to photo of the same subject. The image covariates, on the other hand, were things like facial expression, things that were expected to change from photo to photo. In that study, all of these covariates were hand-labeled. This study supplements those hand-labeled covariates with additional covariates which are measured off the pixel values of a particular image. To make a distinction between these two, we use the terms *subject image covariates* and *image covariates* for these two cases.

5.1 The Hiearchy of Image Predictors

For this study, 52 image predictors were devised. These are divided into three sets:

whole-image predictors are statistics which are generated over all unmasked pixels in a normalized FERET images.

regional predictors are derived from the pixel values corresponding to only a particular region of the face (figure 5.1 illustrates these regions).

regional contrast predictors take two such regions and contrast the statistics of the pixel values in those two areas to each other.

Figure 5.1 provides a summary of the abbreviations which are used for these predictors while the subsequent sections define the exact nature of these predictors.

5.1.1 Whole-Image Predictors

The first four *whole image predictors* are derived from the pixel values in the entire unmasked area of the face. They correspond to four standard statistical measures: the mean, median, standard deviation, and median absolute deviation (a variant of standard deviation which is robust to outliers).

1. Mean intensity of pixels
2. Median intensity of pixels
3. Standard deviation of pixel intensity
4. Median absolute deviation (MAD) of pixel intensity

The remaining two predictors relates to the eye coordinates in the original FERET images before they were aligned to the standard eye coordinates.

1. Source resolution is the number of pixels between the eyes in the original image
2. Facial tilt is the angle off horiontal of the line between eyes

5.1.2 Regional Predictors

The sixteen *regional predictors* apply the four standard statistical measures on the pixel values of four different portions of the face: the combined eye region and the forehead, center and mouth/chin horizontal strips.

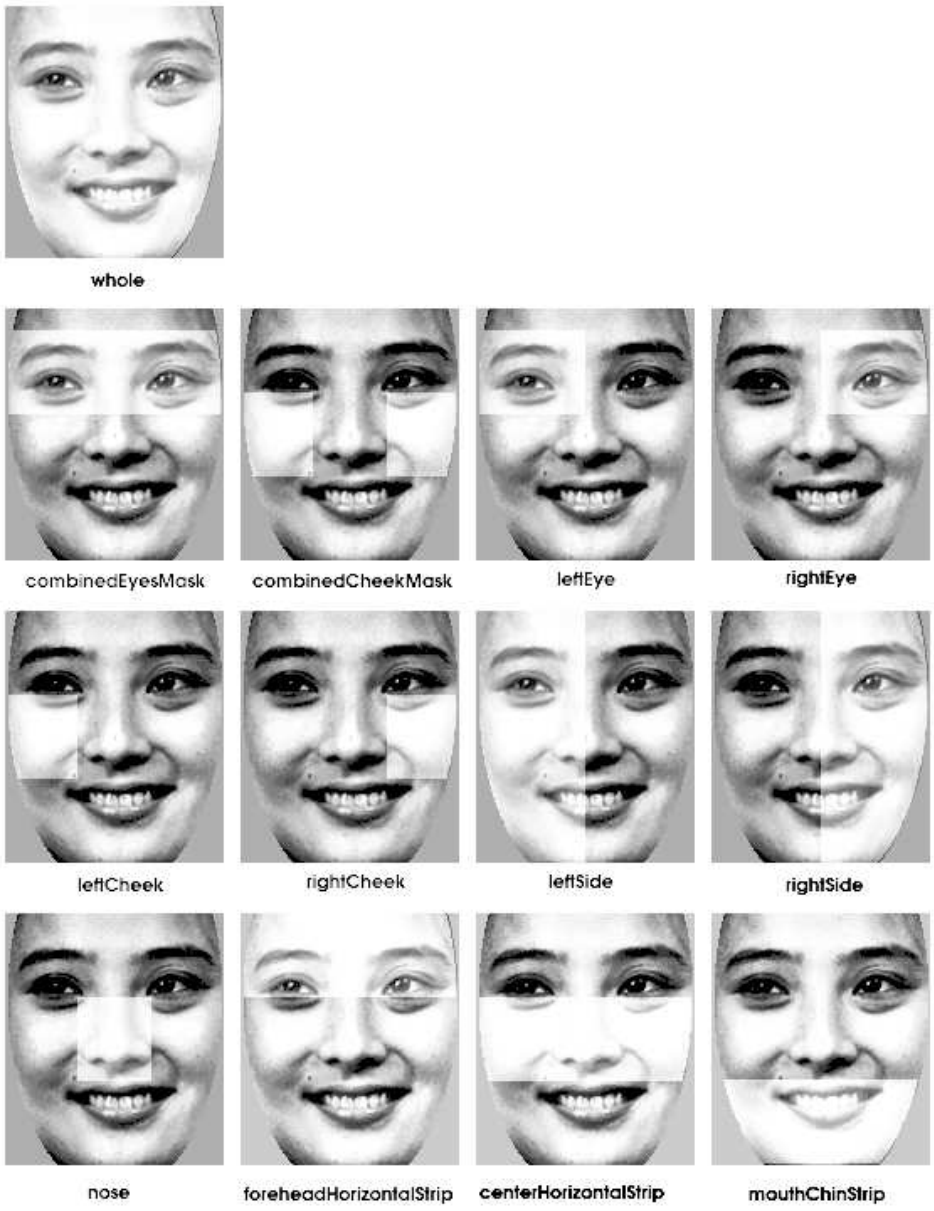


Figure 5.1: Whole face region plus twelve facial feature specific regions

7. Mean intensity for combined eye region

8. Median intensity for combined eye region

9. Standard deviation of intensity for combined eye region

10. MAD intensity for combined eye region

11-14. Like 7-10, for forehead horizonal strip

15-18. Like 7-10, for center horizonal strip

19-22. Like 7-10, for mouth/chin horizontal strip

5.1.3 Regional Contrast Predictors

The 30 *regional contrast predictors* attempt to capture variations between different regions of the face, such as those which might be introduced by lighting conditions. We compute five statistics (difference in mean, difference in median, ratio of SD, ratio of MAD, and t-statistic) over six different pairings of different face regions.

23. Difference in mean intensity, left side vs. right side

24. Difference in median intensity, left side vs. right side

25. Ratio of SD, left side vs. right side

26. Ratio of MAD, left side vs. right side

27. T-statistic=(diff in mean)/SD, left vs. right side

28-32. Like 23-27, for left vs. right eye

33-37. Like 23-27, for forehead vs. center horizontal strips

38-42. Like 23-27, for forehead vs. mouth/chin horizontal strips

43-47. Like 23-27, for center vs. mouth/chin horizontal strips

48-52. Like 23-27, for left cheek vs. right cheek.

5.2 Experiment Setup

The experiment consisted of two parts. The first stage generated the raw data for all the image covariates over all pairs of images. The second stage analyzed the data in order to determine which covariates had any predictive powers over the performance of the algorithms.

5.2.1 Data Generation

The "csuToolsImageStats" tool was used to generate the fifty two predictor values for each of the 3,368 images for which we had eye coordinates and normalized imagery. The source images were 130x150 pixels in size and had been processed by "csuPreprocessNormalize." The output of this stage was an image covariates file which listed for each image the values for the 52 different predictors.

This data was then combined with the face covariate data and then run through the program "csuCovariateStudy." This program pairs each image with every other image of the same subject and generates an output file which contains for each pair the following information:

1. The unique filename identifying each image
2. The subject number identifying the person depicted in the two images
3. The subject covariates, which were common between the images
4. The 52 predictors, once for image 1 and then for image 2, plus the difference of these two values, for a total of 156 predictors.
5. The distance between the images, according to two face recognition algorithms (PCA using the Euclidean measure and Bayesian using the Maximum Likelihood measure).
6. The classification rank of one image with respect to the other. A rank of one indicates a success and means that the pair was correctly classified as coming from the same subject. Any other rank indicates a failure and means that a misclassification occured.

5.2.2 Predicting Success for PCA

The first question the data exploration stage aimed to answer was whether there is any predictor which could help to predict the success or failure of the PCA algorithm. We looked at the distribution of all cases in which the PCA algorithm succeeded and all the cases in which it failed. Our null-hypothesis was that these cases came from the same distribution. The alternative hypothesis was that the two distributions were different.

5.2.3 Predicting When PCA Fails and Bayesian Succeeds

The second question the data exploration stage aimed to answer was whether there was any predictor which could help to predict the cases in which the PCA or the Bayesian algorithms performed better than the other. To answer this question we looked at the distributions of the SF cases (cases in

which the PCA algorithms suceedded and the Bayesian failed) and the FS cases (cases in which the PCA algorithm failed but the Bayesian succeeded).

5.2.4 Data Exploration

For the purpose of evaluating the predictors, we did the following for each predictor:

1. Plotted the histogram of the two distributions, using stacked bars. The histograms allowed us to inspect the data and get a feel for the shape of the distribution. It would also alert us to any non-normal or multi-modal distributions which could adversely affect the statistical tests.

2. We generated box-plots that compared the two distributions for each predictor to make it easy to spot changes in the center and spread of these distributions.

3. We performed a left-sided, two-sided and right-sided t-test to assess the statistical significance of the disparity in the data. The three p-values were shown on each plot.

The results were sorted by statistical significance and then aggregated on a web page to allow for easy inspection (for each plot, we found the minimum p-value of the three t-tests and used that as a basis for sorting). Figures 5.2 and 5.3 show the histogram and box plots for the six top predictors for both cases.

5.3 Results & Conclusion

For the PCA algorithm, we found 125 predictors whose distributions showed a statistically significant difference between the failure and success distributions (at a five percent confidence level). Figure 5.2 shows the first six of these predictors. For the PCA vs. Bayesian results, 50 predictors showed statistical significance. The first six of these results are shown in figure 5.3. In either case, we see that the forehead (FH), center strip (CENTR) and mouth/chin (MC) strips are implicated. It is our suspicion that these measurements capture lighting variations on the face which affect the difficulty of recognition. These study suggests that computing these predictors may help to obtain a confidence value for the recognizer. Future work might involve looking at the ROC curves for each of these predictors and seeing how well they serve to improve the classification accuracy of a hybrid algorithm.

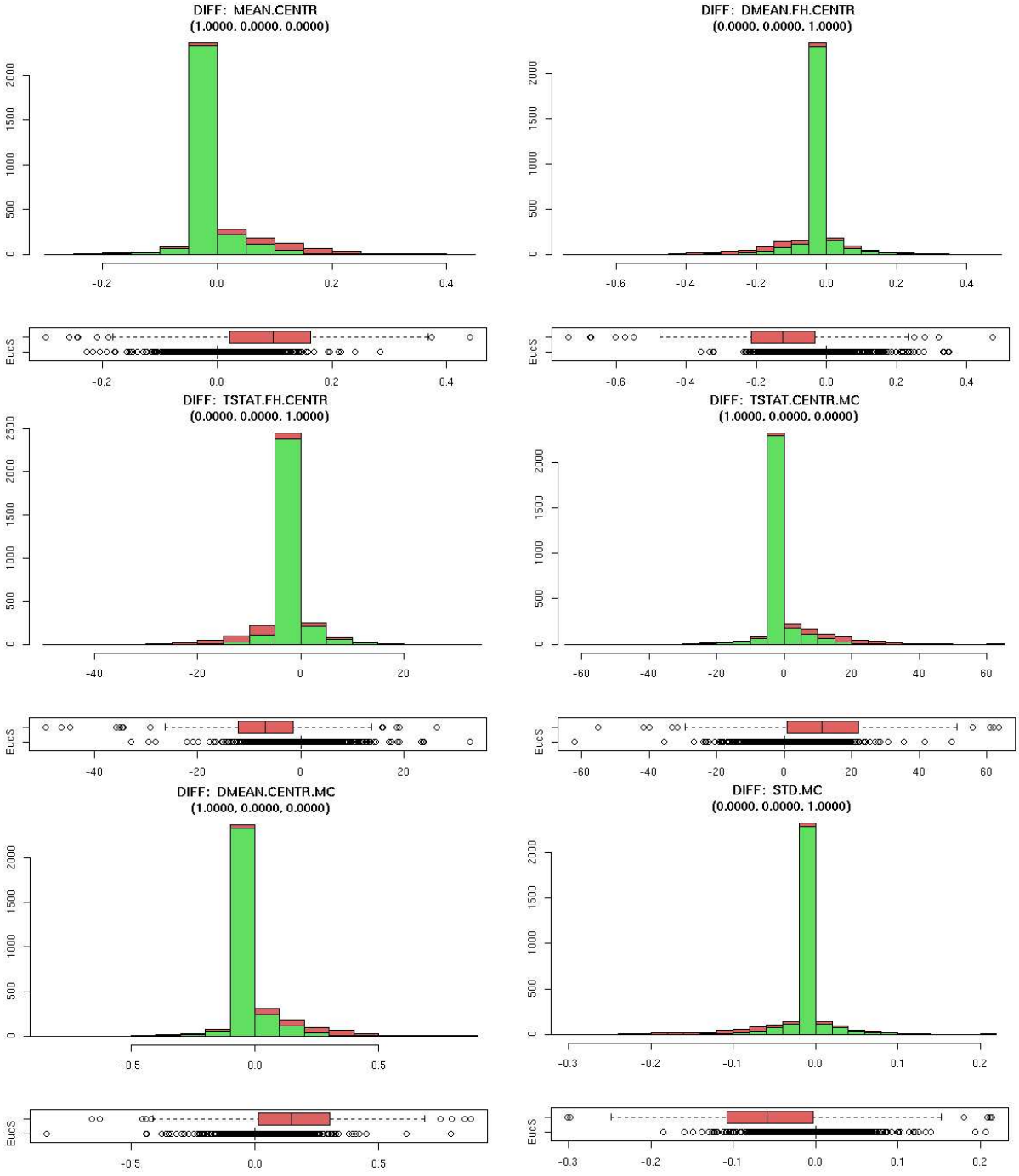


Figure 5.2: A histograms and box plots for the top predictors of the success of the PCA algorithm using the Euclidean distance measure.

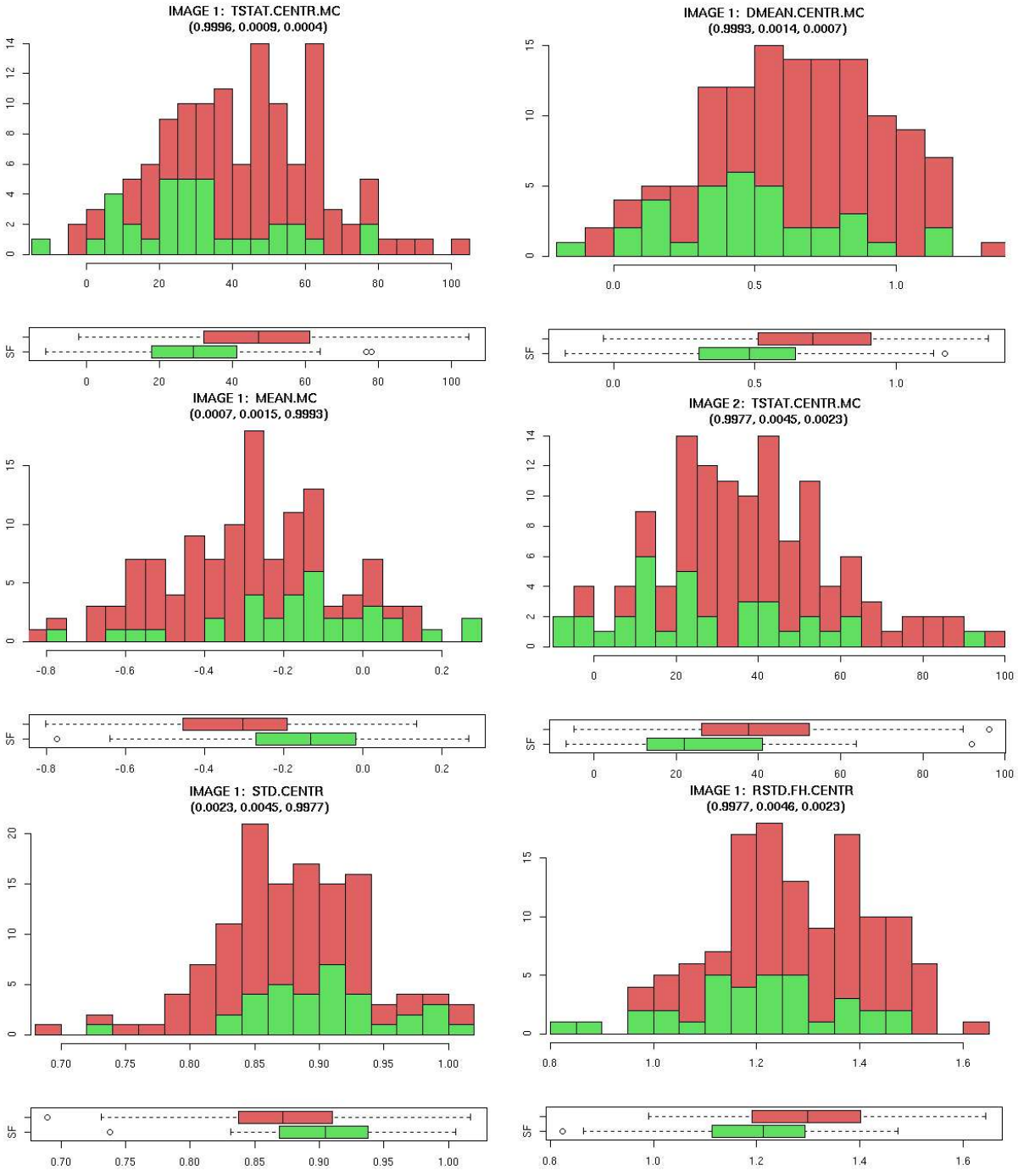


Figure 5.3: Histograms and box plots for the top predictors of the success of BIC algorithm using the ML distance measure over the PCA algorithm using the Euclidean distance measure.

Whole Image Predictors:

Region	Mean	Median	SD	MAD
Whole Face	1) MEAN.WHOLE	2) MED.WHOLE	3) STD.WHOLE	4) MAD.WHOLE

Regional Predictors:

Region	Mean	Median	SD	MAD
Combined Eye	7) MEAN.EYES	8) MED.EYES	9) STD.EYES	10) MAD.EYES
Forehead	11) MEAN.FH	12) MED.FH	13) STD.FH	14) MAD.FH
Center	15) MEAN.CENTR	16) MED.CENTR	17) STD.CENTR	18) MAD.CENTR
Mouth/Chin	19) MEAN.MC	20) MED.MC	21) STD.MC	22) MAD.MC

Regional Contrast Predictors:

Compare Regions		Difference in		Ratio of		
Region 1	Region 2	Mean	Median	SD	MAD	t-statistic
Left Side	Right Side	23) DMEAN.LR.SIDES	24) DMED.LR.SIDES	25) RSTD.LR.SIDES	26) RMAD.LR.SIDES	27) TSTAT.LR.SIDES
Left Eye	Right Eye	28) DMEAN.LR.EYES	29) DMED.LR.EYES	30) RSTD.LR.EYES	31) RMAD.LR.EYES	32) TSTAT.LR.EYES
Forehead	Center	33) DMEAN.FH.CENTR	34) DMED.FH.CENTR	35) RSTD.FH.CENTR	36) RMAD.FH.CENTR	37) TSTAT.FH.CENTR
Forehead	Mouth/Chin	38) DMEAN.FH.MC	39) DMED.FH.MC	40) RSTD.FH.MC	41) RMAD.FH.MC	42) TSTAT.FH.MC
Center	Mouth/Chin	43) DMEAN.CENTR.MC	44) DMED.CENTR.MC	45) RSTD.CENTR.MC	46) RMAD.CENTR.MC	47) TSTAT.CENTR.MC
Left Cheek	Right Cheek	48) DMEAN.LR.CHEEK	49) DMED.LR.CHEEK	50) RSTD.LR.CHEEK	51) RMAD.LR.CHEEK	52) TSTAT.LR.CHEEK

Table 5.1: Abbreviations for the image covariates

Chapter 6

Conclusion

6.1 Summary

In chapter 3, we demonstrated that the probability equation for $P(\Delta|\Omega)$ as defined by Moghaddam and Pentland cannot in many practical cases be computed. We have derived an algebraically equivalent score for the ML classifier that avoids the problem. In addition, by applying the same techniques to the MAP classifier, we were able to derive a simpler score for the classifier than was originally presented in [7]. Our simplified scores circumvent some of the troublesome aspects of the MAP classifier, eliminating, for example, the need to choose values for the priors $\widehat{P}(\Omega_E)$ and $\widehat{P}(\Omega_I)$. This eliminates one potential dimension for algorithm tuning.

Chapter 4 showed that the hybrid system has quite a bit of potential. In all cases, it performs better than mahalanobis angle nearest-neighbor classifier, the most effective of the subspace classifiers. For the FB and FC data sets, we see that choosing a small candidate set in the first stage of the algorithm leads to improved accuracy of the combined classifier over the independent algorithms. This effect is non-existant in the DUP1 and DUP2 data sets: for small candidate sets, the combined algorithm out-performs the nearest-neighbor classifier but performs significantly worse than the bayesian intrapersonal classifier working independently. We have demonstrated that a two-stage classifier is beneficial in cases in which using the bayesian intrapersonal classifier on the entire gallery of images is impratical. In such cases, we can improve on traditional nearest-neighbor classifier by applying the bayesian intrapersonal classifier only to a small set of top-ranked candidate images.

Chapter 5 asked whether there were any image-derived statistics which could help us predict whether the face recognition algorithms would succeed or fail on a pair of images. In this work, 156 predictors were examined under two different situations. Statistical tests were performed to determine whether the failure and success cases were drawn from different distributions. The results were promising. For the PCA algorithm, 125 predictors out of 156 proved significant. For the

Factor	Index	Values
Training Set	t	0,...,15
Gallery Set	g	0,...,29
Probe Image	p	0,...,255
Metric	m	ML or MAP
PCA Space Dimension	d	100% or $E_{60\%}$
Training Set Inclusion Flag	f	Yes or No
Hybrid Training	h	Yes or No

Table 6.1: Summary of the configuration and replication factors used in the experimental design

PCA/BIC comparison, 50 predictors out of 156 proved significant. The aim of these tests was to remove from the candidate pool of predictors those which failed to exibit statistical significance. Further work will attempt to evaluate these predictors in terms of their adequacy for use in a threshold classifier.

6.2 Future Work

6.2.1 Configuration Space Study

An important consideration for any algorithm is the proper selection of configuration parameters. In section 4.4 we began exploring the effects of configuration parameters on algorithm performance. The Bayesian Interpersonal Extrapersonal Classifier (BIC) Configuration is a larger and more comprehensive study that also captures the effect of changing the training or gallery sets.

The configuration study will be run on 204 subjects with 4 replicates per subject, for a total of 816 images. For each of these subjects, the first of the four is chosen as the probe image. The remaining three images are used to build galleries and for algorithm training. Throughout the study 30 galleries are used: these galleries selected for each subject one of the three remaining images. There are 16 different training sets, eight with 16 subjects and another eight with 64 subjects. These training sets contain all three of the replicates for each subject, for a total of 48 and 192 images respectively.

Table 6.1 shows the experimental variables that will be varied. Each probe image is matched $16 \times 30 \times 2 \times 2 \times 2 = 3840$ times, corresponding to changing the variables t, g, m, $d$and h.

REFERENCES

[1] P.Rauss P.J.Philips, H.Moon and S.A.Razvi. The feret evaluation methodology for face recognition algorithms. IEEE Conference on Computer Vision and Pattern Recognition.

[2] "Analyzing PCA-based Face Recognition Algorithms: Eigenvector Selection and Distance Measures," Wendy S. Yambor, Bruce A. Draper and J. Ross Beveridge, 2nd Workshop on Empirical Evaluation in Computer Vision, Dublin, Ireland, July 1, 2000.

[3] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-19(7):696-710, July 1997

[4] B. Moghaddam, W. Wahid, and A. Pentland. Beyond Eigenfaces: Probabilistic Matching for Face Recognition. In *The 3rd IEEE Int'l Conference on Automatic Face and Gesture Recognition*, pages 30-35, Nara, Japan, April 1998

[5] M. Kirby *Geometric Data Analysis: An Empirical Approach to Dimensionality Reduction and the Study of Patterns.* Wiley 2000

[6] M. Kirby. *Dimensionality Reduction and Pattern Analysis: An Empirican Approach.* Wiley, 2000

[7] Kai She. Evaluation of Face Recognition Algorithms. Master's Thesis, Colorado State University, 2002

[8] D. Swets and J. Weng, "Using Discriminant Eigenfeatures for Image Retrieval." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):831-836. 1996

[9] J. Ross Beveridge, D. Bolme, M. Teixeira, B. Draper. The CSU Face Identification Evaluation System User's Guide: Version 5.0. Technical report, Colorado State University, 2003

[10] G. Givens, J. Ross Beveridge, B. Draper, D. Bolme, A Statistical Assessment of Subject Factors in the PCA Recognition of Human Faces