Object Recognition using Boosted Discriminants *

Shyjan Mahamud Martial Hebert Jianbo Shi
Dept. of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We approach the task of object discrimination as that of learning efficient "codes" for each object class in terms of responses to a set of chosen discriminants. We formulate this approach in an energy minimization framework. The "code" is built incrementally by successively constructing discriminants that focus on pairs of training images of objects that are currently hard to classify. The particular discriminants that we use partition the set of objects of interest into two well-separated groups. We find the optimal discriminant as well as partition by formulating an objective criteria that measures the well-separateness of the partition. We derive an iterative solution that alternates between the solutions for two generalized eigenproblems, one for the discriminant parameters and the other for the indicator variables denoting the partition. We show how the optimization can easily be biased to focus on hard to classify pairs, which enables us to choose new discriminants one by one in a sequential manner. We validate our approach on a challenging face discrimination task using parts as features and show that it compares favorably with the performance of an eigenspace method.

1 Introduction

Many approaches to image-based object recognition proceed by detecting a number of (possibly localized) feature responses in an input image and combining the evidence from these responses to determine the presence or absence of an object [1, 12, 14]. Two main issues need to be addressed in such approaches. First, how do we select good features ? Second, what is an appropriate framework for combining the evidence from the various feature responses in a given input image ?

Consider the task of learning a discriminator for a multi-class object recognition problem, given training data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ where $x_i \in X$ are training images of objects of interest and y_i are corresponding class labels $y_i \in Y \equiv \{1, \ldots, m\}$. Our basic approach will be to construct a "code" for an image in terms of the responses to a set of binary discriminants $h : X \rightarrow \{-1, +1\}$ that we learn from the training data. These binary discriminants will be the features that we use. An example for a binary discriminant is an image template mask l for a specific feature of an object class (say the eyes for a face discrimination task) along with a threshold θ and the location r in an input image where the template should be applied. Given an input image, the template responds with $+1$ if the correlation of the subimage at location r of the input image with l is greater than θ and with -1 otherwise. More general discriminants can be used and are discussed in §4. Figure 1 gives an illustration of the approach. Intuitively, a "good" code should have the following properties :

- Images from the same object class should be "close" together in code-space.
- Images of different object classes should be as far apart as possible in code-space.

For binary discriminants, a distance measure between two images x_i and x_j in code space that is simple yet flexible is the weighted correlation function :

$$C(x_i, x_j) = \sum_k \alpha_k h_k(x_i) h_k(x_j) \tag{1}$$

which is related to the usual hamming distance when the weights are all set to 1. Using this distance measure, given an input image the class label corresponding to the training image that has the highest correlation with the input image is reported. As discussed above, the goal at training time is to find a set of discriminants such that training images from the same object class are highly correlated, while training images from different classes are as uncorrelated as possible. In our work, we achieve this goal by minimizing a loss function that penalizes deviations from the above two criteria. To do this, we first reduce the original multi-class problem into a binary classification problem so that we can incorporate both criteria into one loss function.

The original multi-class problem with the training set $\{(x_i, y_i) \mid i = 1, \ldots, n\}$ can be reduced to a corresponding

*This work was supported by NSF Grant IIS-9907142 and DARPA HumanID ONR N00014-00-1-0915

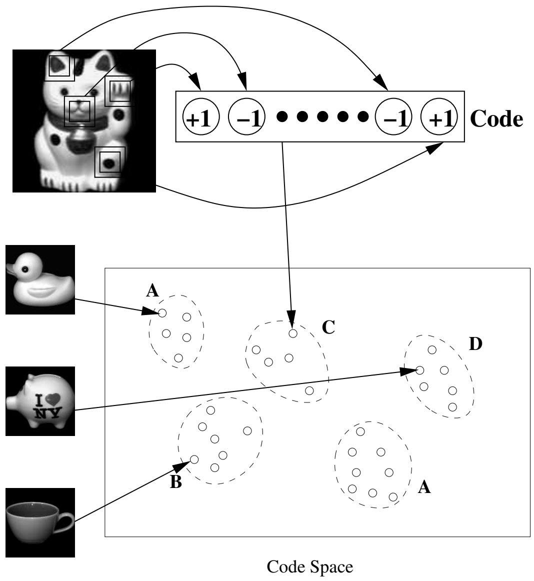

Figure 1: Illustration of a "code"-space for objects. Each image of an object of interest has a "code-word" in terms of the responses to a set of binary discriminants as illustrated at the top of the figure. The bottom shows a $2\mathrm{D}$ embedding of such code-words for a sample of images from various object classes A, B, C, D. The goal is to find codes that cluster together images from the same object class while separating out images from different classes as much as possible.

binary classification problem with the training set :

$$\{((x_i, x_j), y_{ij}) \mid i, j = 1, \ldots, n,\ y_{ij} \in \{-1, +1\}\}$$

where the class labels for the binary problem y_{ij} is given by :

$$y_{ij} = \begin{cases} +1 & \text{if } y_i = y_j \\ -1 & \text{if } y_i \neq y_j \end{cases}$$

Given this reduction and a loss function $L(\cdot)$, we can incorporate the above two criteria by minimizing the following total loss :

$$\begin{aligned} E &\equiv \sum_{i,j} L(y_{ij} C(x_i, x_j)) \\ &= \sum_{i,j} L(\textstyle\sum_k \alpha_k y_{ij} h_k(x_i) h_k(x_j)) \end{aligned} \quad (2)$$

with respect to both discriminants h_k and the corresponding weights α_k. The loss $w_{ij} \equiv L(y_{ij}C(x_i, x_j))$ corresponding to a particular pair of images x_i, x_j is high when the sign of the correlation $C(x_i, x_j)$ between the two images is different from the class label y_{ij}. The particular approach that we use to select robust discriminants that minimize the total loss is the central topic of this paper and is discussed in §2.2.

The energy minimization framework above is most related to the boosting framework from the machine learning community [2]. In boosting, so-called "weak classifiers" are learned sequentially and combined to give a weighted voting rule for discriminating objects from each other. In our approach, the correlation term $y_{ij} h_k(x_i) h_k(x_j)$ in the total loss (2) corresponding to the discriminant h_k is analogous to a weak classifier. The weighted voting rule is analogous to the weighted correlation function in eq (1). At each stage of boosting a running measure is kept of how well the current set of weak classifiers that have been learned so far can discriminate the objects (classes in the learning literature). In the multi-class setting, this measure can be thought of as a confusion matrix and is analogous to the matrix of loss terms $w_{ij} \equiv L(y_{ij}C(x_i, x_j))$ in our approach, where the correlation function C uses the current set of discriminants learned so far. The confusion matrix is then used to bias the weak classifier to focus on discriminating objects that have been hard to classify. Recent applications of the boosting framework to object discrimination can be found in [12, 15].

In our work, we adopt the strategy used in the boosting framework for sequentially learning discriminants in the energy minimization framework. New discriminants are learned one by one that minimize the current loss. The overall scheme is detailed in §2.3.3.

An important issue in the boosting framework is how easy it is to couple the weak classifier with the confusion matrix. Typically, when using standard weak classifiers, the coupling may not be easy or natural. The weak classifier may even need to be modified to work with a confusion matrix. In our approach, as will be detailed in §2.2 the determination of the optimal discriminant is formulated in such a way that a straight-forward and tight coupling between the confusion matrix w_{ij} and the optimization problem for the discriminant can be achieved. Analogous to "boosting" weak classifiers in the boosting literature, we can consider our approach as that of "boosting" discriminants.

In general, the discriminants h_k used can be any function that partitions the space of images X into two. We would like to choose those discriminants that satisfy the following desirable criteria :

I. The discriminant should focus on pairs of training images that have been difficult to classify (i.e. difficult to cluster or discriminate as the case may be) so far. In other words, the discriminant should focus on pairs for which the corresponding entry in the confusion matrix w_{ij} is high.

II. As much as possible, pairs of training images from the same object class (i.e., $y_{ij} = +1$) should be put in the same partition induced by the discriminant, while pairs of training images from different object classes (i.e., $y_{ij} = -1$) should be put in different partitions.

III. The partition induced on the training set is 'well-separated" with each partition "tightly" clustered. This should make the discriminant more robust at run-time if the training data is representative of data to be seen at run-time.

In §2.2, we show how we satisfy the above two criteria by formulating an objective function that can be thought of as an unsupervised generalization of the well known Fisher criteria [4] which can be optimized to find both the optimal discriminant as well as the optimal partition of the training set. Unlike the traditional formulation, we use a purely pair-wise formulation, which allows us to easily bias the optimization to focus on the pairs of training images that are currently hard to classify using the discriminants learned so far (i.e., in the notation above, pairs for which w_{ij} is high).

We summarize our approach by answering the two questions posed at the beginning as well as outline the rest of the paper as follows :

- Various candidate discriminants are constructed by optimizing a pair-wise formulation of a generalization of the Fisher criteria §2.2. The candidate discriminant that reduces the total loss (2) the most is chosen.
- The discriminants chosen so far are weighted and combined to give the final correlation function to be used at run-time (§2.3.2 and §2.3.3).

In §3 we validate our approach on a challenging face discrimination task.

2 Method

2.1 The Loss Function

In our work, we use the exponential loss function $L(z) = e^{-z}$ in the total loss (2), giving us :

$$E = \sum_{i,j} e^{-w_{ij}} \tag{3}$$

$$= \sum_{i,j} e^{-y_{ij} \sum_k \alpha_k h_k(x_i) h_k(x_j)} \tag{4}$$

The resulting total loss is a convex function of the weights α_k with a global minimum. Another loss function that can be used is the logistic cost function $L(z) = \log(1 + e^{-z})$. Either choice for the loss function can be rigorously motivated and justified in the maximum entropy framework [5]. Given a training set and a set of discriminants, the maximum entropy framework seeks the least committed model that is consistent with the statistics of the responses of the discriminants over the training set. In this framework, it can be shown that the exponential loss function is the optimal choice among all loss functions when unnormalized models are sought, while the logistic loss is optimal when conditional probability models are sought. These loss functions are also commonly used in the boosting community though the justification there is from a learning perspective. For simplifying the presentation below, we will use the exponential loss from now on, although all the results below can be adapted for the logistic loss with little difficulty.

2.2 Boosting Discriminants

Learning a good code requires finding good discriminants h_k and the associated weights α_k. We now present our approach to finding good discriminants that satisfy the three criteria outlined in §1.

2.2.1 Finding Good Discriminants

Let us assume that we are given a continuous feature space. For example, the pixel intensities in a localized $m \times m$ window around a given location in an input image lies in the continuous feature space $\mathbb{R}^{m^2}$. We would like to find a discriminant in the feature space that satisfies the criteria outlined in §1. One of the criteria (III) is to find a discriminant along which the training images are partitioned into two well-separated groups, each of which is tightly clustered. The rationale for this criteria is that such a discriminant can be expected to reliably determine the partition that unseen images of objects of interest belong to, assuming that the training data is representative of all the images of objects of interest that will be encountered. In other words, we want to maximize :

$$J \equiv \frac{\text{across-partition separation}}{\text{within-partition separation}}$$

If we know the optimal partition that satisfies the above criteria, then the optimal discriminant can be found by optimizing the Fisher discriminant quotient [4]. Let v_i be the vector corresponding to the training image x_i in the continuous feature space (i.e., $v_i \in \mathbb{R}^{m^2}$ in the example above). The Fisher quotient is usually formulated in the literature in terms of the first and second order statistics of the training data as follows :

$$J(s, l) = \frac{||m^+ - m^-||_2}{{\sigma^+}^2 + {\sigma^-}^2}$$

where m^+, m^- are the means of the projections onto the discriminant l of the v_i's in the two partitions, and similarly σ^+, σ^- are the corresponding variances. In our formulation however, we will instead use a purely pair-wise formulation that will allow us to easily incorporate the other criteria discussed in §1. We denote a partition of the training images by indicator variables $\mathrm{s} = \{s_1, \ldots, s_n\}$ where each $s_i \in \{-1, +1\}$ indicates the partition that v_i belongs to in the feature space. The pair-wise formulation of the Fisher quotient that we use is then given by :

$$J(s,l) = \frac{\sum_{i,j}(1-s_i s_j)K(x_i,x_j)}{\sum_{i,j}(1+s_i s_j)K(x_i,x_j)} \quad (5)$$

where $K(x_i,x_j) \equiv l^T(v_i - v_j)^T(v_i - v_j)l$ is the separation along the discriminant hyperplane l between training images x_i and x_j. Note that as required the term $(1-s_i s_j)/2$ is an indicator function that denotes when x_i and x_j are in different partitions, while $(1+s_i s_j)/2$ denotes when x_i and x_j are in the same partition. The distance function $K(\cdot,\cdot)$ - also known as a kernel - can be generalized to non-linear kernels as will be discussed in §4.

In practice, we will have to determine *both* the optimal partition (i.e. a setting for s that optimizes eq (5)) as well as the optimal discriminant hyperplane l. This is an unsupervised mixed discrete-continuous optimization problem (discrete in s and continuous in l). We derive an iterative solution for this optimization problem in the next subsection. Once the hyperplane l is found, we can form a linear discriminant $h(x) = \text{sgn}(l^T v - \theta)$ where θ is the optimal threshold that separates the two partitions.

We can measure the performance of the discriminant with respect to the binary classification problem at hand. Two training images from two different object classes are classified correctly by the discriminant if they fall in different partitions. Similarly, two training images from the same object class are classified correctly if they fall in the same partition. However, nothing in the criteria optimized by the quotient in eq (5) explicitly encourages finding discriminants with good classification performance. We now do so by encoding the other two criteria (I,II) in §1 into the optimization.

We can constrain the optimization of eq (5) such that training objects that belong to the same object class are encouraged to be in the same partition (criteria (II)). This is done simply by using the same indicator variable for all training images belonging to the same object class, i.e. all training examples x_{k_i} that have the same class label y_i will use the same indicator variable s_i. Thus any assignment to the indicator variables will put all training images from the same object class in the same partition.

We can encode criteria (I) by biasing the optimization to focus on pairs of training images that have been hard to classify with the current set of discriminants that have been learned so far. Let us assume that k discriminants have been learned so far and let $C^k(x_i,x_j)$ be the corresponding weighted correlation function (eq (1)) of the "code" between two images x_i and x_j in terms of the responses to the k discriminants. As discussed in §1 the loss term $w_{ij} \equiv L(y_{ij}C^k(x_i,x_j)$ can be considered as a measure of the difficulty in classifying the two images. The pair-wise formulation of the Fisher quotient eq (5) is readily amenable to incorporating these loss terms by weighting each term in the Fisher quotient by the corresponding loss term. Thus harder to classify pairs of training images will have a correspondingly larger influence on the optimization of the quotient. The modified expression for the quotient is :

$$J(s,l) = \frac{\sum_{i,j}(1-s_i s_j)w_{ij}K(x_i,x_j)}{\sum_{i,j}(1+s_i s_j)w_{ij}K(x_i,x_j)} \quad (6)$$

2.2.2 Iterative Optimization

In practice, direct optimization of J is hard since it is a discrete-continuous optimization problem. To make the optimization feasible, we relax the discrete optimization over s to a continuous optimization problem. With this relaxation, we propose an iterative maximization scheme, by alternating between maximizing J w.r.t s keeping l fixed and maximizing w.r.t. l keeping s fixed. We show below that each of these subproblems leads to a corresponding generalized eigenvalue problem.

First, consider maximizing J keeping l fixed. Define a matrix W with entries :

$$W(i,j) \equiv \sum_{i,j}\sum_{k_i,k_j} w_{k_i k_j} K(x_{k_i}, x_{k_j})$$

where k_i ranges over all the indices of training images that belong to class i and similarly for k_j (the notation takes into account the fact that indicator variables are shared among training images from the same class, i.e. criteria (II) above). Let $\mathbf{1}$ be a vector of 1's with the same number of components as s. Then J can be simplified as follows :

$$J(\mathbf{s}) = \frac{\mathbf{1}^T W \mathbf{1} - \mathbf{s}^T W \mathbf{s}}{\mathbf{1}^T W \mathbf{1} + \mathbf{s}^T W \mathbf{s}}$$

Let D be a diagonal matrix with $D = \text{Diag}(W\mathbf{1})$. Since each component of s takes values in $\{-1,+1\}$, the following equivalence can be verified : $\mathbf{1}^T W \mathbf{1} = \mathbf{s}^T D \mathbf{s}$. Substituting above, we get :

$$J(\mathbf{s}) = \frac{\mathbf{s}^T(D-W)\mathbf{s}}{\mathbf{s}^T(D+W)\mathbf{s}} \quad (7)$$

As mentioned before, instead of solving for the hard discrete optimization problem, we solve for an approximate continuous problem. Specifically, instead of assuming that the indicator variables can take on only binary values $\{-1,+1\}$, we let them take on values in the continuous interval $[-1,+1]$. In other words, we make "soft" instead of hard assignments. For continuous values of s, J is maximized when s is set to the eigenvector corresponding to the largest eigenvalue of the generalized eigen-value problem $(D-W)\mathbf{s} = \lambda_s(D+W)\mathbf{s}$.

Next, we maximize J keeping s fixed. Define the matrices :

$$A \equiv \sum_{i,j}(1 - s_i s_j) \sum_{k_i,k_j} w_{k_i k_j}(v_{k_i} - v_{k_j})(v_{k_i} - v_{k_j})^T$$
$$B \equiv \sum_{i,j}(1 + s_i s_j) \sum_{k_i,k_j} w_{k_i k_j}(v_{k_i} - v_{k_j})(v_{k_i} - v_{k_j})^T$$

with k_i and k_j defined as before. An important issue is the fact that the optimization above for s returns "soft" assignments. These soft assignments need to be normalized such that the largest component (in magnitude) of s is set to 1. This ensures that all the components are in the range $[-1, +1]$. With these assumptions, J can be simplified to :

$$J(l) = \frac{l^T A l}{l^T B l} \tag{8}$$

Once again, J is maximized when l is set to the eigenvector corresponding to the largest eigenvalue of the generalized eigen-value problem $Al = \lambda_l Bl$.

Figure 2 summarizes the iterative scheme. We alternate between maximizing J w.r.t. s and l by solving for the corresponding eigenvector problems, until convergence. Although the iteration is guaranteed to increase J monotonically, it can get stuck in a local minimum. Hence in our experiments, we first find the k most significant principal components of all the vectors v_i for some k that is fixed apriori, then initialize l to each of these principal components and optimize using the iterative scheme just described and choose the hyperplane l among them that maximizes J. Note that the optimal partition s is not required for the rest of the scheme.

Let $u_1, \ldots, u_k$ be the first k PCA components of the set of feature vectors v_i corresponding to training images x_i.

do for $i = 1, \ldots, k$

I. Set $l = u_i$.
II. Iterate between the two eigenproblems $(D - W)\mathrm{s} = \lambda_s (D + W)\mathrm{s}$ and $Al = \lambda_l Bl$ until convergence to s_i, l_i.
III. Set $J_i = J(\mathrm{s}_i, l_i)$.

Output l_i corresponding to max J_i.

Figure 2: Pseudo-code for finding optimal discriminants

Figure 3 is an illustration of the above iterative algorithm on a synthetic example in a continuous 2D feature space. There are two training examples for every class (connected by a dashed line for each class). Both training examples in each class share the same indicator variable in the iteration. The algorithm converged to the optimal discriminant (approximately horizontal) in a few iterations, even though the initialization was far from the optimal solution. Also, the final partition found (denoted by $\bigcirc$ and $\times$) is consistent with what one would expect the optimal partition to be. Note that the variation within classes (approximately along the vertical direction) is more on average than variation across classes (mostly along the horizontal direction). Thus, if we had not specified the class membership of training examples through shared indicator variables, the optimal discriminant found would be almost orthogonal to the one shown in the figure since that would be the direction which maximizes the Fisher quotient.

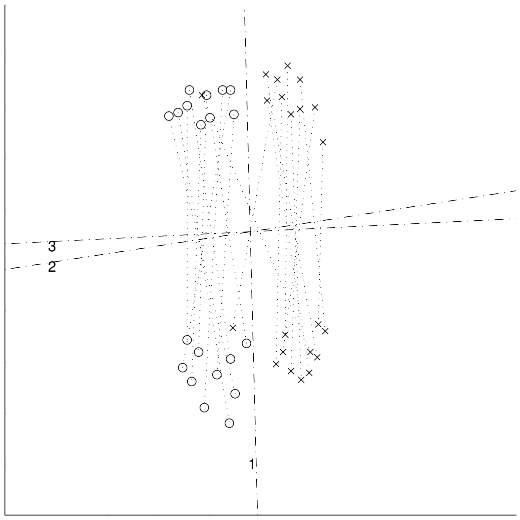

Figure 3: Synthetic example in a continuous 2D feature space illustrating the iterative algorithm for finding optimal discriminants. The number next to a line is the iteration number. The final partition found is denoted by $\bigcirc$ and $\times$.

Choosing θ. Finding the optimal threshold θ is a one-dimensional problem along the discriminant hyperplane l, for which we use a simple brute-force search. The optimal value for θ is that which minimizes the total loss (2). The total loss changes only when θ crosses a vector v_i projected onto l. Accordingly, we determine θ as follows : sort the projections onto the optimal l of all the v_i's, find the total loss for each value of θ that are mid-points (for robustness at run-time) between successive sorted projections, and choose the θ that gives the minimum.

2.3 Learning an Efficient Code

Finally, we discuss the details of the energy minimization framework for learning good codes and present the overall scheme. As discussed in §1, we select discriminants that minimize the total loss (2) sequentially. We first discuss techniques for composing more powerful discriminants from the simple discriminants presented in the previous section to achieve better performance in practice. Once a discriminant h_k has been chosen, the corresponding α_k needs to be optimized. We discuss the practical issues involved in optimizing α_k in §2.3.2. We then summarize the overall scheme in §2.3.3.

2.3.1 Composing Discriminants

The simple discriminants by themselves may not be sufficiently powerful in practice. We can construct more powerful discriminants by composing a set of simple discriminants. In our work, we compose discriminants in a "tree". An input image traverses a path from the root node to a leaf node in the tree. The branch taken by the image at each node along the path is determined by the binary response of the simple discriminant at that node (see figure 4).

Before discussing the details of the composition we use, we first point out the only relevant information about a simple discriminant that the energy minimization framework utilizes (this will also be true at run-time). A discriminant h_k only enters the total loss (2) through the correlation term $h_k(x_i)h_k(x_j)$ on pairs of images x_i and x_j. This correlation term indicates whether the pair of images belong to the same partition induced by the discriminant ($h_k(x_i)h_k(x_j) = +1$) or into different partitions ($h_k(x_i)h_k(x_j) = -1$). Thus we can think of the correlation term as the "partition" function of the corresponding discriminant.

As mentioned above, we will use trees of discriminants T_k in place of simple discriminants h_k to make the scheme more powerful in practice. α_k will then be a weight associated with the whole tree T_k. Consequently, the role of the correlation term induced by the simple discriminant h_k in the energy minimization framework will be replaced by the partition function induced by the tree T_k. T_k partitions the space of all images into those covered by the leaf nodes of the tree (see figure 4). Using a slight abuse of notation, we can denote the partition function induced by the tree T_k by $T(x_i, x_j)$, which is $+1$ if T_k maps both images x_i and x_j to the same partition (i.e. same leaf node of T_k) and -1 otherwise. Using this partition function, we seek to minimize the following total loss :

$$E = \sum_{i,j} L(\textstyle\sum_k \alpha_k y_{ij} T_k(x_i, x_j))$$

2.3.2 Optimizing α_k

Both the exponential and logistic loss function result in a total loss (2) that is convex with a global minimum. Given a set of discriminant trees T_k the optimization of such convex loss functions w.r.t. α_k is straight-forward and can be achieved in practice with Newton-based iterative numerical techniques [11].

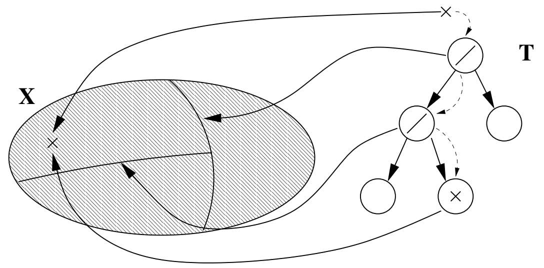

Figure 4: Composing simple discriminants into a tree of discriminants. On the right is shown a tree T composed of two discriminants and on the left the partition induced on the image space X. Also shown is the path taken in the tree by an example image ($\times$) and the corresponding partition that it belongs to.

We can get some intuition for the optimal set of α_k given a set of discriminant trees T_k, by deriving closed-form expressions. By setting the first derivative of the total loss w.r.t. α_k to 0 at the optimum, it can be shown [11] that the optimum α_k^* for each k takes the form :

$$\alpha_k^* = \frac{1}{2} \ln \frac{W^+}{W^-}$$

where W^+ is the total loss of all pairs of training examples that were correctly classified (i.e. $y_{ij}T_k(x_i, x_j) = +1$) by the kth discriminant and W^- is the total loss of all pairs of training examples that were incorrectly classified (i.e. $y_{ij}T_k(x_i, x_j) = -1$). The loss for a pair of training examples in both W^+ and W^- does not include the contribution due to the kth discriminant and thus is a measure of how badly the other discriminants do on that pair. Thus the ratio W^+/W^- (and hence also α_k) is a measure of the classification power of the kth discriminant - the higher the ratio, the better is the discriminant.

In practice, due to limited training data the optimal estimate for α_k^* can be overconfident (i.e. large in value). We can smooth the estimate (thus preferring smaller values) using a regularizer. In our work a simple quadratic regularizer $R(\alpha) = \gamma(\sum_k \alpha_k^2)$ was used where γ is a constant. This regularizer when added to the total loss in (2) still ensures a global minimum.

2.3.3 Summary

We are now in a position to describe the overall scheme. Let $\mathcal{F}$ be a set of continuous feature spaces that will be used for constructing discriminants. For example, in the experiments on face discrimination that we report in §3, the continuous feature spaces are the pixel intensities in a localized window around various feature locations of the face such as the eyes, nose, etc. At the start of each iteration we have a set of discriminant trees $\mathcal{T} = \{T_1, T_2, \dots\}$ and the associated weights $\alpha = \{\alpha_1, \alpha_2, \dots\}$ that were learned in previous iterations. At each iteration we try out various "refinements" $\mathcal{R}$ (to be described shortly) of the discriminant trees in $\mathcal{T}$ and choose the refinement $r \in \mathcal{R}$ that minimizes the total loss 2 the most. This process is repeated for a certain number of iterations.

A refinement of the set $\mathcal{T}$ is defined to be the replacement of some leaf node l belonging to some tree $T_k \in \mathcal{T}$ by a boosted discriminant constructed in some feature space $f \in \mathcal{F}$. The boosted discriminant is constructed as detailed in §2.2 where the training examples are those that reach the leaf node l. We also construct boosted discriminants over the complete training set for various choices of feature spaces $f \in \mathcal{F}$ so that the scheme can choose to start "growing" a new discriminant tree if needed. For simplicity we will treat the addition of such a discriminant (constructed using the complete training set) to the set $\mathcal{T}$ as also a refinement of $\mathcal{T}$. If L is the total number of leaf nodes in all the trees of $\mathcal{T}$, then the number of refinements that we consider at each iteration is $|\mathcal{R}| = (L+1) \cdot |\mathcal{F}|$. Figure 5 shows the pseudo-code for the overall scheme.

Initialize :

I. Let $\mathcal{F}$ be a set of continuous feature spaces.

II. Initialize the set of discriminant trees $\mathcal{T} = \emptyset$.

do for t iterations

I. Construct the set of refinements $\mathcal{R}$ of $\mathcal{T}$ (see text for details).

II. Update $\mathcal{T}$ by the refinement $r \in \mathcal{R}$ that minimizes the total loss in (2). Also update the corresponding weights α.

Output the correlation function :

$$C(x, y) = \sum_k \alpha_k T_k(x, y) \tag{9}$$

At run-time, on input image x, report the object label associated with the training image x_j that maximizes $C(x, x_j)$.

Figure 5: Pseudo-code for the sequential selection of discriminants.

3 Results

We validated our approach on a challenging face discrimination task based on the FERET database [9]. The training

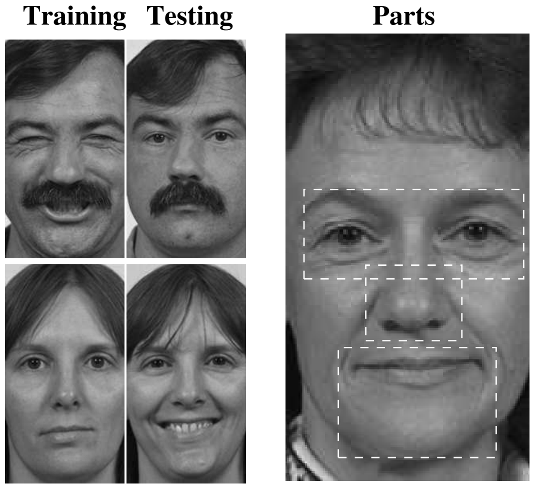

Figure 6: The left shows sample training and testing images for two individuals from the FERET database. The testing images were taken about a month after the training images. Note the difference in expression and hair. The right shows the parts used as features for which discriminants were learned.

images consists of pairs of frontal images of 41 individuals, while the testing images also consists of pairs of frontal images of the same individuals but taken around a month apart from the training images with differences in hair, lighting and expressions. See figure 6 for sample training and testing images. Such images have been considered challenging in the literature [8, 9].

For testing purposes, we rigidly aligned all faces using manually labeled positions of the two eyes. In practice, we can automate this process using face detection and alignment techniques [7, 10]. For our approach, we used the prominent regions around the eyes, nose and mouth as features (see figure 6). The pixel intensities around a localized window around each feature forms a continuous feature space. Candidate discriminants are constructed from each of these feature spaces at each iteration as described in §2.2. Note that the same feature space can contribute many discriminants over iterations, where discriminants constructed in different iterations are in general focused on different pairs of training images that were found difficult to classify using the discriminants constructed so far in previous iterations.

As a baseline, we first determined the performance of a simple eigenspace based method [13]. For each image, all the subimages for the different parts were combined to give one vector. The first 50 PCA components of the resulting vectors for the training data were found to capture over 99% of the energy in the data. Both training and testing data were projected onto these PCA components and a search for the nearest training image for each testing image was performed. The resulting recognition rate was 92.6%.

Our approach requires a few parameters to be set. The total number of discriminants needed can in general be set using cross-validation. However, in our case cross-validation is quite expensive. Instead of using cross-validation, we use a simpler scheme in which we determine twice the number of discriminants that gives a training error of 0. This makes the "code" that is learned more redundant than necessary to classify the training images. This redundancy should help the code to be robust at run-time. Also a regularization constant of $\gamma = 1$ was used to smooth the weights α_k (see §2.3.2). After training, we classified a test image by finding the training image that was most correlated with the test image using the correlation function (9) output by our scheme. In other words, we found the nearest neighbor in code-space. The resulting recognition rate was 95.2%. This compares favorably with the baseline eigenspace technique above. The performance is also similar to the methods reported in the literature for similar datasets [7, 8, 9]. We plan to make more thorough comparisons with these methods in the future. The training time for our approach was around 6 hours, while the run-time was around 2 seconds.

4 Conclusion

We have presented an approach to learning good discriminators that can be thought of as that of learning good codes. Good discriminators are determined sequentially that focus on the currently hard to classify training images. Such discriminators are weighted and combined in an energy minimization scheme.

The work presented in this paper used linear feature spaces where the distance between the representations v_i and v_j in some feature space of two images x_i and x_j was given by the linear kernel $K(x_i, x_j) \equiv l^T(v_i - v_j)^T(v_i - v_j)l$ (see §2.2). We can generalize our approach by using more powerful non-linear kernels that is defined on pairs of images. Such kernels cannot be decomposed as a simple product of two terms each of which is a function of only a single image, as is the case for a linear kernel. Such kernels enable us to use feature spaces in which distance measures can be non-linear. For example, the histogram of some feature like color or gabor filter responses in a localized window of an image forms a feature space. An appropriate distance between two histograms is the χ-square distance which is non-linear. It is known that for kernels that satisfy so-called Mercer conditions [6], we can solve the optimization problem in §2.2. We plan to investigate such extensions in the future.

References

[1] Amit, Y. and Geman, D. and Wilder, K. "Joint Induction of Shape Features and Tree Classifiers", *PAMI*, 19:11, pp. 1300-1305, 1997.

[2] Y.Freund. and R.E. Shapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting." *J. of Computer and System Sciences.* 55:1, pp. 119-139. 1997.

[3] Y.Freund. and L. Mason. "The Alternating Decision Tree Algorithm", *ICML99*, pp. 124–133, 1999.

[4] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, 1990.

[5] J. Lafferty. "Additive models, boosting, and inference for generalized divergences", *Proc. of the Twelfth Annual Conference on Computational Learning Theory (COLT'99).*

[6] S. Mika, G. Rtsch, J. Weston, B. Schlkopf, and K.-R. Mller. "Fisher discriminant analysis with kernels", *Neural Networks for Signal Processing IX*, pp. 41–48, 1999.

[7] B. Moghaddam and A. Pentland. "Probabilistic visual learning for object representation", *PAMI*, 19(7), pp. 696–710, 1997.

[8] "Beyond Eigenfaces: Probabilistic Matching for Face Recognition", B. Moghaddam and A. Pentland. *Intl. Conf. on Automatic Face and Gensture Recogntition*, Nara, Japan, April 1998.

[9] P. Phillips, H. Moon, P. Rauss and S. Rizvi. "The FERET Evaluation Methodology for Face-Recognition Algorithms", *CVPR*, Puerto Rico, pp. 137–143, 1997.

[10] Rowley, H.A. and Baluja, S. and Kanade, T. "Neural Network-Based Face Detection", *PAMI*, 20(1), pp. 23–38, 1998.

[11] R. E. Schapire and Y. Singer. "Improved boosting algorithms using confidence-rated predictions", *Machine Learning*, 37(3), 297–336, 1999.

[12] Schneiderman, H. and Kanade, T. "A Statistical Method for 3D Object Detection Applied to Faces and Cars", CVPR, I:746-751, 2000.

[13] M. A. Turk and A. P. Pentland. "Eigenfaces for recognition", *J. of Cogn. Neur.*, 3(1), pp. 71–86, 1991.

[14] Viola, P.A. "Complex Feature Recognition: A Bayesian Approach for Learning to Recognize Objects", MIT AI, 1995.

[15] Tieu, K. and Viola, P. "Boosting Image Retrieval", *CVPR00*, I, pp. 228–235, 2000.