Parametric and Nonparametric Methods for the Statistical Evaluation of Human ID Algorithms

J. Ross Beveridge, Kai She and Bruce A. Draper
Computer Science Department
Colorado State University
Fort Collins, CO, 80523

Geof H. Givens
Statistics Department
Colorado State University
Fort Collins, CO, 80523

Abstract

This paper reviews some of the major issues associated with the statistical evaluation of Human Identification algorithms, emphasizing comparisons between algorithms on the same set of sample images. A general notation is developed and common performance metrics are defined. A simple success/failure evaluation methodology where recognition rate depends upon a binomially distributed random variable, recognition count, is developed and the conditions under which this model is appropriate are discussed. Some nonparametric techniques are also introduced, including bootstrapping. When applied to estimating the distribution of recognition count for a single set of i.i.d. sampled probe images, bootstrapping is noted as equivalent to the parametric binomial model. Bootstrapping applied to recognition rate over resampled sets of images can be problematic. Specifically, sampling with replacement to form image probe sets is shown to introduce a conflict between assumptions required by bootstrapping and the way recognition rate is computed. In part to overcome this difficulty with bootstrapping, a different nonparametric Monte Carlo method is introduced, and its utility illustrated with an extended example. This method permutes the choice of gallery and probe images. It is used to answer two questions. Question 1: How much does recognition rate vary when comparing images of individuals taken on different days using the same camera? Question 2: When is the observed difference in recognition rates for two distinct algorithms significant relative to this variation? Two important general features of nonparametric methods are illustrated by the Monte Carlo study. First, within some broad limits, resampling generates sample distributions for any statistic of interest. Second, through careful choice of an appropriate statistic and subsequent estimation of its distribution, domain specific hypotheses may be readily formulated and tested.

1 Introduction

Due in part to the FERET evaluations [12], much of the protocol for comparing human identification algorithms, or at least human face recognition algorithms, has been formulated and standardized. This section will review this formulation and establish the mathematical notation needed to address questions of statistical evaluation. This formalization is essential in so far as it gives all of us working in evaluation a common frame of reference. However, it should not be taken as all encompassing: inevitably aspects central to some current and future evaluation tasks will be missing.

Section 2 introduces a compact notation for describing populations and samples. This notation is summarized for convenience in Table 1. It also introduces the three critical image sets used in evaluation: the training set, gallery and probe set. Section 3 reviews how most human identification algorithms are developed such that their performance depends firstly upon how they are trained and secondly upon what gallery images they are provided. This section also reviews the similarity matrix that may be pre-computed for most human identification algorithms and subsequently used to efficiently conduct "virtual" experiments. Section 4 formally defines recognition rate, median rank and median censored rank as distinct performance evaluation metrics.

Section 5 introduces a simple binomial model for the outcomes of testing recognition algorithms applied to human identification data. The assumptions underlying this model are discussed, as well as the practical significance of violating these assumptions under some common scenarios. We show how to use this binomial model to formulate and test hypotheses of the form: "algorithm A is performing better than algorithm B". Specifically, McNemar's test is provided as a simple, direct test when algorithms are tested on common data, i.e. paired testing. McNemar's test is used in an example comparing PCA to ICA face recognition algorithms on the FERET data.

Section 6 introduces some nonparametric techniques based upon resampling. In particular bootstrapping and the

Table 1. Notation Summary

Symbol	Usage
Ω	Target population of images/people to be recognized.
$\omega_{i,j}$	The jth image of person i in Ω.
W	The finite set of images available for performance evaluation; the sampled population.
$w_{i,j}$	The jth image of person i in W.
T	Training images, typically $T \subseteq X$
G	Gallery images, typically $G \subseteq X$
P	Probe images, typically $P \subseteq X$
A	An algorithm
A_T	An algorithm trained on T.
A_{TG}	An algorithm trained on T and using G as exemplars.
U	Union of all gallery and probe sets used for a set of experiments, $U \subseteq W$
$u_{i,j}$	The jth image of person i in U.
υ	A similarity relation between pairs of images.
Υ_U	Pairwise similarity matrix for U.
$x_{i,j}$	A probe image in P.
$y_{k,l}$	A gallery image in G.
$L(x)$	Gallery images sorted by decreasing similarity υ to image x.
L_ℓ	The ℓth element of L.
$\rho(P)$	Recognition rate of an algorithm on probe set P.
$r(x)$	Rank of first correct match for probe image x.
$r_\tau(x)$	Rank censored at maximum value τ.
$\tilde{r}_\tau$	Median of censored rank r_τ over probe set P.
$b(x)$	Success indicator function for an algorithm applied to probe image x.
s_ℓ	The ℓth randomly selected probe image in the sequence S.
S	A random sequence of probe images or a results of indicator function applied to them.
$P[e]$	The probability of some event e.

assumptions underlying it are briefly discussed. In this section, it is observed that applying bootstrapping to the particularly simple dataset consisting of a collection of success/failure outcomes leads back to the binomial model developed in Section 5. Section 7 provides a more detailed example of another Monte Carlo resampling procedure based upon permuting the choice of gallery and probe images. In this context, a difficulty with the direct application of bootstrapping is circumvented.

Section 7 illustrates two advantages of nonparametric methods relative to more traditional parametric methods. First, it illustrates that sample distributions for relevant statistics may be obtained directly without a need to know the exact distribution of the data. Second, this freedom provides wide latitude to develop domain-specific statistics that greatly simplify the direct statement of interesting hypotheses. In other words, we can formalize the general hypothesis "algorithm A is performing better than algorithm B" in the precise manner desired to provide the most interpretable result.

2 Data

Let Ω denote a target population of images over which our goal is to characterize algorithm performance. Denote elements of Ω as $\omega_{i,j}$ where the subscripts have the following meaning: $\omega_{i,j}$ is the jth image of the ith person. Using double subscripting highlights the distinction between images of a single person and images of different people. In practice, we as evaluators have access to only the sampled population, a finite set $W \subseteq \Omega$. For example, when evaluating algorithms on frontal face images using the FERET data set, W contains $3,816$ images of $1,203$ people. Generalization from W to Ω rests on judgment about the nature of each and about the sampling process giving rise to W.

When evaluating algorithms, three subsets of W are of interest:

- The training set T,
- The gallery set G,
- The probe set P.

This trinary distinction differs from the more common binary distinction used in machine learning, where one speaks

of training and test data. Nor does it match the trinary distinction of training, validation and test sometimes used in machine learning. The usage of T and P is essentially identical to that of training and test data in machine learning. However, the gallery contains exemplars for the people to be recognized. It has no direct correlate in the common machine learning nomenclature.

In some cases, T and G may be the same set. In other words, an algorithm may be trained on a set of images and then subesquently the trained algorithm may use the same set as examplars against which to match probe images. However, this need not be the case. For example, in the FERET tests in 1996 and 1997, algorithms were typically trained on a different set of images than those used as the gallery G.

The actual performance of an algorithm is always rated relative to how well the images in P are matched to the images in G. So, for example, performance is perfect on a given probe set P if every image $w_{i,j} \in P$ is matchd to an image $w_{i,k} \in G$. In other words, the probe image $w_{i,j}$ is matched to a different image $w_{i,j}$ of the same person: the ith person. It follows that G and P should be disjoint, otherwise the problem becomes trivial. Further, while not always the case, typically G contains no more than one image of each person. More will be said aobut how T, G and P relate to algorithm and performance measures in the sections that follow.

3 Algorithms

How a given algorithm A will behave depends upon how it is trained and the quality of the exemplars it is given. Thus, an instance of a class of algorithms, such as a principal components analysis (PCA) algorithm, is defined in part by the training and in part by the gallery. For example, the training phase for a PCA algorithm uses the images in T to define a subspace in which to operate a nearest neighbor classifier. The gallery G provides the exemplars for each of the people to be recognized. For example, in a PCA algorithm these images are projected into the subspace and become exemplars to which novel probe images are compared by the nearest neighbor classifier. For other types of face recognition algorithms, the usage of T and G will differ, but for almost all algorithms they are utilized in some manner or another. Thus, for any algorithm A, this dependency is indicated by subscripting, e.g. A_{TG} is algorithm A trained on T and using gallery G.

3.1 Similarity Matrices

Most commonly used recognition algorithms may be characterized by a similarity matrix that represents the information used to perform classification. A similarity measure υ is a function

$$\upsilon \, : \, W \times W \Rightarrow \Re \tag{1}$$

Similarity is used to rank gallery images relative to a specific probe image. Thus, the best match to a probe image $p_i \in P$ is the gallery image $g_j \in G$ such that:

$$\upsilon\left(p_i, g_j\right) > \upsilon\left(p_i, g_k\right) \quad \forall \quad g_k \in G, \; g_j \neq g_k \tag{2}$$

The only condition the similarity relation υ must satisfy to perform this function is that it must induce a complete order on the set of images W. In practice ties may arise. So long as ties are rare, it is probably safe to impose arbitrary choices and otherwise ignore the problem. However, a similarity measure that gives rise to ties gives rise to a weak order rather than a complete order, and this in turn leaves the performance measures such as recognition rate ill-defined.

In most cases, similarity is derived from a distance measure. For example, a PCA algorithm may define similarity as follows:

$$\upsilon\left(w_{i,j}, w_{k,l}\right) = \frac{1}{L_1\left(w_{i,j}, w_{k,l}\right) + \epsilon} \tag{3}$$

where L_1 is the L1, city block, distance between images $w_{i,j}$ and $w_{k,l}$ measured in the PCA subspace. The small constant ϵ prevents the similarity measure from becoming undefined if $L1$ is zero.

The FERET studies made extensive use of the fact that once an algorithm is fixed through training, the similarity υ assigned to a pair of images is a constant. Therefore, for an algorithm A_{TG}, experiments may be conducted in a "virtual" fashion. To illustrate, in the FERET studies [12], most of the results presented utilized one training set, one gallery, and four probe sets. An algorithm was run once on the union of all gallery and probe sets and tests were run using similarity information cached in a similarity matrix. Formally, the FERET studies used the pooled set of images

$$U \; = \; G_{fa} \; \cup \; P_{fb} \; \cup \; P_{dupI} \; \cup \; P_{dupII} \; \cup \; P_{fc}, \tag{4}$$

with corresponding similarity matrix

$$\Upsilon_U \; = \; \upsilon(u_{i,j}, u_{k,l}) \quad \forall \quad u_{i,j}, \; u_{k,l} \in U. \tag{5}$$

A short description of these partitions is given in table 2.

The four probe sets were used in conjunction with the single gallery set to compare algorithm performance. The exact images in each of these sets is enumerated in lists available at [3]. This site also tabulates the overlap between sets both in terms of people in common and images in common. All 234 images in the probe set P_{dupII} are also included in P_{dubI}. This overlap between probe sets in unusual. More often, probe sets are disjoint, as is the case with

Table 2. Five Paritions Used in Most of the FERET 1996/97 Evaluations.

Set	Images	Description
G_{fa}	1,196	Images taken with one of two facial expressions: neutral versus other.
P_{fb}	1,195	Images taken with a other facial expression.
P_{dupI}	722	Subjects taken later in time.
P_{dupII}	234	Subjects taken later in time, this is a harder subset of Dup I.
P_{fc}	194	Subjects taken under different illumination.
T	501	Training Images, roughly 80 percent from G_{fa} and 20 percent from dup I.

the other three: the intersection of P_{fb}, P_{dupI} and P_{fc} is the null set.

The matrix Υ_U is useful in several ways. As was done in the FERET studies, it allows different virtual experiments to be conducted without running the algorithms again. It also supports quickly executing many thousands of virtual experiments associated as required by nonparametric statistical techniques such as bootstrapping [7]. Such nonparametric techniques may compute recognition rates for algorithms subject to thousands of different choices of gallery and probe images. Using Υ_U, these experiments are conducted without actually running the recognition algorithm, and in many cases this makes such repetition computationally tractable.

Unfortunately, when variances in performance associated with changes in the training data T are to be investigated, this trick of precomputing a similarity matrix Υ_U breaks down. This is because two algorithm variants, A_{T_1G} and A_{T_2G}, trained on different sets T_1 and T_2, will typically yield different similarity matrices. So, for example, a PCA algorithm [10] requires that a new subspace projection be constructed for each new set of training images.

4 Performance Measures

4.1 Recognition Rate

Recognition rate is a common measure used to evaluate performance. The first step in defining recognition rate is defining what it means to successfully recognize a probe image $x_{i,j} \in P$. This may be done as follows. For each probe image $x_{i,j}$, sort the gallery images $y_{k,l} \in G$ by decreasing similarity, yielding a list $L = \{L_1, L_2, \ldots\}$. Thus, L_1 is the gallery image closest to the probe image $x_{i,j}$, L_2 is the next closest gallery image, and generalizing, L_ℓ is the ℓth closest gallery image. An algorithm successfully recognizes the ith person from probe image $x_{i,j}$ if, for the closest gallery image $y_{k,l} = L_1$, index i equals index k. In plain English, the algorithm successfully recognizes a person if the probe image and top ranked gallery image are of the same person.

The success criterion may be expressed in the form of an indicator function b. Thus, for an algorithm A_{TG}:

$$b(x_{i,j}) = \begin{cases} 1 & \text{when } A_{TG} \text{ correctly recognizes} \\ & \text{probe image } x_{i,j} \in P \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

The requirement that an image of the same person appear at L_1 may be relaxed such that an algorithm is said to succeed if an image of the same person appears in the closest τ gallery images in L. This gives rise to a family of indicator functions $b_\tau(x_{i,j})$.

For a fixed choice of τ, the recognition rate over a probe set P of size n is:

$$\rho_\tau(P) = \frac{c_\tau}{n} \quad \text{where} \quad c_\tau = \sum_{x_{i,j} \in P} b_\tau(x_{i,j}) \quad (7)$$

What to do in the event of a tie is typically not specified. When using methods such as PCA, where distances are measured in high dimensional subspaces using double floating point arithmetic, ties are unlikely and so seldom considered. However, one can easily imagine other algorithms where ties might require explicit treatment. There is a deeper question as to when differences are meaningful, but such questions are not easily answered in the general case, and if dealt with for specific choices of imagery, the analysis techniques employed will resemble some of the statistical methods set out below.

4.2 Rank and Censored Rank

An alternative to choosing τ in the above definition of recognition rate and then scoring an algorithm as succeeding or failing is to report the rank at which an algorithm first succeeds on a given probe image $x_{i,j}$. Using the same sorted list of gallery images L defined above, let $L_\ell = y_{k,l}$ be the first gallery image in the sequence that is of the same person as the probe image $x_{i,j}$, i.e. index i equals index k. Thus, if ℓ is the rank of the first successful match, then the most obvious rank measure r for evaluating performance on probe image $x_{i,j}$ is:

$$r(x_{i,j}) = \ell \quad (8)$$

At some point, differences between large rank values don't matter: missing by 10 may be considered as bad as missing by 100. This line of thinking gives rise to the censored rank:

$$r_\tau\left(x_{i,j}\right) \;=\; \min(r\left(x_{i,j}\right), \tau) \tag{9}$$

So, for example, r_{10} is the censored rank where the clamp is placed at 10. If $r_{10}(x_{i,j}) \;=\; 1$, then probe image $x_{i,j}$ matches a gallery image of the same person at rank 1. However, if $r_{10}(x_{i,j}) \;=\; 10$, it implies only that probe image $x_{i,j}$ matched a gallery image of the same person at a rank value greater to or equal to 10.

Censored rank is defined relative to a single probe, and unlike recognition rate, is not defined for a set of probes P. Some additional summary statistic must be chosen if a single value based upon censored rank is to be used to characterize performance over a probe set. One such statistic is the median, and this choice gives rise to the median censored rank $\widetilde{r}_\tau$

$$\widetilde{r}_\tau\left(P\right) \;=\; \text{median}\left(\left[r_\tau(x) \quad \forall \quad x \;\in P\right]\right) \tag{10}$$

Median censored rank exemplifies statistics for which traditional parametric techniques are of limited value, since the probability distribution for $\widetilde{r}_\tau$ is not typically of a well understood parameterized form; see section 6.2.

5 Algorithm Testing as Bernoulli Trials

All recognition algorithms considered here are deterministic. This follows from the presumption that the distance between two images is fully defined once algorithm A is trained on training imagery T. Thus, at the risk of stating the obvious, A_{TG} either succeeds or fails on $x_{i,j}$ and there is nothing random or uncertain about it.

Now consider what happens when A_{TG} is applied to a collection of probe images drawn at random from a larger population of possible probe images. Let s_ℓ denote the ℓth randomly selected probe image [1]. Suppose that the population from which the s_ℓ are drawn has the property that

$$P\left[b\left(s_\ell\right) = 1\right] = p \tag{11}$$

In words, this equation states that the probability of algorithm A_{TG} recognizing a randomly selected probe image s_ℓ is p.

To provide some intuition for this model, let us relate our selection of probe images to a classic and simple experiment. You are given a jar containing a mix of red and green marbles. You reach into the jar and select a marble at random, record a 1 if the marble is green and a 0 otherwise. You then place the marble back into the jar. If you do this n times, you are conducting Bernoulli trials, and the number of 1's recorded is a random variable described by a binomial distribution. The distribution is parameterized by the number of marbles drawn, n, and by the fraction of all the marbles that are green, p. The equation for our marble drawing experiment that is analogous to equation 11 above is:

$$P\left[\text{marble drawn is green}\right] = p \tag{12}$$

It should be obvious for the jar of marbles that the probability of any given randomly selected marble being green is the ratio p of green marbles to the total number of marbles. Likewise for the randomly selected probe images. Some fraction p of the probe images in the population of possible probe images P are correctly recognized by algorithm A_{TG}.

The fraction p dictates how the indicator function b will behave over a sequence of independently selected probe images s_ℓ. It also characterizes how the recognition rate defined in equation 7 behaves. Restating this equation for rank $\tau \;=\; 1$.

$$\rho\left(P\right) \;=\; \frac{c}{n} \quad \text{where} \quad c \;=\; \sum_{x_{i,j}\,\in P} b(x_{i,j}) \tag{13}$$

It follows directly from the assumptions above that c is a binomially distributed random variable,

$$P\left[c = k\right] \;=\; \left(\frac{n!}{k!\;(n-k)!}\right) p^k\, q^{n-k} \quad p + q = 1 \tag{14}$$

5.1 Appropriateness of a Binomial Model

This binomial model is extremely simple, and there are several aspects where further consideration is warranted. The following discussion may help eliminate some areas of concern.

5.1.1 Sampling With and Without Replacement

In terms of the mathematically correctness of a model, the difference between sampling with replacement and sampling without replacement seems fundamental. This is a concern because most human identification experiments do not sample probe images with replacement. Thus, the question arises as to how a model built upon the assumption of sampling with replacement may be used in experiments where sampling is done without replacement.

This concern is of little practical significance. If one assumes the target population is large relative to the sample size, then the probability of sampling the same instance twice is low and the difference between sampling with and

[1] The switch from double to single subscripting of images is intentional. The single subscript emphasizes that the index refers to placement in the collection of randomly selected images. Clearly $s_\ell \;=\; x_{i,j}$ for some i and j, but random sampling means that which i and j is not known.

without replacement negligible. While not always the case, the norm in human identification experiments is that the number of samples (probe images) is much much smaller than the population over which we are attempting to draw statistical inferences about the performance of a recognition algorithm.

5.1.2 Some People are Harder to Recognize

Without question, some individual people are harder for an automated algorithm to recognize than others. Is this a concern for the success/failure model? The answer hinges upon what one assumes about sampling. If sampling of probe images is indeed done independently and at random, and if the sampled population, P, is representative of the target population, Ω, then the fact that some people are harder to recognize than others is irrelevant. However, one can easily imaging situations where this is not true.

A helpful way to illustrate that the binomial model proposed above applies even when we know that some individuals are harder to recognize than others is to extend the marble example. Instead of selecting a marble from a jar, consider selecting one from a cabinet with many drawers. Selection operates by first selecting a drawer at random, and next selecting a marble from that drawer at random.

Assume different drawers have different fractions of green and red marbles. Thus, for this experiment, the following is true:

$$P\left[\text{marble drawn is green} \mid \text{marble is drawn from drawer } i\right] = p_i \tag{15}$$

However, since the drawer itself is selected at random, this experiment is equivalent to an experiment where first all the marbles are poured out of the drawers into a single jar, are mixed, and then a marble is selected from the jar. Thus, it becomes equivalent to the earlier experiment. If there are m drawers with n_i marbles in the ith drawer, then the probability of drawing a green marble from a randomly selected drawer is a weighted average of the probabilities for each drawer, i.e.

$$p = P\left[\text{marble drawn is green}\right] = \frac{1}{m}\sum_{i=1}^{m} n_i p_i \Big/ \sum_{i=1}^{m} n_i \tag{16}$$

To finish the analogy, consider that each person in the probe set corresponds to a drawer, and each image of that person corresponds to a marble in that drawer. Then, although people have unequal probabilities (p_i) of being successfully recognized by A_{TG}, the unconditional probability that A_{TG} correctly recognizes a randomly selected probe image is p. As long as P is representative of Ω, it is not a problem if there is only one probe image per person in P because this argument can equally be applied to the sampling of P from Ω.

	Outcome of A_{TG}	
Outcome of B_{TG}	S	F
S	73	27
F	2	23

Table 3. Hypothetical summary of paired recognition data suitable for McNemar's test. Here 'S' means that the algorithm correctly identified a probe, and 'F' represents a failure.

Of course, when individual drawer attributes (i.e. the p_i) are known, elementary sampling techniques [4] show that more powerful tests and estimators than the ones we will propose can be constructed. However, it is unreasonable in the human identification context to assume known p_i, and it would eliminate the distinction between probe and gallery sets if the p_i were considered estimable.

5.2 Hypothesis Testing using the Binomial Model

Given the binomial model, a simple and natural approach for the comparison of performance of two algorithms is McNemar's test [8, 14]. This test is suitable for paired data: for example, data generated from testing two trained algorithms with the same gallery and probe sets. One can imagine cases where a comparison is made using two independent datasets. In this case, standard statistical methods for comparing two population proportions (eg. [6]) could be applied.

The paired data from applying two recognition algorithms, say A_{TG} and B_{TG}, to the same set of gallery and probe images is naturally summarized as in table 3. The outcomes tabulated here can be labeled as SS, SF, FS and FF, where SS means both algorithms succeed, SF means that A_{TG} succeeds and B_{TG} fails, and so on. There were 125 probes in this hypothetical example, of which 73 were correctly identified by both algorithms and 23 were incorrectly identified by both algorithms. The recognition rate for A_{TG} was $75/125 = 0.60$, and $100/125 = 0.80$ for B_{TG}. Clearly, the comparison between the algorithms boils down to comparing the relative frequencies of SF and FS.

5.2.1 Paired Success/Failure Trials: McNemar's Test

McNemar's test begins be discarding those cases where the outcome is either SS or FF. For the remaining outcomes, SF and FS, a sign test is used. The null hypothesis, H_0, is that the probability of observing SF is equal to that of observing FS. Let n_{SF} denote the number of times SF is observed

and n_{FS} denote the number of times FS is observed. We are interested in the one sided version of this test, so without loss of generality consider the alternative hypothesis, H_{Alt}, to be that $P[SF] > P[FS]$, i.e. A_{TG} fails less often than B_{TG}. Under H_0, a mismatched outcome (i.e. either SF or FS) is equally likely to favor A_{TG} or B_{TG}. Therefore, under H_0,

$$\begin{array}{rcl} P[\text{at least } n_{SF} \text{ mismatches favor } A_{TG}] & = & \\ P[\text{at most } n_{FS} \text{ mismatches favor } B_{TG}] & = & \\ \sum_{i=0}^{n_{FS}} \frac{n!}{i!(n-i)!} 0.5^n & & \end{array} \quad (17)$$

where $n = n_{SF} + n_{FS}$. This probability is the p-value for rejecting H_0 in favor of H_{Alt}.

5.3 Illustrating McNemar's Test: Comparing PCA and ICA

As an example of using a binomial evaluation methodology, we compare recognition rates for principal components analysis (PCA) and independent components analysis (ICA) for faces in the FERET face data base. This example is a short summary of results presented in [9]. The results of comparing PCA and ICA on the FERET data set are given in Table 4. The algorithms are compared by measuring how often a probe image matches the nearest gallery image. (i.e. the comparison is for rank 1). The distance measure used was L1 norm for PCA, and cosine for ICA (as recommended by [1]).

The first two columns in Table 4 indicate the gallery and probe set respectively. Next the number of correctly recognized images over the total number of images is shown both as a rational number and a percentage for first the PCA and then the ICA algorithm. Next, the count for each possible outcome for the paired tests are shown. The four possible outcomes are:

SS Both the PCA and ICA algorithm recognize the image.

SF The PCA algorithm recognizes the images and the ICA algorithm does not.

FS The ICA algorithm recognizes the images and the PCA algorithm does not.

FF Both algorithms fail to recognize the image.

Finally, the p-value for the null hypothesis H_0 is shown for each of the four experiments. When p-value is below 0.0001 this is indicated by < 0.0001. Under the binomial model, all four tests indicate that the PCA algorithm is significantly better than ICA on every probe set.

5.4 Simplicity of the Binomial Model

The strength of the binomial model for comparing recognition algorithms is its relative simplicity. We are confident others can quickly understand the nature of the model and how to perform tests such as McNemar's as a first order check on the statistical significance of an observed difference in recognition performance. That said, the model has weaknesses.

A weakness of the binomial model is reliance upon assumptions that may or may not be reasonable for a given experiment. When assumptions are violated, will significant errors ensue that in turn lead to false conclusions? As discussed above, there are circumstances under which violating an assumption, for example sampling without replacement, is not of significant practical concern. However, under other circumstances, a violated assumption may lead to erroneous results. Finally, and perhaps most troubling, it is not always easy for researchers with only modest statistical training to distinguish between these two cases.

6 Nonparametric Methods Generally and Bootstrapping Specifically

Computer-intensive nonparametric methods can free us from limiting assumptions about distributions [5]. This can be attractive if one does not feel comfortable with the binomial model. Arguably the most informative data derived from recognition experiments are not binary, and such data cannot be arranged in a simple contingency table or analyzed with McNemar's test. For example, the median censored rank statistic isn't suitable for such an analysis. Also, the simple binomial model we have presented is conditional on the gallery used, but the most relevant conclusions are unconditional. In other words, we usually wish to draw conclusions about the overall relative performance of algorithms, generalized from the single (or small number of) gallery set(s) used in testing. For all these reasons, we now turn attention to more flexible, nonparametric techniques. These methods represent a means of directly estimating probability distributions for statistics of interest.

We have recently been examining a Monte Carlo method to estimate probability distribution functions for recognition rate subject to variation in the choice of probe and gallery images for a set of 256 individuals in the FERET data set [2]. Ross J. Micheals and Terry Boult have conducted a related type of non-parametric sampling test using a technique called balanced-replicate resampling [13].

This section briefly describes bootstrapping in general and then illustrates bootstrapping on a particularly simple problem: bootstrapping the distribution of success/failure counts by sampling a set of succeeded/failed values obtained from running an algorithm A_{TG} on a single probe

Table 4. Performance of PCA and ICA. The probe sets are ordered easiest (fafb) to hardest (fafc).

		PCA		ICA		Paired Outcomes				McNemar's
Gallery	Probe Set	Corr/Total	Pct	Corr/Total	Pct	SS	SF	FS	FF	test p-value
G_{fa}	P_{fb}	928/1195	78	864/1195	72	824	104	40	227	< 0.0001
G_{fa}	P_{dupI}	277/722	38	255/722	217	60	38	407	35	0.0164
G_{fa}	P_{dupII}	52/234	22	38/234	30	22	8	174	16	0.0080
G_{fa}	P_{fc}	53/194	27	10/194	9	44	1	140	5	< 0.0001

set. For this particularly simple illustration we observe the form of the resulting distribution is already known: it is precisely the binomial distribution discussed above.

6.1 Bootstrapping

In general, a bootstrapping procedure would proceed as follows. Let S be an i.i.d. sample of size n from a larger population. Let $\widehat{\theta}$ be a statistic, namely $\widehat{\theta}(S)$, estimating a quantity of interest, θ. So, for example, θ might be the population median and $\widehat{\theta}$ the sample median. Bootstrapping allows us to estimate the probability distribution function for $\widehat{\theta}$. (Of course, the distribution of the median is well known, but in general many statistics have distributions that are difficult to determine analytically, as exemplified below.)

The distribution is estimated by repeatedly drawing pseudosamples, S^*. Each pseudosample is formed by selecting n elements of S independently, completely at random, and with replacement. For each psuedosample S^*, a value for the associated statistic $\widehat{\theta}(S^*)$ is computed; denote one as $\widehat{\theta}^*$. The psuedosampling is repeated many times. A normalized histogram of the resulting values for $\widehat{\theta}^*$ will often be a good approximation to the probability distribution function for $\widehat{\theta}$. A much more careful and thorough introduction to bootstrapping is given by Efron [7].

6.2 Bootstrapping Performance Measures for Fixed A_{TG}

In terms of human identification, and recognition rates in particular, bootstrapping could be applied to the problem of estimating the probability distribution for recognition rate given fixed training and gallery sets. Consider again the success indicator function from equation 6. Given a probe set P of size n that is representative of a larger population, a sample of success/failure outcomes for algorithm A_{TG} may be expressed as the sequence:

$$S = [b(x_{i,j}), \quad \forall \quad x_{i,j} \in P] \tag{18}$$

The recognition rate ρ for a sample S is:

$$\rho(P) = \frac{c}{n} \quad \text{where} \quad c = \sum_{s_\ell \in S} b(s_\ell) \tag{19}$$

where s_ℓ is the ℓth element of the sequence S. Now, to bootstrap the statistic ρ, generate $10,000$ pseudosamples S^* from S and, for each, compute ρ^*:

$$\rho^*(P) = \frac{c^*}{n} \quad \text{where} \quad c^* = \sum_{s_i \in S^*} b(s_i) \tag{20}$$

The normalized histogram of ρ^* values is the bootstrap distribution for ρ.

Of course, application of bootstrapping to the recognition rate problem is somewhat pointless, since it can be shown analytically that c^* is a binomially distributed random variable. The bootstrapping in this case amounts to nothing more than a Monte Carlo approximation to the integral whose exact value is given in equation 17.

However, there are related statistics whose distribution is not so easily known. For example, consider bootstrapping to determine the probability distribution for the median censored rank statistic, $\widetilde{r}_\tau$, defined in Section 4.2, equation 10. Again, the procedure would be to draw, say, $10,000$ pseudosamples from the original probe set. For each of these $10,000$ pseudo-probe sets, compute a value of $\widetilde{r}^*_\tau$. The normalized histogram of these resulting values represents the bootstrap distribution for median censored rank.

7 Permuting Gallery and Probe Choices

The above example of evaluation methods left the choice of gallery fixed. Thus, performance variability due to variations in the gallery were not measured. This makes designing experiments easier, but it understates the amount of variation that will be seen in practical circumstances where galleries vary. For many applications, evaluation studies accounting for variation in gallery imagery are more appropriate.

Here we present a summary of a Monte Carlo study designed to compare PCA [11, 10] and PCA+LDA [15] algorithms under varying choices of gallery and probe images [2]. The algorithm descriptions and data preprocessing steps are omitted here, since the exact nature of the algorithms and data is less important than the methodology being illustrated. These details may be found in [2].

The study assumes that, under the null hypothesis, each person's gallery images are exchangeable, as are each per-

son's probe images. Under this assumption, the evaluation results seen in the actual experiment are distributionally equivalent to those that would have been obtained in any hypothetical experiment using different probe and gallery images for these people. Therefore, by considering the outcomes of such hypothetical experiments, it is straightforward to derive the null distribution of the test statistic of interest.

Initially, we endeavored to design a bootstrapping [7] study, but difficulties described below led us to instead favor a different Monte Carlo method. This different method generates samples that always contain one instance of each person by permuting the choice of gallery and probe images.

7.1 The Research Question and Associated Data

Two related questions give rise to the study presented here:

1. How much variation in recognition rate can be expected when comparing gallery images of these individuals taken on one day to probe images taken on another day?

2. Does one algorithm perform significantly better than another relative to the variance induced by perturbing gallery and probe images?

We arrived the choice of imagery by noting first that the complete FERET database includes $14,051$ source images, but only $3,819$ have the subject facing directly into the camera. Further, of these, there are $1,201$ distinct individuals represented. For 481 of these people, there are 3 or more images, and for 256 there are 4 or more images. Being more precise, of the 256 people with four or more images, there are 160 where the first pair was taken on a single day, and the second pair on a different day. Of the images taken on the same day, the subject was instructed to pick one facial expression for the first image and another for the second [2] Thus, we have 160 people, 4 image per person, appropriate for testing the questions posed above.

For training the algorithms, we consider the arguably most difficult case of precluding any overlap between training and test data. So, it was decided to use the imagery of the 225 people for whom there are at least 3, but not 4, images each for training. Consequently, the PCA algorithm was trained using 675 images. In keeping with common practice in the FERET evaluation, the top 40 percent of the eigenvectors were retained. The LDA algorithm was trained on the same images partitioned into 225 classes, one class per person. Additional details regarding data preprocessing and algorithm training appear in [2]

	Day 1 Expression		Day 2 Expression	
Person	**One**	**Another**	**One**	**Another**
0	$u_{0,0}$	$u_{0,1}$	$u_{0,2}$	$u_{0,3}$
1	$u_{1,0}$	$u_{1,1}$	$u_{1,2}$	$u_{1,3}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
159	$u_{159,0}$	$u_{159,1}$	$u_{159,2}$	$u_{159,3}$

Table 5. Illustrating the organization of the 640 test images organized by person, day and facial expression.

In order to investigate how recognition rate ρ_τ varies with different choices of probe images P and gallery images G, we will permute the assignment of images to P and G. Because our research question concerns algorithm performance for probe and gallery images taken on different days, our Monte Carlo process must preserve the multiple day separation property. This is easily done, and the process is perhaps best explained by first arraying the test data W as shown in Table 5. However, before describing the Monte Carlo technique, let us explain why we did not use bootstrapping for this study.

7.2 Bootstrapping Recognition Rate is Difficult

An obvious way to perform bootstrapping on the image data presented in Table 5 is to begin by sampling from the population of 256 people with replacement. Sampling with replacement is a critical component of bootstrapping in order to properly infer generalization to a larger population of people [5]. Indeed, we went down this road a few steps before encountering the following difficulty.

When sampling with replacement, some individuals will appear multiple times and these duplicates cause a problem for the scoring methodology. To see this clearly, it is necessary to go one level deeper into the sampling methodology. Once an individual is selected, it still remains to select a pair of images to use for testing: one as the gallery image and one as the probe image.

For the sake of illustration, assume individual 0 is duplicated 4 times [3]. Also assume for the moment that the gallery image is selected at random from columns 0 and 1 and the probe image from columns 2 and 3. Thus, one possible selection might be:

$$\left\{\left(u_{0,0}, u_{0,2}\right), \left(u_{0,1}, u_{0,2}\right), \left(u_{0,0}, u_{0,3}\right), \left(u_{0,1}, u_{0,3}\right)\right\}$$

[2] It might surprise some readers to note that no further instruction was given. Specifically the subjects were not coached as to what sort of expression to adopt, for example smile of frown, happy or sad. So, it is incorrect to assume anything other than that the expressions are different.

[3] Indeed, the chances of at least one individual being duplicated 4 times is over 99%.

where the pairs are ordered, gallery image then probe image. The intent with bootstrapping is that when a given pair is selected, for example $(u_{0,0}, u_{0,3})$, then the recognition score should pertain specifically to that pairing. However, it could easily happen that probe image $u_{0,3}$ is close to gallery image $u_{0,1}$, but not to $u_{0,0}$. So, strict adherence to the bootstrapping requirements dictates a near match to $u_{0,1}$ should be ignored, and the algorithm should be scored based upon whether or not $u_{0,0}$ is in the set of k nearest gallery images. Clearly this is not how our scoring was defined above. Making this change alters the measure we are attempting to characterize, so is not an option. However, if the match between $u_{0,3}$ and $u_{0,1}$ is counted, as would happen with normal application of the recognition rate defined above, the bootstrapping assumptions are violated.

It is not immediately obvious how to preserve the recognition rate scoring protocol and simultaneously satisfy the needs of bootstrapping. The matter is certainly not closed and we are continuing to consider alternatives. However, for the moment this problem represents a significant obstacle to the successful application of bootstrapping and we therefore turn our energies to a Monte Carlo based approach that does not require sampling with replacement.

7.3 Permuting Probe-Gallery Choices

As with many nonparametric techniques, the idea of our Monte Carlo approach is to generate a sampling distribution for the statistic of interest by repeatedly computing this statistic from different datasets that are somehow equivalent. In our approach, the key assumption is that the gallery images for any individual are exchangeable, as are the probe images. If this is true, then, for example, $(u_{0,0}, u_{0,2})$ is exchangeable with $(u_{0,1}, u_{0,2})$, $(u_{0,0}, u_{0,3})$, or $(u_{0,1}, u_{0,3})$. The statistic of interest is the recognition rate rho_τ and the samples are obtained by permuting the choice of gallery and probe images among the exchangeable options for each of the 160 people. This might be done by going down the list of people selecting at random a gallery image from one day and a probe image from the other as illustrated in Table 6a.

This is an unbalanced sample: not all columns are equally represented. A balanced sampling is easily obtained by first permuting the personal identifiers and then using a fixed pattern of samples for the columns, as illustrated in Table 6b. This guarantees equal sampling from all columns. Most experiments below use balanced sampling. However, Section 7.6 discusses the empirical difference, which is little, and presents an example.

Id.	G	P
0	$u_{0,3}$	$u_{0,1}$
1	$u_{1,1}$	$u_{1,3}$
2	$u_{2,3}$	$u_{2,0}$
3	$u_{3,1}$	$u_{3,3}$
4	$u_{4,2}$	$u_{4,0}$
5	$u_{5,1}$	$u_{5,2}$
6	$u_{6,2}$	$u_{6,1}$
7	$u_{7,1}$	$u_{7,3}$
⋮	⋮	⋮
159	$u_{159,2}$	$u_{159,1}$

(a)

Id.	G	P
154	$u_{154,0}$	$u_{154,2}$
130	$u_{130,0}$	$u_{130,3}$
69	$u_{69,1}$	$u_{69,2}$
80	$u_{80,1}$	$u_{80,3}$
128	$u_{128,2}$	$u_{128,0}$
72	$u_{72,2}$	$u_{72,1}$
82	$u_{82,3}$	$u_{82,0}$
42	$u_{42,3}$	$u_{42,1}$
⋮	⋮	⋮
108	$u_{108,3}$	$u_{108,1}$

(b)

Table 6. Illustrating unbalanced, (a), and balanced, (b), sampling. These sampling strategies permute the choices of gallery and probe images. In both tables, the first column is the integer indicating a person, the second column is the gallery image and the third column the probe image.

7.4 Distributions and Confidence Intervals on Recognition Rate ρ

As mentioned above, the "virtual" experiments using the randomly selected probe and gallery sets may be run without running the recognition algorithms themselves if first the distance matrix Υ_U is computed. This was done once for the 640 test images in U and each of the following eight algorithms:

1. PCA using L1 distance.
2. PCA using L2 distance.
3. PCA using angle between normalized image vectors as the distance measure.
4. PCA using Mahalanobis distance.
5. LDA using L1 distance.
6. LDA using L2 distance.
7. LDA using angle between normalized image vectors as the distance measure.
8. LDA using soft weighted variant of L2 distance.

The first seven variants are more fully described in [2]. The eigth uses a weighted distance measure proposed by WenYi Zhao [16]. Essentially, it is the L2 norm with the modification that each dimension is scaled by the associated LDA eigenvalue raised to the power c. In this test, $c = 0.2$.

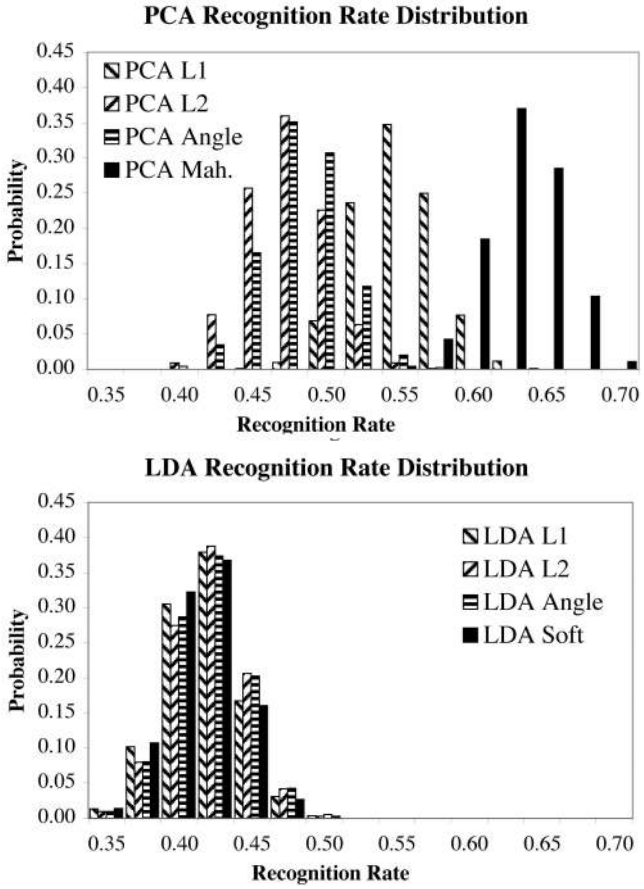

Figure 1. Rank 1 recognition rate distributions for PCA and LDA variants.

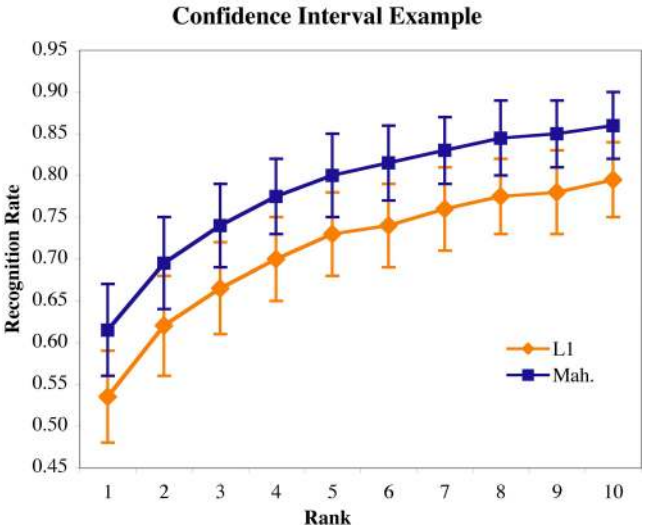

Figure 2. The 95% confidence intervals for PCA using L1 and Mahalanobis distance.

The balanced sampling described above was used to simulate $10,000$ experiments where different combinations of probe and gallery images were selected. For each of these $10,000$ trials, the recognition rate ρ_τ for $\tau = 1, \ldots 10$ were recorded. These $10,000$ values are then histogrammed to generate the sample distribution for ρ_τ. The distribution of these recognition rates represents a good approximation to the probability distribution for the larger population of possible probe and gallery images.

Figures 1 show these distributions for the PCA and LDA algorithm variants at rank 1. To explain the recognition rate labels along the x axis, there are only 160 images in the probe sets. This means not all recognition rates are possible, but instead recognition rate runs from 0 to 1 in increments of $1/160$. To avoid the problem of unequal allocation of samples to histogram bins, histogram bins are $4/160 = 1/40$ units wide. When histogrammed in this fashion, the distributions are relatively smooth and, to a first order, unimodal.

Looking at the PCA algorithm variants, there is a clear ranking: Mahalanobis distance, followed by L1 distance, followed by the remaining two. We will take up shortly the question of how to further refine the question of relative performance between these variants. Looking at the LDA algorithm variants, two things stand out. First, there is very little difference between them. Second, they are all clustering around recognition rates slightly lower than for the PCA algorithm using L2 or angle, and clearly worse than PCA using L1 or Mahalanobis distance.

The simplest approach to obtaining one- and two-sided confidence intervals is the percentile method. For example, a centered 95% confidence interval is determined by coming in from both ends until the accumulated probability exceeds 0.025 on each side. This is best done on the most finally sampled version of the histogram: one with bin width equal to $1/160$.

Figure 2 shows the 95% confidence intervals obtained in the manner just described for ranks 1 through 10. To keep the figure readable, the confidence intervals for only the PCA algorithm using Mahalanobis and L1 distance are shown. Keep in mind that these are pointwise intervals for each rank that are not adjusted for multiple comparisons. These plots are elaborations of the CMS plots commonly used in the FERET evaluation with the notable exception that now intervals rather than single curves are shown.

Both the distributions and confidence intervals call attention to the differences between PCA using Mahalanobis distance, L1 and the other distance measures. For example, based upon the overlapping confidence intervals shown in Figure 2, one might be drawn to conclude there is no

significant difference between PCA using L1 versus PCA using Mahalanobis distance. However, as the next section will show, there are more direct and discriminating ways to approach such questions, and simply looking to see if confidence intervals overlap can be somewhat misleading.

7.5 Hypothesis Testing: Is A_T Better than B_T ?

The question typically asked is: Does algorithm A perform better than algorithm B? This gives rise to a one sided test of the following form. Formally, the hypothesis being tested and associated null hypothesis are:

H1 The recognition rate ρ_τ for algorithm A is higher than for algorithm B.

H0 The recognition rates are identical for both algorithms.

To establish the probability of H0 a new statistic $D_\tau(A, B)$ is introduced that measures the signed difference in recognition rates:

$$D_\tau(A, B) = \rho_\tau(A) - \rho_\tau(B) \tag{21}$$

The same Monte Carlo method used above to find the distribution for ρ_τ may be used to find the distribution for $D_\tau(A, B)$. In other words, gallery and probe sets are selected according to the same randomized procedure $10,000$ times and the difference $D_\tau(A, B)$ is computed each time. Figure 3 shows the distributions for $D_1(A, B)$ for the PCA algorithm using three pairs of distance measures: Mahalanobis minus L1, L1 minus L2 and L2 minus angle.

For two of the differences, Mahalanobis minus L1 and L1 minus L2, the distribution is highly skewed with respect to zero. For the Mahalanobis minus L1 case, D_1 is equal to or less than zero only 35 out of $10,000$ times. For the L1 minus L2 case, D_1 is equal to or less than zero only 13 out of $10,000$ times. For the third comparison, L2 to angle, the distribution is centered more closely about zero. In this case, D_1 is equal to or less than zero $9,014$ out of $10,000$ times. These distributions may be used to test H0. Table 7 shows the probabilities for the observed differences given H0. With very high confidence, H0 may be rejected in favor of H1 for the first two comparisons, and not for the third. Observe these probabilities derive directly from the ratio stated above.

At first glance it might appear wise to carry out all 42 possible pairwise tests using D_τ. However, doing so invites false associations. The common practice of rejecting H0 at probability level 0.05 implies that it is very likely that one will mistakenly reject H0 a few times. Multiple comparison procedures could be employed to remedy this problem, but a full analysis of variance [5] would provide a richer model for inference. In future work we plan to pair the analysis of variance model with the Monte Carlo inferential paradigm to provide a complete analysis of such experimental data. In lieu of such a procedure, looking at individual performance measures and making a small set of salient pairwise tests is a reasonable strategy.

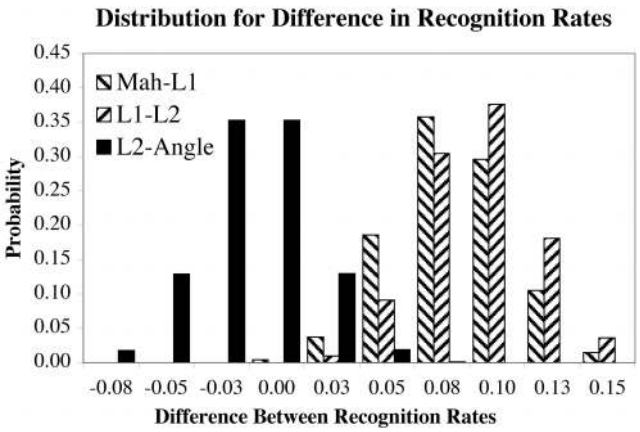

Figure 3. Rank 1 distribution for recognition rate difference.

Alg. A	Alg. B	$P(D_1(A, B) < 0)$
Mah.	L1	0.0080
L1	L2	0.0013
L2	Angle	0.9014

Table 7. Probability of H0 at rank 1 given observed difference in recognition rate.

7.6 Balanced versus Unbalanced Sampling

Section 7.3 indicated that sampling may be done in either a balanced or unbalanced fashion. Does the distinction matter in our context? Figure 4 shows the result of one such comparison: the recognition rate probability distribution for the PCA algorithm using Mahalanobis distance obtained using balanced versus unbalanced sampling. The distinction does not appear to matter: the two distributions are essentially indistinguishable. The other distributions presented above were also essentially unchanged when unbalanced sampling was compared to balanced. More work is needed to fully explore the implications of the two alternative sampling methods, but at least using the definitions of balanced versus unbalanced sampling introduced above, the distinction appears to matter little.

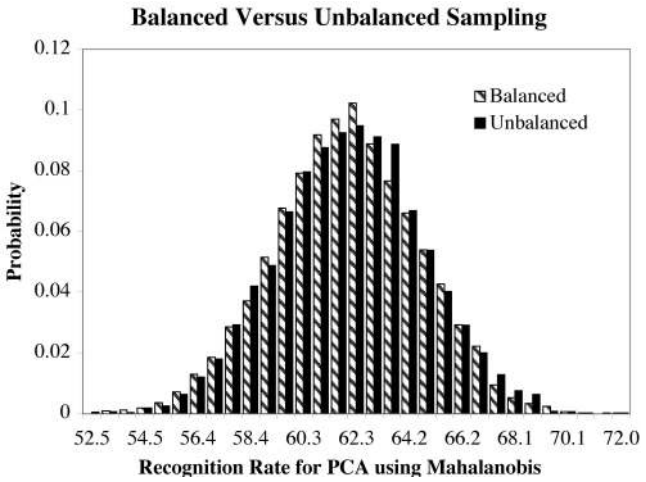

Figure 4. Distributions obtained using balanced versus unbalanced sampling.

8 Conclusion

A framework for making statistical comparisons between different human face recognition algorithms has been presented and two statistical evaluation methods are developed. The first is a parametric method that equates success or failure of algorithms on probe images to Bernoulli trials. The method is simple to use and captures variation arising from the size of the sample, .i.e., the number of probe images tested. More precisely, it captures the uncertainty associated with estimating the true probability that an algorithm succeeds based upon a finite number of samples.

The second method is a nonparametric Monte Carlo sampling technique that samples the space of possible gallery and probe sets. This method approximates the probability distribution of a statistic such as recognition rate or difference in recognition rate. Given a desire in practice to know how algorithms behave when changes are made to gallery and probe sets, this latter methodology is arguably the more interesting.

The use of each method is illustrated with examples comparing well known algorithm types from the literature. The first example shows that using the FERET data under the test conditions specified, a standard PCA classifier performs recognition better than an ICA classifier. The second example shows that a standard PCA classifier performs recognition better than a PCA followed by LDA classifier. In both cases, there is a literature suggesting the results should have come out the other way. More work is needed to better understand these results.

This paper is part of larger effort to understand how certain common face recognition algorithms behave, and to develop better statistical evaluation methodologies for comparing algorithms. A web site is being developed [3] to report progress. The source code for the PCA algorithm and the PCA followed by LDA algorithm is available through this site.

Acknowledgements

This work supported by the Defense Advanced Research Projects Agency under contract DABT63-00-1-0007.

References

[1] Bartlett, M. S., H. M. Lades, et al. Independent component representations for face recognition. In *SPIE Symposium on Electronic Imaging: Science and Technology; Conference on Human Vision and Electronic Imaging III*, San Jose, CA, 1998.

[2] J. R. Beveridge, K. She, B. Draper, and G. H. Givens. A nonparametric statistical comparison of principal component and linear discriminant subspaces for face recognition. In *Proceedings of the IEEE Conference on Pattern Recognition and Machine Intelligence*, page (to appear), December 2001.

[3] R. Beveridge. Evaluation of face recognition algorithms web site, data section. `http:\\cs.colostate.edu\evalfacerec`.

[4] W. Cochran. *Sampling Tecniques*. Wiley and Sons, New York, 1953.

[5] P. Cohen. *Empirical Methods for AI*. MIT Press, 1995.

[6] J. Devore and R. Peck. *Statistics: The Exploration and Analysis of Data, Thrid Edition*. Brooks Cole, 1997.

[7] B. Efron and G. Gong. A Leisurely Look at the Bootstrap, the Jackknife, and Cross-validation. *American Statistician*, 37:36–48, 1983.

[8] IFA. Statistical tests, http://fonsg3.let.uva.nl:8001/service/statistics.html). Website, 2000.

[9] Kyungim Baek,Bruce A.Draper,J.Ross Beveridge and Kai She. Pca vs.ica:a comparison on the feret data set. In *IEEE Conference on Computer Vision and Pattern Recognition*, page (submitted), 2001.

[10] M. A. Turk and A. P. Pentland. Face Recognition Using Eigenfaces. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 586 – 591, June 1991.

[11] M. Kirby and L. Sirovich. Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(1):103 – 107, January 1990.

[12] P. Phillips, H. Moon, S. Rizvi, and P. Rauss. The FERET Evaluation Methodology for Face-Recognition Algorithms. *T-PAMI*, 22(10):1090–1104, October 2000.

[13] Ross J. Micheals and Terry Boult. Efficient evaluation of classification and recognition systems. In *IEEE Computer Vision and Pattern Recognition 2001*, page (to appear), December 2001.

[14] B. A. D. Wendy S. Yambor and J. R. Beveridge. Analyzing pca-based face recognition algorithms: Eigenvector selection and distance measures. In *Second Workshop on Empirical Evaluation in Computer Vision*, Dublin, Ireland, July 2000.

[15] W. Zhao, R. Chellappa, and A. Krishnaswamy. Discriminant analysis of principal components for face recognition. In *In Wechsler, Philips, Bruce, Fogelman-Soulie, and Huang, editors, Face Recognition: From Theory to Applications*, pages 73–85, 1998.

[16] W. Zhao, R. Chellappa, and P. Phillips. Subspace linear discriminant analysis for face recognition. In *UMD*, 1999.