Combining Simple Discriminators for Object Discrimination *

Shyjan Mahamud, Martial Hebert, and John Lafferty

Dept. of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213

Abstract. We propose to combine simple discriminators for object discrimination under the maximum entropy framework or equivalently under the maximum likelihood framework for the exponential family. The duality between the maximum entropy framework and maximum likelihood framework allows us to relate two selection criteria for the discriminators that were proposed in the literature. We illustrate our approach by combining nearest prototype discriminators that are simple to implement and widely applicable as they can be constructed in any feature space with a distance function. For efficient run-time performance we adapt the work on "alternating trees" for multi-class discrimination tasks. We report results on a multi-class discrimination task in which significant gains in performance are seen by combining discriminators under our framework from a variety of easy to construct feature spaces.

1 Introduction

A object discriminator can in general be thought of as any function that induces a partition of the space of images X. For example, a very crude example of a discriminator would be a function that tests whether the average intensity or some other simple statistic of a fixed subwindow of the input image crosses a threshold - in this case the image space is partitioned into two. Examples of more complex discriminators would be decision trees where each decision node is a simple discriminator as in the example above and the set of leaf nodes of the tree corresponds to the partition of the image space. Yet another example is a nearest prototype discriminator, in which a set of prototypes in some feature space (like color, shape, etc.) induces a partition, where each subset of image space in the partition contains the images that are closest to one of the prototypes (in other words, the partition is the voronoi diagram induced by the set of prototypes, see figure § 1). An ideal discriminator will induce a partition in which each subset of image space in the partition contains images of objects from a single class. Such an ideal discriminator might be hard to construct if the partition required is complex.

In practice, it might be easier to construct discriminators that induce simple partitions but which are far from ideal. We can then think of combining such

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

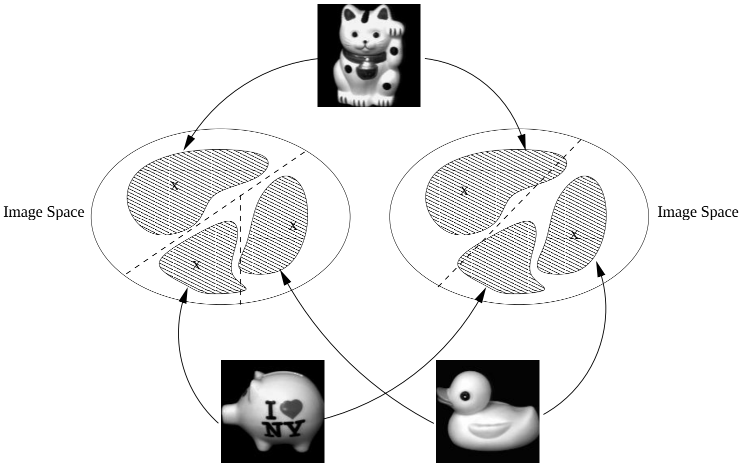


Fig. 1. The partition of image space induced by discriminators. Three classes of objects are shown within the image space (depicted as an ellipse). The discriminator on the left (a nearest 3-prototype discriminator for illustration, constructed in some feature space like color, edge, shape, etc., the prototypes are marked by ×) discriminates the three classes quite well but is complex. The discriminator on the right (with 2 prototypes) does not discriminate as well as the one on the left, but is simpler to implement. The partition boundaries in each case is given by the voronoi diagram induced by the prototypes which are at the center of each cell.

simple discriminators to create a more powerful discriminator. In this paper, we show how the maximum entropy (ME) framework [6, 8] can be used to combine simple discriminators. Under the ME framework, we seek a combined discriminator that is constrained to be consistent (in a manner that will be made precise shortly) with each of the simple discriminators, but is otherwise least committed. The resulting scheme is very similar to boosting [2] which was initially derived from computational learning principles (see § 3 for a discussion of the similarities and differences). The ME framework can also be generalized to handle regularization to avoid over-fitting in a principled manner [1] compared with standard boosting. In related work in computer vision, the framework has been used in the context of selecting good features for texture synthesis [10]. The framework is also known to be dual to the more familiar maximum likelihood (ML) framework for exponential distributions, but the justification for our approach is more natural under the ME framework.

A second issue we address is that of selecting which simple discriminators to combine. Given a large collection of N simple discriminators, we want to select a fixed number $K \ll N$ from this collection which when combined in the above framework gives the best discriminator possible. In practice, K might be constrained by the run time performance required. Two schemes have been proposed in the literature, one under the ME framework [10], and another under the ML framework [8]. Using the known duality between the two frameworks, we show that these two seemingly different selection schemes are in fact the same (see § 4).

For good run-time performance we would like to combine the simple discriminators in an efficient structure like a tree. For this purpose, we adapt the work on alternating trees [3] - a generalization of the more familiar decision trees - for multi-class discrimination (see § 6). In § 7 we illustrate our scheme on a multi-class discrimination task where significant gains in performance are seen by combining simple discriminators based on different types of features.

2 Formulation

For simplicity of implementation and wide applicability, in our work we use the nearest prototype (NP) discriminators as the simple discriminators that we choose to combine. As described earlier, the partition in image space induced by such a discriminator corresponds to the voronoi diagram (see figure § 1) with prototypes at the center of each cell. An NP discriminator has wide applicability since it can be constructed in any feature space that has a distance function. For example, the distance function for intensities in an image patch is typically just Euclidean, the distance for histograms is the non-linear χ^2 distance, the distance for edge maps is the hausdorff distance, etc. An NP discriminator is specified by the feature space that it is constructed in and the number and locations of the prototypes in that feature space. A particularly simple implementation that we use in our work, chooses the location of the prototypes from those derived from the set of training images rather than searching for the best prototypes in the whole feature space. This has the benefit of easy implementation even in feature spaces like histograms or edge maps where the distance function is non-linear and a search over the whole feature space may be infeasible. See § 5 for the details of our implementation of NP discriminators.

Our task is to learn a multi-class discriminator for objects of interest, given a training set $S = \{(x_1, c_i), (x_2, c_2), \ldots, (x_m, c_m)\}$, where $x_i \in X$ are the training images and $c_i \in \{1, \ldots, l\}$ are class labels. We assume that the objects are centered in the training images (see Fig.(4) for training examples from a discrimination task that we report in § 7).

Let us assume that a discriminator h has been trained on S using some procedure. Recall that a discriminator partitions the image space X. In the usual formulation, at run-time on input $x \in X$ a discriminator outputs the class labels c_j of all images x_j in the training set S that fall in the same subset of the image space partition induced by h that contains the input x. These class labels c_j can then be post-processed and weighted by the relative number w_j of training examples from each class that fall in the same partition as x to give a confidence value for each label. Alternatively a simpler implementation would report the class label(s) that is associated with the most number of training images that fall in the same partition as x.

In our work, we use an alternate formulation where the original multi-class discrimination task is reduced to an equivalent two-class problem as follows. This formulation simplifies the development of the framework as well as allows a finer-grained selection of "good" discriminators compared with the original multi-class framework. Instead of thinking of a discriminator as outputting the class label(s) given an input image, we can think of a discriminator that determines if a given *pair* of input images belongs to the same class or different classes of objects. Formally, given a pair of images (x_i, x_j), the discriminating function $h : X \times X \rightarrow \{-1, +1\}$, outputs the label $h(x_i, x_j) = +1$ if the pair belongs to the same partition induced by h, otherwise it outputs $h(x_i, x_j) = -1$. Under this formulation, the training set S gets transformed into a corresponding training set (which we will continue to denote as S) for the two-class problem :

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

where the new class labels y_{ij} is assigned $+1$ if the condition $c_i = c_j$ holds for the original class labels, and -1 otherwise. After learning a discriminator h from this set, at run-time on input x, just as for the original multi-class formulation described above, the discriminator outputs the class labels c_j corresponding to all the training examples x_j for which $h(x, x_j) = +1$, i.e. x_j falls in the same partition as x. Again, the class labels can be weighted by the relative number w_j of training examples from each class to give a confidence value for each label, or the class label(s) associated with the most number of training examples x_j can be reported. In the next section we consider the task of combining the outputs of a set of such simple discriminators under the maximum entropy framework.

3 Maximum Entropy Framework for Combining Discriminators

We can think of the tuple $((x_i, x_j), y_{ij}) \in S$ as an assertion which states that the pair of images (x_i, x_j) belongs to the same class (i.e., $y_{ij} = +1$) or in different classes ($y_{ij} = -1$). For a discriminator h, the expression $f(x_i, x_j, y_{ij}) = y_{ij} h(x_i, x_j)$ can be considered as a test which determines if the assertion is true ($f = +1$) or not ($f = -1$) for the discriminator h. We will call f the testing function corresponding to the discriminator h.

Suppose we wish to combine a given set of simple discriminators $h_k, k = 1, \ldots, K$. As discussed above, for each of the discriminators h_k, we can derive a testing function f_k that determines whether an assertion is true or not for the discriminator h_k. Similarly, we would like to derive a testing function F for the combined discriminator H. Intuitively, F should combine the outputs of each of the f_k on an input assertion to determine the truth of that assertion. Rather than derive a binary testing function F (with outputs in $\{-1, +1\}$ as is the case for each of the f_k), we will seek a more general probability distribution p over the possible output values $\{-1, +1\}$ that gives the likelihood that an input assertion is true or not given the outputs of the testing functions f_k on that assertion. The likelihood will give a measure of the confidence in a prediction (truth or falsity of an assertion) made by the combined discriminator. Formally for our purposes, we would like to derive a conditional probability distribution $p(y_{ij} \mid (x_i, x_j))$ that measures the likelihood that $y_{ij} = +1$ or $y_{ij} = -1$ given the outputs of $f_k, k = 1, \ldots, K$ on any assertion $((x_i, x_j), y_{ij})$.

What should the constraints be in selecting a probability measure p from the space of all probability measures ? The outputs of the testing functions f_k on all the assertions in the training set S provide one source of constraints. Obviously since S is finite and the space of probability measures is infinite, we need additional a priori constraints to properly constrain the selection of p. The maximum entropy (ME) framework [6, 8] is based on the principle that the only constraint that we can reasonably impose on the probability measure p a priori is that it be *consistent* with the outputs of the tests f_k over all training assertions for each k, but is otherwise "least-committed". The probability measure p is defined to be consistent with a test f_k for a given k if a set of chosen statistics of f_k over all possible assertions $z \in X \times X \times \{-1, +1\}$ under the probability measure p matches the empirical statistics over the training set S. Intuitively, we want to select the probability measure that is at least in agreement with the training data. The simplest statistic is the mean. The empirical mean is defined by :

$$\langle f_k \rangle \equiv \frac{1}{|S|} \sum_{((x_i,x_j),y_{ij}) \in S} f_k(x_i, x_j, y_{ij})$$

Note that in principle one can impose additional constraints corresponding to higher order moments. But in practice, typically only the constraint for the mean is imposed since estimating the higher-order moments reliably requires more and more training data as the order of the moment increases.

Given these constraints, the ME framework searches for the "least committed" probability measure. Intuitively, the least committed measure under no constraints is the uniform probability measure. As we add constraints, we would like to keep the measure as close to uniform as possible while satisfying the constraints. More generally, we would like to be as close to a prior measure q_0 that may not be uniform, and is task dependent but data independent. For our task, the distance between two conditional probability measures p and q can be measured by the following conditional Kullback-Leibler (KL) divergence [8] :

$$D(p, q) = \frac{1}{|S|} \sum_{(x_i,x_j) \in S} \sum_{y_{ij} \in \{-1,+1\}} p(y_{ij} \mid (x_i, x_j)) \log \frac{p(y_{ij} \mid (x_i, x_j))}{q(y_{ij} \mid (x_i, x_j))}$$

which is non-negative and 0 iff $p = q$.

Let $\mathcal{M}$ be the space of all conditional probability measures. Define the *feasible* set $\mathcal{F} \subset \mathcal{M}$ as :

$$\mathcal{F} \equiv \{p \in \mathcal{M} \mid E_p[f_k] = \langle f_k \rangle \text{ for all } k\}$$

where $E_p[\cdot]$ denotes the expected value operator under p. Then the ME framework entails the solution of the following problem : minimize $D(p, q_0)$ subject to

$p \in \mathcal{F}$ and a fixed prior measure q_0. In our task, we assume a uniform prior for q_0. In this case it can be shown [7] that by setting up an appropriate Lagrangian, the optimal solution denoted by p_{ME} is the logistic function :

$$p_{\mathrm{ME}}(y_{ij} \mid (x_i, x_j)) = \frac{1}{1 + \exp\left(-\sum_k \lambda_k f_k(x_i, x_j, y_{ij})\right)} \tag{2}$$

where λ_k is the set of Lagrange multipliers, one for each of the constraints $E_p[f_k] = \langle f_k \rangle$. These multipliers can be determined by minimizing the following objective function :

$$J(\lambda, f) = \sum_{z \in S} \log(1 + \exp(-\langle \lambda, f(z) \rangle)) \tag{3}$$

where $\langle \cdot, \cdot \rangle$ denotes vector dot product.

The dual objective looks very similar to the exponential cost function that is minimized in boosting [2]. In fact it has been recently shown [7] how boosting can be derived from the ME framework by using slightly different constraints than those used here. The boosting framework was originally inspired by computational learning principles where the exponential cost function is a continuous upper bound to the true discrete error function for the boosting classifier. Here we have derived the objective to be minimized from first principles under the ME framework. Furthermore, the ME framework allows us to generalize the scheme in a principled manner to avoid over-fitting [1] if needed.

4 Selecting Good Features

First we note a duality between the ME framework and and the maximum likelihood (ML) framework over exponential measures. Consider the family of conditional exponential probability measures :

$$Q \equiv \left\{ p \in \mathcal{M} \;\mid\; p(y_{ij} \mid (x_i, x_j)) \propto q_0(y_{ij} \mid (x_i, x_j)) e^{\langle \lambda, f(x_i, x_j, y_{ij}) \rangle},\ \lambda \in \mathbb{R}^K \right\}$$

Let $\tilde{p}((x_i, x_j), y_{ij})$ be the empirical distribution over the assertions determined by the training set S (since by construction each assertion in S is unique, $\tilde{p}$ will simply take the value $1/|S|$ over each assertion in S and 0 elsewhere). The log-likelihood L of a probability measure p with respect to $\tilde{p}$ is defined as :

$$L(\tilde{p}, p) \equiv -D(\tilde{p}, p)$$

It has been shown [8] that the probability measure p_{ML} that maximizes the likelihood over the exponential family Q is the same as p_{ME}. Thus the two optimization problems are dual to each other.

Now we consider the problem of selecting good discriminators. Assume that we are given a large but finite collection $\mathcal{C}$ of discriminators $h_k, k = 1, \ldots, N$. This is the case in our work where we use nearest prototype discriminators for which the prototypes are chosen from the finite set of prototypes in the training set (see § 5). We wish to choose $K \ll N$ discriminators from this collection that gives the best combined discriminator. In practice, K will be determined for example from run-time performance considerations.

We can formulate two criteria for choosing the best K discriminators, one for each of the two frameworks. Under the more familiar ML framework [8], for a fixed family of probability measures Q, the criterion is to choose the K discriminators that maximize the likelihood function $L(\tilde{p}, p), p \in Q$. Formally the ML criterion can be stated as follows :

Criterion ML. *For a fixed choice of K discriminators $\{h_1, \ldots, h_K\} \subset \mathcal{C}$, let p^* be the probability measure that maximizes the likelihood $p^* = \underset{p \in Q}{\operatorname{argmax}} L(\tilde{p}, p)$. Choose the K discriminators for which $L(\tilde{p}, p^*)$ is maximum over all choices of K discriminators from $\mathcal{C}$.*

Under the ME framework on the other hand, the authors in [10] propose the use of the *mini-max* entropy criterion. The context of their work was selection of good features for texture synthesis. In their original formulation, the criterion assumes a uniform prior model q_0 and chooses the K features such that the resulting maximum entropy probability measure p_{ME} has *minimum* entropy over all choices of K discriminators. This criterion might seem less intuitive at first than the ML criteria. It is based on the notion that the entropy of the probability measure determined by a given choice of K discriminators indicates how "informative" the discriminators are, the discriminators being more informative the lower the entropy. Thus the mini-max entropy criterion chooses the K most informative discriminators. Since minimizing (maximizing) the entropy of a distribution p is the same as maximizing (minimizing) the KL divergence $D(p, q_0)$ where q_0 is set to the uniform distribution, the original mini-max entropy criterion can be generalized for arbitrary priors q_0 and formally stated as follows :

Criterion ME. *For a fixed choice of K discriminators $\{h_1, \ldots, h_K\} \subset \mathcal{C}$, let p^* be the maximum entropy probability measure with constraints determined by the corresponding testing functions $f_1, \ldots, f_K$, i.e. $p^* = \underset{p \in F}{\operatorname{argmin}} D(p, q_0)$. Choose the K discriminators for which $D(p^*, q_0)$ is maximum over all choices of K discriminators from $\mathcal{C}$.*

We show in the following theorem that due to the duality between the ME and ML framework, these two seemingly different criteria are in fact the same when the ML criterion is applied to the exponential family Q :

Theorem 1. *A set of K features optimize the ME criterion iff they also optimize the ML criteria for the exponential family.*

Proof. We first state an analogue of the Pythagorean theorem for the KL divergence [8] :

$$D(p, q) = D(p, p^*) + D(p^*, q), \text{ forall } p \in \mathcal{F}, q \in \bar{Q}$$

where $\bar{Q}$ is the closure of Q and by the duality theorem [8],

$$p_{\mathrm{ML}} = \underset{p \in \bar{Q}}{\operatorname{argmin}} D(\tilde{p}, p) = p^* = \underset{p \in F}{\operatorname{argmin}} D(p, q_0) = p_{\mathrm{ME}}$$

We set $p = \tilde{p}$, the empirical distribution from the training set S, and $q = q_0$ a prior measure, both of which are fixed for a given learning task and thus $D(\tilde{p}, q_0)$ is constant. Also since the log-likelihood is given by $L(\tilde{p}, p) = -D(\tilde{p}, p)$, we have :

$$L(\tilde{p}, p^*) = D(p^*, q_0) + const$$

Thus choosing the K discriminators that maximize the likelihood $L(\tilde{p}, p^*)$ also maximize the KL distance $D(p^*, q_0)$ and thus the K discriminators that optimize the ML criterion also optimize the ME criterion and vice-versa.

In practice, instead of searching for the best K discriminants all at once, we select each discriminant in a greedy fashion one at a time. At iteration k, we have $k-1 < K$ discriminants that have been chosen in the previous iterations. We then choose to add the discriminant that along with the previous $k-1$ discriminants gives the biggest gain in the ML or ME criteria. Concretely, this means we choose the discriminant that along with the previous discriminants minimizes the objective function J in (3). Optimization of J is a convex optimization problem in $\lambda_1, \ldots, \lambda_k$ that can be done using standard numerical techniques [9].

5 Nearest Prototype Discriminators

In this section, we describe in more detail the particular type of discriminators that we use in our work. As mentioned in the introduction, nearest prototype discriminators are easy to implement and widely applicable. Such a discriminator is specified by the feature space the prototypes come from (color, histograms, shape, etc.), as well as the number and locations of the prototypes. For example, in the experiments that we report later, we use edge maps of images as one of the features. In this case, a prototype in the discriminator will be an edge map. The image space will be partitioned by the set of prototypes in the discriminator where each partition corresponds to the subset of the image space that is closest to one of the prototypes. The distance function that determines this partition is the hausdorff distance between edge maps.

If we restrict the choice of the locations of the prototypes to be those in the training images, we get a finite collection of discriminators $\mathcal{C}$ to choose from : for each feature space, given m training images, there are a total of $\mathrm{O}(r^m)$ discriminators with r prototypes. At each iteration of the greedy search, we can in principle exhaustively search $\mathcal{C}$ for the best discriminator (that which minimizes J the most along with the other discriminators that were chosen in the previous iterations).

In practice, this does not scale well when m is large. Instead, we use a simple sampling technique that trades off the quality of the discriminator found for a speed-up in the search process. Rather than searching for the best discriminator at each iteration, we will search for a discriminator in the top s percentile (for example $s = 0.1\%$), where the discriminators are ranked according to the how much they minimize J. Under this search strategy, we can show that with high probability we can find a discriminator in the top s percentile by uniformly sampling the finite set of discriminators $\mathcal{C}$ a fixed number of times n that is *independent* of the total number m of training examples and the number of prototypes r required. More precisely, let $0 < \delta < 1$, then it can be shown that we need to sample :

$$n > \frac{\log(1/\delta)}{s}$$

discriminators such that at least one of them is in the top s percentile with probability at least $1 - \delta$.

6 Efficient Organization of Discriminators

For run-time performance, we would like to combine the discriminators that we choose in an efficient structure. For this purpose, we adapt the work on "alternating trees", which is a generalization of decision trees (see Fig. (2)). The salient feature that distinguishes alternating trees from regular decision trees is that a node in an alternating tree can have multiple decision nodes as children. The term "alternating" refers to alternating levels of two types of nodes : *prediction* nodes, which in our task predicts whether two images belong to the same class or not, and *discriminator* nodes which correspond to the discriminators that we choose. A predictor node is associated with a subset U of the image space X that pass through the the sequence of discriminators from the root to the predictor node. In other words, the subset U is determined by the conjunction of discriminators from the root to the predictor node. We can think of the rest of the image space $X - U$ as the subset that the predictor node "abstains" from. The root node of the alternating tree is a predictor node associated with the entire image space X. A predictor node can have a multiple number of discriminators as children. Each discriminator node partitions the image space subset U associated with its parent predictor node. In turn, a discriminator node has predictor nodes as children, each of which corresponds to a subset of the image space in the partition induced by the parent discriminator node. The possibility of multiply partitioning the image space gives the alternating tree more flexibility compared with standard decision trees. In fact, by constraining each predictor node to have at most one discriminator node as a child, we get a standard decision tree after combining each predictor node with its sole discriminator child (if any).

In order that an alternating tree of discriminators can be incorporated under the maximum entropy framework, we need to derive the testing functions f_k for each discriminant h_k in the tree. Recall from § 2 that the testing function f is derived from the partition induced by a discriminator h : if $h(x_i, x_j)$ denotes whether two images are in the same ($h(x_i, x_j) = +1$) or in different ($h(x_i, x_j) = -1$) partitions induced by the discriminator, then $f(x_i, x_j, y_{ij}) = y_{ij} h(x_i, x_j)$ tests whether the assertion $((x_i, x_j), y_{ij})$ is true for discriminator h. The partition associated with a discriminator h_k in an alternating tree is the partition induced on the subset U of image space associated with its parent predictor node, as well as the compliment subset $X - U$ that the parent abstains from. Concretely, if

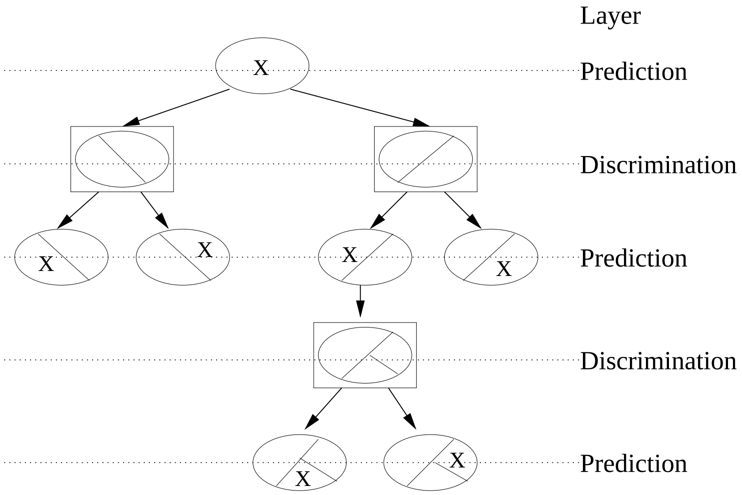


Fig. 2. Alternating Trees. The tree alternates between prediction nodes (ellipses) and discriminator nodes (boxes). Each prediction node is associated with a subset U of the image space (marked by ×), which is partitioned by any discriminator child of the prediction node. A prediction node outputs $+1$ when a pair of images are both in either the subspace U associated with the node or in the compliment $X - U$, otherwise it outputs -1. Each prediction node can have multiple discriminator nodes as children, each of which partitions the subset U of image space associated with its parent predictor node.

the discriminator h_k partitions U into U_1 and U_2 for example, then the partition $\{X - U, U_1, U_2\}$ is associated with h_k, from which the corresponding testing function f_k is derived.

6.1 Summary

We now outline the iterative scheme for selecting nearest prototype discriminators in a greedy manner while composing them in an alternating tree. Assume that we can extract a variety of features from images (intensity, edge maps, histograms, etc.). Thus each image belongs to a set of feature spaces $g \in \mathcal{G}$. Denote by $\mathcal{C}(g, S)$ the finite collection of nearest prototype discriminators constructed in feature space g, where the locations of the prototypes are chosen from those in the training set S. Let $h^*(g, S) \in \mathcal{C}(g, S)$ be the nearest prototype discriminant optimizing the objective function J by the sampling scheme described in § 5.

The alternating tree is initialized to a predictor node that corresponds to the whole image space X. At the start of iteration k, let T be the alternating tree constructed so far in the previous iterations. Let $S_i \subset S$ denote the subset of training examples that reach the predictor node P_i in T. At iteration k, we choose the discriminant h^* that minimizes the objective function J from among the set of all $h^*(g, S_i)$ over all choices of feature space $g \in G$ and training sets S_i associated with each predictor node P_i in the tree. Note that since a predictor node P_i can have multiple children, each predictor node will participate in all iterations, unlike the case for a standard decision tree where only the current leaf nodes are considered.

At the end of a fixed number of iterations, we output the final alternating tree with discriminators h_k along with the optimal Lagrange multipliers λ_k. The corresponding testing functions f_k along with λ_k determines the maximum entropy conditional probability distribution $p_{\mathrm{ME}}(y_{ij} \mid (x_i, x_j))$ given by (2). p_{ME} can be used to determine if an input image x_i belongs to the same class of objects as that of a training example x_j as follows. Define the log-ratio for a pair of images (x_i, x_j) with respect to the probability distribution p_{ME} as :

$$\rho(x_i, x_j) \equiv \frac{1}{2} \log \frac{p_{\mathrm{ME}}(y_{ij} = +1 \mid (x_i, x_j))}{p_{\mathrm{ME}}(y_{ij} = -1 \mid (x_i, x_j))} = \sum_k \lambda_k h_k(x_i, x_j) \tag{4}$$

The log-ratio indicates whether the input image x_i is in the same class as the training image x_j if $\rho(x_i, x_j) > 0$, otherwise they belong to different classes. The magnitude of the log-ratio can be thought of as the margin or confidence in making the prediction. Using this observation, on input x_i we output the class label c_j of the training example x_j that is predicted to be in the same class as x_i ($\rho(x_i, x_j) > 0$) and whose margin is largest ($|\rho(x_i, x_j)|$ is maximum over all x_j). Figure 3 gives the pseudo-code for the whole scheme.

```
Initialize :
  1. Initialize the alternating tree $T$ with a root predictor node.
  2. Let $G$ be the collection of feature spaces in which discriminators
     will be constructed.
  3. Let $S_i \subset S$ denote the training set that reaches a predictor node
     $P_i \in T$ from the root.
do for $K$ iterations
  1. For each $g \in G$ and each $S_i$ corresponding to predictor node
     $P_i \in T$, find the optimal discriminant $h^*(g, S_i)$ using the sampling
     scheme in § 5.
  2. Add the best discriminator $h^*$ that minimizes $J$ from among all
     $h^*(g, S_i)$ to the alternating tree $T$.
Output : The discriminating rule 4 with the Lagrange multipliers $\lambda_k$
that optimizes $J$.
```

Fig. 3. Pseudo-code for the iterative greedy scheme.

7 Results

We illustrate our approach on a multi-class discrimination task, where we show the benefits of combining simple discriminants from different feature spaces (edge

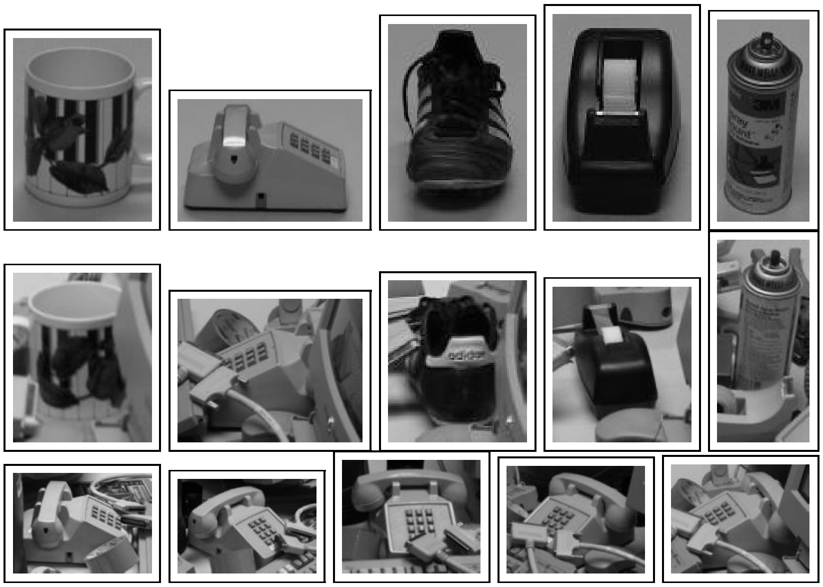


Fig. 4. Example training (top row) and testing (middle row) views of 5 objects. For each object 12 training views were taken in a circle around the object, and 24 testing views were taken with varying clutter and occlusion. The bottom row shows more testing examples for one the objects.

maps and intensity histograms). In our task, we have five objects that need to be discriminated from each other (see Fig. (4)). For each object, we collected 12 grey valued training images spanning the viewpoints around the object placed at a desk. The training images have the objects at the center and are taken from a fixed distance to the camera.

For our experiments, nearest prototype discriminators constructed from the following two types of feature spaces are combined to construct the alternating tree during training :

Edge Maps Edge contours are quite insensitive to some illumination and viewpoint changes. Two edge maps can be compared using the robust Hausdorff measure [5]. The Hausdorff distance $h(A, B)$ between two edge maps A and B is given by :

$$h(A, B) \equiv \max\,(\max_{a \in A} \min_{b \in B} ||a - b||, \max_{b \in B} \min_{a \in A} ||a - b||)$$

The distance is made robust to occlusion and clutter by replacing the maximum distance over all points in one set to the closest neighbour in the other set in the above expression by the distance at some percentile. In our experiments we use a percentile of 75%.

Second-Order Histograms A second order or correlation histogram [4] measures the distribution of grey value pixel intensities while taking into account the relative positions of the pixels. Formally the histogram $C_\tau(g, g')$ measures the number of pixel pairs in a subwindow centered around the object (80×80 in our experiments) that are a certain distance τ apart and which have intensities g, g' respectively. The intensities in our experiments are quantized to 15 levels. A second order histogram is more informative than a simple histogram of pixel intensities due to the inclusion of the relative positions of the pixels. In our experiments τ is set to 5 or 10 pixels either horizontally or vertically. Two histograms can be measured by the χ^2 distance :

$$\chi^2(C_1, C_2) = \sum_{b \in \text{bins}} \frac{(C_1(b) - C_2(b))^2}{C_1(b) + C_2(b)}$$

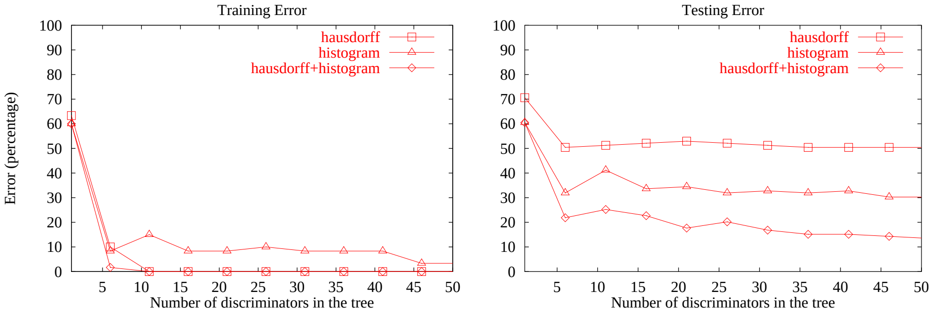
where b runs over the set of bins in a histogram.

In general, we can consider other types of feature spaces like color, shape from shading etc.

For testing, we took 24 images of each object under varying levels of occlusion for a total of testing 120 images. (see Fig. (4) for examples). In general, a complete recognition system would typically be divided into two stages : an object detection stage, which might be based on grouping principles or other pattern recognition techniques, will first identify regions of the image that might possibly contain objects of interest, and a second stage where these regions are then passed to a discriminator for the identification of a specific object. This is the approach for example in face recognition, where a general face detector can be used to first identify regions in the input image that contains a face which can then be recognized by a separate module. An object detection stage is beyond the scope of this paper, and so in our work we will assume that the objects of interest are roughly centered in the test images. Note that the testing images still have significant clutter and occlusion.

Fig. (5) shows the training and testing error versus the number of discriminators in the alternating tree (or equivalently the iteration number in the greedy selection scheme). In practice, the number of discriminators in the tree will be dictated by the trade-off between the run-time performance and the testing error desired. For both the testing and training, three curves are shown based on using discriminators from both the feature spaces described above ("hausdorff+histogram" in the plot), or using discriminators from just one of the feature spaces ("hausdorff" and "histogram" in the plot). As expected, using discriminators from multiple feature spaces can significantly enhance the performance of the algorithm. In our experiment, at the end of 50 iterations the testing error was reduced from 50.42% when using only edge maps or from 30.25% when using only intensity histograms to 13.45% when using both. Most of the error is due to one of the objects (phone) which does poorly with discriminators from either of the feature spaces. It is quite possible that other feature spaces that we haven't considered can bring the error down even further.

We note that in our approach it is conceptually no more difficult to combine discriminants from multiple feature spaces than it is to combine discriminants



(a) (b)

Fig. 5. Training (a) and testing (b) error vs. number of discriminators in the alternating tree. The testing error when using discriminators from both edge maps and histograms ("hausdorff+histogram") is significantly better (13.45% error for 50 nodes in the tree) than when using discriminators from either feature space alone ("hausdorff" at 50.42% error and "histogram" at 30.25% error).

from just one feature space. This is one of the advantages inherent in our approach compared with other approaches like neural networks that have difficulty in working with disparate feature spaces.

8 Conclusion

We have shown how the maximum entropy framework can be used to combine simple discriminants from a variety of feature spaces in a principled manner. The nearest prototype discriminator is simple to implement and can be constructed in any feature space with a corresponding distance function. We have shown that two criteria for selecting discriminants - one in the maximum entropy framework and the other in the dual maximum likelihood framework - are in fact equivalent. For run-time performance, we have adapted the work on alternating trees for combining simple discriminators into an efficient structure. Experiments on a multi-class discrimination task validated the approach and showed significant gains by combining simple discriminants constructed in different feature spaces.

References

1. Chen, S., Rosenfeld, R.: A survey of smoothing techniques for ME models. IEEE Transactions on Speech and Audio Processing **8**(1) (2000)
2. Freund, Y., Shapire, R.E.: A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. J. of Computer and System Sciences **55**(1) (1997) 119–139
3. Freund, Y., Mason, L.: The Alternating Decision Tree Algorithm. ICML99 (1999) 124–133

4. Haralick, R.M.: Statistical and Structural Approaches to Texture. Proc. 4th Intl. Joint Conf. Pattern Recognition (1979) 45–60
5. Huttenlocher, D., Klanderman, G., Rucklidge, W.: Comparing Images Using the Hausdorff Distance. IEEE Transactions on Pattern Analysis and Machine Intelligence **15**(9) (1993) 850–863
6. Jaynes, E.T.: Information theory and statistical mechanics. Physical Review **106** (1957) 620–630
7. Lebanon, G., Lafferty, J.: Boosting and Maximum Likelihood for Exponential Models. Advances in Neural Information Processing Systems **14** (2001)
8. Della Pietra, S., Della Pietra, V., Lafferty, J.: Inducing features of Random fields. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**(4) (1997)
9. Schapire, R. E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Machine Learning **37**(3) (1999) 297–336
10. Zhu, S. C., Wu, Y., Mumford, D.: Filters, Random Fields and Maximum Entropy (FRAME). IJCV **27**(2) (1998) 1–20