Discriminative Distance Measures for Object Detection

Shyjan Mahamud

CMU-CS-02-161

December 2002

School of Computer Science
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Martial Hebert, co-Chair
Reid Simmons, co-Chair
Takeo Kanade
Jianbo Shi
Pietro Perona

Submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy

This research was sponsored by NSF Grant IIS-9907142 and DARPA HumanID ONR N00014-00-1-0915.

Keywords: object detection, nearest neighbor search, optimal distance measure, efficient search, discrimination.

Abstract

The reliable detection of an object of interest in an input image with arbitrary background clutter and occlusion has to a large extent remained an elusive goal in computer vision. Traditional model-based approaches are inappropriate for a multi-class object detection task primarily due to difficulties in modeling arbitrary object classes. Instead, we develop a detection framework whose core component is a nearest neighbor search over object parts. The performance of the overall system is critically dependent on the distance measure used in the nearest neighbor search.

A distance measure that minimizes the mis-classification risk for the 1-nearest neighbor search can be shown to be the probability that a pair of input measurements belong to different classes. This pair-wise probability is not in general a metric distance measure. Furthermore, it can out-perform any metric distance, approaching even the Bayes optimal performance.

In practice, we seek a model for the optimal distance measure that combines the discriminative powers of more elementary distance measures associated with a collection of simple feature spaces that are easy and efficient to implement; in our work, we use histograms of various feature types like color, texture and local shape properties. We use a linear logistic model combining such elementary distance measures that is supported by observations of actual data for a representative discrimination task. For performing efficient nearest neighbor search over large training sets, the linear model was extended to discretized distance measures that combines distance measures associated with discriminators organized in a tree-like structure. The discrete model was combined with the continuous model to yield a hierarchical distance model that is both fast and accurate.

Finally, the nearest neighbor search over object parts was integrated into a whole object detection system and evaluated against both an indoor detection task as well as a face recognition task yielding promising results.

Contents

Chapter 1

Introduction

The reliable detection of an object of interest in an input image with arbitrary background clutter and occlusion has to a large extent remained an elusive goal in computer vision since the beginning. In the most common formulation of a multi-class object detection task, we would like to detect the presence or absence of an object of interest in an input image, given a prior training set (2D or 3D data) for the objects of interest. The factors that confound reliable detection include background clutter, occlusion of the objects of interest and the variability in viewing conditions. Figure 1.1 shows examples of the kind of objects that we would like to detect as well as examples of clutter that we would like the detection scheme to be robust against.

Previous approaches to object detection can be grouped under various criteria. For our purposes, we shall make the distinction between *model-based* or generative-based approaches on the one hand (Roberts, 1965; Chin and Dyer, 1986; Kane et al., 1991; Arman and Aggarwal, 1993b; Huttenlocher and Ullman, 1990) and *exemplar-based* or appearance-based approaches on the other hand (Mel, 1997; Murase and Nayar, 1997; Nayar et al., 1996; Shapiro and Costa, 1995; Selinger and Nelson, 2001; Nelson and Selinger, 1998; Worthington and Hancock, 2000; Schiele, 1997; Huang et al., 1999). Broadly speaking, in the former class of approaches, a model for each object of interest is assumed that can generate new images of the objects by varying the parameters of the model. An extreme example is a 3D CAD model for each object of interest (Arman and Aggarwal, 1993a) along with a model-independent imaging process parametrized by

Figure 1.1: Sample object classes (top row) along with sample scenes (middle row) with one of the objects of interest under clutter and occlusion. The bottom row shows more sample scenes for one the objects.

viewing and lighting conditions. New views of the object are generated by specifying parameters for the viewing and lighting conditions. As another example, the class of faces can be modeled quite well by a low-dimensional linear subspace in image space (Turk and Pentland, 1991). New views of a face are generated by linearly combining the basis vectors spanning the subspace. As a last example, objects can be modeled using a linear combination of views (Ullman and Basri, 1991) where the model is a set of prototype images for the object along with the locations for a set of features in the prototype views. The locations of the features in novel views of the object can be obtained by linearly combining the locations of the features in the prototype views.

The main difficulty in such generative approaches is the development of good generative models and their estimation from training data. This is especially a problem for a general object detection task in which we are interested in detecting an arbitrary set of objects. Each object of interest might require a different generative model, each of which needs to be estimated. For example, it is reasonable to expect that the class of chairs might require a different type of generative model than the class of cars. Furthermore, it is not clear that all objects of interest can be easily modeled with some generative model.

Exemplar-based approaches on the other hand avoid the need for explicit models of objects. Instead, a training set of images under various viewing directions and scene illumination is acquired for each object of interest. Perhaps the simplest exemplar-based approach is to use the training images as templates that are matched against the input image. The object class label of the training image that best matches the input image is reported. In other words, the input image is classified by a nearest neighbor search among the training images, where the distance between the template and the input image is based on some feature space like color, texture or shape, or more generally a combination of elementary features. Unlike generative approaches where different object classes might in general require different generative models, exemplar-based approaches can be typically applied uniformly to all objects of interest.

1.1 Nearest Neighbor Framework

Nearest neighbor search is one of the simplest forms of an exemplar-based method (Dasarathy, 1991). Formally, we are given a training set $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ where the x_i are training images and y_i are corresponding class labels. We are also given a distance measure $d(x, x')$ that is used to find the nearest neighbor in S of an input image. The one nearest neighbor rule reports the class label y_i of the training image x_i that is the nearest neighbor of the input image. Thus the classification performance of the nearest neighbor rule is solely determined by the training set S and the distance measure d. More generally, the nearest neighbor rule can depend on the K nearest neighbors.

Most work on nearest neighbor search assumes a fixed distance measure. However, it is easy to show that the choice for a distance measure can significantly affect the classification performance of the nearest neighbor rule. More recent work (Short and Fukanaga, 1981; Fukanaga and Flick, 1984; Hastie and Tibshirani, 1996; Blanzieri and Ricci, 1999; Friedman, 1994) has begun to exploit the gain in classification performance possible by using good distance measures. The optimal distance measure depends on the task at hand. In the case of object detection, the search for an optimal distance measure is confounded by the fact that we might want to use a combination of features to discriminate objects, since in a multi-class object detection task, no one feature type will likely be suitable for discriminating all objects from each other. Instead, it is more likely that different feature types and/or their combinations are required for discriminating different pairs of object classes from each other. For example, two different object classes A and B may be distinguished by color alone, while class A and yet another object class C maybe of the same color but can be distinguished by shape properties. It is not clear a priori how to construct a single optimal distance measure between images when the representation uses a combination of different features like color and shape. Furthermore, different features may have differing discriminative powers and a good distance measure should take into account such differences.

What should the optimal distance measure be ? Intuitively a distance measure that ignores variations within the same class (for example, variations due to lighting and viewing conditions) while enhancing variations between images from different classes should be ideal for use in a nearest neighbor search. Objectively, *the optimal distance measure should be the one that maximizes the classification performance of the nearest neighbor rule*. We will show in the next chapter that a distance measure that directly optimizes the classification performance can be expressed simply in terms of the odds ratio that a pair of images x and x' belong to the same class :

$$d(x, x') \equiv \log \frac{p(y \neq y' \mid x, x')}{p(y = y' \mid x, x')} \tag{1.1}$$

where y and y' are the corresponding class labels. Clearly, this distance measure satisfies the intuitive requirement that pairs of images from the same object class should be close to each other compared with pairs of images from different object classes.

Thus the problem of finding the optimal distance measure reduces to the problem of modeling and estimating the probability distribution that a pair of images belong to the same class or to different classes. This pair-wise "discriminative" distribution $p(y \neq y' \mid x, x')$ can in principle be computed from a generative model $p(x|y)$ for each class. So in principle, the problem can further be reduced to first estimating the generative models for each class. However, we are then faced with all the pitfalls of modeling and estimating generative models discussed above.

In our work instead, *we propose to model and estimate the pair-wise distribution directly*. The basic intuition for why this direct approach should be more feasible in practice is that the pair-wise likelihood depends only on the discriminative features whereas estimating a generative model first requires modeling the role of all features irrespective of their discriminative value.

How do we model the pair-wise distribution directly ? In general, for an arbitrary multi-class detection task, the optimal distance measure cannot be expected to assume any particular parametric model. Any choice for modeling the distance measure should be dictated by what the data suggests for a particular detection task as well as other factors like ease of implementation and analyzability.

Our basic approach will be to model the optimal distance measure by combining more "elementary" distance measures. An elementary distance measure is defined on simple feature spaces like color, local shape properties or texture. Our motivation for basing our approach on combining such elementary distance measures is primarily the ease of implementation for such an approach since there are plenty of choices for such simple feature spaces that have been well-studied in the literature that are easy and efficient to implement in practice. For example, we can consider simple histograms of features, for which one choice for the elementary distance measure is the χ^2 distance. Other simple feature spaces include edge maps with the Hausdorff distance measure (Huttenlocher et al., 1993), shape contexts (Belongie et al., 2002), or normalized pixel intensities with the simple euclidean distance measure (Nayar et al., 1996).

In general, each of the simple feature spaces by itself cannot be expected to be sufficient at the discrimination task at hand. Thus we seek to combine the discriminative powers of a set of such simple feature spaces in our model. The ideal set of feature spaces to use is that which complements each other well for the discrimination task at hand.

How should the elementary distance measures be combined ? We can motivate our answer to this question by first taking a look at some actual data from an indoor discrimination task that we are interested in. In this thesis, we will use histograms of various features like color, local shape properties and texture as the simple feature spaces that we would like to combine in our model. Histograms were chosen since they can be efficiently computed from an input image and are stable representations with respect to a fair amount of distortions in viewing conditions. See Chapter 6 for details.

The distribution $p(y \neq y' \mid x, x')$ that we wish to model is a function of pairs of images. Figure 1.2(a) shows the distribution of distances in a local shape histogram feature space between images of object parts sampled from a collection of 15 objects and randomly sampled image patches of background clutter (see Chapter 7 for a description of these objects, and Chapter 6 for how objects are decomposed into parts). The elementary distance measure chosen is the simple L_1 distance measure. See $\S$ 3.2 for the distribution of distances in the other feature spaces that we use, namely color and texture.

As can be seen from the figure, the distance scores between images falls into two distributions depending on whether the pair of images come from the same object part class or from different classes (including clutter). The distance score in this feature space can be roughly divided into three intervals along the x-axis. It can be claimed with high confidence that if a pair of images has a distance score that falls in either of the two extreme intervals, then the images come from either the same class (in the case of the left-most interval) or from different classes (in the case of the right-most interval). For the middle interval, the within-class or out-of-class membership is more uncertain.

Figure 1.2(b) plots the empirically determined log odds ratio (1.1) which is the transform of the pair-wise distribution $p(y \neq y' \mid x, x')$ that we wish to model. As illustrated in Figure 1.2(c), the uncertain middle interval of the log odds ratio plot can be well-modeled as a linear function of the distance score. Similar observations hold true for each of the other feature spaces that we use (color and texture), see $\S$ 3.2. These observations are used to justify approximating the optimal distance measure by *linearly* combining the elementary distance measures associated with the simple feature spaces that we use in our work. See Chapter 3.

It will turn out that the optimal coefficients in the linear combination of such

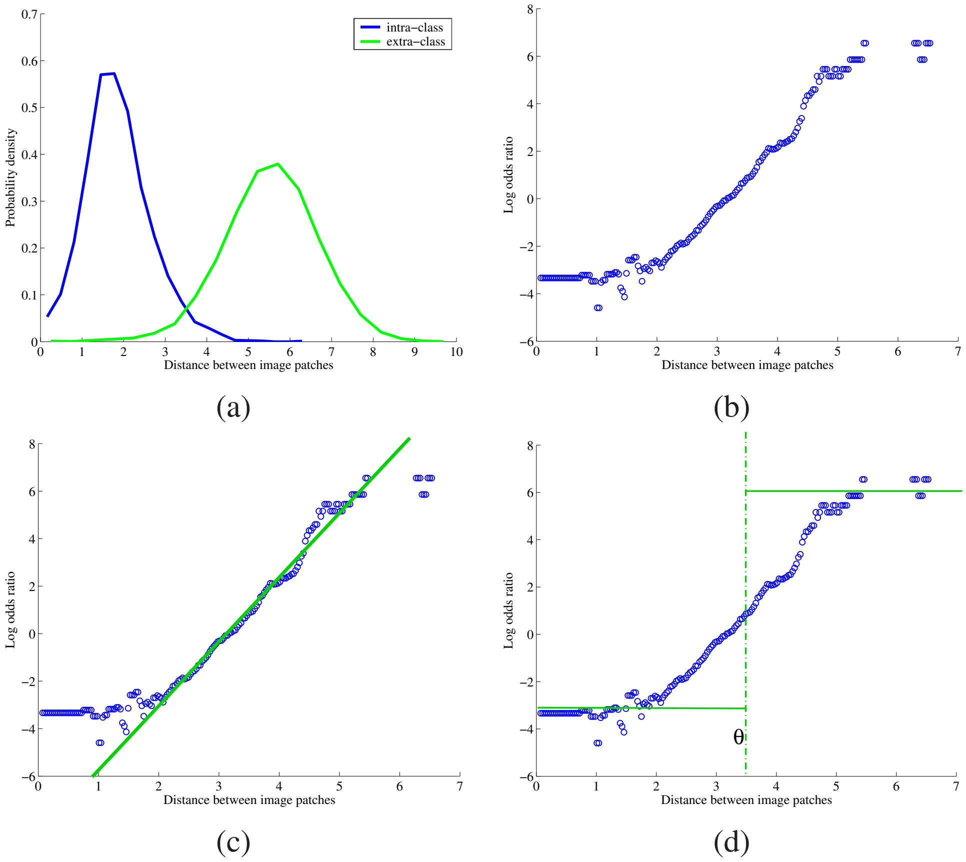


Figure 1.2: (a) Distribution of distances in a local shape histogram feature space between images of object parts from a collection of 15 objects described in Chapter 7 and randomly sampled image patches of background clutter. The distance scores fall into two distributions labeled "intra-class" and "extra-class". The distance score can be split roughly into three intervals along the x-axis: the middle interval is where uncertainty is greatest as to which distribution the distance score comes from. (b) plot of the log odds ratio (1.1). Note that the plot is quite linear in the middle uncertain interval. (c) a linear model fits the middle interval quite well. (d) a discretization of the distance measure that is induced by a simple discriminator that uses a threshold θ on the distance score.

elementary distance measures can be interpreted as indicating the discriminative power of each elementary distance measure. See Chapter 4.

The need for a hierarchical distance measure. We have thus far described a *continuous linear model* for the optimal distance measure. Although we find that in practice this continuous model is accurate at retrieving the nearest neighbor, it is expensive to use at run-time when searching over a large training set. Any kind of efficient nearest neighbor search implicitly requires a *discrete* distance measure. Consequently we will investigate the construction of discrete distance measures that are appropriate for efficiently performing the nearest neighbor search for our discrimination tasks.

Although we can show that, in theory, the optimal distance measure can be replaced by using only a discrete distance measure without sacrificing the classification performance (see Chapter 3), in practice we find that discrete distance measures are only useful for *coarse* discrimination among object classes. Thus in practice, discrete distance measures are most useful for reporting a small set of candidate neighbors, one of which is likely to be the optimal nearest neighbor. On the other hand, we show how the nearest neighbor search can be implemented efficiently by using a discrete distance measure that combines elementary discrete distance measures associated with *discriminators* in a tree-like structure, where each of the discriminators is constructed in simple feature spaces like color, texture or local shape properties. The elementary discrete distance measures will turn out to be discretizations of the same elementary distance measures over simple feature spaces used in the continuous linear model, and where the discretization is induced by discriminators. Returning to our one-dimensional example feature space in Figure 1.2, we can construct a simple discriminator that thresholds the distance between a pair of images. The optimal threshold will be such that image pairs with distance scores that fall below the threshold most likely belong to the same class, otherwise they most likely belong to different classes. The corresponding discretized distance measure associated with such a discriminator is shown in Figure 1.2(d). Again, just as in the case for continuous distance measures, we will consider a linear model for combining the discrete distance measures.

Compared with the discrete model, the continuous model is more expensive to use at run-time for searching over a large training set but also more accurate as noted above. Thus both distance models are problematic to use in an efficient as well as accurate nearest neighbor search for different reasons, when each is used in isolation. Instead, *our strategy will be to combine the complementary aspects of the two models to create a distance measure that is both accurate and efficient to compute at run-time*. The basic idea will be to first use the discrete model to efficiently search for a small list of candidate neighbors, which is then further pruned using the finer discriminative power of the continuous distance measure (see § 3.3).

How do we estimate the distance measure from training data ? A linear combination model for the distance measure, either discrete or continuous, implies an *exponential* family for the pair-wise discriminative distribution $p(y \neq y' \mid x, x')$ in (1.1). Thus we seek to estimate the optimal model for the distribution from the family of exponential models given the training data. We use the maximum likelihood framework (see Chapter 4) for estimating the parameters of the optimal exponential model.

1.2 Sketch of our Detection Scheme

We have thus far discussed only the issue of utilizing an optimal distance measure for nearest neighbor search for object detection. In practice, there are several other issues that need to be addressed when using a nearest neighbor search framework in the context of an overall scheme for object detection. Since the main focus of this thesis is on developing an optimal distance measure for object detection, for the rest of the object detection system, we will seek the simplest implementation that we can get away with, but yet which is sufficient and realistic enough for evaluating the distance measures that we develop.

Figure 1.3 outlines our overall scheme for object detection. In general, we might use attentional mechanisms or interest operators (Grimson et al., 1994; Burt, 1988; Abbott and Zheng, 1995; Westlius et al., 1996; Grove and Fisher, 1996; Stough and Brodley, 2001; Culhane and Tsotsos, 1992; Itti et al., 1998; Baluja and Pomerleau, 1997; Tomasi and Shi, 1994; Ruzon and Tomasi, 1999; Mikolajczyk and Schmid, 2002) to focus on only the locations in the input image that are likely to correspond to an object of interest. However, such techniques are beyond the scope of this thesis whose main focus is on using the nearest neighbor framework for object detection. Instead, we use a simple strategy where we sub-sample locations in the image at various positions and scales and classify the sub-image at each location. Such a "brute" force approach has been used in the literature with reasonable run-time performance (Rowley et al., 1998; Schneiderman, 2000; Viola and Jones, 2001). Clearly, any attentional mechanism will be complementary to such a naive approach and can only improve run-time performance.

In practice, the objects that we are interested in detecting can be of varying sizes and shapes. The naive approach of performing a nearest neighbor search at each location over a training set with whole object views will result in poor performance since no single choice for the size and shape of the support window to be used when performing the nearest neighbor search can be expected to be optimal for all objects. A single choice for the support window will typically be either too small for some objects, in which case some discriminative information will likely be lost, or will be too large in which case the object can be confounded with background clutter.

The solution that we pursue is to find a decomposition of object training images in terms of parts, each of which has a support window with the same size and shape. The nearest neighbor search is then performed over parts rather than whole object views. A decomposition into parts is also useful for robustness against partial occlusion which is expected to affect only some but not all of the parts. Since different parts will in general have different discriminative powers, and we would like to use as few parts as possible for run-time efficiency, an important issue that we need to deal with is that of finding a good decomposition of training views into a few parts. See Chapter 6 for details.

Our detection scheme is composed of the following steps (detailed in Chapter 6):

- An input image is first pre-processed to extract the various histograms (color, shape, texture) at each location.

- The sub-image at every location is labeled by the nearest neighbor part classifier with a few number of parts from the training data that are nearest to

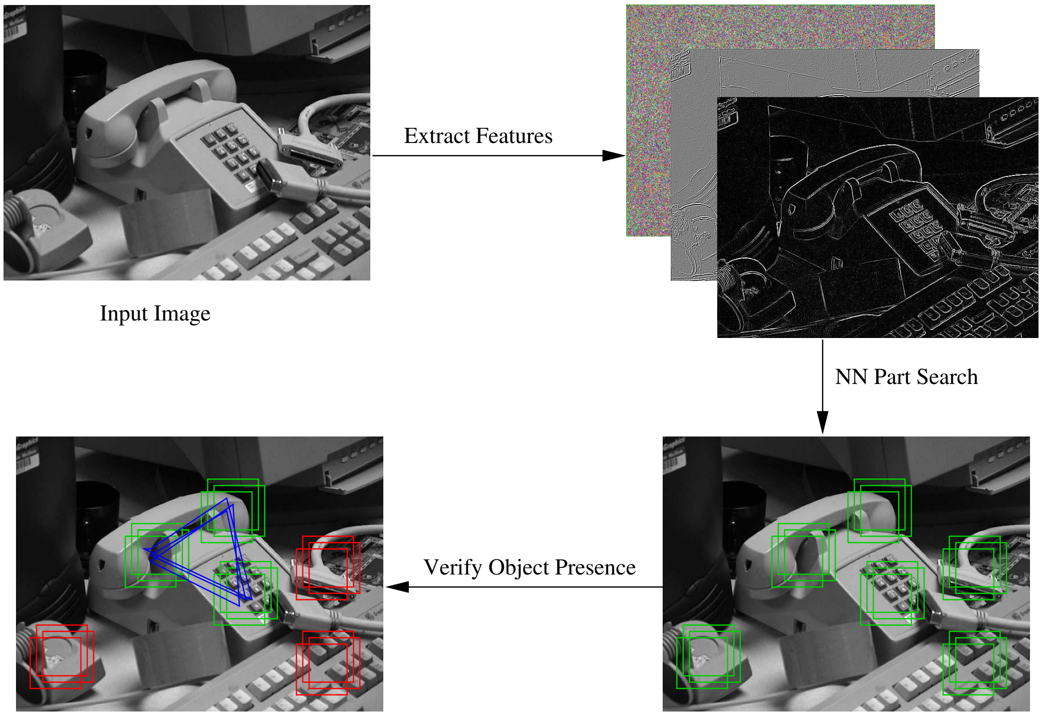


Figure 1.3: Outline of our approach. The input image is pre-processed to extract features at various locations sub-sampled across the image. In our work, we extract color, local shape and texture histograms. Next the nearest neighbor part classifier is run at each location. As outlined in Figure 3.3, the NN search first uses the efficient but coarse discretized distance measure to return a small list of candidate neighbors for each location. This list is then pruned by the more accurate continuous distance measure. Note that in the illustration, only a few parts detected are shown. Also note that neighboring locations can give multiple part detections that overlap. Each part is used to generate a hypothesis for an object of interest at that location. The locations of the other parts (shown by the triangles) in the hypothesized object class is searched for the corresponding part expected at those locations. Possible occlusions of parts are handled by rejecting outliers. The scores for all such non-outlier parts are accumulated and thresholded to give an object detection. In the illustration, part hypotheses that could not be verified are shown in red.

the sub-image. The distance measure used is the hierarchical distance measure discussed above. Part labels corresponding to clutter training samples are ignored in subsequent processing.

- Each part label at a location is used to generate a hypothesis for the presence of an object viewed under conditions closest to a training image containing the part. A "score" for the hypothesis is computed by first predicting the locations of all other parts belonging to the same training image and accumulating the scores (the nearest neighbor similarity) of all the parts.

- Finally, the various object hypotheses at each location are pruned by thresholding their scores, after which local non-maximal suppression is performed resulting in non-overlapping hypotheses. The operating characteristic (characterized by the false positive and detection rates) of the whole detection scheme is determined by the threshold used for the pruning. Thus the final output consists of one or more non-overlapping locations in the image labeled with an object of interest.

1.3 Outline of the Thesis

We conclude this chapter with an outline of the rest of the thesis.

Chapter 2 discusses in detail the nearest neighbor framework. We first derive the optimal nearest neighbor distance measure that maximizes the classification performance, in terms of the probability distribution that a pair of images belong to the same class. We then show how precisely the optimal distance measure is different from the more familiar metric distance measures that are commonly used in the literature. We also compare the classification performance of the optimal distance measure with the Bayes optimal risk as well as the best performance possible for any metric distance. Finally, we survey related work in the literature on finding optimal distance measures for nearest neighbor search.

Chapter 3 discusses how we model and estimate the optimal distance measure in practice. We first argue for the advantages of directly modeling the pair-wise distribution rather than the alternative approach of first estimating a generative model for each class and then deriving the pair-wise distribution. We then consider a linear model for the optimal distance measure that combines elementary distance measures acting on simple feature spaces. Discrete and continuous linear models are then considered in detail as well as their use in a hierarchical distance measure that is both efficient and accurate.

A linear model for the distance measure implies an exponential model for the pair-wise distribution, the estimation of which is considered under the maximum likelihood framework in **Chapter 4**. We then note the relationship with the maximum entropy framework that gives us an alternative view of our approach. We re-examine a natural selection scheme under the maximum entropy framework that has been proposed in the literature in a different context (Zhu et al., 1998) and show that, although they look very different, the maximum entropy selection procedure is the same as the selection procedure under the maximum likelihood framework. We also discuss similarities between our work and the boosting framework.

In **Chapter 5**, we discuss the construction of candidate discriminators required for the maximum likelihood selection scheme presented in **Chapter 4** for discrete distance measures. We first present a very general approach for constructing discriminators that is simple to implement and applicable to any feature space equipped with an arbitrary distance measure : the nearest prototype discriminator. To generate such discriminators efficiently, we develop a simple sampling strategy with provable performance guarantees. For linear feature spaces (for example, normalized pixel intensities), we propose another approach for generating good discriminators that can be posed as optimizing an objective function encoding various criteria for good discrimination. The optimization can be performed by iteratively solving two associated eigenproblems.

Chapter 6 deals with the training phase for the nearest neighbor classifier. We first discuss the choice of feature types that will be used. We discuss the efficient construction of histograms of various feature types (color, contour, texture). Next, we discuss the decomposition of each training image into a few spatially non-overlapping discriminative parts. The chapter also discusses how discriminators that are used to form the discrete distance measure can be organized in a tree-like structure for run-time efficiency. The chapter concludes by describing in detail the complete object detection system that we have implemented to test our approach.

Chapter 7 reports results on two detection tasks: an indoor task and a face recognition task. The chapter includes a detailed empirical analysis of the various parameters and issues that affect classification performance when using the proposed hierarchical distance measure.

Chapter 8 concludes with a discussion on possible directions for future research.

Chapter 2

Optimal NN Distance Measure

In this chapter we present our approach for finding good distance measures that maximize the classification performance or equivalently minimize the mis-classification risk for the nearest neighbor search. The optimal distance measure that minimizes the risk is the pair-wise distribution that indicates how likely two images come from the same or different object classes. In general, this distance measure is not a metric distance which is the most popular distance measure assumed in the literature. We will investigate precisely where and how the metric axioms are violated. Next, we will study the nearest neighbor classification performance under the optimal distance measure and compare it with the performance of metric distances as well as the Bayes optimal classification performance. We conclude the chapter by surveying prior work on optimal distance measures for nearest neighbor search.

2.1 The Setting

We assume that we have a training set $S_n = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ of size n where each tuple (x_i, y_i) is chosen i.i.d. from some unknown distribution over $X \times Y$ where X is the space of all image measurements and Y is some discrete finite set of class labels. A measurement is the representation of the image in terms of a set of features like color, shape or texture. We are also given a distance measure $d : X \times X \rightarrow \mathbb{R}$ between any two image measurements. The distance measure is assumed to be symmetric and has the following *qualitative* interpretation: for three images x, x', x'', if $d(x, x') < d(x, x'')$, then x' is considered to be "closer" to x than x''. In a nearest neighbor search, only such relative values of the distance measure are of interest and thus we do not impose any restrictions on a distance measure other than symmetry. In particular, we do not assume a metric distance (for definition see discussion below), in contrast with most work in the literature (Dasarathy, 1991).

When given a new input image $x \in X$, the 1-*nearest neighbor rule* reports the class label y' associated with the training image $x' \in S_n$ that is closest to x according to the distance measure d. Let $L(y, y')$ be some loss function that gives the loss incurred by the NN rule if y is the true class label of x rather than y'. Let $p(x, y)$ be the joint distribution over image measurements and class labels. Given that $x' \in S_n$ is the nearest neighbor to input x, we can then define the *conditional risk* $r(x, x')$ to be the conditional expectation over the loss function L as follows:

$$\begin{aligned} r(x, x') &\equiv \mathrm{E}_{y,y'} L(y, y') \\ &= \sum_{y,y'} L(y, y') p(y, y' \mid x, x') \\ &= \sum_{y,y'} L(y, y') p(y \mid x) p(y' \mid x') \end{aligned} \tag{2.1}$$

where the last equation follows from the i.i.d. assumption.

The n-sample NN risk $R(n)$ is defined as:

$$R(n) \equiv \mathrm{E}_{(x,y),S_n}[L(y, y')] \tag{2.2}$$

where the expectation is taken over all inputs x *as well as* all training sets S_n of size n. Note that x' is the nearest neighbor of x in S_n and therefore x' is a function of x, however x' does vary with S_n. Due to the i.i.d. assumption, given a nearest neighbor x', the corresponding class label y' is dependent only on x'. Thus we can express the NN risk in terms of the conditional risk $r(x, x')$ as follows:

$$R(n) = \mathrm{E}_{x,X_n}[r(x, x')] \tag{2.3}$$

where X_n is the set of only the training measurements x_i from S_n, excluding the corresponding class labels y_i. The large sample or asymptotic risk is defined as:

$$R \equiv \lim_{n \to \infty} R(n) \tag{2.4}$$

2.2 Optimal 1-NN Distance Measure

Consider a 0-1 loss function given by $L(y, y') = 1$ if $y \neq y'$ and $L(y, y') = 0$ otherwise. Then the conditional risk $r(x, x')$ measures the probability of mis-classifying x if x' is assigned as its nearest neighbor, while the risk $R(n)$ measures the average mis-classification error of the NN rule for a training set of size n. It can be verified that for the 0-1 loss, the conditional risk (2.1) reduces to the following:

$$\begin{aligned} r(x, x') &= \sum_{y \neq y'} p(y \mid x) p(y' \mid x') \\ &= p(y \neq y' | x, x') \end{aligned} \tag{2.5}$$

The risk $r(x, x') = p(y \neq y'|x, x')$ defined on any two measurements x and x' can be thought of as a "discriminative" measure between the two image measurements, since it indicates the probability that the two measurements comes from the same object class or not.

For a given training set size of n, the risk $R(n)$ depends only on the distance measure d used for the nearest neighbor search. Thus, it is natural to ask for the distance measure that minimizes the risk. The discriminative distribution $p(y \neq y'|x, x')$ can itself be thought of as a distance measure for which two images are "closer" to each other if they are both likely to come from the same class rather than from different classes. We can in fact easily show that this discriminative distribution when considered as a distance measure minimizes the NN risk.

For a given input x and training set S_n, using $d \equiv p(\cdot \mid \cdot, \cdot)$ as the distance measure gives the training example x' that minimizes the conditional risk $r(x, x') = p(y \neq y'|x, x')$ over the training set S_n since by construction the distance measure used is also the conditional risk and thus finding $x' \in S_n$ that minimizes the distance measure also minimizes the conditional risk. Since the conditional risk $r(x, x')$ is minimized for any input x by the chosen distance measure, the unconditional risk $R(n)$ is also minimized. We have thus shown the following:

Theorem 1 *The distance measure $d(x, x') \equiv p(y \neq y'|x, x')$ minimizes the risk $R(n)$ for any n.*

Note that the above result remains true even if we transform the discriminative distribution by any monotonically strictly increasing function f. This is true because using $d \equiv f(p(\cdot|\cdot,\cdot))$ as the distance measure returns the same nearest neighbor as when using $d \equiv p(\cdot|\cdot,\cdot)$. We will use this fact later when modeling the optimal distance measure (see § 3.2).

2.2.1 The Pair-Wise Distribution is not a Metric Distance

Most previous work in the literature (Dasarathy, 1991) was interested in finding an optimal metric distance. A distance measure $d(x, x')$ is a metric distance if it satisfies positivity: $d(x, x') \geq 0$ with equality iff $x = x'$, symmetry: $d(x, x') = d(x', x)$ and the triangle inequality: $d(x, x') + d(x', x'') \geq d(x, x'')$. In general, there is no reason to expect that the pair-wise distribution is a metric distance. Nevertheless, it is instructive to see which of the conditions above are not satisfied by the pair-wise likelihood when considered as a distance measure. Typically, it is assumed that the most common reason that a distance measure is non-metric is because it violates the triangle inequality. Surprisingly, this is not the case for the pair-wise distribution.

As before, let $p(x, y)$ be the distribution over $X \times Y$ under which measurements x and corresponding class labels y are drawn i.i.d. An expression for the pair-wise distribution that is equivalent to the one in (2.5) but is more convenient for the present discussion is given by:

$$p(y \neq y' \mid x, x') = \sum_y p(y|x)(1 - p(y|x')) \tag{2.6}$$

Positivity. It can easily be shown that positivity is not satisfied by the pair-wise distribution in general. As a simple counter-example, let $p(y|x) = 1/|Y|$ be uniformly distributed over all class labels for all $x \in X$ ($|Y|$ is the number of classes). Then $p(y \neq y'|x, x') = 1 - 1/|Y| > 0$ even when $x = x'$.

More generally, when $x = x'$, $p(y \neq y'|x, x') = 0$ iff x belongs to one of the classes with complete certainty, i.e. $p(y|x) = 1$ for some y and $p(y'|x) = 0$ for $y' \neq y$. The if part is immediate from the r.h.s. of (2.6). For the converse, we have $\sum_y p(y|x)(1 - p(y|x)) = 0$, from which $p(y|x)(1 - p(y|x)) = 0$ for each y since each term in the sum is non-negative. Thus either $p(y|x) = 0$ or $p(y|x) = 1$ for each y. Finally, since $\sum_y p(y|x) = 1$, we have the desired result. In other words, lack of positivity for any measurement x is due to lack of complete certainty about its class membership which will be the case in most real tasks.

Lack of positivity leads to the the most important difference between the pair-wise distribution and any metric distance measure: the nearest neighbor of a given measurement x over the whole space X under the pair-wise distribution distance measure need *not* be x itself. This property will turn out to be the reason why the optimal distance measure out-performs any metric distance measure in general, as discussed in the next subsection.

Symmetry. Next, symmetry is satisfied since the order of the two measurements x and x' in the pair-wise distribution is immaterial.

Triangle Inequality. Lastly, it might seem that the triangle inequality will not be satisfied by the pair-wise distribution distance measure in general for an arbitrary distribution $p(x, y)$. Surprisingly, this is not the case as we show next.

Since p is a probability measure and thus takes values in $[0, 1]$, $p(y|x) \geq p(y|x)(1 - p(y|x''))$ as well as $(1 - p(y|x'')) \geq p(y|x)(1 - p(y|x''))$. Using these two relations,

$$\begin{aligned} & p(y|x)(1 - p(y|x')) + p(y|x')(1 - p(y|x'')) \\ \geq \; & p(y|x)(1 - p(y|x''))(1 - p(y|x')) + p(y|x')p(y|x)(1 - p(y|x'')) \\ = \; & p(y|x)(1 - p(y|x'')) \end{aligned}$$

Summing over y on both sides and using (2.6) yields the desired triangle inequality for the pair-wise likelihood.

Symmetry and the triangle inequality implies that if x' is close to both x and x'', then x and x'' should also be close to each other. This property is useful for some applications like efficient image retrieval (Berman and Shapiro, 1997; Barros et al., 1996).

Finally, we note that Jacobs et al. (2000) have investigated the properties of robust distance measures used in computer vision. They show that most robust distance measures do not satisfy the metric axioms - in particular the triangle inequality. However they were not concerned with the issue of whether the distance measure used is optimal as is the case in our work.

2.2.2 Classification Performance Comparison

As mentioned before, most of the work in the literature has assumed a metric distance. An important question is if the pair-wise distribution distance measure can outperform any metric distance in the limit as the size of the training set grows to infinity. It was shown in (Cover and Hart, 1967) that the asymptotic risk for any *metric* distance is at most twice the Bayes optimal risk. Given an input x, the Bayes optimal decision assigns x to the class y that maximizes the posterior $p(y|x)$. Of course, in general the posterior distribution is not known in practice, but the Bayes risk indicates the optimal performance that any classifier can hope to achieve. Denoting the Bayes optimal risk by R^B, (Cover and Hart, 1967) showed the following when the distance measure used is any metric:

$$R^B \leq R^M \leq 2R^B$$

where R^M is the asymptotic risk defined in (2.4) for the nearest neighbor rule using *any* metric distance. Since the class of metric distance measures is a subset of the class of all distance measures, and since the pair-wise distribution distance measure $p(y \neq y' \mid x, x')$ minimizes the risk over all distance measures (see theorem 1 where no restrictions on the distance measures were made), we conclude that no metric distance can outperform the pair-wise distribution distance measure. On the other hand we give an example where the pair-wise distribution distance measure outperforms any metric, in fact it achieves the Bayes optimal risk for the example.

Example. We use the same example presented in (Cover and Hart, 1967) for which the NN asymptotic risk as well as the Bayes optimal risk can be easily determined. The measurements x are real-valued and come from two classes y_1 and y_2 with triangular densities $p(x|y_1) = 2x$, $p(x|y_2) = 2(1-x)$ respectively with priors $p(y_1) = p(y_2) = 1/2$. For these densities and priors, the density on x ($p(x)$) is uniform on $[0, 1]$. See Figure 2.1.

The pair-wise distribution for two measurements x, x' is then given by:

$$p(y \neq y' \mid x, x') = x(1-x') + (1-x)x' \tag{2.7}$$

Let S_n be a training set of size n. For two measurements, x_1 and x_2 from the training set, the conditions under which another measurement x is closer to x_1 than it is to x_2 when using the pair-wise distribution as the distance measure is given by:

$$\begin{aligned} x(1-x_1) + (1-x)x_1 &< x(1-x_2) + (1-x)x_2 \\ \implies (x_1 - x_2)(1-2x) &< 0 \end{aligned}$$

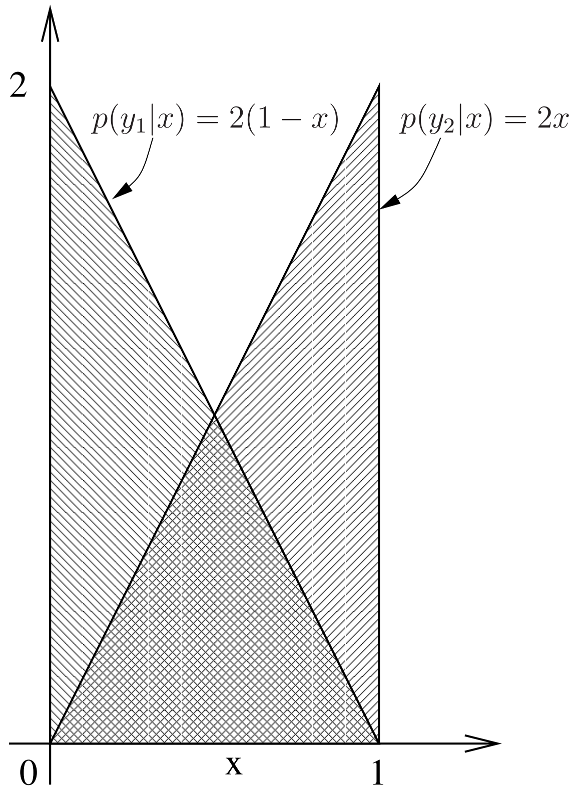


Figure 2.1: A two class example from (Cover and Hart, 1967), that we reuse for illustrating the classification performance of nearest neighbor performance under various distance measures. Note that at $x = 0$ and 1, the class membership is completely certain.

Thus when $0 \leq x < 1/2$, the smaller of x_1 and x_2 is closer to x, whereas for $1/2 < x \leq 1$, the greater of x_1 and x_2 is closer to x. Thus given a training set of size n with measurements $X_n = \{x_1, \ldots, x_n\}$, the nearest neighbor x'_n in X_n for a query point x is given by the rule:

$$x'_n = \begin{cases} \min\limits_i x_i, x_i \in X_n & \text{if } x < 1/2 \\ \max\limits_i x_i, x_i \in X_n & \text{if } x \geq 1/2 \end{cases}$$

It can be seen that since the density for x is uniform on $[0, 1]$, in the limit as $n \to \infty$ $\min\limits_i x_i, x_i \in X_n$ converges to 0 with probability one. Similarly, $\max\limits_i x_i, x_i \in X_n$ converges to 1 with probability one. This example illustrates a claim we made in the previous subsection, namely the lack of positivity for the pair-wise distribution implies that in general the nearest neighbor for a measurement x over the whole space X need not be x itself. In this example, only 0 and 1 are their own nearest neighbors. Note that 0 and 1 have complete certainty as to their class membership. More generally, for a discrimination task with more than two classes, the nearest neighbor of an input measurement will be the measurement from the training set whose class membership is most certain.

With these limits, we have from (2.7):

$$p(y \neq y' \mid x, x') = \begin{cases} x & \text{if } x < 1/2 \\ 1 - x & \text{if } x \geq 1/2 \end{cases} = \min\{x, 1 - x\}$$

The expression on the right hand side above can also be shown to be the conditional Bayes risk for a given input x. The Bayes decision assigns x to the class that minimizes the mis-classification probability. In other words, it assigns x to the class y that minimizes $1 - p(y|x)$. It can be verified that the Bayes risk incurred is indeed the right hand side of the equation above.

The total risk R using the pair-wise distribution as distance measure (or equivalently the Bayes risk for this example) is given by:

$$R^B = R = \int_0^1 \min\{x, 1 - x\} dx = \frac{1}{4}$$

On the other hand, using any metric as a distance measure, the nearest neighbor x' can be shown to converge to x as $n \to \infty$ under quite general conditions (Cover and Hart, 1967). Thus from (2.7), which is also the expression for the risk incurred when x' is the nearest neighbor of x, the conditional risk incurred for a given input x when using any distance metric is $2x(1-x)$ in the limit as $n \to \infty$. The total asymptotic risk R^M for any metric is then given by:

$$R^M = \int_0^1 2x(1-x)dx = \frac{1}{3}$$

Thus $R^B = R < R^M$ for this example.

In summary, for this example the pair-wise distribution distance measure outperforms any metric distance measure in the large sample limit and furthermore attains the least possible risk that can be achieved by any classification procedure, namely the Bayes optimal risk R^B. The reason why the pair-wise distribution distance was able to outperform any metric distance measure was precisely because of its lack of positivity. In the example, there was at least one measurement (0 and 1 in this example) for which there was complete certainty as to which class it belongs to and the nearest neighbor under the optimal distance measure approaches one of these two measurements in the large sample limit. As expressed in (2.6), the mis-classification risk can be seen to be proportional to the class label uncertainty of the nearest neighbor x' as well as the uncertainty of the query x. Since the query is given, the only strategy for reducing the risk is to choose the the nearest neighbor with the least class label uncertainty, which is precisely what the optimal distance measure does. Any metric distance measure on the other hand returns a nearest neighbor x' that approaches the query x in the large sample limit, whose class label uncertainty is thus given and cannot be reduced.

In general, there need not be any measurement with complete class label certainty for a given task. Thus the asymptotic risk attained by using the optimal distance measure can be anywhere between the Bayes optimal risk and the risk obtained using a metric distance, i.e.:

$$R^B \leq R \leq R^M$$

In practice, we will have to estimate the optimal distance measure from training data. Before taking up this issue, we first survey previous work on finding good distance measures for the nearest neighbor rule.

2.3 Prior Work

Our survey can be considered to be quite comprehensive since little work has been done on finding good distance measures compared with other aspects of the nearest neighbor rule. Most work to date has focused on finding good *metric* distance measures. Typically, the metric distance assumed is euclidean for which a linear transform that optimizes some criterion is found. In our case, we cannot assume that the measurements are embedded in any metric space, especially since an image may be represented using measurements from different feature spaces (color, shape, texture) that cannot be combined using a common metric distance measure.

Short and Fukunaga (1981) find a metric distance measure that reduces the discrepancy between the finite sample NN risk (2.2) and the asymptotic risk (2.4). The distance measure is approximated by a local metric that is estimated from the training data for every query point. Since estimating a local metric anew for every query point is expensive, in subsequent work (Fukanaga and Flick, 1984) the authors presented a globally optimal quadratic metric that minimizes the same error criteria above.

Hastie and Tibshirani (1996) also find a local metric for a given query point. Their approach draws inspiration from the traditional work on linear discriminant analysis (LDA) but applied locally. The local metric is derived from local estimates of the within class and between class scatter matrices just as for LDA. The local metric emphasizes between class variations while suppressing within class variations.

Friedman (1994) estimates the relevance of each component of the measurement or linear combinations of the components for any given classification task. The relevance is estimated locally for each query point using a tree-structured recursive partitioning technique. The relevance of a component is proportional to how useful the component is for discriminating classes. Essentially, the method finds a locally adapted metric for each query point.

Mel (1997) approaches the object detection task using the nearest neighbor framework just as we do. Object views are represented in terms of color, shape and texture histograms, which is the same basic representation that we will also use in our work (see Chapter 6 for details). The author determines a weighted

L_1 distance measure using the intuitive heuristic that the optimal metric should cluster object views from the same class while separating views from different classes. However, the metric is determined using an intuitive but ad-hoc objective function that encodes the heuristic above. The objective function is optimized for the optimal weights for the L_1 distance measure using gradient descent. The weighted L_1 metric found is global with no local adaptation to a query point.

Blanzieri and Ricci (1999) propose to use the same pair-wise distribution distance measure as we do. However, they justify using the pair-wise distribution as a simpler alternative compared with estimating the distance measure in Short and Fukunaga's (1981) work. The authors do not seem to have realized that the pair-wise distribution measure is in fact the optimal measure to use. Furthermore in their work, the pair-wise distribution distance measure is constructed by first estimating a generative model $p(x|y)$ for each class from the training data and then using (2.5) to express the pair-wise distribution distance measure in terms of the posteriors $p(y|x)$ (which can be obtained from the generative models $p(x|y)$ and the priors $p(y)$ using Bayes rule).

Lastly, we survey work done on the so-called Canonical Distance Measures (CDM) (Baxter and Bartlett, 1998; Minka, 2000). The motivation for this work is to find a distance measure for use in a nearest neighbor rule that minimizes the mis-classification risk over a *distribution of classification tasks* rather than just a single task. For example, the measurement space might be the height of a person, and two classification tasks might then be the gender and ethnicity of the person.

Similar in spirit to the argument we made for theorem (1), the optimal distance measure, called the CDM in (Baxter and Bartlett, 1998), that finds the nearest neighbor that gives the least mis-classification risk when using the nearest neighbor rule was shown to be the expected risk over all classification tasks:

$$d(x, x') = \mathrm{E}_f[L(f(x), f(x'))]$$

where each f gives the class label for an input measurement for a given task, and L is a loss function.

We are not wholly convinced of the need for a distance measure that is optimal over a distribution of classification tasks. Certainly at run-time, we will know which particular classification task that we need to tackle. Thus, at training time, if we had estimated the optimal distance measure for each classification task and use these individually tailored distance measures at run-time, the resulting average classification performance over all tasks will be better than the average classification performance of the CDM. Nevertheless, for a single task, the CDM framework is related to our work as follows.

In the original formulation, the classifiers f are assumed to be perfect, that is, they give the true class label for each input measurement. More recently, this requirement has been relaxed and generalized such that the classifiers can give a distribution over class labels for each input measurement.

If we assume that we have only one classification task, then under the 0-1 loss function, the above generalization to the CDM framework can be shown to give the pair-wise distribution (2.6)– which is the optimal distance measure in our work– as also the optimal distance measure in the CDM framework, see (Baxter and Bartlett, 1998; Minka, 2000). However, just as in (Blanzieri and Ricci, 1999) discussed above, this pair-wise distribution is still determined in (Minka, 2000) by first estimating a generative model $p(y|x)$ for each class.

We argue in the next chapter that if the generative models $p(y|x)$ can be estimated reliably, then we are better off using the Bayes optimal decision rule to assign an input measurement x to the class with the highest posterior $p(y|x)$. If the generative models are learned using an unbiased estimator, then asymptotically as the number of samples in the training set increases, we will achieve the Bayes optimal risk. Thus, there is no advantage in using a 1-NN decision rule. In fact, non-parametric decision rules like the nearest neighbor rule are used precisely when we cannot hope to reliably estimate generative models for each class. This is certainly the case for object detection tasks where it is not obvious what a good generative model would be for an arbitrary object class, much less obvious whether we will be able to reliably estimate the model from training data.

Chapter 3

Modeling the Optimal Distance Measure

Unlike previous approaches, we will directly model and estimate the pair-wise distribution from training data, using a simple additive logistic model. The logistic model linearly combines elementary distance measures, each of which is defined over simple feature spaces like color, texture and local shape properties. Two types of distance models are investigated: discrete and continuous models. Discrete distance models combine discretized elementary distance measures that are associated with discriminators constructed in simple feature spaces. Even though we show the somewhat surprising result that there exists discrete distance measures that give the same performance as the optimal distance measure, in practice the linear discrete model will only be good enough for performing coarse discrimination. On the other hand, they also permit an implementation that leads to efficient neareast neighbor search. In comparison, continuous distance model are typically more accurate in practice but more expensive when used for searching over a large training set. Thus the two models complement each other. We use this fact to develop a hierarchical distance measure which combines the two models to yield a nearest neighbor search that is both efficient as well as accurate.

3.1 Our Approach

As noted at the end of the last chapter, one approach to estimating the pair-wise distribution $p(y \neq y'|x, x')$ is to first estimate a generative model $p(x|y)$ for each class and then use (2.5). Instead, in our approach we directly estimate the pair-wise distribution $p(y \neq y'|x, x')$ from training data. We will argue that this direct approach is more appropriate and stable for the object detection task than the indirect approach where the generative models $p(x|y)$ are first estimated.

Specifying a generative model $p(x|y)$ might require many more parameters than is required for specifying the pair-wise distribution distance measure that we are ultimately after. The classic example is the two class case $y \in \{+1, -1\}$, where the generative model for each class is assumed to be Gaussian $p(x|y) = \mathcal{N}(x; \nu_y, \Sigma)$ parametrized by a mean ν_y for each class and a covariance matrix Σ that is the same for both classes. Suppose the measurements x lie in an n dimensional vector space, then we require $O(n^2)$ parameters to specify the mean and covariance. However, it can be shown that only $O(n)$ parameters is sufficient to specify the pair-wise distribution distance measure. For two classes, the pair-wise distribution distance measure is given by:

$$p(y \neq y'|x, x') = p(y = +1|x)p(y' = -1|x') + p(y = -1|x)p(y' = +1|x') \tag{3.1}$$

again under the i.i.d assumption. The posteriors $p(y|x)$ are expressed in terms of the generative models as follows:

$$\begin{aligned}
p(y = +1 \mid x) &= \frac{p(x \mid y = +1)p(y = +1)}{p(x)} \\
&= \frac{1}{1 + a\exp(-l^T x + b)} \\
l &= \Sigma^{-1}(\nu_{+1} - \nu_{-1}) \\
b &= \nu_{-1}^T \Sigma^{-1} \nu_{-1} - \nu_{+1}^T \Sigma^{-1} \nu_{+1} \\
a &= \frac{p(y = -1)}{p(y = +1)} \\
p(y = -1 \mid x) &= 1 - p(y = +1 \mid x)
\end{aligned}$$

In the above, the hyper-plane l, known as the Fisher discriminant (Bishop, 1995;

Duda et al., 2001), and thus also the pair-wise distribution distance measure, needs only $O(n)$ parameters to specify.

In general, given a limited amount of training data, the estimation of model parameters from the data is more well-conditioned, the fewer the parameters in the model (Bishop, 1995). For a more in-depth argument for directly estimating parameters for a discriminative task rather than first estimating generative models as an intermediate step, see (Vapnik, 1999). Below we corroborate this claim with a simple synthetic experiment.

We consider two classes with equal prior, each of which have Gaussian distributions with the same unit covariance defined over a vector space. The dimension n of the the vector space was varied from 5 to 100 in steps of 5 in the experiments below. In each case, the means of the two Gaussians were separated by two units. A trial experiment consisted of a training set of 20 samples from each of the two classes and a testing set of 500 samples. The results reported below were averaged over 20 such trials.

For each dimension n of the measurement space, the maximum likelihood estimates for the two means and the common covariance of the Gaussian distributions for the two classes were estimated. As mentioned before, this required the estimation of $O(n^2)$ parameters. The resulting estimated generative models for the two classes were used to classify the testing set using the Bayes decision rule. For comparison, we also estimated the maximum likelihood parameters for the optimal NN distance measure (3.1) directly from the training data. This required the estimation of only $O(n^2)$ parameters. The resulting estimated distance measure was then used to classify the testing set using the NN rule.

Figure 3.1 compares the performance for the generative versus the direct approach as the dimension of the measurement space is varied. As can be seen, both approaches perform quite a bit worse than the ground truth performance due to the very limited number of training examples. However, as the dimension increases the direct approach quickly outperforms the generative approach.

In the case of an object detection task, the above considerations are even more pertinent. Typically, for a general object detection task we can easily think of a few features that might be sufficient for discrimination while these same features may not be sufficient for specifying a generative model for any class of objects. For example, cars and humans may be sufficiently discriminated from each other

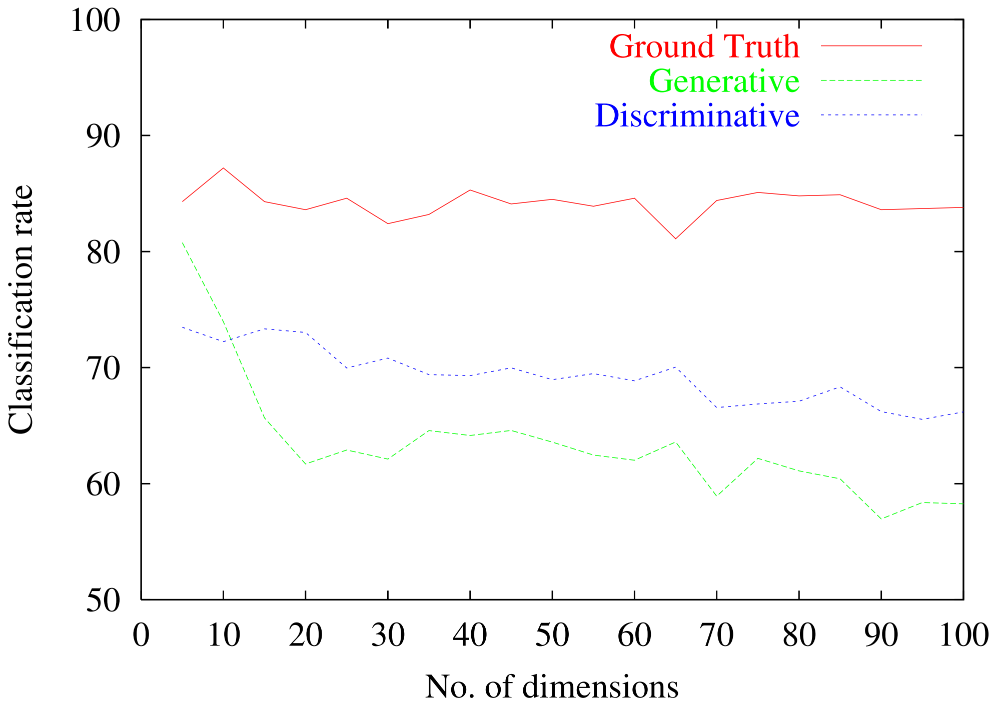


Figure 3.1: Comparison of the generative vs. discriminative approach in a synthetic experiment. See text for details.

by the presence or absence of wheels or legs. However, wheels and legs alone are not sufficient for specifying a generative model for cars and humans respectively. More generally, we have the further difficulty of being unable to easily come up with a generative model for an arbitrary object class in the first place. In the worst case, each new object of interest might require a different generative model. On the other hand, the same few features (say based on color, shape and texture) might be sufficient for discriminating all object classes of interest from each other.

3.2 Modeling the Optimal Distance Measure

Keeping in mind the arguments in the previous section, we now discuss our approach for directly modeling the pair-wise distribution $p(y \neq y'|x, x')$. A probability measure is constrained to lie between 0 and 1 on the real line. Instead of working with the pair-wise distribution directly, we will instead find it more convenient to work with a transform of the distribution that is unconstrained on the real line. Recall from theorem 1 that we can use any monotonically strictly increasing transform without changing the nearest neighbor returned.

The particular transform of the distance measure that we will use is the logit transform (Hastie and Tibshirani, 1990; McCullagh and Nelder, 1989):

$$H(x, x') \equiv \log \frac{p(y \neq y' \mid x, x')}{p(y = y' \mid x, x')}$$

As desired, the logit transform is unconstrained on the real line $-\infty < H(x, x') < \infty$, and will thus be easier to work with. Inverting the transform, the pair-wise distribution and its inverse can be expressed in terms of H as:

$$p(y \neq y' \mid x, x') = \frac{e^{H(x,x')}}{1 + e^{H(x,x')}} \tag{3.2}$$

$$p(y = y' \mid x, x') = \frac{1}{1 + e^{H(x,x')}} \tag{3.3}$$

We now discuss how we model the distance measure $H(x, x')$. Chapter 4 will discuss the estimation of the model from training data.

For a general object detection task with an arbitrary collection of objects of interest, there is no prior expectation that the optimal distance measure will assume any particular form. Different discrimination tasks may require different models. On the other hand, whatever model we consider should be feasible to implement in practice. One simple approach that we adopt in this thesis is to approximate the optimal distance measure by combining more “elementary” distance measures, each of which is defined over simple feature spaces like color, local shape or texture. One advantage of adopting such an approach is the ease with which such simple feature spaces can be implemented in practice, along with the variety of simple feature spaces that we can choose from. For example, we can consider simple histograms of features as we do in this thesis, for which one choice for the elementary distance measure is the χ^2 distance or we could use the simpler L_1 distance as we do in this thesis. Other simple feature spaces include edge maps with the Hausdorff distance measure (Huttenlocher et al., 1993), shape contexts (Belongie et al., 2002), or normalized pixel intensities with the simple euclidean distance measure (Nayar et al., 1996).

We seek to combine a set of simple feature spaces since no one feature space can be expected to be sufficient for an arbitrary discrimination task. The ideal set of feature spaces will complement each other for the discrimination task at hand. Given a set of feature spaces, we next turn to the issue of what is an appropriate model for combining the elementary distance measures associated with the feature spaces.

In general, the appropriate model will depend on the discrimination task at hand as well as the choice for the feature spaces in which images are represented. Thus we next motivate the appropriate model that we use by first looking at actual data for the discrimination task at hand.

In our thesis, we will use histograms of various features like color, local shape properties and texture as the simple feature spaces that we would like to combine in our model. Histograms were chosen since they can be efficiently computed from an input image and are stable representations with respect to a fair amount of distortions in viewing conditions. See Chapter 6 for details.

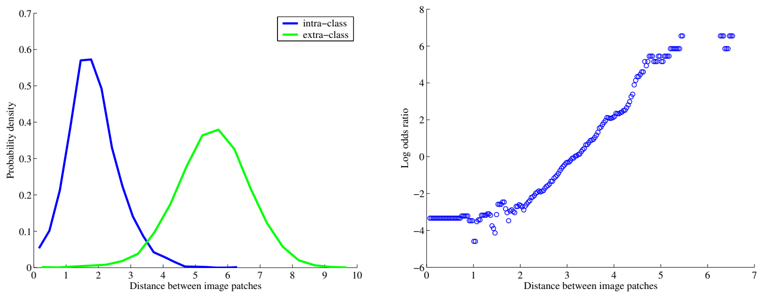
We wish to model the logit transform $H(x, x')$ (3.2) or the log odds ratio which is a function of pairs of images. To get an idea for what should be an appropriate model for combining elementary distance measures associated with simple feature spaces, we plot in Figure 3.2 the distribution of distance scores in such feature spaces between images of object parts from a collection of 15 objects of interest from an indoor detection task described in Chapter 7 and randomly sampled image patches from background clutter. The feature spaces considered are histograms of color, texture and local shape properties. The elementary distance measure used in these feature spaces is the L_1 distance.

As can be seen from the left column in the figure, the distance scores between images fall into one of two overlapping distributions that depend on whether the pair of images came from the same object part class or from different classes (including clutter). The distance score can be divided roughly into three intervals along the x-axis. The middle interval is where distance scores are hardest to classify as to whether they come from images belonging to the same object part class or to different classes.

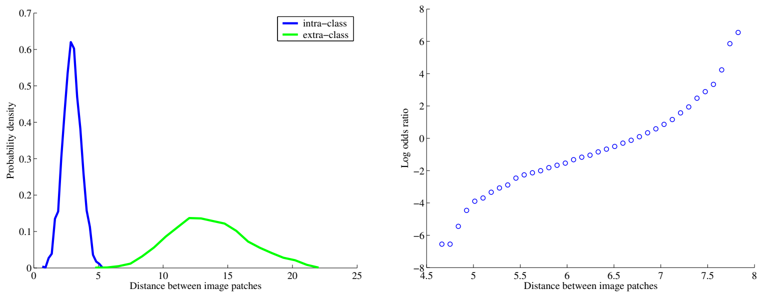
The right column of the figure plots the empirically determined log odds ratio $(H(x, x'))$. As can be seen from the plots, in the uncertain middle interval for each feature space, the log odds ratio is close to linear as a function of the distance score. Thus at least for this interval, we are justified in using a linear model. Modeling this region is what is most important for a discrimination task compared with modeling the other regions where one can be sure of the within-class or without-class membership of a distance score with high confidence. Thus a linear model can afford to fit these outer regions poorly compared with fitting the middle region. It remains to be shown however that the estimation procedure that we use for learning such a linear model from training data does in fact fit the middle region at the expense of the outer regions. See Chapter 4.

The above observations hold for each of the feature spaces that we use in our work. We can thus be justified in approximating the optimal distance measure with a *multi-dimensional model that linearly combines* the elementary distance measures in all of the feature spaces that is used. More formally, we are assuming an additive logistic model for the pair-wise distribution $p(y \neq y' \mid x, x')$. Before proceeding however, it should be emphasized that the observations that led to the consideration of a linear model in our case need not be valid more generally when different feature spaces and or their associated distance measures are used or when the discrimination task is different. The usefulness of such a linear model for arbitrary choices of discrimination tasks and or feature spaces remains to be seen.

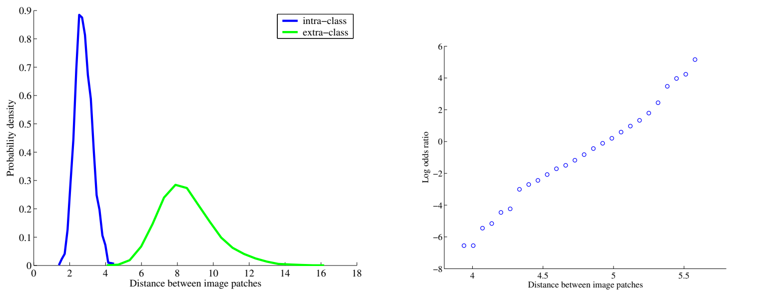
More generally, let $\mathcal{C}$ be a possibly large collection of elementary distance measures associated with simple feature spaces. We wish to select K elementary



Local Shape



Color



Texture

Figure 3.2: The left column plots the distribution of distances in various histogram feature spaces that we use in our work, between pairs of images of object parts from 15 objects described in Chapter 7 and randomly sampled image patches from background clutter. The distance scores fall into two distributions depending on whether the pair of images come from the same part class or not. The distance score can be split roughly into three intervals along the x-axis. The middle interval is where uncertainty is greatest as to which of the two distributions the distance score comes. The right column plots the log odds ratio (3.2). Note the linearity of the middle interval.

distance measures $d_k \in \mathcal{C}$ from the collection that best approximate the optimal distance measure using the following linear model:

$$H(x, x') \approx \tilde{H}(x, x') \equiv \alpha_0 + \sum_k^K \alpha_k d_k(x, x') \tag{3.4}$$

In practice, the choice for K will be based on run-time performance considerations.

For a given choice of K elementary distance measures from $\mathcal{C}$ the corresponding linear model for H implies a conditional exponential model for the pair-wise distribution. To see this more clearly, we can rewrite the expressions in (3.3) as follows:

$$\begin{aligned}
p(y \neq y' \mid x, x') &= \frac{e^{H(x,x')}}{1 + e^{H(x,x')}} && (3.5)\\
&= \frac{1}{Z(x, x')} e^{H(x,x')/2} && (3.6)\\
p(y = y' \mid x, x') &= \frac{1}{1 + e^{H(x,x')}} && (3.7)\\
&= \frac{1}{Z(x, x')} e^{-H(x,x')/2} && (3.8)\\
& && (3.9)
\end{aligned}$$

where $Z(x, x') = e^{-H(x,x')/2} + e^{H(x,x')/2}$ is a normalizing constant given a pair of images x and x'. Thus when H is approximated by a linear model $\tilde{H}$, we get a conditional exponential model since the exponent is linear in the parameters $\alpha_0, \ldots, \alpha_K$.

3.3 Discrete and Continuous Distance Models

We now consider the types of elementary distance measures that will be considered in our work. Examples of elementary distance measures include the simple Euclidean distance measure in a feature space for pixel intensities in an image, the χ^2 distance (Schiele, 1997; Press et al., 1992) between histograms of feature types like color, shape or texture, the Hausdorff distance (Huttenlocher et al., 1993) between edge maps, etc. All of the above elementary distance measures are continuous, the resulting model for the optimal distance measure is thus also continuous.

We will see in Chapter 7 that the use of the continuous distance model in a nearest neighbor search leads to good detection performance. However, continuous distance measures can only be used to search over a training set in a brute-force manner. Such a search is prohibitive for large training sets. Thus we seek alternative distance models that can be used for efficient NN search.

The basic idea behind most previous attempts (Beis and Lowe, 1997; P. Indyk, 1998) at efficient NN search is to (possibly recursively) *partition* the measurement space X. For example, in Kd-trees (Beis and Lowe, 1997), each node of the tree recursively partitions X based on the component of the measurement with maximum variance over the training set. However, Kd-trees are not appropriate in our case since the image measurement will be composed of measurements from different feature types like color, texture and shape. It does not make sense to compare variances of measurements from different feature spaces as required for the construction of Kd-trees.

In (P. Indyk, 1998), the space of measurements is partitioned by a collection of random hash functions. Our strategy is similar in spirit, but instead uses a collection of discriminators each of which is constructed in some simple feature space. Furthermore, the choice of discriminators is not random but is tuned to the particular discrimination task at hand. As we shall show later, a set of discriminators can be associated with a hamming distance measure. Thus a set of discrimintors induces a discrete distance model for the optimal distance measure. Such a discrete distance model can be used to implement an efficient nearest neighbor search by combining the associated discriminators in a tree-like structure as discussed below and in detail in Chapter 6.

In practice, the discrete distance model, though efficient, will not be as accurate as the continuous model. The continuous distance model on the other hand will be expensive to use for performing a nearest neighbor search when the training set size is large. We thus seek a distance measure that is both accurate and efficient to compute at run-time. Our strategy will be to combine the best of both models while overcoming the shortcomings of both at the same time as follows. We first use the discrete distance model for performing a coarse but efficient nearest neighbor search to return a small list of candidate neighbors for an input measurement, rather than just the nearest neighbor. This small list of candidate neighbors is then further pruned to find the nearest neighbor by using the more accurate but expensive to use continuous distance model. We will call this combined model the *hierarchical distance model*. See Figure 3.3.

3.3.1 Discrete Distance Model

In the rest of the chapter, we discuss in more detail the discrete linear distance model. We first make the somewhat surprising observation that there exists a discrete distance measure that gives the same classification performance as the optimal distance measure. However, the functional form of this discrete distance measure need not in general be linear. We then discuss a practical linear model that combines elementary discretized distance measures associated with discriminators, each of which act on simple feature spaces.

What is the best possible discrete distance measure that maximizes the classification performance for a given training set ? We can easily show that for a given training set, the optimal distance measure can be replaced by a discretized distance measure that has the *same* classification performance. For any distance measure H and training set S_n, the discrete distance measure — which we denote as H^d — that has the same classification performance as H can be constructed from H as follows. Given a distance measure H, the Voronoi diagram is a partition of the image space X such that the closest training measurement to each $x \in X_i$ under H is x_i. Let $X = X_1 \cup X_2 \cup \ldots \cup X_n, X_i \cap X_j = \emptyset, i \neq j$ be the Voronoi diagram induced in measurement space X by the distance measure $H(x, x')$ and the training measurements $\{x_1, x_2, \ldots, x_n\}$. We now define the discrete distance measure H^d that has the same classification performance as H by discretizing H as follows:

$$H^d(x, x') \equiv H(x_i, x_j), \quad \text{if } x \in X_i, x' \in X_j$$

In words, the discrete distance measure assigns to any given two measurements, the distance between the training measurements associated with the Voronoi partitions containing the given two measurements. Thus it can be verified that by construction, H^d assigns the same nearest neighbor from the training set to an input measurement as does the original distance measure H. Since the same nearest

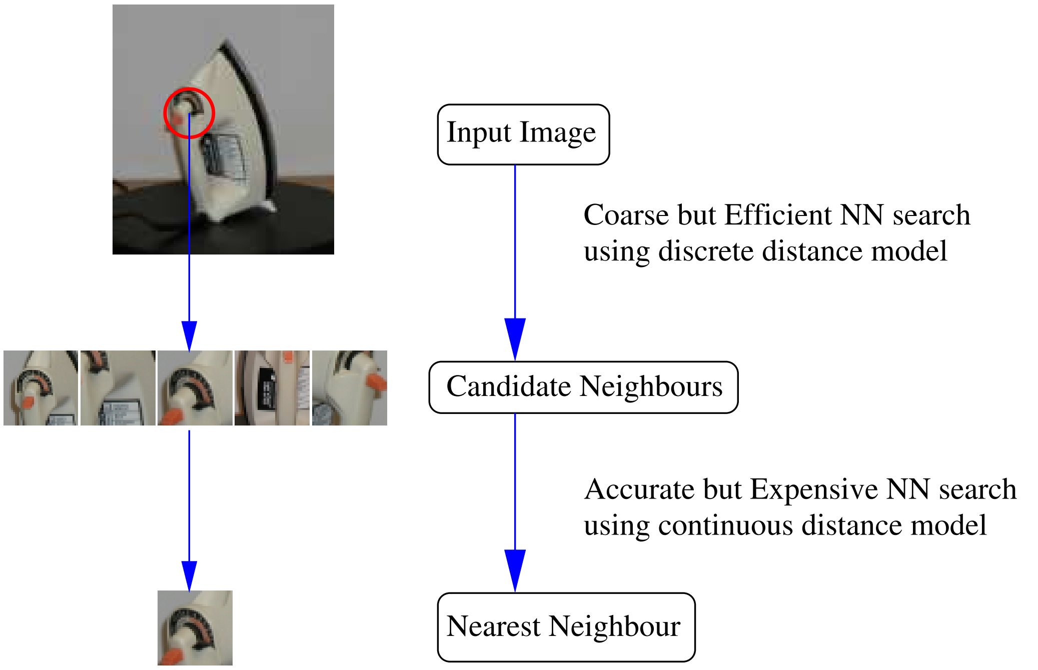


Figure 3.3: Our strategy for efficient and accurate nearest neighbor search. An input measurement is matched against each training measurement using the coarse but efficient discrete approximation to the optimal distance measure, yielding a small list of candidate neighbors. These candidate neighbors are then searched for the closest neighbor using the more accurate but expensive to use continuous model for the optimal distance measure. On the left is shown an actual example from our experiments (see Chapter 7). We only show the nearest neighbors for the sub-image from the input that is circled.

neighbor is returned when using either of the distance measure H and H^d, they both have the same classification performance for the given training set.

The above construction is obviously not useful in practice since the construction of H^d requires knowledge of the optimal distance measure H. Thus we seek a practical model for discrete distance measures. Our approach approximates the optimal distance measure by linearly combining a set of elementary *discretized* distance measures associated with discriminators acting on various feature spaces (color, shape, texture) as detailed below. Even though the above construction for H^d was only of theoretical interest, we will reuse the idea behind the construction when associating distance measures with discriminators as detailed later. We choose to use elementary distance measures associated with discriminators so that we can compose such elementary distance measures in a tree-like structure for efficient run-time nearest neighbor search, to be used in our hierarchical distance model (see § 3.3). In Chapter 6, we discuss the details for implementing such a tree-like structure.

Any discriminator can be characterized by the partition in measurement space that it induces. For example, a simple discriminator might test whether the average intensity or some other simple statistic of the input image crosses a threshold, in which case the the measurement space is split into two parts. A decision tree, on the other hand, partitions the measurement space into many parts, where each part corresponds to a leaf node of the decision tree. Another type of discriminator which we use in our work due to its ease of implementation and wide applicability is the nearest prototype discriminator (see Chapter 5). A nearest prototype discriminator is specified by the number and locations of a set of prototypes in some given feature space. The partition induced is the Voronoi diagram associated with the set of prototypes where each partition contains measurements in the given feature space that is closest to one of the prototypes. See Figure 3.4 for examples of nearest prototype discriminators.

A "good" discriminator induces a partition that is aligned well with the class boundaries, i.e. ideally two measurements from the same class will likely be contained within the same partition while two measurements from different classes will likely be in different partitions (see Figure 3.4). It is easy to construct a distance measure associated with a discriminator that shares the same property. The distance measure is a discretization induced by the discriminator of the underlying

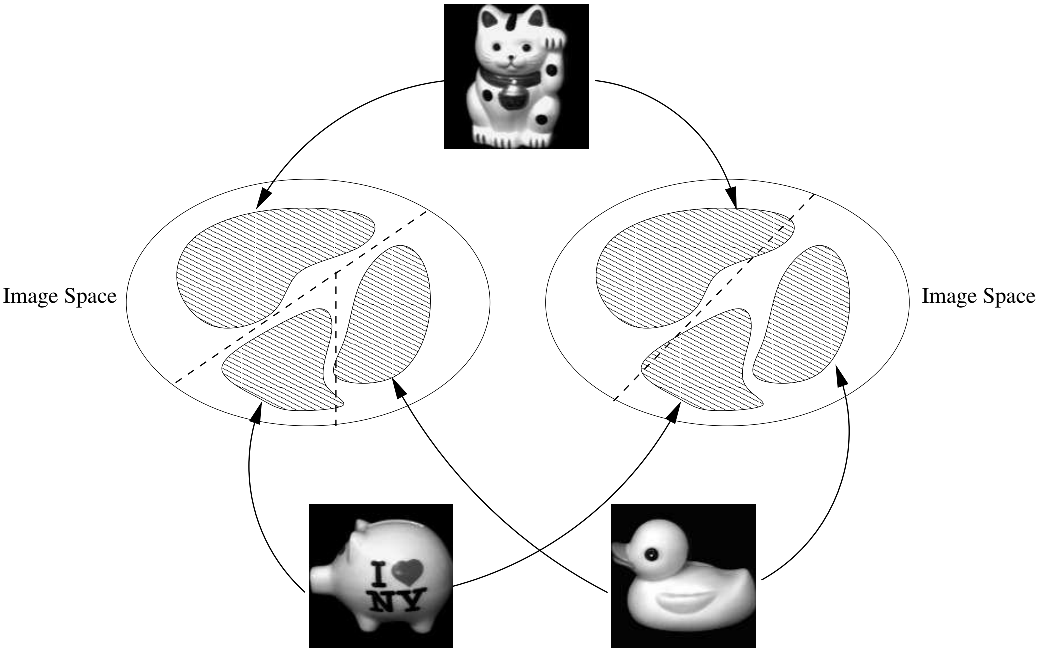


Figure 3.4: Discriminators are characterized by the partitions induced by them in image space. Shown here are three classes of objects and two simple discriminators, the one on the left partitions the image space into three parts while the one on the right partitions the image space into two parts. The image space is denoted by an ellipse. The discriminator on the left is good with respect to the three object classes since different object classes are more or less contained in different partitions, while the discriminator on the right confuses two of the object classes in the same partition. Each discriminator can be associated with an elementary discretized distance measure indicating whether a pair of images belong to the same partition or not. Such elementary distance measures corresponding to a set of simple discriminators are combined to approximate the optimal distance measure for the nearest neighbor search.

distance measure of the feature space in which the discriminator is constructed. Such a discrete distance measure will inherit the "goodness" of the discriminator, i.e. two measurements falling in the same partition induced by the discriminator will have a lower distance score than if they fall in different partitions.

The Voronoi construction used above for finding the discrete distance measure H^d that has the same classification performance as the optimal distance measure illustrated how a distance measure H and a set of training measurements induces a partition of image space and an associated discretized distance measure H^d. The idea behind this construction can also be used to find the discretized distance measure associated with a discriminator as follows.

The idea behind the construction for H^d from H is to design a discretized distance measure that is smallest for two measurements in the same partition compared with two measurements in different partitions. We can apply the same idea for associating a distance measure with a discriminator. The distance measure that we seek should be designed such that two measurements in the same partition induced by the discriminator is given a lower distance score compared with two measurements that fall in different partitions. A simple distance measure that satisfies the above requirement can be designed as follows. Let the discriminator h induce the partition $X = X_1 \cup X_2 \cup \ldots \cup X_n, X_i \cap X_j = \emptyset, i \neq j$. On input x, let $h(x)$ denote the partition X_i that x falls under. The discretized distance measure associated with discriminator h, denoted by $[h(x) = h(x')]$, is defined by:

$$[h(x) = h(x')] \equiv \begin{cases} -1 & \text{if } h(x) = h(x') \\ +1 & \text{otherwise} \end{cases}$$

Note that the above distance measure is just one of many such distance measures that can be used. All that is required is for the distance measure to assign a lower distance score between two measurements from the same partition compared with two measurements from different partitions. The above function is the simplest such distance measure.

We would also like to note the relationship between the elementary distance measures used in the discrete distance model and those used in the continuous distance model. The elementary distance measures used in the discrete model are discretizations induced by discriminators of the same elementary distance measures used for the continuous distance model. Different discriminators induce different discretizations of the same elementary distance measure. Obviously, the discretizations induced by good discriminators will be better elementary distance measures compared with poor discriminators.

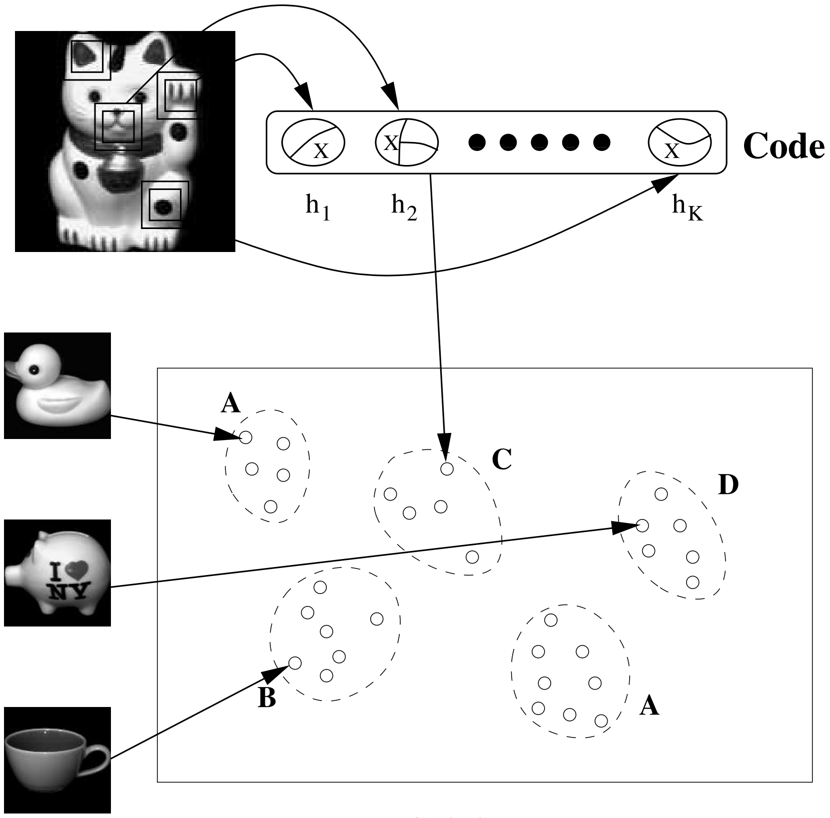
In general, we can assume that we have a possibly large collection of discriminators $\mathcal{H} = \{h_1, h_2, \dots\}$, each of which is constructed in some simple feature space like color, shape or texture. Corresponding to $\mathcal{H}$, we have the collection of elementary distance measures $\mathcal{C} = \{[h(x) = h(x')] \mid h \in \mathcal{H}\}$. The K best discriminators $h_k \in \mathcal{H}, k = 1, \dots, K$ are chosen whose corresponding elementary distance measures in $d_k \in \mathcal{C}$ give the best linear discretized approximation to H (3.4):

$$H(x, x') \approx \alpha_0 + \sum_{k=1}^{K} \alpha_k d_k(x, x') \tag{3.10}$$

$$= \alpha_0 + \sum_{k=1}^{K} \alpha_k [h(x) = h(x')] \tag{3.11}$$

One can think of the set of partition labels $\{h_k(x)\}$ output by each of the discriminators on a measurement x as a "code" for x. Viewed in this light, the above linear approximation can be thought of as a weighted hamming distance measure between the "codes" $\{h_k(x)\}$ and $\{h_k(x')\}$ for two measurements x and x'. Thus we seek the K discriminators and combining coefficients that give the best hamming distance measure in "code" space, i.e. separates measurements from different classes as much as possible in code space while clustering measurements in the same class, see Figure 3.5.

In the next chapter, we discuss the selection of the best K elementary distance measures from $\mathcal{C}$ as well as estimating the best corresponding combining coefficients under the maximum likelihood framework for exponential models.



Code Space

Figure 3.5: Illustration of the "code" space induced by a set of discriminators. At the top is the encoding of an input image by a set of discriminators $\{h_1, h_2, \ldots, h_K\}$. As explained in Figure 3.4, each such discriminator is characterized by the partition in image space that it induces, shown here above each discriminator with the image space denoted by an ellipse. The partition in which the input image falls under is marked by a $\times$ for each discriminator. The partition can also be thought of as the label given to the input image by the discriminator. The resulting set of partition labels over all discriminators can be thought of as a code for the input image. Our goal is to find a set of good discriminators and corresponding weights such that in the corresponding code space, the hamming distance measure clusters together images from the same object class while separating away as much as possible images from different object classes. Shown above is a code space with such a "good" hamming distance measure.

Chapter 4

Estimating the Optimal Distance Measure

In the last chapter, we presented a linear model for the optimal distance measure for nearest neighbor search that combines elementary distance measures. As noted, this model implies a conditional exponential model for the pair-wise distribution (3.9). In this chapter, we first deal with the issue of estimating the parameters in the linear model for a given set of elementary distance measures by employing the maximum likelihood estimation framework for exponential models. We also discuss the maximum likelihood selection criterion for the optimal set of elementary distance measures themselves, given a large collection of such elementary distance measures. We then discuss the maximum entropy framework that is the dual of the maximum likelihood framework and show that a natural selection criterion under this framework that was proposed in the literature is equivalent to the maximum likelihood criterion. Finally, we describe the relationship of our work with boosting (Freund and Shapire, 1997; Schapire and Singer, 1999).

4.1 Maximum Likelihood Estimation

In the previous chapter, we had presented two types of linear models: discrete and continuous. In both cases, the model combines a set of elementary distance measures. In the case of the continuous model, the elementary distance measures act upon simple feature space like color, local shape properties, and texture. In the case of the discrete model, the elementary distance measures are associated with discriminators constructed in various simple feature spaces. These distance measures are discretizations induced by the discriminators of the same elementary distance measures used for the continuous distance model. The estimation framework that we present below requires as input only a collection of elementary distance measures, discrete or continuous. Even though the basic estimation framework is the same for both cases, for concreteness of presentation, we will assume a collection of discretized elementary distance measures. This will also allow us to explore issues that are specific to the discrete distance model. We will in any case point out at the appropriate places, how the presentation below is essentially the same for a collection of continuous distance measures as well.

Thus for concreteness, we will assume that we are given a collection $\mathcal{C} = \{[h(x) = h(x')] \mid h \in \mathcal{H}\}$ of elementary distance measures associated with a large finite set of candidate discriminators $\mathcal{H} = \{h_1, h_2, \ldots h_N\}$, each of which is constructed in some simple feature space. Recall from (3.3.1) that $[h(x) = h(x')]$ denotes the distance measure associated with the discriminator h. The next chapter will discuss how we can generate such a collection of simple discriminators based on various feature spaces. We wish to choose $K \ll N$ discriminators from this collection that gives the best discrete approximation to the distance measure H. In practice, K will be limited for example by run-time performance considerations.

How good is an approximation to the distance measure? Since the distance measure H is related to the pair-wise distribution $p(y \neq y'|x, x')$ through the logit transform (3.2), the task of finding the best approximation reduces to modeling the distribution using the best K discriminators. We will use the maximum likelihood framework (Duda et al., 2001; Bishop, 1995) for finding the best discriminators.

First we introduce some useful notation. If y_i and y_j are the class labels of two measurements x_i and x_j respectively, then let y_{ij} be a binary variable taking the value -1 if $y_i = y_j$ and $+1$ otherwise. Using the binary variable y_{ij} we can rewrite the two pair-wise distributions in equation(3.3) more compactly as follows:

$$p(y_{ij} \mid x_i, x_j) = \frac{1}{1 + e^{-y_{ij} H(x_i, x_j)}} \tag{4.1}$$

In the following we will denote the linear approximation (3.4) to H by $\hat{H}$ and the corresponding approximation to the pair-wise distribution by $\hat{p}$.

We seek the best approximation $\hat{p}$ to the pair-wise distribution from a set of training data. However, the training data that we will be presented with in our task is a set of measurements and associated class labels $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$. For estimating $\hat{p}$, we need a training set which consists of pairs of measurements (x_i, x_j) associated with the label y_{ij} indicating whether the pair comes from the same class ($y_{ij} = -1$) or not ($y_{ij} = +1$). We can easily create such a training set from the given training set S. One such set which we denote by S^2 considers all possible pairs of training measurements from S:

$$S^2 \equiv \{((x_i, x_j), y_{ij}) \mid y_{ij} = -1 \quad \text{if } y_i = y_j \text{ else } +1, \ i, j = 1, \ldots, N\} \tag{4.2}$$

However, this leads to a new training set of size N^2 which can be computationally expensive to use for training. In practice, we sample some manageable number of pairs instead of all possible pairs of training measurements (see Chapter 6).

Let $\mathbf{h} = \{h_0, \ldots, h_K\}$ be our current selection of discriminators from the collection $\mathcal{H}$, where for compactness in the notation below, $h_0 \equiv -1$ is the trivial discriminator that corresponds to the bias and is always assumed to be chosen. As described in the last chapter, each such discriminator h_k is associated with a discretized elementary distance measure $[h_k(x_i) = h_k(x_j)]$ that takes the value -1 if a pair of images x_i and x_j falls under the same partition of measurement space induced by h_k, and takes the value $+1$ otherwise. Let $\boldsymbol{\alpha} = \{\alpha_0, \ldots, \alpha_K\}$ be our current choice for the combining coefficients in the linear approximation to H. The current choice for $\mathbf{h}$ and $\boldsymbol{\alpha}$ determines a particular distribution $\hat{p}$. The log-likelihood $l(\boldsymbol{\alpha}, \mathbf{h}|S^2)$ indicates how well the current choice for $\mathbf{h}$ and $\boldsymbol{\alpha}$ model the training data S^2 and is defined as:

$$l(\boldsymbol{\alpha}, \mathbf{h} \mid S^2) \;\; \equiv \;\; \frac{1}{|S|^2} \sum_{i,j}^{N} \log \hat{p}(y_{ij} \mid x_i, x_j) \tag{4.3}$$

Substituting the linear approximation (3.4), the above expands to:

$$\begin{aligned} l(\boldsymbol{\alpha}, \mathbf{h}|S^2) &= -\frac{1}{|S|^2} \sum_{i,j}^{N} \log(1 + e^{-y_{ij}\hat{H}(x_i, x_j)}) \\ &= -\frac{1}{|S|^2} \sum_{i,j}^{N} \log\left(1 + \exp\left(-\sum_{k=0}^{K} \alpha_k y_{ij} [h_k(x_i) = h_k(x_j)]\right)\right) \end{aligned} \tag{4.4}$$

4.1.1 Estimating the Continuous Model

The estimation for a linear continuous distance model is exactly the same as above with $[h_k(x_i) = h_k(x_j)]$ replaced by $d_k(x_i, x_j) \in \mathcal{C}$ where $\mathcal{C}$ is now a collection of elementary continuous distance measures:

$$l(\boldsymbol{\alpha}, \mathbf{d}|S^2) \quad = \quad -\frac{1}{|S|^2}\sum_{i,j}^{N} \log\left(1 + \exp\left(-\sum_{k=0}^{K} \alpha_k y_{ij} d_k(x_i, x_j)\right)\right) \tag{4.5}$$

where $\mathbf{d} = \{d_0, \ldots, d_K\}$ is the current choice of elementary distance measures.

Each choice for the set of discriminators $\mathbf{h}$ can be associated with a score that indicates how well $\mathbf{h}$ models the training data. Under the maximum likelihood estimation framework, the score for $\mathbf{h}$ is the maximum likelihood of the data attained by $\mathbf{h}$ over all choices of $\boldsymbol{\alpha}$. Overloading the notation, we denote the score for $\mathbf{h}$ by $l(\mathbf{h}|S^2)$:

$$l(\mathbf{h} \mid S^2) \equiv \max_{\boldsymbol{\alpha}\in\mathbb{R}^K} l(\boldsymbol{\alpha}, \mathbf{h} \mid S^2) \tag{4.6}$$

We can now state the maximum likelihood criterion for choosing the best K discriminators:

Criterion ML. *Choose the K discriminators from the collection $\mathcal{H}$ that maximize $l(\mathbf{h} \mid S^2)$:*

$$\begin{aligned} \mathbf{h}^{\mathrm{ML}} &= \operatorname*{argmax}_{\mathbf{h}\subset\mathcal{H}, |\mathbf{h}|=K} l(\mathbf{h} \mid S^2) \\ &= \operatorname*{argmax}_{\mathbf{h}\subset\mathcal{H}, |\mathbf{h}|=K} \max_{\boldsymbol{\alpha}\in\mathbb{R}^K} l(\boldsymbol{\alpha}, \mathbf{h} \mid S^2) \end{aligned} \tag{4.7}$$

where $|\cdot|$ denotes the size of a set.

In the remainder of the section, we consider various issues that are important in practice: (a) optimization, (b) interpreting α_k and (c) regularization.

4.1.2 Optimization

Note that the above selection criterion involves two types of optimization. One is an optimization over a discrete space $\mathcal{H}$ for the best discriminators. The other is an optimization over a continuous space $\mathbb{R}^K$ for the combining coefficients $\boldsymbol{\alpha}$ for each choice of discriminators $\mathbf{h}$ in the discrete optimization above. We discuss the practical issues involved in these two types of optimization.

Optimization over Discriminators

For the discrete optimization, searching for the best K discriminators from the collection $\mathcal{H}$ in a brute-force manner will in general be computationally prohibitive. The brute-force approach in which every choice of K discriminators is evaluated takes $O(|\mathcal{H}|^K)$ evaluations. Instead we propose a simple sequential greedy scheme that takes $O(K|\mathcal{H}|)$. At the start of each iteration of the greedy scheme, we have a set of discriminators $\mathbf{h^k} = \{h_1, \dots, h_k\}, k < K$ that were selected in the previous iterations. We choose the discriminator $h_{k+1} \in \mathcal{H}$ that along with the previously chosen discriminators $\mathbf{h^k}$ maximizes the likelihood score of the data. More precisely, letting $\mathbf{h^{k+1}} = \mathbf{h^k} \cup \{h_{k+1}\}$, we choose the discriminator $h_{k+1} \in \mathcal{H}$ that maximizes the score $l(\mathbf{h^{k+1}}|S^2)$ defined in (4.6).

Here for simplicity, we have assumed that the collection $\mathcal{H}$ of discriminators is fixed over all iterations. In Chapter 6 we discuss how to compose discriminators in a tree-like structure for efficient run-time performance. We will see that this will lead to choosing discriminators from a collection $\mathcal{H}_k$ that can vary with each iteration in the greedy scheme.

Optimization over $\boldsymbol{\alpha}$

The continuous optimization for the optimal combining coefficients $\boldsymbol{\alpha}$ for a given selection of discriminators $\mathbf{h}$, on the other hand, leads to a convex optimization problem. This fact is well-known in the literature (Della Pietra et al., 1997; Schapire and Singer, 1999; Lebanon and Lafferty, 2001), but for completeness and better insight we prove the convexity result for our task. Using the expanded form (4.4) for the likelihood l, maximizing the likelihood of the data for a fixed $\mathbf{h}$ amounts to minimizing the following cost function (for convenience, we have dropped the normalizing term $\frac{1}{|S|^2}$ which is constant for a given training set):

$$\begin{aligned}
J_{\mathbf{h}}(\boldsymbol{\alpha}) &\equiv \sum_{i,j}^{N} \log(1 + e^{-y_{ij}\hat{H}(x_i,x_j)}) && (4.8)\\
&= \sum_{i,j}^{N} \log\left(1 + \exp\left(-\sum_k \alpha_k y_{ij}[h_k(x_i) = h_k(x_j)]\right)\right) && (4.9)\\
&= \sum_{i,j}^{N} \log\left(1 + \exp\left(-\sum_k \alpha_k u_{ij}^k\right)\right) && (4.10)\\
& && (4.11)
\end{aligned}$$

where we have used the notation $u_{ij}^k \equiv y_{ij}[h_k(x_i) = h_k(x_j)]$ for compactness.

The first derivative of this cost function is given by:

$$\frac{\partial J}{\partial \alpha_k} = -\sum_{i,j}^{N} u_{ij}^k \sigma(-y_{ij}\hat{H}(x_i, x_j))$$

where $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function. It can be verified that $\sigma'(x) = \sigma(x)(1-\sigma(x))$. Thus $\sigma'(x) > 0$ since $\sigma(x)$ has range in $(0,1)$ for $-\infty < x < \infty$. We then get the following for the Hessian:

$$\frac{\partial^2 J}{\partial \alpha_r \partial \alpha_s} = \sum_{i,j}^{N} u_{ij}^r u_{ij}^s \sigma'_{ij} \qquad (4.12)$$

where we have used the notation $\sigma'_{ij} \equiv \sigma'(-y_{ij}\hat{H}(x_i, x_j))$. Since each $\sigma'_{ij} > 0$ as shown above, the Hessian of J is seen to be positive definite as follows: for any $\boldsymbol{\alpha}$, we have:

$$\begin{aligned}
\sum_{r,s}^{K} \alpha_r \frac{\partial^2 J}{\partial \alpha_r \partial \alpha_s} \alpha_s &= \sum_{i,j}^{N} \sigma'_{ij} \sum_{r,s}^{K} \alpha_r u_{ij}^r \alpha_s u_{ij}^s\\
&= \sum_{i,j}^{N} \sigma'_{ij} (\boldsymbol{\alpha} \cdot \mathbf{u}_{ij})^2 > 0
\end{aligned}$$

where $\mathbf{u}_{ij} \equiv (u_{ij}^1, \ldots, u_{ij}^K)$. Thus J is convex in $\boldsymbol{\alpha}$ whose minimum can be found using well-established iterative techniques like Newton's method (Press et al., 1992).

4.1.3 Interpreting α_k

Given a choice of K discriminators, we might expect that if a discriminator h_r is "better" than another discriminator h_s at the discrimination task, then the optimal value for the corresponding combining coefficient α_r should be higher than α_s, or in other words, α_r indicates the relative utility of the discriminator h_r at the discrimination task. In this section, we give some analytical justification for this intuition. We will see that the best choice for the K discriminators are those that best "complement" each other in a sense that will be made precise below.

With respect to a given discriminator h_k and a fixed pair of training measurements x_i and x_j, we use the following notations in what follows:

$$\begin{aligned}
z_{ij}^0 &= \sum_{r \neq k} \alpha_r^* y_{ij}[h_r(x_i) = h_r(x_j)] \\
\epsilon_{ij} &= \alpha_k y_{ij}[h_k(x_i) = h_k(x_j)] \\
z_{ij} &= z_{ij}^0 + \epsilon_{ij} \\
g_{ij}(z_{ij}) &= \log\left(1 + e^{-z_{ij}}\right) \\
J(\alpha_k) &= \sum_{ij} g_{ij}(z_{ij})
\end{aligned}$$

where $\alpha_r^*, r \neq k$ are the optimal values minimizing the cost function J in (4.11) and where $g_{ij}(z_{ij})$ corresponds to one term in the cost function J with all of the α_r except α_k set to its optimal value. Also, J has been re-written as a function of only α_k.

We would like to find a closed form expression for the optimal value of each α_k. As it stands, this is not possible with the cost function J. Instead, we will find a closed form expression to a quadratic approximation to J.

Consider the quadratic approximation to each term $g_{ij}(z_{ij}^0 + \epsilon_{ij})$ about z_{ij}^0:

$$\begin{aligned}
\hat{g}_{ij}(z_{ij}) &= g_{ij}(z_{ij}^0) + \epsilon_{ij} g'(z_{ij}^0) + \frac{\epsilon^2}{2} g''(z_{ij}^0) \\
&= g_{ij}(z_{ij}^0) - \epsilon_{ij}\sigma(-z_{ij}^0) + \frac{\epsilon^2}{2}\sigma(-z_{ij}^0)(1 - \sigma(-z_{ij}^0))
\end{aligned}$$

where as in § 4.1.2, $\sigma(\cdot)$ is the sigmoid function. The approximation to the cost function J is then:

$$\hat{J}(\alpha_k) = \sum_{ij} \hat{g}_{ij}(z_{ij})$$

Minimizing the quadratic approximation $\hat{J}$ for the optimal value for α_k by setting the derivative to 0, we get:

$$\begin{aligned}
(\sum_{ij} -y_{ij}[h_k(x_i) = h_k(x_j)]\sigma(-z_{ij}^0)) && (4.13)\\
+\alpha_k^*(\sum_{ij}(y_{ij}[h_k(x_i) = h_k(x_j)])^2\sigma(-z_{ij}^0)(1-\sigma(-z_{ij}^0)) &= 0 & (4.14)\\
\Longrightarrow (\sum_{ij} -u_{ij}\sigma(-z_{ij}^0)) + \alpha_k^*(\sum_{ij}\sigma(-z_{ij}^0)(1-\sigma(-z_{ij}^0)) &= 0 & (4.15)
\end{aligned}$$

where as in § 4.1.2, $u_{ij} \equiv y_{ij}[h_k(x_i) = h_k(x_j)]$.

We introduce some further notation:

$$\begin{aligned}
W_k^+ &= \sum_{u_{ij}=+1} \sigma(-z_{ij}^0) > 0\\
W_k^- &= \sum_{u_{ij}=-1} \sigma(-z_{ij}^0) > 0\\
W_k^0 &= \sum_{ij} \sigma(-z_{ij}^0)(1-\sigma(-z_{ij}^0)) > 0
\end{aligned}$$

The term z_{ij}^0 depends only on the other discriminators $h_r, r \neq k$ and can be rewritten as $z_{ij}^0 = y_{ij}\hat{H}_k(x_i, x_j)$ where $\hat{H}_k(x_i, x_j) \equiv \sum_{r\neq k} \alpha_r[h_r(x_i) = h_r(x_j)]$ is the distance measure using all discriminators except h_k. With this rewrite, z_{ij}^0 can be seen as measuring how well the other discriminators have classified the pair x_i and x_j, larger values indicating better classification. The term $\sigma(-z_{ij}^0)$ can then be thought of as a weight associated with the pair x_i and x_j that indicates how well the other discriminators have classified the pair. Pairs that are incorrectly classified by the linear combination of the other discriminators are associated with a large weight. Note that since the sigmoid is bounded above by 1, it does not over-penalize incorrect classifications. Next, the term u_{ij} indicates whether the discriminator h_k classifies a pair of measurements x_i and x_j correctly ($u_{ij} = +1$)

or incorrectly ($u_{ij} = -1$). Thus W_k^+ denotes the total weight associated with all pairs that are correctly classified by h_k and similarly W_k^- denotes the total weight associated with all pairs that are incorrectly classified by h_k. W_k^0 on the other hand is independent of h_k and is thus a constant.

Continuing with the minimization of α_k in (4.15), we get:

$$\begin{aligned} W_k^- - W_k^+ + \alpha_k^* W_k^0 &= 0 \\ \Longrightarrow \hat{\alpha}_k^* &= \frac{(W_k^+ - W_k^-)}{W_k^0} \end{aligned}$$

where we have denoted the optimum to the quadratic approximation by $\hat{\alpha}_k^*$ to distinguish it from the the true optimum α_k^* obtained by minimizing the true cost function J (4.11). Substituting this optimum back into the quadratic approximation $\hat{J}$, we get:

$$\hat{J}(\alpha_k^*) = \hat{J}_k^0 - \frac{(W_k^+ - W_k^-)^2}{2W_k^0}$$

where $\hat{J}_k^0$ is the cost due to all the discriminators except h_k.

Thus under the quadratic approximation, intuitively speaking, $\hat{J}$ is minimized by a choice for the discriminator h_k that correctly classifies pairs associated with large weights while affording to incorrectly classify pairs associated with low weights. In this sense, the best choice for h_k is that which "complements" the other discriminators the most. Since the optimal value for α_k is proportional to the difference $W_k^+ - W_k^-$ in the total weight associated with pairs that h_k correctly classifies and the total weight associated with pairs that h_k incorrectly classifies, we can think of $\hat{\alpha}_k^*$ as measuring how well h_k correctly classifies those pairs that were not classified well enough by the other discriminators.

4.1.4 Regularization

Maximum likelihood estimation can suffer from over-training (Duda et al., 2001; Bishop, 1995; Lebanon and Lafferty, 2001; Chen and Rosenfeld, 2000). As shown in the last section, in our case this means the optimal estimate for any of the α_k can be overly confident about the discriminative power of the corresponding discriminator h_k if its value is large in magnitude. The standard approach to dealing with over-training is to use priors on the possible values for the parameters being optimized. This leads to the maximum a posteriori estimation (MAP) framework. Under MAP, the likelihood (4.4) is replaced by:

$$l(\boldsymbol{\alpha}, \mathbf{h} \mid S^2) \equiv \frac{1}{|S|^2} \sum_{i,j}^{N} \log \hat{p}(y_{ij} \mid x_i, x_j) + \sum_{k}^{K} \log q_k(\alpha_k) \tag{4.16}$$

where q_k is the prior distribution over the parameter α_k.

What should be an appropriate choice for the prior q_k? The prior should penalize large values of α_k, since as discussed above, large values likely indicate over-confidence about the discriminative power of the corresponding discriminator h_k. Other than that, we would like to preserve the convexity of the resulting optimization problem just as was the case for the ML framework (see § 4.1.2 above). A simple prior that satisfies both these constraints is the Gaussian:

$$q_k(\alpha) \sim e^{-\frac{\alpha_k^2}{2\sigma_k^2}}$$

where the choice for the variance σ_k limits the effective range of the parameter α_k. It can be seen that the cost function that needs to be minimized under the MAP framework is simply the cost function J (4.11) for the ML framework plus a quadratic term due to the priors on $\boldsymbol{\alpha}$:

$$J_{\mathbf{h}}(\boldsymbol{\alpha}) \equiv \sum_{i,j}^{N} \log(1 + e^{-y_{ij}\hat{H}(x_i,x_j)}) + \sum_{k}^{K} \frac{\alpha_k^2}{2\sigma_k^2} \tag{4.17}$$

This new cost function is also convex as was the case for the ML framework as the Hessian of J is still positive definite since the contribution of the quadratic term is only an additional positive quantity $1/\sigma_k^2$ along the diagonal of the Hessian (4.12) under the ML framework.

4.2 Maximum Entropy Formulation

In this section, we consider an alternative formulation for estimating the pair-wise distribution $p(y_{ij} \mid x_i, x_j)$ that is dual to the maximum likelihood framework discussed in the previous section. The main reason we consider the dual framework is to present new insights into the estimation problem. We also consider a natural criterion for selecting the best discriminators for modeling the pair-wise distribution under this framework. This criterion has been previously used in the vision literature for texture synthesis (Zhu et al., 1998). We are interested in knowing the relationship between this criterion and the maximum likelihood criterion presented in the previous section in the hope of using the superior one in practice. On the surface, the two criteria look quite different. Nevertheless, we prove that they are in fact the same criterion seen from different perspectives.

For a fixed pair of measurements x_i and x_j, let y_{ij} be a sample from the pair-wise distribution $p(y_{ij} \mid x_i, x_j)$. Recall that $y_{ij} = -1$ if x_i and x_j belong to the same class and $y_{ij} = +1$ otherwise. For a given discriminator h, the function $f(y_{ij}, x_i, x_j) = y_{ij}[h(x_i) = h(x_j)]$ can be considered as a test as to whether the discriminator h "classifies" the pair of measurements x_i and x_j correctly. That is, $f(y_{ij}, x_i, x_j) = +1$ if either the pair belongs to the same class (i.e., $y_{ij} = -1$) while also falling in the same partition induced by the discriminator h (i.e., $[h(x_i) = h(x_j)] = -1$) or if the pair belongs to different classes (i.e., $y_{ij} = +1$) while also falling in different partitions (i.e., $[h(x_i) = h(x_j)] = +1$). On the other hand, $f(y_{ij}, x_i, x_j) = -1$ indicates that the discriminator h did not classify the pair of measurements correctly.

The average classification performance of a discriminator h is the expected value of f. If the expected value is $+1$, we have perfect classification, if it is 0 the performance is random, and if it is -1, the classification is always wrong. In practice, we do not know the true pair-wise distribution, but the expected value of the classification performance can be estimated from a training set. Formally, we define the empirical performance $\langle f \rangle$ as:

$$\langle f \rangle \equiv \frac{1}{|S^2|} \sum_{((x_i,x_j),y_{ij}) \in S^2} f(y_{ij}, x_i, x_j)$$

We seek to estimate the true pair-wise distribution $p(y_{ij} \mid x_i, x_j)$ from the space of all probability measures. What should the constraints be?

Suppose we are given K discriminators $h_k, k = 1, \ldots, K$ for each of which we can determine the empirical performance $\langle f \rangle_k$ from the training data. Clearly, one set of constraints on the estimated distribution $\hat{p}(y_{ij} \mid x_i, x_j)$ is that the classification performance of the K discriminators under the estimate is the same as the empirical performance $\langle f \rangle_k$ determined from the training data (ignoring noise in the estimates). However, we still only have a finite number K of such constraints and thus the optimal choice for the estimate of the distribution from an infinite set of possible probability measures is still ill-defined. Clearly we need some other criterion that is not data driven. The maximum entropy (ME) principle (Jaynes, 1957; Della Pietra et al., 1997) states that we should choose the probability distribution that satisfies the given constraints, but otherwise should be the "least committed" probability distribution.

Intuitively, the least committed probability distribution when there are no constraints is the uniform probability distribution. As we add constraints, we would like to keep the distribution as "close" to uniform as possible while satisfying the given constraints. More generally, we might like to be as close to a prior distribution q_0 that may not be uniform and which is task dependent but data independent. For our task, the "closeness" or distance between two conditional pair-wise distributions $p(y_{ij} \mid x_i, x_j)$ and $q(y_{ij} \mid x_i, x_j)$ can be measured by the following conditional Kullback-Leibler (KL) divergence (Della Pietra et al., 1997):

$$D(p, q) = \frac{1}{|S^2|} \sum_{(x_i, x_j) \in S^2} \sum_{y_{ij} \in \{-1, +1\}} p(y_{ij} \mid x_i, x_j) \log \frac{p(y_{ij} \mid x_i, x_j)}{q(y_{ij} \mid x_i, x_j)}$$

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

Let $\mathcal{M}$ be the space of all possible conditional pair-wise distributions $p(y_{ij} \mid x_i, x_j)$. Define the *feasible* set $\mathcal{F} \subset \mathcal{M}$ as:

$$\mathcal{F} \equiv \{p \in \mathcal{M} \mid E_p[f_k] = \langle f_k \rangle \text{ for all } k\} \tag{4.18}$$

where $E_p[\cdot]$ denotes expectation under the likelihood p. Then, the ME framework requires the solution to 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 (Della Pietra et al., 1997; Lebanon and Lafferty, 2001) that by setting up an appropriate Lagrangian, the optimal pair-wise distribution which we denote by p_{ME} takes the form of the logistic function:

$$p_{\mathrm{ME}}(y_{ij} \mid (x_i, x_j)) = \frac{1}{1 + \exp\left(-\sum_k \alpha_k f_k(x_i, x_j, y_{ij})\right)} \tag{4.19}$$

where α_k is the set of Lagrange multipliers, one for each of the constraints $E_p[f_k] = \langle f_k \rangle$.

The ME solution $p_{\mathrm{ME}}(y_{ij} \mid (x_i, x_j))$ takes the same form as the exponential model (4.1) in the previous section. In fact it is known (Della Pietra et al., 1997) that the ME solution is dual to the maximum likelihood (ML) exponential model. We discuss this duality in more detail below.

Consider the family of conditional exponential probability distributions:

$$Q \equiv \left\{ p \in \mathcal{M} \;\mid\; p(y_{ij} \mid (x_i, x_j)) \propto q_0(y_{ij} \mid (x_i, x_j)) e^{\sum_k \alpha_k f_k(x_i, x_j, y_{ij})},\ \alpha \in \mathbb{R}^K \right\} \tag{4.20}$$

where as before q_0 is a prior measure. The exponential model (4.1) considered in the previous section is a special case where the prior q_0 is uniform. Let $\tilde{p}(y_{ij} \mid x_i, x_j)$ be the empirical distribution determined by the training set S^2; $\tilde{p}(y_{ij} \mid x_i, x_j)$ simply takes the value 1 if $((x_i, x_j), y_{ij}) \in S^2$ and 0 otherwise. The log-likelihood L of a probability measure p with respect to the empirical distribution $\tilde{p}$ is defined as:

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

It can be verified that when q_0 is uniform and $p \in Q$, the above definition reduces to the likelihood defined in (4.4). It has been shown (Della Pietra et al., 1997) that the probability distribution 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.

4.2.1 ME Selection Criterion

We next consider the problem of selecting good discriminators under the ME framework just as we did for the ML framework in § 4.1. As in the case for the ML framework, we assume that we are given a large but finite collection $\mathcal{H}$ of discriminators. We wish to choose $K \ll N$ discriminators from this collection that is in some sense "optimal" under the ME framework. We will reexamine a selection scheme under the ME framework that has been recently proposed (Zhu et al., 1998). We will then show that despite the very different appearance of this selection criterion from the ML selection criterion, they are in fact equivalent. Thus from a practical point of view, there is no gain in considering the ME selection criterion, although it does bring new perspective to the issue of selecting the best discriminators.

Zhu et al. (1998) proposed the use of what they called the *mini-max* entropy criterion. The context of their work was the selection of good features for texture synthesis. In their formulation, the criterion assumes a uniform prior model for q_0 and chooses the K features such that the resulting maximum entropy probability distribution p_{ME} has *minimum* entropy over all choices of K features. This criterion might seem less intuitive at first than the ML criterion presented in the previous section. It is based on the notion that the entropy of the probability distribution determined by a given choice of K discriminators indicates how "informative" the discriminators are in specifying the pair-wise distribution, the discriminators being more informative the lower the entropy. Thus the mini-max entropy criterion chooses the K most informative discriminators. Since minimizing the entropy of a distribution p is the same as maximizing 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 $\mathbf{h} \equiv \{h_0, \dots, h_K\} \subset \mathcal{H}$, let $p^*(\mathbf{h})$ be the maximum entropy probability measure with constraints determined by the corresponding testing functions $f_1, \dots, f_K$, i.e. $p^*(\mathbf{h}) = \operatorname*{argmin}_{p \in F} D(p, q_0)$. Choose the K discriminators for which $D(p^*(\mathbf{h}), q_0)$ is maximum over all choices of K discriminators from $\mathcal{H}$.*

As before, we are assuming that the trivial discriminator $h_0 \equiv -1$ is always chosen.

At first reading, the ME criterion looks quite different from the ML criterion of the previous section. Nevertheless, we show next 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. First, we generalize and restate the ML criterion from the previous section for arbitrary priors q_0:

Criterion ML(restated). *For a fixed choice of K discriminators $\mathbf{h} \equiv \{h_0, \dots, h_K\} \subset \mathcal{H}$, let $p^*(\mathbf{h})$ be the probability measure that maximizes the likelihood $p^*(\mathbf{h}) = \operatorname*{argmax}_{p \in Q} L(\tilde{p}, p)$. Choose the K discriminators for which $L(\tilde{p}, p^*(\mathbf{h}))$ is maximum over all choices of K discriminators from $\mathcal{H}$.*

It can be verified that this reduces to the same criterion presented in the previous section if we assume the prior q_0 to be uniform.

Theorem 2 *A set of K discriminators optimizes the ME criterion iff they also optimize the ML criterion for the exponential family.*

Proof. We first state an analogue of the Pythagorean theorem for the KL divergence (Della Pietra et al., 1997):

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

where $\bar{Q}$ is the closure of Q and where by the duality theorem (Della Pietra et al., 1997):

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

We set $p = \tilde{p}$, the empirical distribution from the training set S^2, 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:

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

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

Figure (4.1) is a cartoon illustration of the proof. For clarity of presentation, we assume a collection of discriminators $\mathcal{H} = \{h_1, h_2\}$ containing just two discriminators. We seek the best linear model to the optimal distance measure H based on just one discriminator, either h_1 or h_2. Corresponding to each discriminator h_i, the figure shows the feasible set (4.18) F_i induced by h_i under the ME framework as well as the one-dimensional exponential family (4.20) Q_i under the ML framework. Note that the two feasible sets intersect at the empirical distribution $\tilde{p}$, while the two exponential families intersect at the prior model q_0.

The sets Q_i and F_i intersect at the unique distribution p_i^* as required by the duality theorem (Della Pietra et al., 1997). The three points $\tilde{p}$, p_i^* and q_0 form the triangle in the analogue of the Pythagorean theorem above. ML likelihood is related to the KL divergence between p_i^* and $\tilde{p}$, while the relative entropy is

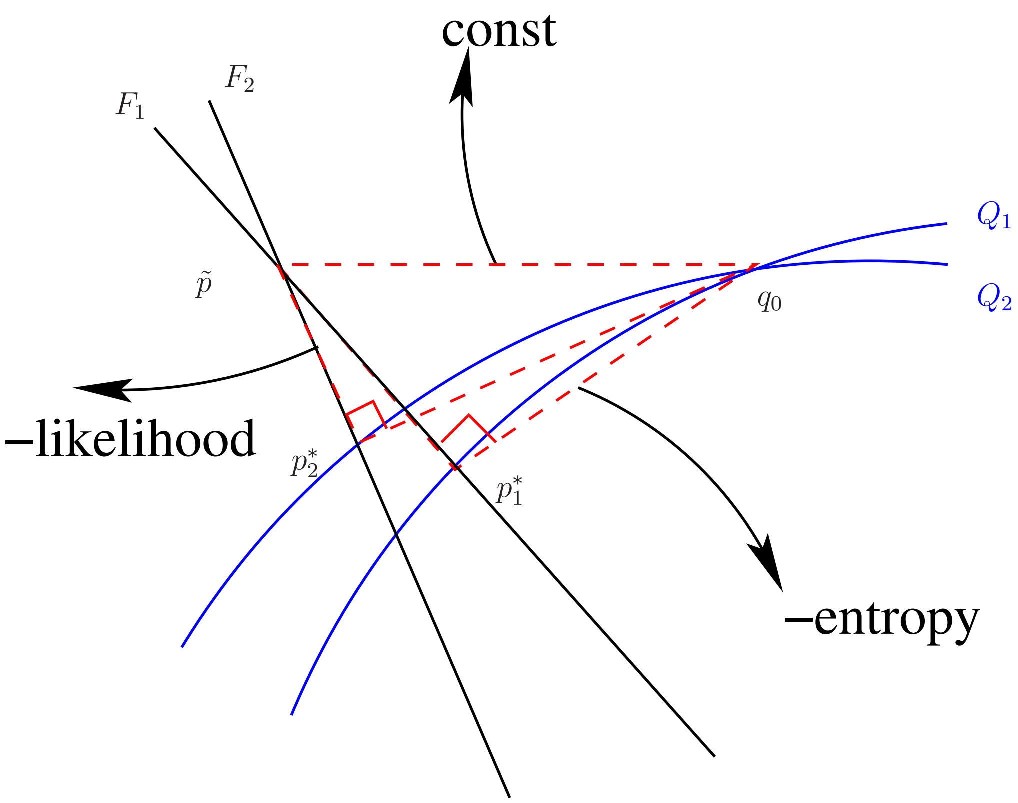


Figure 4.1: Illustrating the proof of theorem 2. See text for details.

related to the KL divergence between p_i^* and q_0. Changing the discriminator h_i changes only p_i^*, thus keeping the opposite leg of the triangle constant. Since the sum of the other two legs equals the leg opposite p_i^*, maximizing the likelihood is equivalent to minimizing the relative entropy. Thus in the figure, both selection criteria under the two frameworks would choose the distance model based on the discriminator h_1.

In closing it should be emphasized that the equivalence between the two criteria is not solely caused by the duality between the ML and ME frameworks. To see this, suppose we modified our selection criteria such that we could also choose the best prior models q_0 from some collection in addition to choosing the best set of discriminators. For such selection criteria, all the legs of the triangle can now vary and thus there is no guarantee that maximizing the likelihood will necessarily minimize the entropy simultaneously, even though the duality between the two frameworks of course still holds.

4.3 Connections with Boosting

The distance measure $h(x_i, x_j) \equiv [h(x_i) = h(x_j)]$ corresponding to a discriminator h can also be thought of as a binary classifier on pairs of measurements (for convenience, we have abused the notation h to indicate both a discriminator which acts on a measurement and a classifier that acts on a pair of measurements, the correct interpretation should be clear from the context). A pair of measurements is classified with the label $h(x_i, x_j) = -1$ if both belong to the same partition induced by the discriminator h, otherwise they are classified with the label $h(x_i, x_j) = +1$. A good classifier is one that will more likely output -1 for a pair of images that belong to the same class, while $+1$ is more likely to be output if they belong to different classes.

Consider K such binary classifiers $h_k, k = 0, \ldots, K$ that we wish to combine using a linear combination:

$$F(x_i, x_j) = \sum_{k=0}^{K} \alpha_k h(x_i, x_j)$$

Using this combination, a pair of measurements is classified with sign$(F(x_i, x_j))$.

Let $S^2 \equiv \{((x_i, x_j), y_{ij})\}$ be a training set on pairs of measurements as before (4.2). The error of the combined classifier on the training set is given by:

$$E \equiv \sum_{((x_i,x_j),y_{ij}) \in S^2} [\![\text{sign}(F(x_i, x_j)) \neq y_{ij}]\!]$$

where for a predicate π, $[\![\pi]\!]$ is 1 if π is true, else 0. We wish to find the classifiers h_k and corresponding combining coefficients α_k that minimize the training error E. However, E is a discrete function and thus hard to work with. The boosting framework (Freund and Shapire, 1997; Schapire and Singer, 1999) uses instead a continuous upper bound that is easier to work with. The exponential cost function is commonly used to bound the discrete training error above. Using the exponential cost function, it can be easily verified that:

$$[\![\text{sign}(F(x_i, x_j)) \neq y_{ij}]\!] < e^{-y_{ij}F(x_i,x_j)}$$

Thus we can replace the discrete training error E with the continuous upper bound:

$$\begin{aligned} E &< \sum_{((x_i,x_j),y_{ij}) \in S^2} e^{-y_{ij}F(x_i,x_j)} \\ &= \sum_{((x_i,x_j),y_{ij}) \in S^2} e^{-\sum_k \alpha_k y_{ij} h(x_i,x_j)} \end{aligned}$$

In boosting, this upper bound is minimized with respect to the choice of classifiers h_k and the corresponding α_k.

Comparing the upper bound above with the cost function J, we see that the only difference is the cost function: the upper bound uses the exponential cost function whereas the maximum likelihood framework results in the log cost function. The log cost function is better behaved compared with the exponential cost function as it does not over-penalize bad classifications. In fact, more recent work (Mason et al., 2000) uses arbitrary cost functions including the log cost function for the upper bound above. The criterion for choosing one cost function over another is based on which one gives a tighter upper bound. On the other hand, in our case, the ML framework gives rise to a particular cost function that also happens to be a good choice under the boosting framework. Furthermore, the ML framework can be generalized to avoid over-fitting by regularization (see § 4.1.4), whereas there is no known regularization framework in boosting (however, see (Lebanon and Lafferty, 2001)).

Ignoring the choice of cost functions and the issue of regularization however, from a computational point of view there is no essential difference between the boosting framework and the ML framework. In fact, it has been recently shown that the ML framework for exponential models can be precisely related to the boosting framework by specifying a particular class of constraints in the maximum entropy formulation that is the dual of the maximum likelihood problem (Lebanon and Lafferty, 2001).

However, from a conceptual viewpoint, for our task we argue that the nearest neighbor framework is more natural than the boosting framework. In the boosting framework, the basic primitives are the simple (or "weak") classifiers that are combined. In the development above, the simple or base classifiers act on pairs of measurements outputting a label indicating whether they are in the same class or not, while typically one thinks of a classifier as acting on one measurement and outputting a class label. Furthermore, we are able to interpret the distance measure $[h(x_i) = h(x_j)]$ for each discriminator h as a pair-wise classifier only because of our choice of simple distance measures that give binary values $+1$ and -1 for a pair of input measurements. It is not clear whether more general classes of elementary distance measures that need not be binary can also be interpreted as pair-wise classifiers.

On the other hand, the nearest neighbor framework naturally leads to the consideration of optimal distance measures that are obviously defined on pairs of measurements. It was only after we assumed a particular discrete model induced by discriminators for the optimal distance measure, as well as using the maximum likelihood framework for estimating the parameters of such a model, that we were able to draw the connection to boosting. The connection would not have resulted had we either chosen to model distance measures differently (for example with a continuous model) or used a different parameter estimation framework. Viewed in this light, for our task the consideration of distance measures is motivated from first principles in a nearest neighbor framework, whereas casting the task in a boosting framework is only coincidental and contingent upon particular choices made during modeling and estimation.

Chapter 5

Generating Candidate Discriminators

The last chapter assumed a collection of candidate discriminators $\mathcal{H}$ from which $K \ll |\mathcal{H}|$ discriminators were chosen in a greedy manner under the maximum likelihood estimation framework. In this chapter, we discuss the details for generating such a collection of discriminators.

We present two approaches to generating discriminators. The first one presented in § 5.1 is a general approach that can generate candidate discriminators using any feature space like color, shape or texture in which some distance measure can be defined. The approach is more appropriate for coarse discrimination tasks for which gross feature differences are sufficient for discriminating different classes of objects of interest. However, this approach is computationally expensive primarily because the search space is discrete. The second approach presented in § 5.2 generates discriminators in linear feature spaces, for example pixel intensities in a window for which the distance measure between two points in this feature space is given by the Euclidean distance. The approach takes advantage of the linearity of the feature space to generate discriminators in a computationally efficient manner. Both approaches have been implemented and tested in Chapter 7.

5.1 Nearest Prototype Discriminator

As discussed in § 3.3.1, discriminators can be characterized by the partition they induce in image space. This relationship between discriminators and partitions works both ways. Given a partition, we can associate a discriminator with the partition. The discriminator classifies two measurements as belonging to the same class if they fall in the same partition, otherwise they are classified as belonging to different classes. Thus one approach to generating discriminators is to find ways of partitioning the image space, where each such partitioning corresponds to a discriminator.

Perhaps the simplest means of creating partitions is to specify the locations of some number of prototypes in some feature space with a distance measure. An example of a distance measure is the χ^2 distance for histograms (Schiele, 1997; Press et al., 1992). The Voronoi diagram induced by the prototypes and the distance measure in the feature space is a partition of image space. We call the discriminator associated with the Voronoi diagram a *nearest prototype discriminator*, which we first introduced in Chapter 3. The construction of the nearest prototype discriminator is similar in spirit to vector quantization in signal processing (Gersho and Gray, 1992).

Since a nearest prototype discriminator is completely specified by the number and locations of a set of prototypes in some feature space, we next discuss how the prototypes are generated.

Let us assume we are interested in constructing a nearest prototype discriminator with r prototypes in some feature space. For a continuous feature space, the set of locations for any one of the prototypes is infinite, thus the set of candidate nearest prototype discriminators $\mathcal{H}$ is also infinite. Recall from § 4.1 that under our greedy scheme, we seek the best discriminator from a set of candidate discriminators that minimizes the cost function J (4.11). However, if the set of candidate discriminators is continuous then efficiently searching for the best discriminator from such a set may not be feasible or maybe difficult in general since the possibility for performing an efficient search in a continuous space will depend on the distance measure used which may be nonlinear and also the parametrization used for measurements in the feature space. For example, if the feature space is histograms over some feature, then the parameters are positive real values, one for each bin of the histogram representation, and which are constrained to sum to one. A possible distance measure would be the nonlinear χ^2 distance (Schiele, 1997; Press et al., 1992). Searching for the best discriminator under such a parametrization is made difficult due to the huge space of parameters, one for each bin, and is further complicated by the nonlinear distance measure. For example, common search techniques that utilize some form of gradient descent over the parameter space are susceptible to getting trapped in local minima.

To overcome such issues and achieve the widest possible applicability for our approach, we will adopt a simple-minded approach in which we sample a discrete number of possible prototype locations from the feature space rather than search through all possible locations in the continuous feature space. This gives rise to a finite set $\mathcal{H}$ of candidate nearest prototype discriminators.

The simplest approach is to sample the parameter space of measurements in the feature space (for example, real values for each bin for histograms) uniformly. Another approach is to sample the same training set S that is used to estimate the optimal distance measure under the maximum likelihood framework. For r prototypes, the set $\mathcal{H}$ of all possible nearest prototype discriminators where each prototype is chosen from the training set S has size $|\mathcal{H}|r^{|S|}$. Exhaustively searching such a set for the best discriminator (that which minimizes J (4.11)) will not scale well if the size $|S|$ of the training set is large.

Instead, we will use a simple sampling technique that trades off the quality of the discriminator found for a speed-up in the search process. Rather than exhaustively searching over all nearest prototype discriminators that are possible from a training set, we will instead be satisfied with a discriminator that is among the top percentile of discriminators minimizing the cost function J. More precisely, say we want to find a discriminator that is in the top s percentile, that is if we rank all the discriminators according to how much the cost function J is minimized, then we want to find a discriminator such that no more than s percent of all possible discriminators have a lower cost J than the selected discriminator. We can show that with high confidence we can find a discriminator in the top s percentile by uniformly sampling the finite set of all possible discriminators $\mathcal{H}$ a fixed number of times n that is *independent* of the size of the training set $|S|$ and the number of prototypes r required. Our approach is similar in spirit to the RANSAC algorithm for the robust estimation of model parameters (Fischler and Bolles, 1981).

For $0 < \delta < 1$, we would like to find the number of samples n such that there exists at least one discriminator that is in the top s percentile with probability at least $1 - \delta$ (in other words, with confidence δ). Since each sample is drawn uniformly from the set of all discriminators, the probability that a given sample does not fall in any fixed fraction s of the set of all discriminators is $1 - s$. This is true irrespective of which fraction s is chosen. In particular, it is true when the top s percentile is chosen. Since each sample is chosen independently from each other, if n samples are drawn, the probability that none of them fall in the top s percentile is $(1 - s)^n < e^{-sn}$. Thus the probability that at least one of the n samples does fall in the top s percentile is greater than $1 - e^{-sn}$. Thus, at least one of the samples is in the top s percentile with probability $1 - \delta$ if $1 - e^{-sn} > 1 - \delta$. Thus:

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

For example, for $s = 0.1\%$ we need $n > 46$ samples to meet a confidence level of 99%, for $s = 0.01\%$ we need $n > 460$ samples to meet a confidence level of 99%. The above analysis for our sampling strategy is similar in spirit to that for the RANSAC algorithm (Fischler and Bolles, 1981).

Note that as stated before, we have shown that the number of samples n that meet a particular confidence δ neither depends on the size of the training set $|S|$ nor the number of prototypes r. However, the evaluation of the cost function J for each discriminator that is sampled does depend on the training set size and the number of prototypes.

5.2 Candidate Discriminators in a Linear Feature Space

The last section presented an approach for constructing candidate discriminators in a feature space with an arbitrary parametrization and distance measure. Even if the feature space is continuous, we noted that it might be difficult to use continuous optimization strategies to find the best discriminator that minimizes the cost function J.

In this section, we will consider linear feature spaces, for example pixel intensities in a sub-window of the image, where the distance between two measurements in this feature space is the weighted euclidean distance. Instead of sampling prototypes that are restricted to the training set as in the last section which results in the consideration of only a discrete set of candidate discriminators, we will instead construct "good" candidate discriminators where the search for such good discriminators is done over the whole continuous feature space. This global search is made possible due to the linearity of the feature space.

There will be one such candidate discriminator constructed for each linear feature space. These discriminators will be the set of candidates $\mathcal{H}$ for the greedy selection scheme presented in § 4.1.2. A discriminator is considered "good" if it satisfies the following criteria that are relevant to the task at hand:

I. Assume that a set of discriminators has already been selected by the maximum likelihood greedy scheme detailed in § 4.1.2. We want to choose a new discriminator that we would like to add to this set. A good discriminator should focus on classifying pairs of training measurements that have been difficult to classify using the previous discriminators selected by the greedy scheme so far. For a given set of discriminators $\{h_0, \ldots, h_k\}$, the probability that a pair of measurements x_i and x_j is mis-classified is given by:

$$w_{ij} \equiv 1 - p(y_{ij} \mid x_i, x_j) = \sigma(-y_{ij} \sum_k \alpha_k [h_k(x_i) = h_k(x_j)]) \qquad (5.1)$$

Thus in terms of the probability of mis-classification w_{ij}, we want to find a discriminator in the feature space that focuses on classifying pairs for which w_{ij} is high.

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

III. A good discriminator induces a partition such that the training measurements in the different partitions are separated well, while training measurements in the same partition are tightly clustered. This should make the discriminator more robust at run-time in deciding which partition a measurement falls under if the training set is representative of data to be seen at run-time.

The first two criteria deal only with the training set, while the last one is a heuristic criterion for finding a discriminator that generalizes well to future unseen data.

Our approach to finding a discriminator which satisfies the above criteria will be to encode them in an objective function that can be thought of as an unsupervised generalization of the well known Fisher quotient (Fukunaga, 1990). This objective function is minimized to find a linear discriminant, i.e. a hyper-plane in the linear feature space for which the projections of training measurements on the hyper-plane are maximally separated into two groups, while also satisfying the other criteria above. The linear discriminant along with an optimal threshold will form the desired candidate discriminator for the linear feature space under consideration. Unlike the traditional formulation of the Fisher criterion, 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 (criterion (I) above).

Formulating the Objective Function

For concreteness below, we assume an example feature space $\mathbb{R}^{m^2}$ of pixel intensities in a sub-window of size $m \times m$ in an input image. We would like to find a discriminant $\mathbf{l}$ in this feature space that satisfies the three criteria discussed above. One of the criteria (III) is to find a discriminant that partitions training measurements into two well-separated groups, each of which is tightly clustered. The rationale for this criterion 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 at run-time. In other words, we want to maximize:

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

If we know the optimal partition that satisfies the above criteria, then the optimal discriminant can be found by optimizing the Fisher discriminant quotient (Fukunaga, 1990). Let $\mathbf{v}_i$ be the representation of training image x_i in the continuous feature space (i.e., $\mathbf{v}_i \in \mathbb{R}^{m^2}$ in the example above). For a given partition of the training data, the Fisher quotient is usually formulated in the literature in terms of the first and second order statistics of the training data as follows:

$$F(\mathbf{l}) = \frac{||m^+ - m^-||_2}{{\sigma^+}^2 + {\sigma^-}^2}$$

where m^+, m^- are the means of the projections onto the discriminant $\mathbf{l}$ of the $\mathbf{v}_i$'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 two criteria discussed above. We denote a partition of the training images by indicator variables $\mathbf{s} = \{s_1, \ldots, s_n\}$ where each $s_i \in \{-1, +1\}$ indicates the partition that $\mathbf{v}_i$ belongs to in the feature space. The pair-wise formulation of the Fisher quotient that we use is then given by:

$$F(\mathbf{s}, \mathbf{l}) \quad = \quad \frac{\sum_{i,j}(1 - s_i s_j) K(x_i, x_j)}{\sum_{i,j}(1 + s_i s_j) K(x_i, x_j)} \tag{5.2}$$

where $K(x_i, x_j) \equiv \mathbf{l}^T(\mathbf{v}_i - \mathbf{v}_j)^T(\mathbf{v}_i - \mathbf{v}_j)\mathbf{l}$ is the separation along the discriminant hyper-plane $\mathbf{l}$ between training images x_i and x_j. Note that, as required, the term $(1 - s_i s_j)/2 \in \{0, 1\}$ is an indicator function that denotes when x_i and x_j are in different partitions, while $(1 + s_i s_j)/2 \in \{0, 1\}$ denotes when x_i and x_j are in the same partition.

In practice, we will have to determine *both* the optimal partition (i.e., a setting for $\mathbf{s}$ that optimizes eq (5.2)) as well as the optimal discriminant hyper-plane $\mathbf{l}$. This is an unsupervised mixed discrete-continuous optimization problem (discrete in $\mathbf{s}$ and continuous in $\mathbf{l}$). We derive an iterative solution for this optimization problem in the next subsection. Once the hyper-plane $\mathbf{l}$ is found, we can form a linear discriminant $h(x) = \text{sgn}(\mathbf{l}^T\mathbf{v} - \theta)$ where θ is the optimal threshold that separates the two partitions.

With the pair-wise formulation, it is now a simple matter to encode the other two criteria (I,II) into the optimization.

We can constrain the optimization of eq (5.2) 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 discriminators have been learned so far and let w_{ij} be the corresponding probability of mis-classification of a pair of images x_i and x_j by the k discriminators, as defined in (5.1). The pair-wise formulation of the Fisher quotient eq (5.2) can readily bias the optimization to focus on the hard to classify pairs of images, by weighting each term in the Fisher quotient by the corresponding probability of mis-classification. 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:

$$F(\mathbf{s}, \mathbf{l}) = \frac{\sum_{i,j}(1 - s_i s_j) w_{ij} K(x_i, x_j)}{\sum_{i,j}(1 + s_i s_j) w_{ij} K(x_i, x_j)} \tag{5.3}$$

Iterative Optimization

In practice, direct optimization of F 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. This approximation is similar in spirit to the normalized-cut approach for segmentation (Shi and Malik, 2000). With this relaxation, we propose an iterative maximization scheme, by alternating between maximizing F with respect to s keeping l fixed and maximizing F with respect to l keeping s fixed. We show below that each of these sub-problems leads to a corresponding generalized eigenvalue problem.

First, consider maximizing F 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 F can be simplified as follows:

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

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:

$$F(\mathbf{s}) \quad = \quad \frac{\mathbf{s}^T (D - W)\mathbf{s}}{\mathbf{s}^T (D + W)\mathbf{s}} \tag{5.4}$$

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, F is maximized when s is set to the eigenvector corresponding to the largest eigenvalue of the generalized eigenvalue problem $(D - W)\mathbf{s} = \lambda_s (D + W)\mathbf{s}$.

Next, we maximize F with respect to l while keeping s fixed. Define the matrices:

$$\begin{aligned} A &\equiv \sum_{i,j} (1 - s_i s_j) \sum_{k_i,k_j} w_{k_i k_j} (\mathbf{v}_{k_i} - \mathbf{v}_{k_j})(\mathbf{v}_{k_i} - \mathbf{v}_{k_j})^T \\ B &\equiv \sum_{i,j} (1 + s_i s_j) \sum_{k_i,k_j} w_{k_i k_j} (\mathbf{v}_{k_i} - \mathbf{v}_{k_j})(\mathbf{v}_{k_i} - \mathbf{v}_{k_j})^T \end{aligned}$$

with k_i and k_j defined as before. With these definitions, F can be simplified to:

$$F(\mathbf{l}) \quad = \quad \frac{\mathbf{l}^T A \mathbf{l}}{\mathbf{l}^T B \mathbf{l}} \tag{5.5}$$

Once again, F is maximized when l is set to the eigenvector corresponding to the largest eigenvalue of the generalized eigenvalue problem $A\mathbf{l} = \lambda B\mathbf{l}$.

Figure 5.1 summarizes the iterative scheme. We alternate between maximizing F w.r.t. s and l by solving for the corresponding eigenvector problems, until convergence. Although the iteration is guaranteed to increase F 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 $\mathbf{v}_i$ for some k that is fixed a priori, then initialize $\mathbf{l}$ to each of these principal components and optimize using the iterative scheme just described and choose the hyper-plane $\mathbf{l}$ among them that maximizes F. Note that the optimal partition $\mathbf{s}$ is not required for the rest of the scheme.

Let $\mathbf{u}_1, \ldots, \mathbf{u}_k$ be the first k PCA components of the set of feature vectors $\mathbf{v}_i$ corresponding to training images x_i.

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

I. Set $\mathbf{l} = \mathbf{u}_i$.
II. Iterate between the two eigen-problems $(D - W)\mathbf{s} = \lambda_s(D + W)\mathbf{s}$ and $A\mathbf{l} = \lambda_l B\mathbf{l}$ until convergence to $\mathbf{s}_i, \mathbf{l}_i$.
III. Set $F_i = F(\mathbf{s}_i, \mathbf{l}_i)$.

Output $\mathbf{l}_i$ corresponding to max F_i.

Figure 5.1: Pseudo-code for finding optimal discriminants

Figure 5.2 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 converges to a good 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 on average more 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 that maximizes the Fisher quotient.

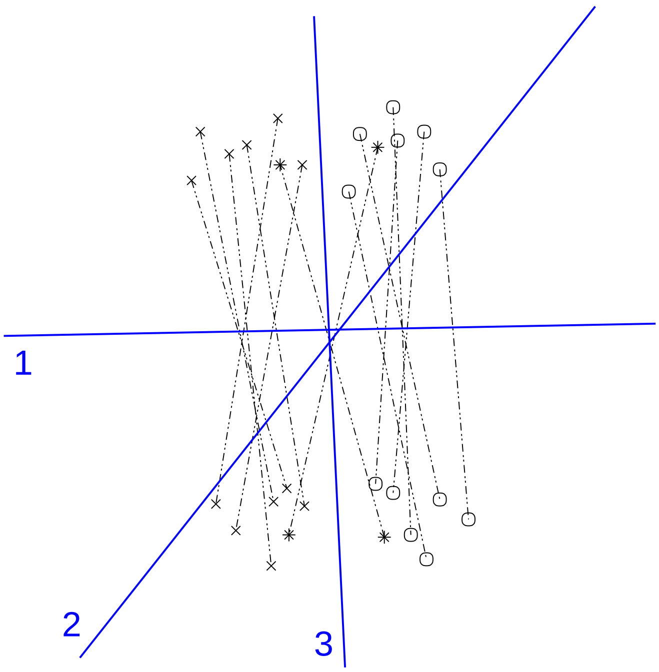


Figure 5.2: Synthetic example in a continuous 2D feature space illustrating the iterative algorithm for finding optimal discriminants. The numbers in the figure refer to the iteration number. The final partition found is denoted by $\bigcirc$ and $\times$.

Choosing θ. Finding the optimal threshold θ is a one-dimensional problem along the discriminant hyper-plane $\mathbf{l}$, for which we use a simple brute-force search. The optimal value for θ is that which minimizes the total cost function J (4.11). The total cost as a function of θ changes only when θ crosses a vector $\mathbf{v}_i$ projected onto $\mathbf{l}$. Accordingly, we determine θ as follows: sort the projections onto the optimal $\mathbf{l}$ of all the $\mathbf{v}_i$'s, find the total cost J 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.

Chapter 6

Implementation

In this chapter we will discuss several issues that are important for a practical and efficient implementation of our approach. In § 6.1 we discuss the various feature spaces that we use in our work. In § 6.2 we describe the need for decomposing an object view in terms of a set of parts. We present our approach for selecting a set of such parts that are good for the discrimination task at hand. We noted in § 3.3 that even though a discrete model to the optimal distance measure will in practice be less accurate than a continuous model, the discrete distance model can still be useful in practice if it permits the possibility of efficiently narrowing down the set of possible neighbors to an input measurement. This set can then further be pruned by the more accurate continuous distance model. In § 6.3 we discuss how to compose the discriminators that are used in forming the discrete distance model in a tree-like structure for efficient run-time performance. Finally § 6.4 describes in detail the overall scheme for detecting objects of interest in an input image. The scheme first detects candidate parts at various locations in the input image using the nearest neighbor classifier, then accumulates support for each candidate part from other parts that are consistent with the candidate part, and finally performs local non-maximal suppression.

6.1 Feature Spaces

As discussed in § 3.2, in our work we approximate the optimal distance measure by a linear combination of elementary continuous or discretized distance measures in simple feature spaces based on color, shape and texture. In this section, we describe the details of the types of feature spaces that we use in our experiments.

The histogram of various image feature types is a widely used feature space in computer vision (Schiele, 1997; Swain and Ballard, 1991; Worthington and Hancock, 2000; Schneiderman, 2000; de Bonet et al., 1998; Comaniciu et al., 2000). In our work, we consider the histogram of continuous feature types. Each such feature type can be multi-dimensional. For example, color is typically expressed in terms of three bands (red, green and blue, or equivalently hue, saturation and value). Formally, a histogram is a discretization of a probability density $p(f|x)$ for a feature type f in an image x. In the simplest such discretization, each dimension of the feature type f is discretized into a fixed number of bins. For example, in our work we choose to discretize the color of a pixel into 8 levels for each of the three color bands - red, green and blue. The histogram is then constructed by finding the frequency count of the pixels in the input image with a feature value that falls in each bin. In our work, we use a 32×32 pixel window of support, centered around the point of interest in the input image for constructing the histogram.

Two observed histograms C_1 and C_2 of the same feature type can be compared using various distance measures. For example, the χ^2 distance is defined by:

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

where b runs over the set of bins in a histogram for the particular feature type under consideration. Another distance measure is the simple L_1 distance. Both the above distance measures can be related to the KL distance measure between two distributions. The χ^2 is a quadratic approximation to the KL distance, while the L_1 is an upper bound for the KL distance (Cover, 1991). Yet another distance measure between two observed histograms is the *intersection* distance (Swain and Ballard, 1991):

$$\bigcap(C_1, C_2) = \sum_{b \in \text{bins}} \min\{C_1(b), C_2(b)\}$$

For a performance comparison of some of these distance measures, see (Schiele, 1997). In our work, we use the simple L_1 distance for run-time efficiency.

Histograms are popular in the computer vision literature since they are typically efficient to create from an input image as well as being robust to a fair amount of geometric transformations (Schiele, 1997; Swain and Ballard, 1991; Comaniciu et al., 2000). On the down side, histograms based on a single feature type cannot be expected to be discriminative enough for all objects of interest. Recently, multi-dimensional histograms have been shown to be highly discriminative (Schiele, 1997; Rao and Ballard, 1995). However, they typically require a large support window for reliable estimation (the "curse of dimensionality" (Schiele, 1997; Duda et al., 2001; Bishop, 1995)) and are expensive to compute at run-time.

In our work, for run-time efficiency considerations, we consider only very low-dimensional (1- or 2-dimensional) histograms. As noted above, each such low-dimensional histogram will in general be insufficient for the discrimination task at hand. Our approach gets around this short-coming by combining the discriminative power of several such low-dimensional histograms. More precisely, under the nearest neighbor framework, we assume a linear combination model for the optimal distance measure in terms of a set of elementary distance measures, each of which is defined on histograms constructed in a particular low-dimensional feature space.

In our work, we also utilize the spatial location of the feature to further improve the discriminative power of low-dimensional feature spaces. Typically, the spatial location of the feature in the support window is ignored when constructing a histogram. We encode crude spatial information by discretizing the spatial location. In other words, the histogram that we use is a discretization of the probability density $p(f, l|x)$ over the joint space of both the feature type f and its location l (specified by the coordinates of the pixels in the support window) with respect to the center of the support window around the point of interest in an input image x. This is similar in spirit to the work on shape context (Belongie et al., 2002). In our work, we choose a 2×2 discretization of the spatial locations, centered around the point of interest in the input image.

We conclude this section by listing all of the specific feature types that we use in our work. The feature types belong to the following three categories:

Color Three single dimensional feature spaces are considered corresponding to the red, green and blue bands. Each band is first normalized by the average value in the support window. Each band is then discretized into 8 bins.

Texture A simple characterization of texture is in terms of the Gaussian derivative filter responses (Schiele, 1997; Viola, 1995; Greenspan et al., 1994). Specifically, we first convolve the image with the Gaussian derivative kernels g_x, g_y along the two coordinate axes. We use the Deriche implementation for the convolution (Deriche, 1992) with the width of the Gaussian set to $\sigma = 2.0$ pixels. Each Gaussian derivative gives us a single dimensional feature space. Additionally, we use the magnitude of the derivative $g_x^2 + g_y^2$. For characterizing textures in an efficient as well as rotation-invariant manner, see (Greenspan et al., 1994).

Local Shape Lastly, we consider histograms of local shape properties. First, contours are detected by using the Canny edge detector followed by contour growing using hysteresis (Canny, 1986). Two types of local shape properties are considered. The simplest is the orientation of the edges (on the contours) that falls within the support window. The orientation is discretized into 6 directions, 30° apart. We also estimate the local curvature at each edge point that fall within the support window. A simple estimate can be obtained at each edge point that is not at the boundary of a contour from the orientations at the edge and its two neighbors in the contour containing the edge. This estimate is discretized into 6 levels.

6.2 Decomposition into Parts

An important issue in constructing the feature spaces described in the previous section is the optimal size and shape for the support window. Ideally, the support window should cover the whole of the object. Since at run-time the object of interest is not known, we will then need to choose an optimal size for the support window that can be used for all objects of interest. Since different objects will in general have different shapes and thus different optimal sizes for the support windows, choosing one size to fit all cannot be expected to perform well in practice. Any one choice for the size will likely be too small for some objects or too big for other objects for which some of the background will be considered along with the object of interest. Also different objects will have different shapes and thus no one shape for the support window will be optimal.

One approach to overcoming the above shortcoming is to decompose each object of interest into a set of parts such that each part has a support window that is entirely or mostly contained within the object of interest. The support window for each part need not be as large as when only one support window is used for the whole object. Furthermore, with such an approach, even non-convex objects can be reasonably covered with a set of parts, see Figure 6.2 for an example. Decomposing an object into parts and using both the part matching scores as well as their spatial configuration for object detection has been quite well-studied in the literature (Weber et al., 2000; Burl et al., 1998; Leung et al., 1995; Schneiderman, 2000; Viola, 1995).

Another important reason for using a part decomposition is to enable object detection that is robust to detection failure or partial occlusion. A detection scheme that does not depend on the detection of all the parts, but instead requires the detection of only some fraction s of the parts will be robust if the detection failure or partial occlusion only affects at most $1 - s$ fraction of the parts. We will describe such a scheme in more detail in § 6.4.

In the rest of the section we discuss several important issues that arise when decomposing an object training image into parts.

6.2.1 Part Classes

Instead of performing a nearest neighbor search over whole object training views, we perform a nearest neighbor search over object part training views. We first define our notion of a *part class*. Conceptually, a part class corresponds to image measurements of some surface patch of an object of interest, taken under differing viewpoints and lighting conditions, just as in the case for whole object classes.

For our purposes, training images for a part class are obtained as follows. First we assume a sample view of the part class is given, which we refer to as the "center" view (see Figure 6.1). This view corresponds to some surface patch of an object of interest and is selected from a training view of the whole object. The next subsection will detail how such views for each part class are selected.

We sample additional training views of this surface patch as follows. We can easily sample new training views under small translations, rotations or scalings from the original whole object view from which the center view was selected. For translations, we extract 4 new training views that are ± 4 pixels from the center view along either coordinate axis. For rotations and scalings, we first create new image views by geometrically transforming the original object image view under a set of rotations and scalings with linear interpolations of pixel values and then sample new training images for the part from the transformed locations of the center view. We consider rotations of $\pm 10^\circ$ and scale variations of 0.9 and 1.1. See Figure 6.1 for an example of a "center" training view of a part class along with corresponding training views obtained under the transformations discussed above.

Ideally, we would also like to sample training images of a part under viewpoint changes in depth. In principle, we could extract them from additional object views around the object view containing the "center" view of the part. However, unlike the case for rotations, translations and scalings, we cannot easily determine the expected location of the part view under viewpoint changes. One way around this difficulty would to search additional images for parts that are most "similar" to the center view of the part. This requires a distance measure and a threshold. The optimal distance measure is of course one that ignores within part variations. However, we are then faced with a chicken-and-egg problem. Furthermore, part views may not be detectable due to self-occlusion and modeling errors.

We get around these difficulties by adopting the following simple approach. We select a set of parts (the selection criteria is discussed in the next section) and model variations in translation, rotation and scale as discussed above, *independently* for each whole object training view. For neighboring whole object training views, it is possible that the same underlying surface patch is represented by different part classes selected in each of the whole object training views. If we had a reliable means of detecting such corresponding part classes, we would of course want to group all the part training views in all those part classes as training views for a single part class. Instead, we avoid this correspondence problem which is difficult to perform in practice, by letting each underlying surface patch to be represented by a *redundant* number of part classes, one for each training view in which the surface patch is visible. The down-side to this simple approach is the extra storage space required for the redundancy and the fact that viewpoint variations in depth for a surface patch are not taken into account when estimating the optimal distance measure for the nearest neighbor rule.

6.2.2 Part Selection

For run-time considerations, it is desirable to decompose an object view into only a few parts. One criterion for choosing a particular part should be its discriminative power. Let S_z be the training set for some part class z. The training set is chosen as described in the previous subsection. Let $\bar{S}_z$ be a random sample of training views of parts that do not belong to the same object class as z. Then a natural measure for the discriminative power of a part view is the log-likelihood $l(S_z, \bar{S}_z)$ that a view from S_z and $\bar{S}_z$ belong to different classes:

$$l(S_z, \bar{S}_z) \equiv \frac{1}{|S_z||\bar{S}_z|} \sum_{z_i \in S_z, z_j \in \bar{S}_z} \log p(y_{ij} = -1 \mid z_i, z_j) \tag{6.1}$$

We model the pair-wise distribution $p(y_{ij} \mid z_i, z_j)$ using a linear continuous model for its logit transform, i.e. we use the continuous linear model for the optimal distance measure $H(z_i, z_j)$ (see § 3.2). A global continuous linear model of $H(z_i, z_j)$ is estimated under the maximum likelihood framework (see § 4.1) from a random training sample of part classes from all whole object training views.

Two part classes that are very discriminative but whose underlying surface patches overlap on the object will be redundant for the discrimination task. Thus a second criterion that we use for selecting good parts is to choose parts that are "non-overlapping". In addition to the fact that such parts will have discriminative powers that are not redundant, such a part selection scheme will lead to a detection scheme that is more robust to occlusion. In our work, we select parts at two different scales (see 6.2), the original scale of the training images and a lower resolution scale that is $1/2$ the original scale. The non-overlapping condition is imposed only within each scale, not across scales. This is because two parts from the same location but at different scales can have non-redundant discriminative power.

We use a greedy scheme for selecting a set of parts from a whole object training view that satisfies the above two criteria at two different scales. First, for each scale, the set of all possible parts that are valid candidates are constructed from the object training view, sub-sampled every 4 pixels along both coordinate axes. A part is a valid candidate for the selection scheme if more than 80% of its support covers the object view rather than the background. For the purpose of determining the valid candidate parts, the training images are manually segmented into object and background.

Each of these candidate parts in both scales are scored by the log-likelihood score defined in (6.1). At each iteration, we select the part that has the highest score across both scales and which do not "overlap" with the parts selected in the previous iterations. We consider two parts as "overlapping" each other if their supports intersect by more than 50%. In our work, we select up to 10 such non-overlapping parts. See Figure 6.2 for the final set of parts selected for sample training images.

6.3 Efficient Composition of Discriminators

As discussed in $\S$ 3.3.1, we have chosen to discretize the optimal distance measure using a linear combination of distance measures associated with the partitions in image space induced by simple discriminators. As mentioned in that section, this choice permits the possibility of coarse, but efficient, nearest neighbor search at run-time that yields a small list of candidate neighbors that can be further pruned by the more accurate, but computationally expensive, continuous model for the optimal distance measure. Efficient search is possible if we select discriminators such that they can be organized into an efficient tree-like structure. In this section, we detail our approach for composing discriminators in such a structure.

6.3.1 Alternating Trees

For composing the discriminators into an efficient structure, we adapt the work on "alternating trees" (Freund and Mason, 1999) which is a generalization of decision trees (see Fig. (6.3)). This is also similar in spirit to "option trees" (Buntine, 1993). 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:

Partition Nodes: which indicates the subset of the image space $U \subset X$ that reaches the node after passing through the sequence of discriminator nodes from the root to the partition node. We can think of the rest of the image space $X - U$ as the subset of image space that the partition node "abstains" from.

In the original presentation of alternating trees in (Freund and Mason, 1999), these were called "predictor nodes", but we prefer the more instructive term of "partition nodes" for our task.

Discriminator Nodes: are children of partition nodes and that correspond to discriminators that partition the subset of image space associated with the parent partition node.

The root node of the whole alternating tree is a partition node associated with the entire image space X. A partition node can have a multiple number of discriminators as children. In turn, a discriminator node has partition nodes as children, each of which corresponds to one of the subsets of the image space in the partition induced by the parent discriminator node.

The possibility of partitioning the subset of image space associated with each partition node by a possibly multiple number of discriminators gives the alternating tree more flexibility and redundancy compared with standard decision trees. The standard decision tree is recovered if the alternating tree is constrained to have at most one discriminator node as a child for each partition node in the tree and collapsing each partition node with its sole discriminator child (if any). The redundancy in the alternating tree leads to more robustness at run-time compared with decision trees since an input leads to multiple paths from the root to leaf nodes unlike in decision trees where only one path is possible. An error at any point along the single path of a standard decision tree leads to the wrong result, whereas an alternating tree can recover from a few errors due to its reliance on multiple paths.

6.3.2 Trees and the Linear Distance Model

In § 3.3.1 we discussed a discrete model for the optimal distance measure in terms of elementary distance measures corresponding to simple discriminators (3.11). On first thought, it might not seem that we can incorporate the simple discriminators composed in an alternating tree into a linear model since the discriminators in a tree have dependencies on each other. However, recall that the only manner in which a discriminator h_k enters into the linear model (3.11) is through the elementary distance measure $[h_k(x) = h_k(x')]$ associated with the partition of image space induced by the discriminator h_k. The binary distance measure indicates whether two image x, x' belong to the same partition induced by h_k (i.e. $[h_k(x) = h_k(x')] = -1$) or belong to different partitions (i.e. $[h_k(x) = h_k(x')] = +1$).

In an alternating tree, a discriminator h_k only partitions the subset of images U that reaches its parent partition node. Clearly, the distance measure $[h_k(x) = h_k(x')]$ can be defined as before if its domain is restricted to pairs of images $(x, x') \in U \times U$. Our approach to incorporating discriminators in an alternating tree is to extend the domain for the distance measure $[h_k(x) = h_k(x')]$ to all of the image space $X \times X$.

Accordingly consider the case when either or both of x and x' belong to $X - U$, that is the images belong to the subset of image space that the discriminator h_k "abstains" from. First, let both $x, x' \in X - U$. How should $[h_k(x) = h_k(x')]$ be defined ? As far as the discriminator h_k is concerned, both x and x' cannot be discriminated by h_k, thus we should let $[h_k(x) = h_k(x')] = -1$. On the other hand, if one of the image measurements belong to U while the other belongs to $X - U$, then the pair can be considered to be discriminated by h_k and thus we should let $[h_k(x) = h_k(x')] = +1$. Put another way let $U = \{U_1, \ldots, U_l\}$ be the partition induced by h_k on U, then the above extension of $[h_k(x) = h_k(x')]$ to all image space is the same as defining a distance measure on the *extended* partition $X = \{X - U, U_1, \ldots, U_l\}$ over the whole measurement space.

6.3.3 Building the Tree

We end this section by describing how an alternating tree of discriminators is built at training time. Recall from § 4.1 that we want to select K discriminators in a greedy manner from a given collection of candidate discriminators $\mathcal{H}$ under the maximum likelihood framework, or more specifically we want to select the K discriminators $h_k \in \mathcal{C}, k = 1, \ldots, K$ that minimize the cost J (4.11). Each candidate discriminator in $\mathcal{H}$ is constructed in some feature space by either of the procedures (the nearest prototype discriminator or fisher like discriminator) outlined in Chapter 5 using a training set S.

The above greedy selection scheme for choosing discriminators remains largely unchanged in the context of building an alternating tree, but with important differences. At any iteration of the greedy scheme, let us assume that we have built some alternating tree that contains the discriminators selected so far in the previous iterations. At the current iteration, we have a choice of adding a new discriminator to *any* partition node in the alternating tree (recall that in an alternating tree, a partition node can have multiple discriminator nodes). The candidate discriminators available for each partition node P_i is constructed using the procedures in Chapter 5 in various feature spaces as before but trained on only the subset of training examples $S_i \subset S$ reaching the partition node P_i.

The greedy scheme for building the alternating tree is outlined in Figure 6.4. The alternating tree is initialized to a partition 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. As before, let $S_i \subset S$ denote the subset of training examples that reach the partition node P_i in T, and let $\mathcal{H}(S_i)$ denote the set of candidate discriminators available to partition node P_i using the procedures for constructing discriminators in Chapter 5 and the training set S_i. At iteration k, we choose the discriminator h^* that minimizes the objective function J (4.11) from among the set of all candidate discriminators $h \in \bigcup_i \mathcal{H}(S_i)$ over all choices of training sets S_i associated with each partition node P_i in the tree. This discriminator h^* is added to the tree as a child of the partition node P_i for which h^* came from the corresponding set of candidate discriminators $\mathcal{H}(S_i)$. Note that since a partition node P_i can have multiple children, each partition 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 combining coefficients α_k.

6.4 Tying it all Together

In this final section, we will walk through our scheme for detecting objects of interest in an input image. Figures 6.5- 6.6 are the accompanying illustrations for the following discussion.

An object of interest might be present at any location in the input image. Attentional mechanisms or interest operators have been used in the literature (Grimson et al., 1994; Burt, 1988; Abbott and Zheng, 1995; Westlius et al., 1996; Grove and Fisher, 1996; Stough and Brodley, 2001; Culhane and Tsotsos, 1992; Itti et al., 1998; Baluja and Pomerleau, 1997; Tomasi and Shi, 1994; Ruzon and Tomasi, 1999; Mikolajczyk and Schmid, 2002) for focusing on those locations in the input image that might correspond to an object of interest. These locations are then further analyzed for the possible presence or absence of an object of interest. Such techniques for narrowing down the set of all locations to a manageable number is necessary since typically the object detection procedures are computationally expensive.

However, the state of the art for such attentional mechanisms leaves much to be desired and is beyond the scope of this thesis whose main focus is on the principled formulation and the various issues involved in developing an efficient nearest neighbor framework for object detection. Instead, for simplicity we adopt a more "brute-force" approach where we sub-sample all possible locations in the input image and classify the sub-image at each location. Such a brute force approach has been successful in certain restricted domains like face detection (Rowley et al., 1998; Schneiderman, 2000; Viola and Jones, 2001). Good run-time performance with current compute power using such a brute-force strategy is possible whenever the detection process for an object of interest at each location is reasonably cheap. In our case, the hierarchical nearest neighbor search scheme presented in § 3.3 leads to such an efficient object detection scheme. Nevertheless, any reliable attentional mechanism can complement such a naive brute-force approach and will only improve the run-time performance.

Accordingly, for our experiments reported in the next chapter we chose to sub-sample locations in the input image along both coordinate axis every 4 pixels. We could in principle also choose to sample rotations in the image plane along with some amount of scale at each sampled position but instead we employ an alternate strategy, which is to expand the training set by adding rotated and scaled versions of each training image. Thus we trade-off training time for improved run-time performance.

In the rest of the section, we describe each step in detail for detecting the presence or absence of an object of interest.

Pre-processing. The various features mentioned in § 6.1 are extracted from the input image. Histograms at each of the sampled locations (along both coordinate axes as well as at two scales) are constructed for each feature type. We have chosen to use histograms of various feature types precisely because they can be constructed at each location of the input image efficiently by making one pass from left to right and from top to bottom for each scale that is sampled. Such a scheme is applicable for any desired quantity like simple moments of feature values (averages, variances) whenever the quantity is a function of only the feature values but not its position in a support window. See (Viola and Jones, 2001) for similar applications of such a scheme. For completeness, we describe the scheme for efficiently constructing histograms in more detail below.

Consider a location x in the input image at which we assume that the histogram $C(x)$ for some feature type has already been constructed. The histogram $C(x + dx)$ for the same feature type at any of the neighboring positions $x + \Delta x$ along either of the coordinate axes can be computed by updating the histogram at x with only feature values from the appropriate leading and trailing strips at the border of the support window for the histogram, as illustrated in Figure 6.5. Thus with appropriate initializations, all the histograms can be efficiently constructed in a single sweep from left to right, and top to bottom.

NN Part Detection. Once the histograms have been constructed, object parts are detected at each location by the hierarchical nearest neighbor search described in § 3.3. As described there, the hierarchical scheme first utilizes a discrete distance model based on discriminators organized in a tree like structure (see § 6.3). This discrete distance model is not very accurate in practice but is efficient to compute, thus it is used to search for a short list of K_d possible neighbors that is further refined in the next stage. Obviously the longer the list, the more likely the true nearest neighbor is within the list. See Chapter 7 on how classification performance depends on K_d.

The next stage further prunes this list of K_d neighbors using the more accurate but expensive to compute continuous distance model. Once again, we do not find just the nearest neighbor but instead report a shorter list of $K_c < K_d$ nearest neighbors for the next step which accumulates scores for whole object hypothesis formed from each of the K_c parts. Figure 6.6, step 1 shows the first 5 nearest neighbors found by the hierarchical distance measure at a few sample locations in an input image.

Object Detection. Each part detected at each location is used to form a hypothesis for an object training view that is closest to the view of the object in the input image. A score is accumulated for the hypothesis from the scores for all the parts from the same object training view as described later. A part detection at a given location generates a hypothesis for an object training view as follows. Recall from $\S$ 6.2 that each part class is formed from some training view of an object of interest. Thus it is natural to hypothesize the presence of the same object viewed under conditions similar to that of the training view from which the part class was formed. If the hypothesis is true, then the other parts from the same training view can also be expected to be found in the input image whose locations can be predicted from their locations in the training image and the scale and location of the detected part that generated the hypothesis, see Figure 6.6, steps 2 and 3.

These predicted locations are searched for the other parts from the training view. For robustness against some viewpoint changes as well as some modeling error in assuming rigid object classes, the predicted parts are searched in a small window around the corresponding predicted locations. The distance scores from the nearest neighbor search of the predicted parts that are found at the expected locations are accumulated to form the score for the hypothesis. *Crucially*, for robustness against occlusion and/or false negatives while finding the predicted parts, we only accumulate the scores of a pre-determined number of the topmost parts ranked by their scores, including the score of the parts that generated the hypothesis. In our experiments we have a total of up to 10 parts for each training view of an object class, and we choose to score each hypothesis with the 5 topmost parts detected in the input image. Thus our scheme is robust to occlusion or false negatives that affect up to 5 parts.

Thresholding. The scores for all the hypotheses are thresholded (see the experiments in the Chapter 7 for the dependence of the classification performance on varying thresholds). Finally, non-maximal suppression is performed to remove any hypotheses that have lower scores than any other hypothesis that is spatially overlapping. The spatial extent of an object class hypothesis in the input image is estimated from the extent of the object in the training view corresponding to the hypothesis and the location and scale of the hypothesis in the input image. The final output contains one or more object class detections with corresponding scores.

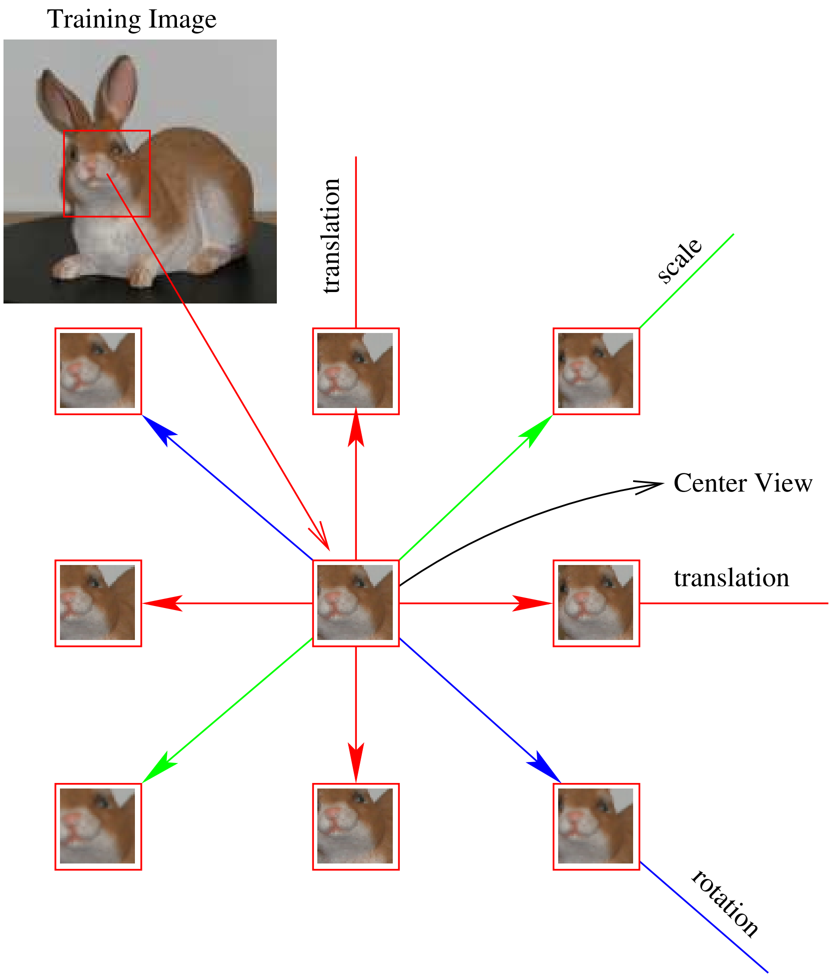


Figure 6.1: Example of a part class formed from a training image. A "center" view of the part class is selected from the training image as detailed in 6.2.1. Additional training views of the part class are sampled from the training image by translating, scaling and in-plane rotation of the part. Viewpoint changes due to rotation in depth are not modeled in a part class. Instead, the same underlying surface patch is redundantly represented by multiple part classes in different training images. See the text for details.

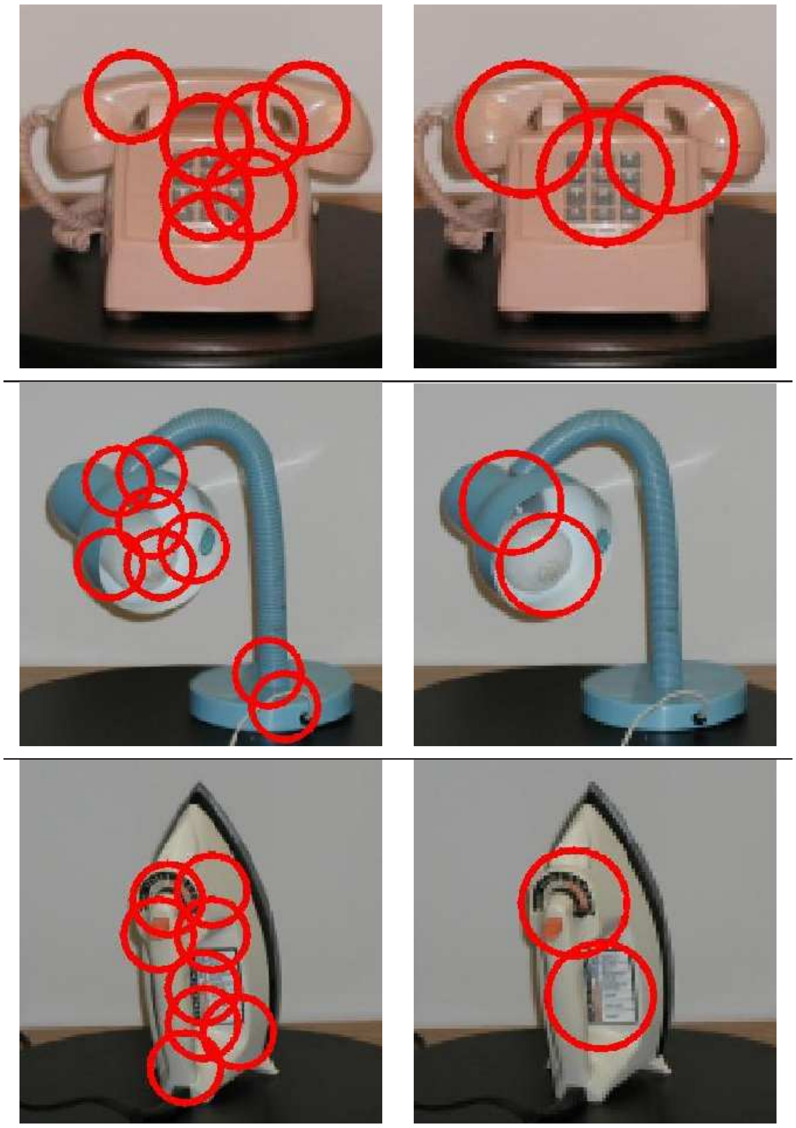


Figure 6.2: Optimal selected parts for sample training images. In our work, parts can be selected at two different scales. The left column shows parts selected from the original scale, while the right column shows parts selected from $1/2$ the original scale, back-projected to the original scale for ease of illustration.

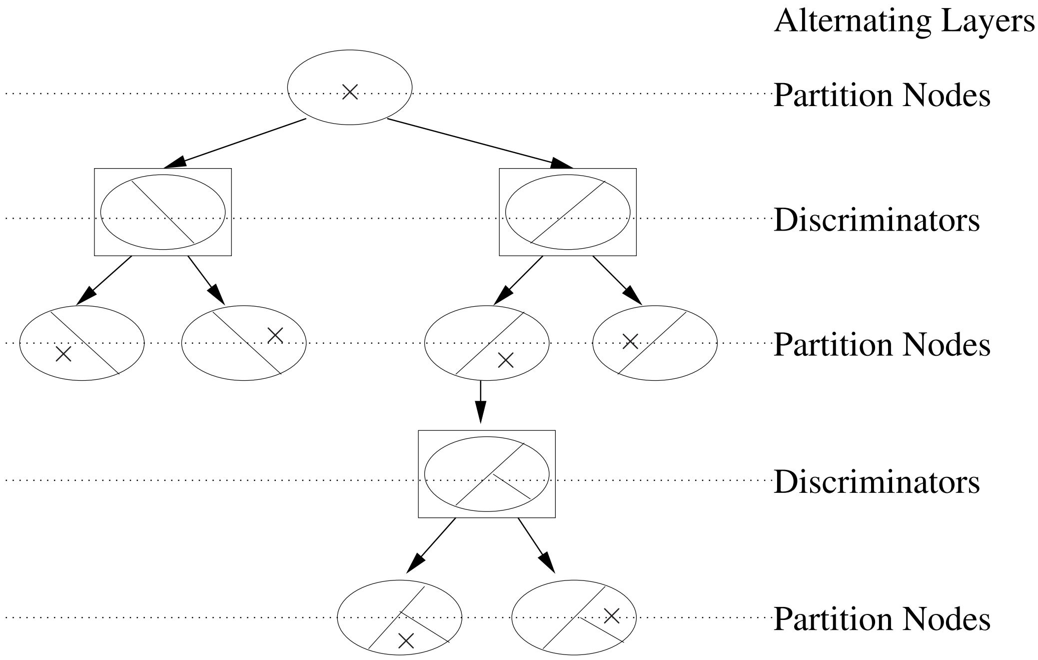


Figure 6.3: Alternating Trees. The tree alternates between partition nodes (ellipses) and discriminator nodes (boxes). Each partition node is associated with a subset U of the image space (marked by $\times$) that reaches the node through the sequence of discriminator nodes from the root to the node. Each partition node can have multiple discriminator nodes as children, each of which partitions the subset U of image space associated with its parent partition node.

Initialize:

I. Initialize the alternating tree T with a root partition node.

II. Let $\mathcal{H}(S_i)$ denote the set of candidate discriminators constructed from the training set $S_i \subset S$ that reaches a partition node P_i from the root.

do for K iterations

I. Find the discriminator $h^* \in \bigcup_i \mathcal{H}(S_i)$ that minimizes the cost function J (4.11).

II. Add h^* to the alternating tree T as a child of the partition node for which $h^* \in \mathcal{H}(S_i)$.

Figure 6.4: Pseudo-code for building the alternating tree.

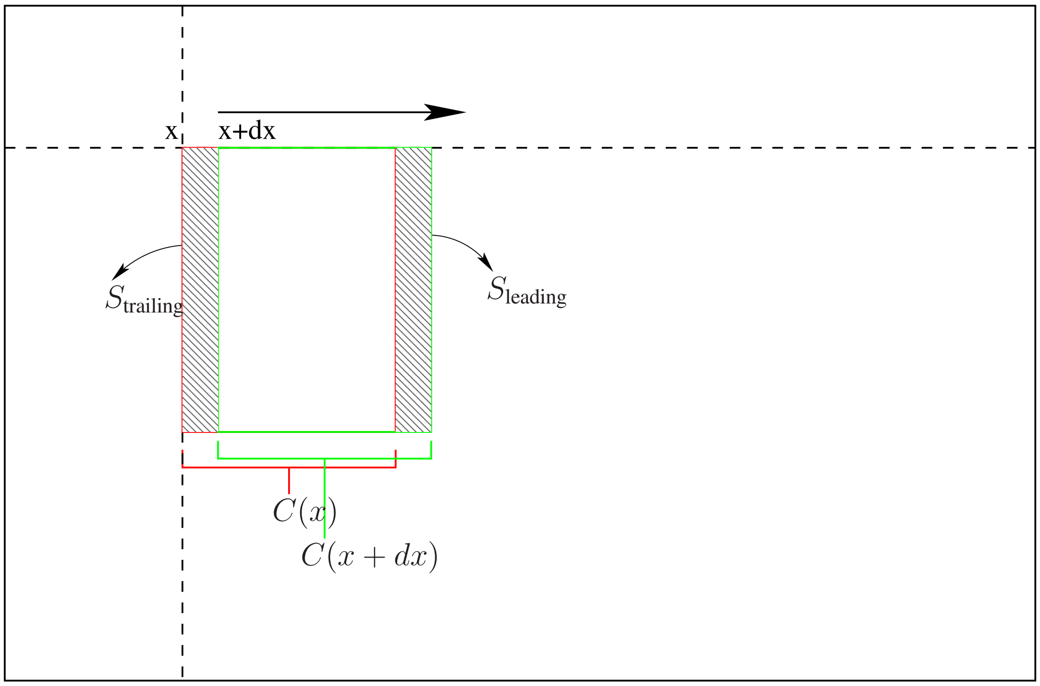


Figure 6.5: Constructing histograms efficiently. Assume that the histogram $C(x)$ for some feature type has already been constructed at location x. The histogram for $C(x + dx)$ at a neighboring location $x + dx$ can be efficiently computed from $C(x)$ and the histograms in the leading strip S_{leading} and trailing strip S_{trailing}.

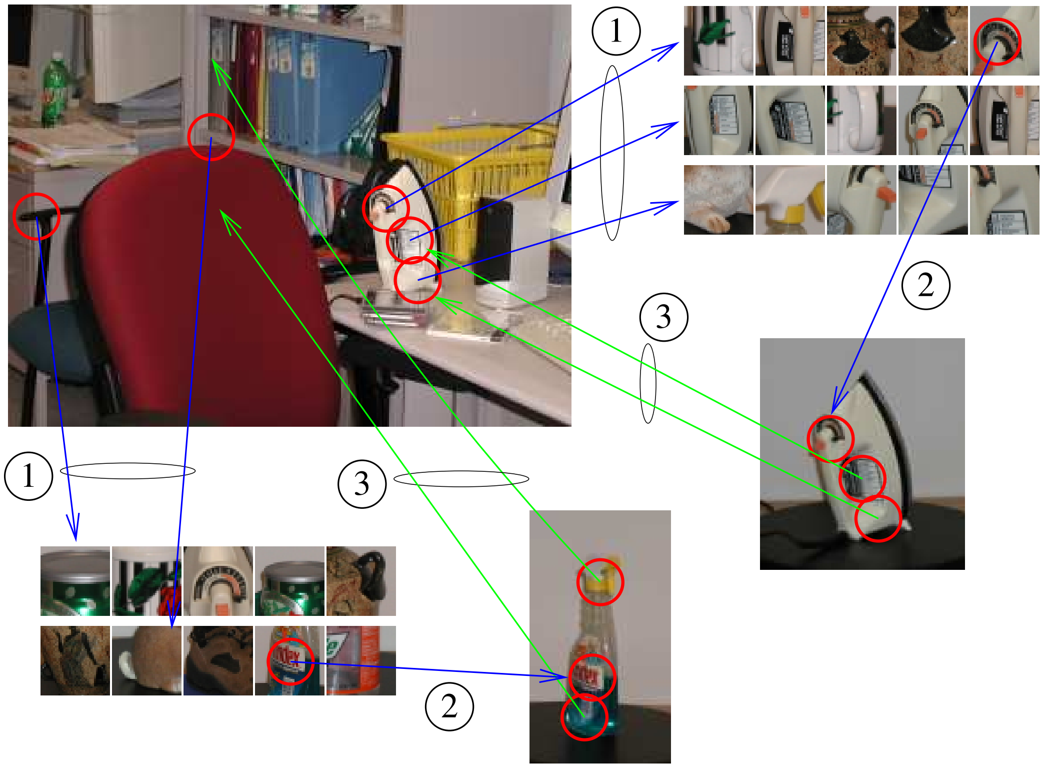


Figure 6.6: Illustration of our detection scheme on an actual test input (see § 7.1). Step (1): After pre-processing the image to extract histograms of various features, the nearest neighbor parts from the training set are determined at each sampled location using the hierarchical distance measure (§ 3.3). Shown here are the top 5 parts for a few locations. Step (2): Each part forms an object training view hypothesis. Step (3): The locations of the other parts in the training view for each hypothesis is determined and the corresponding parts are searched in the input image. The hypothesis is scored by accumulating the NN scores of these parts along with NN score for the part that generated the hypothesis. Shown here are 2 object view hypotheses formed from parts detected at two locations. In the actual system, each part detected at each location forms a hypothesis, each of which is scored. Finally, object detections are reported after thresholding the scores for the hypotheses and performing non-maximal suppression.

Chapter 7

Experiments

Most of this chapter will be devoted to the investigation of the classification performance of our detection scheme for a collection of everyday objects in an indoor environment. In addition, we will also present results on a difficult face recognition task.

Section 7.1 introduces the indoor detection task where we have a collection of 15 objects of interest. Recall from $\S$ 6.4 that we use a hierarchical nearest neighbor search for detecting parts at each sampled location in an input image, in which we first use a tree-based efficient but coarse discrete distance model to determine a short list of candidate neighbors that is further pruned by the more accurate but expensive to compute continuous distance model. Before presenting results on this hierarchical scheme, we first report performance when we use only the continuous distance model discussed in $\S$ 3.3. Since using the continuous model alone is more accurate in practice, this performance will be used as a benchmark to gauge the performance of the full hierarchical scheme. This section also presents the relative discriminative powers of the various feature spaces (color, texture and local shape) and shows how the discriminative information from these feature spaces when used together complement each other to a substantial degree compared with just using each feature space in isolation. In $\S$ 7.3, we report the significant increase in run-time performance that is gained when using the hierarchical scheme, while sacrificing little in detection performance. We conclude the chapter with results on a difficult face detection task with varying facial expressions. This detection task will illustrate the use of linear discriminators that are generated using the unsupervised Fisher-like criterion that was presented in § 5.2. We will also report the performance when a continuous distance measure learned on one set of training images is used for detecting faces that are not represented in the training set. Such “transfer” of distance measures is useful in practice when the set of faces that needs to be detected at run-time need not all be known at training time.

7.1 The Indoor Detection Task

7.1.1 Training Set

Figure 7.1 shows a collection of 15 objects that we are interested in detecting in images taken under an indoor office setting. Training images for each object were taken at two elevations that were $10°$ apart and which were close to the height of a person at a distance of approximately 7 ft from the object. At each elevation, training images were taken over a $180°$ sweep horizontally around the object at intervals of $20°$. Only half the horizontal sweep was taken since most of the objects are symmetric about the vertical axis. Objects were manually segmented from the background in each training image. Figure 7.2 shows some of the training images for one of the objects. As described in § 6.2.1, up to 10 discriminative parts are selected in each training image. Additional training views for the selected parts are sampled synthetically from the raw training image at different scales and rotations (see § 6.2.1). Furthermore, the training images were taken under illumination conditions that were natural and kept constant for an indoor setting. Rather than collecting more training images under varying illumination conditions, we chose to use the normalization procedures described in § 6.1 that were found to be sufficient in compensating for the moderate amount of illumination variation encountered in typical indoor settings.

7.1.2 Testing Set

We wanted to collect a large set of testing images with a large number of backgrounds as well as with a large number of viewpoint changes for the objects of interest. Collecting testing data satisfying both criteria at the same time would

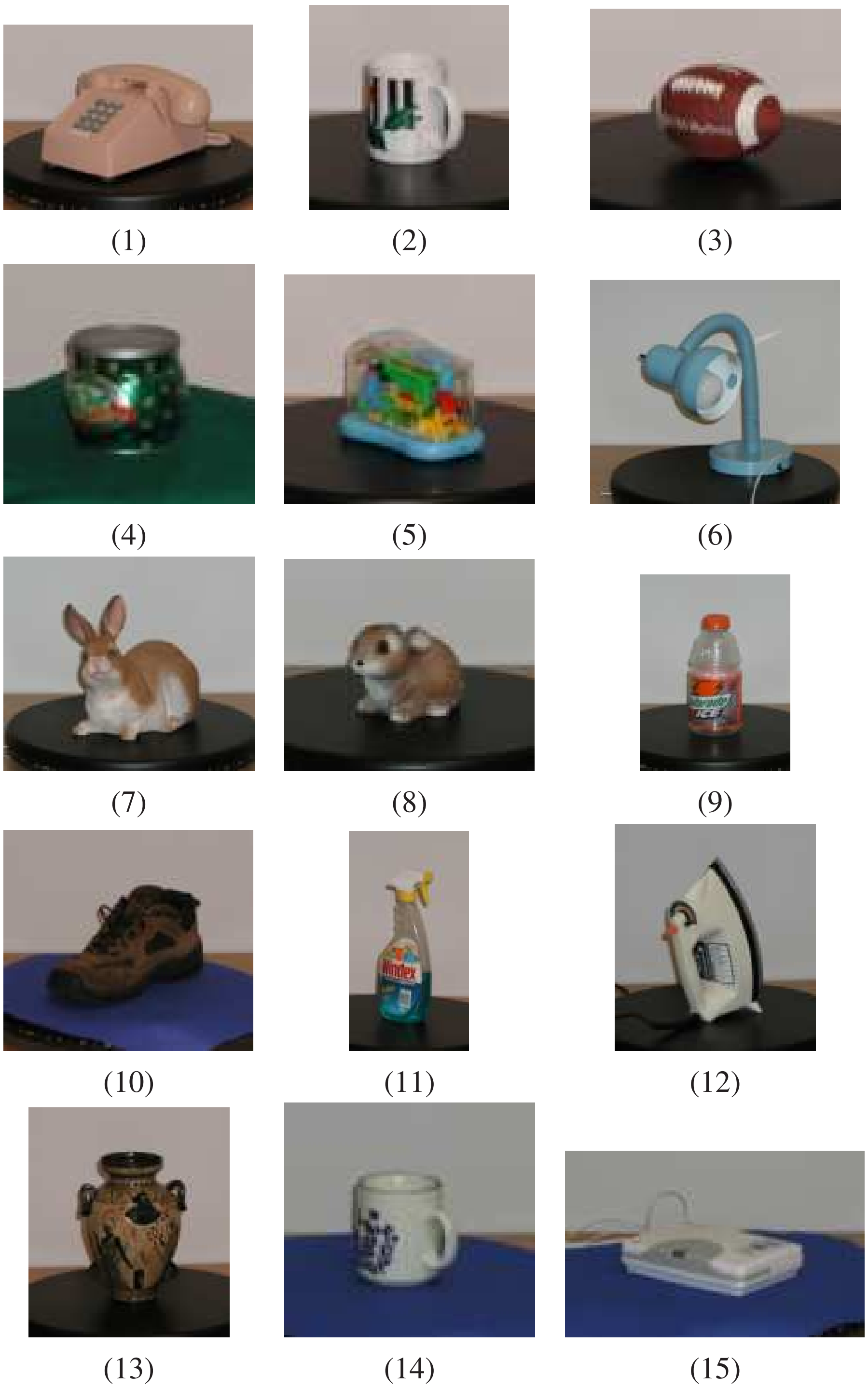


Figure 7.1: The 15 objects of interest for the indoor detection task.

Figure 7.2: Sample training views for one of the objects.

be prohibitively expensive. Instead, we collected two sets of testing images: one set varied the viewpoint that the object of interest was taken under more than the background, while the other set varied the background more than the viewpoint of the objects of interest.

The first set was taken using a tripod and contains images of objects of interest taken with 3 different backgrounds. For each of the 3 backgrounds, images of each object of interest were taken under varying viewpoints at around the same distance from the camera as was the case when the training images were taken. This set contained a total of 315 images with 21 images for each object of interest.

The second test set was taken with a hand-held camera and contains images with 15 different backgrounds, one for each object of interest. This set contains a total of 60 images with 4 images for each object of interest. Thus we have a combined total over both sets of 375 test images with 25 images for each object of interest.

Figure 7.3: Sample test images for the indoor detection task.

See 7.3 for a sample of the test images. As seen from the sample, the test set includes variations in scale, elevation and viewpoint for the objects of interest.

7.2 Continuous Distance Model Performance

Recall from § 6.4 that our scheme first finds a small set of K_c candidate object parts at each sampled location in the input image through a nearest neighbor search over the training set using some distance measure. Each of these candidate parts at a given location generates a hypothesis for an object at that location, for which scores are accumulated from all the parts belonging to that object found at the corresponding locations in the input image predicted by the hypothesis. The scores for each hypothesis are then thresholded and the surviving hypotheses are reported after performing local non-maximal suppression, see § 6.4 for details.

In this section, we investigate the performance for our scheme when only the continuous linear model (§ 3.3) for the optimal distance measure is used in the nearest neighbor search for parts at each location. As discussed in § 3.3, we find that the continuous model is more accurate than the discrete model in practice, albeit at more expense to compute at run-time compared with the discrete model. For good run-time performance as well as good detection performance, we combine the two models in a hierarchical scheme as detailed in § 3.3. Since the continuous model is more accurate in practice, we will use the performance reported in this section as a benchmark against which the detection performance for the full hierarchical scheme will be judged in the next section. We will also empirically evaluate the relative discrimination powers of the various feature types (color, texture and local shape) and show that in practice they complement each other to a substantial degree for the detection task at hand.

7.2.1 The Continuous Model Benchmark

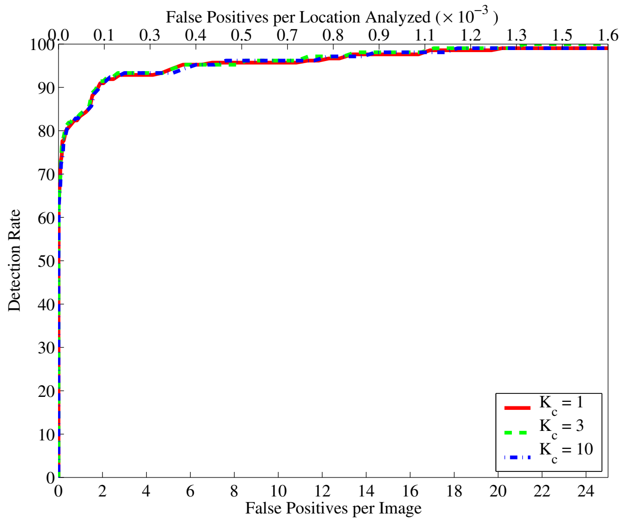
The performance of our detection scheme outlined above and detailed in § 6.4 depends on two parameters: (a) K_c the number of nearest neighbor parts reported at each location and (b) the threshold θ that is used after accumulating scores for each hypothesis generated by the detected parts. A given setting for these parameters (K_c, θ) will give rise to some performance for each object of interest, which can be empirically characterized by the correct detection rate for that object along with the false positive rate over the set of 25 test images for the object described in § 7.1.2. An object is considered to be detected in a test image if our scheme reports a detection of an object with the correct object label and falls within a 32×32 pixel neighborhood of the actual location of the object in the test image that was manually labeled beforehand. Plotting the detection vs. false positive rate while varying the two parameters gives us a receiver operating characteristic (ROC) plot (Egan, 1975; Green and Swets, 1966).

Each object will give us a corresponding ROC plot. Obviously, different objects will in general have different ROC plots as some objects will be harder to detect than others. We summarize the performance of our detection scheme by plotting the average ROC curve over all objects in Figure 7.4 as well as plotting the individual ROC plots for each object in Figure 7.5.

An objective unit for the false positive rate is the total number of false positives over all test images divided by the total number of locations tested by the detection scheme over all test images. We plot this unit along the top margin in all the ROC plots reported here. However, this unit can make the ROC plot seem too optimistic (note the scale factor of 10^{-3} for the unit in the plots). In contrast, we also use the average number of false positives per test image. This unit is plotted along the bottom margin in the ROC plots and is more subjective since it depends on the size of the field of view that the input image covers, unlike the case for the unit described above. Nevertheless, we feel that the second unit gives a more intuitive handle on the detection performance of our scheme.

In Figure 7.4, the ROC plot is represented by a set of ROC curves, one for each setting for K_c, the number of candidate parts returned by the nearest neighbor search using the continuous distance model. Each curve is generated by varying the threshold θ. As a representative point, we get a detection rate of 82% for a false positive rate of 0.5 per test image corresponding to $K_c = 3$.

Surprisingly, the detection performance does not vary much with the number of neighbors K_c. This insensitivity can be explained as follows. A given whole object training view is decomposed into a certain number of parts (up to 10 in our experiments) as discussed in § 6.2. Consider a test image which contains the object at some location under viewing conditions close to that in the training image. The location of the object will determine the locations where the parts



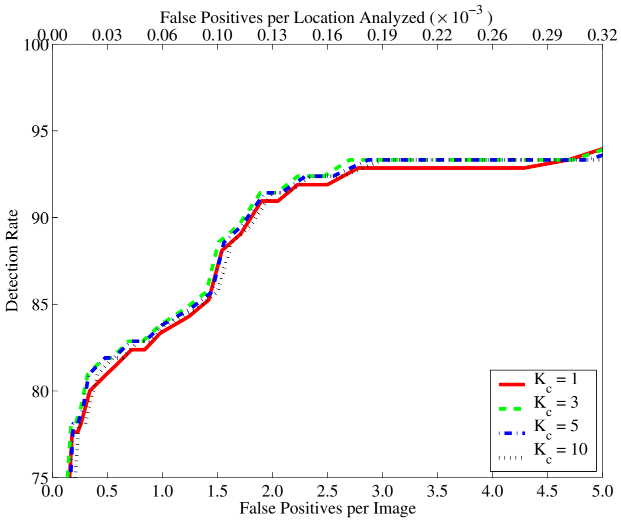


Figure 7.4: Average ROC plot for the indoor detection task using only the continuous distance model. The x-axis is labeled using two units, the more objective unit shown along the top margin is the false positive rate per location tested, while the more subjective unit shown along the bottom is the false positive rate per test image, where both units are averaged over all test images. The ROC is represented by a set of ROC curves, one for each setting for the number of candidate parts K_c that is returned by the nearest neighbor search using the continuous distance model. The detection performance is surprisingly quite insensitive to K_c. See text for discussion. The second plot above details the top left hand corner of the first plot. The ROC curve corresponding to $K_c = 3$ is used as a reference for comparison purposes in subsequent plots.

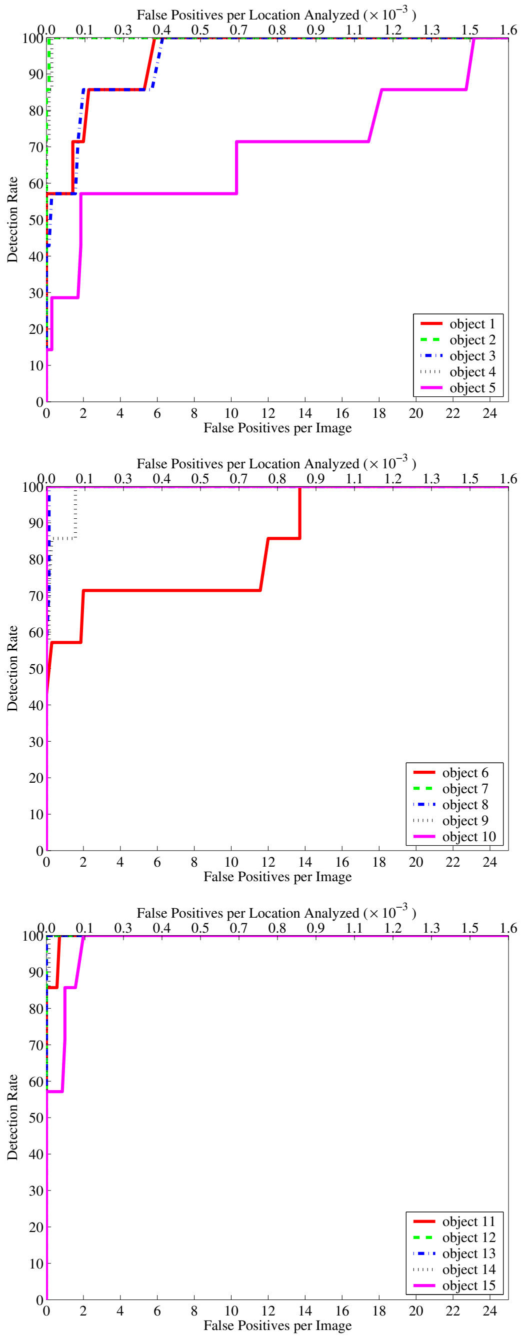


Figure 7.5: The individual ROC plots for each object. For clarity, the set of plots is broken into 3 graphs with 5 objects each. The numbering for the objects is the same as in Figure 7.1.

corresponding to the training view will be expected. *Any* of these locations in the test image can trigger a hypothesis for the given object under consideration if the corresponding part is reported among the top K_c neighbors at those locations. Put another way, for the hypothesis to be triggered, *only one* of these parts need to be reported in the top K_c neighbors at the corresponding expected location in the test image. Thus the hypothesis will likely be triggered with high probability since the probability that *all* the parts fail to be reported in the top K_c neighbors will be low.

To make this intuition more precise, assume the following very simple model: let the probability that a part fails to be reported in the top K_c neighbors be $p(K_c)$ which we assume is the same for all the parts. Obviously, this probability will be some monotonically increasing function of K_c since the set of parts reported for any value for $K_c = k$ is a subset of the set of parts for all values of $K_c > k$. Furthermore, let the probability of failure for the different parts be *independent* of each other. This assumption is not unrealistic if we assume that the parts are non-overlapping. Under this assumption, the probability that the hypothesis for the object under consideration will not be triggered exponentially decreases with the number of parts. Thus for a large enough number of parts, the hypothesis will likely be triggered by at least one part. Note that the subsequent verification step where scores are accumulated for the hypothesis does not depend on K_c.

In Figure 7.6 we compare the detection performance when using the optimal estimate for the continuous distance model with the performance when using a "naive" distance model where each of the elementary distance measures are equally weighted. As a representative point, we get a detection rate of 76% for a false positive rate of 0.5 per test image corresponding to $K_c = 3$ for the naive distance measure compared with a detection rate of 82% for the optimal estimate for the continuous distance model. Note that the comparison is not an evaluation of the distance measures in isolation, rather it is an evaluation of the distance measures in the context of the whole detection scheme. Other factors like the parts selected and part integration also influence performance. We report the influence of some of these factors on detection performance later on.

Figure 7.7 shows some examples of correct detection at the representative point mentioned above, whereas Figure 7.8 shows examples of false negatives. Both sets of examples also show some false positives.

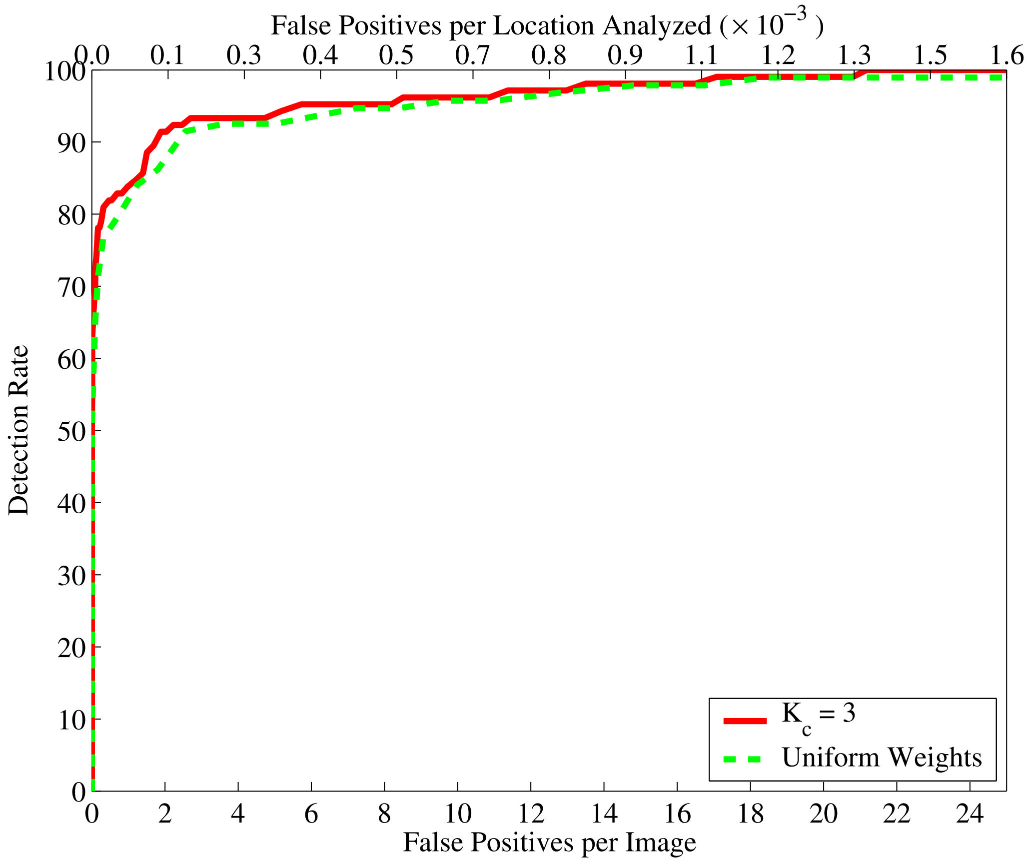


Figure 7.6: Comparison of the detection performance when using the optimal estimate for the continuous distance model with the performance when using a “naive” distance model where each of the elementary distance measures are equally weighted. See text for details.

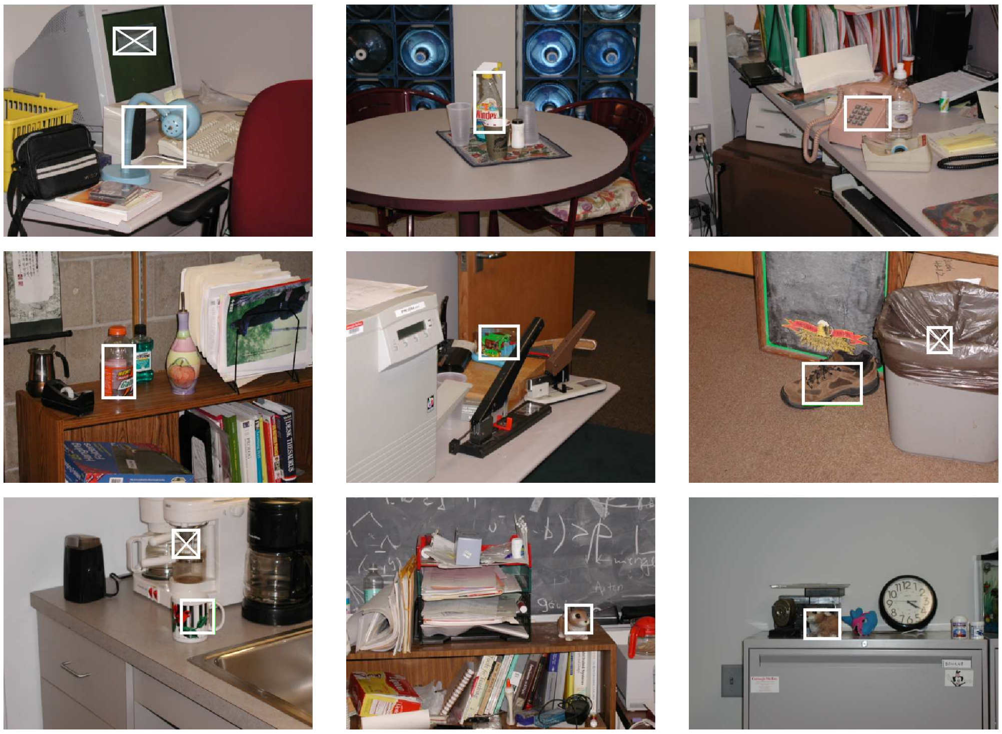


Figure 7.7: Examples of correct detections corresponding to a threshold that gives an average false positive rate of 0.5 per test image. Correct detections are shown as empty white boxes while false positives are shown as crossed boxes.

Finally, Figure 7.9 shows anecdotal results on a few test images with more than one object of interest.

In the remainder of the chapter we will use the ROC curve corresponding to $K_c = 3$ as a reference for comparison in subsequent sections.

7.2.2 The Relative Discriminative Powers of the Features

The previous subsection utilized all of the feature types (color, texture and local shape) in the continuous distance model. Here we systematically compare the relative discriminative powers of the various feature types by determining the empirical detection performance when only one or two feature types are used.

Figure 7.10 shows the relative performance of the various feature types when used in isolation. Note that each feature type is comprised of more than one feature space (3 for color, 3 for texture and 2 for local shape, see § 6.1). All of the feature spaces comprising a given feature type are used when that feature type is tested in isolation.

For reference, we also show the performance when all three feature types are used (called the "reference" ROC curve corresponding to $K_c = 3$ in Figure 7.4). As can be seen, both color and texture are quite discriminative on their own, while local shape is the least discriminative. This need not mean that local shape is not a useful feature type in general since our implementation for extracting local shape properties (local orientation and curvature) is quite simple and not very robust (see § 6.1 for details of the implementation). More robust implementations and/or more global shape properties should result in better detection performance.

As a representative point, corresponding to a false positive rate of 0.5 per test image, color gives a detection rate of 5.7%, texture gives a rate of 12.1% and shape gives a rate of 4.08%. These detection rates are however far lower than the 82% detection rate obtained when using all the feature types together. Thus we see that the various feature types complement each other to a substantial degree when used together, especially at operating points with low false positive rates, which is precisely the region that is useful in practice.

Figure 7.11 shows the relative performance when we choose all combinations of only two feature types together. Once again, as should be evident by studying the ROC plot where the corresponding feature type has been dropped, both color and texture have good discriminative powers, while local shape has the least discriminative power.

7.2.3 Importance of Hypothesis Verification

One interesting question is how important is the accumulation of scores from multiple parts for detecting an object of interest, which we will call "part integration" in the following, compared with just using the parts directly for detecting the object. We can effectively test this empirically by comparing the performance when part integration is enabled vs when it is disabled. By disabled, we mean that each hypothesis generated is scored by only using the score of the part that generated the hypothesis and not the scores of the other parts predicted by the hypothesis.

Figure 7.12 shows the result of such an experiment. As can be seen, parts by themselves are quite capable of predicting the presence of an object in an input image. Nevertheless, part integration provides quite a boost to the resulting detection performance. As a representative point, without part integration we get a detection rate of only 60% corresponding to a false positive rate of 0.5 per test image, compared with an 82% detection rate for the same false positive rate when part integration is enabled.

7.3 Hierarchical Distance Measure Performance

In this section, we report the detection performance for the full hierarchical distance measure scheme. Recall from § 3.3 that in the hierarchical scheme, we first use an efficient but coarse tree-based discrete distance measure for the searching for the nearest neighbor parts at each sampled location of the input image. We search for the K_d nearest neighbors that are then further pruned by the continuous distance measure that is accurate but expensive to compute, to yield $K_c < K_d$ nearest neighbors. The resulting K_c parts are further processed by generating object hypothesis from these parts, followed by accumulating and thresholding scores for each hypothesis, as detailed in § 6.4.

The detection performance when using the hierarchical distance measure depends on two parameters associated with the discrete distance measure, in addition to the parameters K_c (number of nearest neighbors reported by the continuous distance) and the threshold θ discussed in the previous section. The two parameters for the discrete distance measure are: (a) K_d, the number of nearest neighbors reported by the discrete distance measure and (b) $|T|$, the size of the tree T implementing the discrete distance measure (see § 6.3).

Before exploring the detection performance for the hierarchical scheme, we first report the performance when using only the discrete distance model and compare it with the performance when using only the continuous distance model that was studied in § 7.2. Figure 7.13 shows the detection performance for the discrete distance model with $K_d = 3$ and $|T| = 80$. This performance is compared with the the reference ROC curve from § 7.2.1 for the continuous distance model with $K_c = 3$. As can be seen, the discrete distance model performs poorly when used in isolation. This is our main motivation for combining the discrete model with the continuous model to yield a hierarchical scheme that is both efficient as well as accurate.

We will now explore the detection performance for the hierarchical scheme as we vary both K_d and $|T|$. In practice, we will choose the settings for these parameters that will satisfy the operating requirement (characterized by the false positive and detection rate) that is desired for the task at hand. Figure 7.14(a) shows the ROC plot when we vary K_d while fixing $|T| = 80$, whereas Figure 7.14(b) shows the ROC plot when we vary $|T|$ while fixing $K_d = 3K_c = 9$.

Table 7.15 shows the time performance corresponding to Figure 7.14(b) as $|T|$ varies. For each value of $|T|$, we quantify the time performance by taking the ratio of the average time taken by the hierarchical scheme over all test images and the time taken when using just the continuous distance measure. We also report the absolute time taken per image on a 1.5 GHz CPU x86 machine. The absolute time taken when using just the continuous distance measure was around 13 minutes and 10 seconds. The ratio should be considered as the more useful measure of time performance since to a first order approximation, it does not depend on the absolute speed of the machine.

As can be seen, we get an order of magnitude speed-up when using the hierarchical scheme while sacrificing only a little bit in detection performance. As a representative point, for $K_d = 9, |T| = 80$, we get a speed-up by a factor of about 20 corresponding to a detection performance characterized by a detection rate of 77% and false positive rate of 0.5 per test image. On the other hand, the representative point mentioned in $\S$ 7.2.1 when using only the continuous distance measure is characterized by a detection rate of 82% and false positive rate of 0.5 per test image.

7.4 Experiments on Faces

In this last section, we report results on a challenging face recognition task. The domain of face recognition gives us an opportunity for illustrating the use of the technique outlined in $\S$ 5.2 for generating candidate discriminators, used to form the discrete distance measure, based on a Fisher-like criterion.

We chose a subset of frontal face images from the FERET (Phillips et al., 1997) database that had varying expressions and some illumination changes. Specifically, we chose a subset corresponding to 200 individuals, for each of which there were 3 images with varying expression and illumination, labeled as 'fa', 'fb' and 'fc' in (Phillips et al., 1997). Figure 7.16 shows a sample of the selected images.

The selected images were pre-processed as follows. Each of the images were aligned using a similarity transform (rotation, translation and scale) such that the locations of the eyes, whose positions in the original image were provided in the FERET database, fell on pre-specified pixel locations in the transformed image. Next, the images were cropped with a common mask to exclude background and hair. The non-masked pixels were then histogram equalized and the resulting pixels were further processed to have zero mean and unit variance. Figure 7.17 shows an image before and after pre-processing.

Two of the three images for each individual were chosen as training images, while the remaining image was used as a test image. Before we construct the hierarchical distance measure, we first develop and benchmark a continuous distance measure that we can use to gauge the performance of the hierarchical distance measure, just as we did for the indoor discrimination task in the previous section.

7.4.1 Continuous Distance Model

There are several possible continuous distance measures that we can develop. Our choice will be dictated by simplicity of the resulting implementation. The simplest is to just use the euclidean distance measure in the linear feature space of all the non-masked pixels. A more robust version will be to first project this space onto the principal components using PCA thus ignoring the dimensions in the feature space that are likely to correspond to noise (Turk and Pentland, 1991; Nayar et al., 1996).

The above PCA approach gives us only one distance measure for the whole linear feature space. All directions in the PCA subspace chosen are given equal weight in the euclidean distance measure for that subspace. We can hope to get more discriminative distance measures if we combine more elementary distance measures, all of which are defined in the same feature space. The elementary distance measures we choose to use are distances between projections of images along different directions in the feature space. We then learn a distance measure that linearly combines such elementary distance measures. The resulting weights will indicate the relative discriminative powers of each direction of projection.

What are good directions to project? Since we are interested in discriminating among faces, we can think of finding directions within the subspace of the linear feature space in which faces vary. This subspace can be expected to be most important for discrimination purposes. Such a subspace can be conveniently obtained by finding the PCA of all the differences between face images. Such a so-called *image difference space* has been used previously in the literature (Moghaddam and Pentland, 1998; Phillips, 1999). Note that this PCA decomposition is different from the PCA decomposition described above which was for the original image space.

Let the PCA decomposition of the image difference space be an N-dimensional subspace. We use each of the N principal components of the the PCA decomposition of image difference space as directions along which we create the elementary distance measures that we can use in our linear model for a discriminative distance measure. We then use the maximum likelihood greedy scheme developed in 4.1 to select the $K < N$ most discriminative elementary distance measures for our linear model.

Figure 7.18 compares the performance of our continuous distance model with the baseline PCA algorithm described earlier as we vary the number of components K for each algorithm. In the case of the baseline PCA algorithm, K corresponds to the number of the most significant principal components chosen, whereas for the continuous distance measure, K is the number of elementary distance measures that we choose from among the $N = 200$ available distance measures using the greedy selection scheme. As can be seen, the continuous distance model performs very well in comparison with standard PCA while using only a few components.

7.4.2 Hierarchical Distance Model

Next, a hierarchical distance measure was learned for the face discrimination task. The discrete component for the hierarchical distance measure was constructed from discriminators learned using the approach detailed in § 5.2 for constructing discriminators in the linear feature space. Here the linear feature space is formed by the set of all non-masked pixels.

Recall that in this approach, we first generate candidate linear discriminators that satisfy the three criteria given in § 5.2. Note that in the first iteration, since there is only a single feature space, only a single discriminator is generated which forms the root of the alternating tree (see § 6.3). However, in all subsequent iterations more than one candidate discriminators are generated, one each for every partition node in the tree, even though all of them are constructed in the *same* linear feature space.

We learned an alternating tree with 40 discriminator nodes. For the continuous component of the hierarchical distance measure we used the distance measure developed in the previous section with $K = 30$ components. The resulting hierarchical distance measure gave a recognition rate of 93% compared with a rate of 94% when using just the continuous distance measure. On the other hand, we get around a factor of 9 speed-up when using the hierarchical distance measure compared with using just the continuous distance measure.

Figure 7.8: Examples of false negatives corresponding to a threshold that gives an average false positive rate of 0.5 per test image. False positives are shown as crossed boxes.

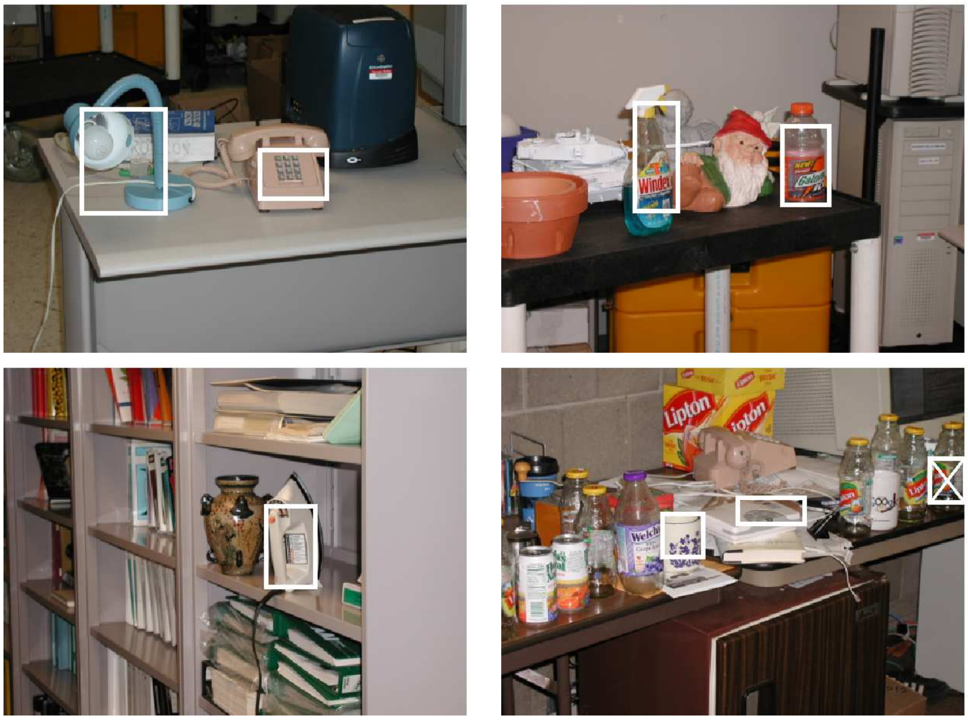


Figure 7.9: Anecdotal results with more than one object of interest per test image. Correct detections are shown as empty white boxes while false positives are shown as crossed boxes.

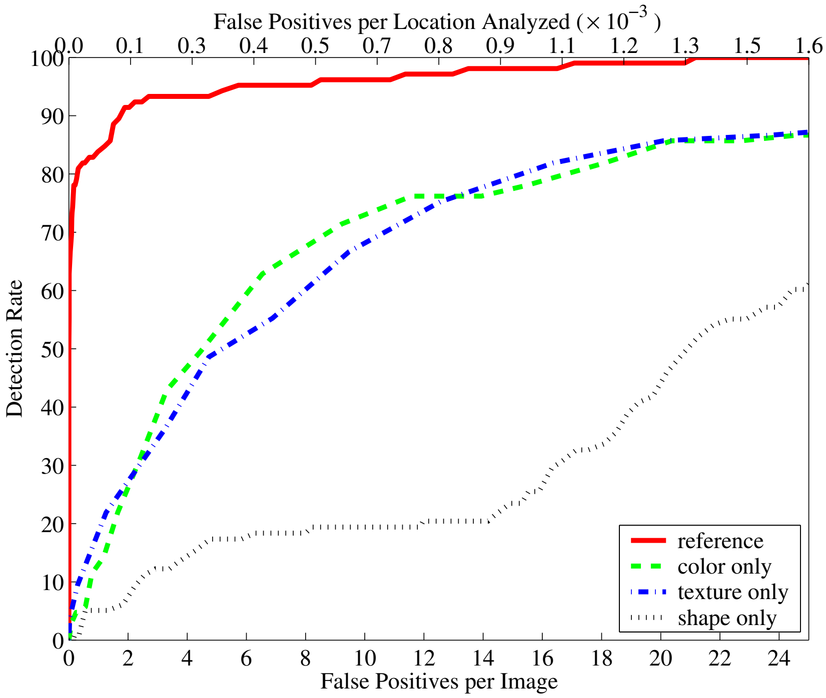


Figure 7.10: Detection performance when the feature spaces are used in isolation. For comparison, we also show the reference curve from section § 7.2.1 with $K_c = 3$ that utilizes all of the feature spaces.

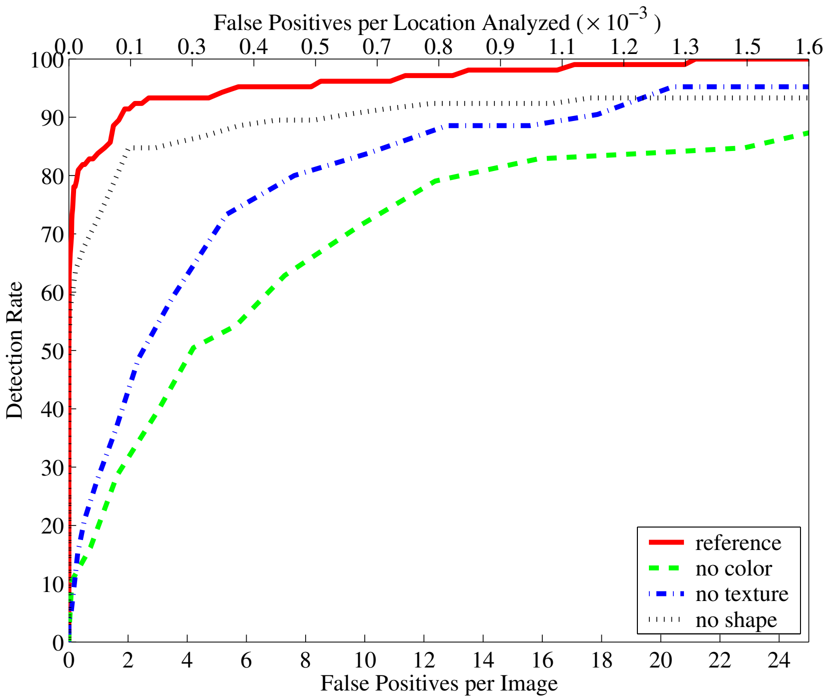


Figure 7.11: Detection performance when only two feature spaces are used together. The ROC curves are labeled by the feature type that has been dropped.

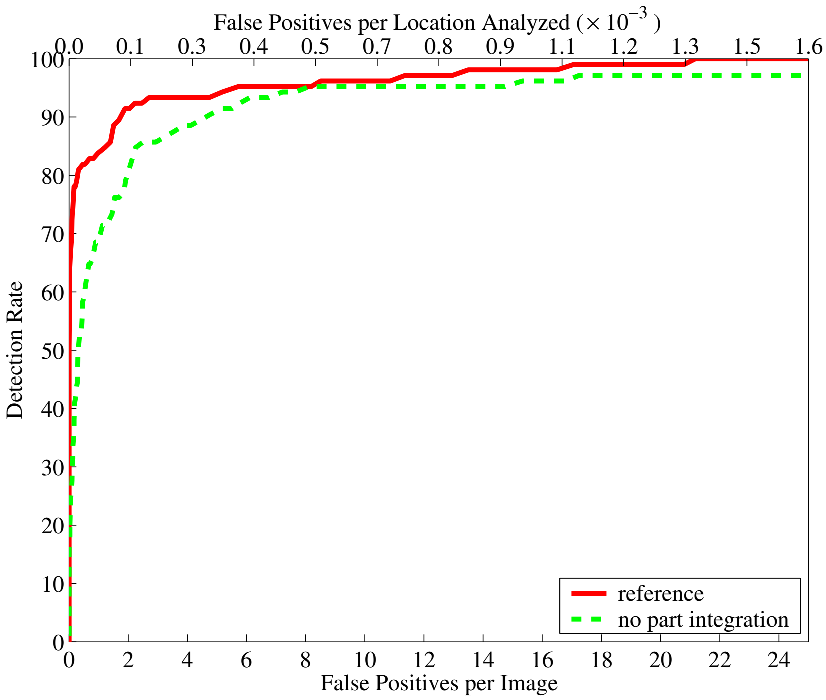


Figure 7.12: Detection performance when part integration is enabled vs when it is disabled. Part integration provides quite a boost to the detection performance.

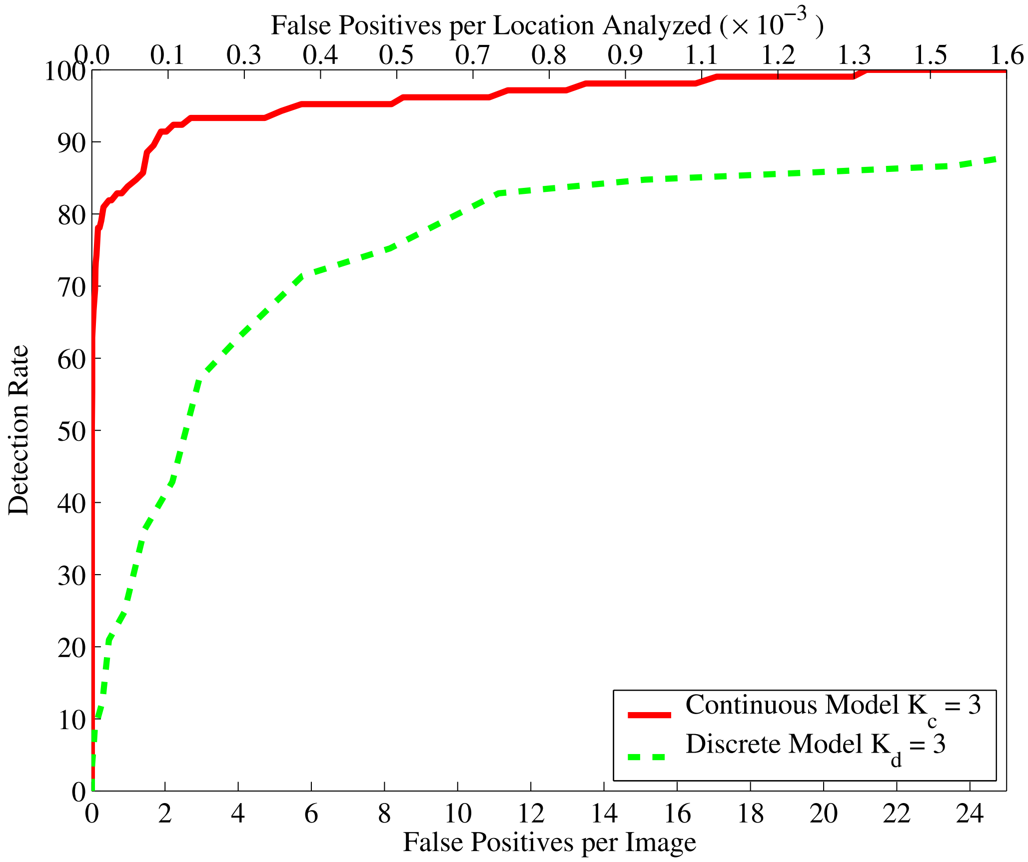
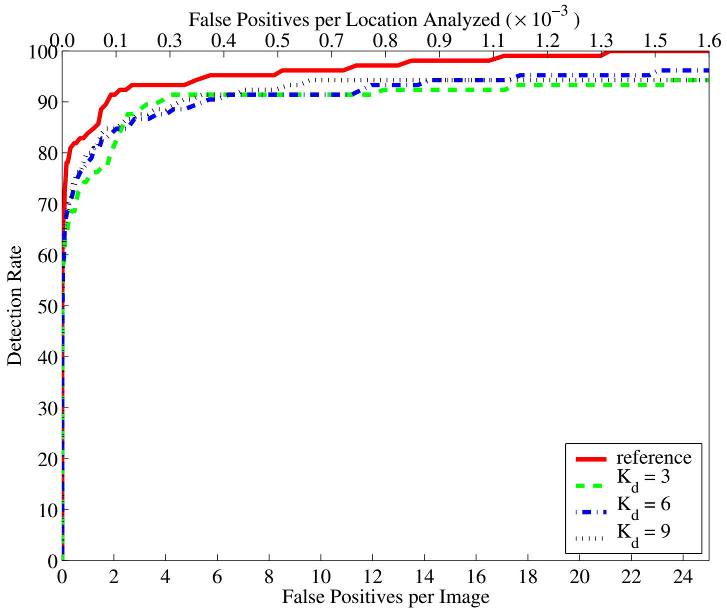
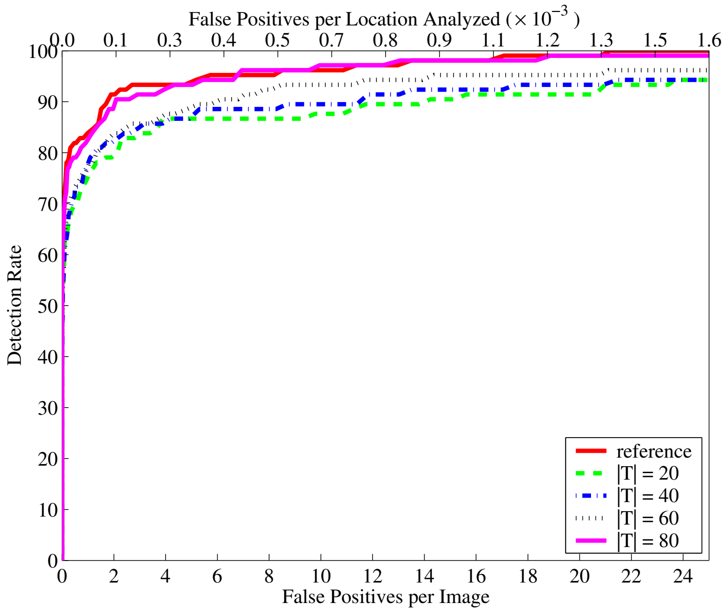


Figure 7.13: Comparison of the detection performance when using the the continuous distance model with the performance when using the discrete distance model. See text for details.



(a)



(b)

Figure 7.14: (a) Detection performance against varying K_d, the number of nearest neighbours returned by the tree-based discrete distance measure. The size of the tree is fixed to $|T| = 80$. (b) Detection performance against varying tree size $|T|$ while fixing $K_d = 3K_c = 9$.

# of Nodes $|T|$	Absolute Time (sec)	Speed-up
20	34.9	22.5
40	36.9	21.3
60	38.4	20.5
80	39.3	20.0

Figure 7.15: Time performance corresponding to Figure 7.14(b) as $|T|$ varies. The second column is the absolute time on a 1.5 GHz x86 machine. The third column is the speed-up over the average time taken per image when only the continuous distance measure is used.

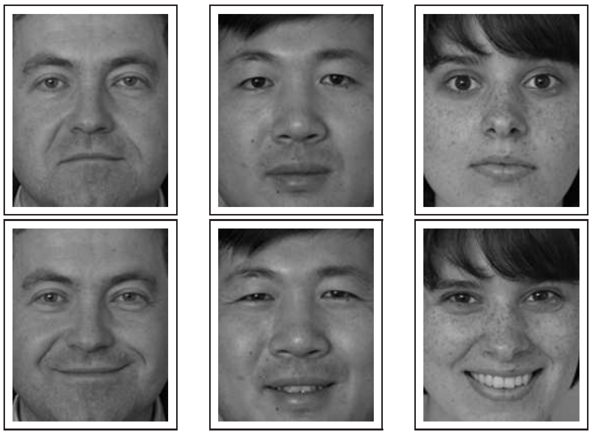


Figure 7.16: Sample images from the FERET database that we use in our discrimination task.

Figure 7.17: A face image before and after pre-processing. See text for details.

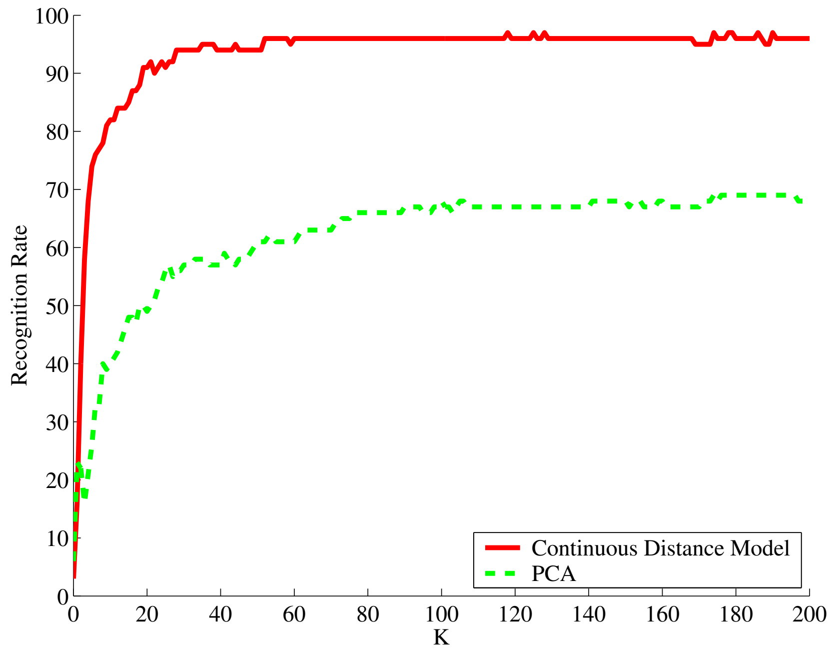


Figure 7.18: Recognition performance of our continuous distance model as the number K of elementary distance measures in the PCA difference space that is chosen by the greedy selection scheme is varied. For comparison, we also plot the performance with a baseline PCA algorithm in the original face space. For the latter K is the number of the most significant PCA components chosen.

Chapter 8

Conclusion

In this thesis, we investigated the design and implementation of good distance measures for a nearest neighbor framework for object detection. We first derived the optimal distance measure for the nearest neighbor search. Unlike most previous approaches, we modeled the optimal distance measure directly rather than first estimating intermediate generative models. We then investigated modeling the optimal distance measure by combining elementary distance measures associated with simple feature spaces. A simple linear combination model was motivated after observing actual data for a representative discrimination task.

For a given set of elementary distance measures, the parameters in the linear distance model were estimated under the maximum likelihood framework. Also a greedy scheme was presented under the same framework for selecting the best set of elementary distance measures chosen from a large collection of such distance measures. We investigated a selection scheme already proposed in the literature for the maximum entropy framework which is dual to the ML framework and showed that the two selection schemes are in fact the same.

For performing efficient nearest neighbor search over large training sets, we also developed a discrete distance measure that combined elementary distance measures associated with discriminators organized in a tree-like structure.

Finally, the nearest neighbor framework described above was integrated into an object detection system and evaluated in an indoor detection task as well as a face recognition task.

Future Work

Local Distance Models. In the work reported so far, the various distance models that we considered were all global models, that is the distance score output by these models did not depend on where in measurement space they were used. Clearly, the optimal distance measure can vary from place to place. Thus it is natural to think of adapting a distance model locally. One can then think of two approaches for estimating local distance models.

In the first approach, we can estimate a local distance measure for each query measurement. We can adopt the same maximum likelihood estimation framework that we developed for global linear models to find local distance models with the added restriction that only the subset of the training data that is "near" the query point is used in the estimation. This raises a chicken-and-egg problem since we do not know what is "near" and what is "far" from the query point until we have estimated the local distance model. We can get around this difficulty by first estimating a global distance model, and then finding the training data that is closest to the query point using this global model. We can even think of iterating this procedure by using the newly found local distance model to find again the nearest training data to the query point and use this new training subset to estimate yet another local distance model that hopefully should be better than the first. Such a procedure will be iterated until convergence. Similar ideas have been proposed in (Hastie and Tibshirani, 1996) for estimating locally optimal linear discriminants.

The obvious drawback of such an approach is that of poor run-time efficiency since a new local distance measure has to be estimated for every new query point. Motivated by the need to overcome such a drawback, the second approach for estimating local distance models would be to adapt a distance model for each *training* point rather than the query point. This can be done at training time and the estimated distance models can be stored for use at run-time. Given a query point, a nearest neighbor search is performed over the training set, in which the distance measure used between the query and a training point is the local distance measure estimated at training time for that training point. While obviously solving the run-time efficiency issue faced by the first approach, we are now faced with the problem of how to compare the different distance scores between the query and the training points since each distance score was determined by using a different distance measure. Intuitively, it is likely to be the case that the “further” the query is from a given training point, the less reliable is the corresponding local distance measure associated with the training point. Thus we need to know the “confidence region” for each distance measure for such an approach to work. Pursuing such ideas will be a future goal of our work.

Better Part Integration. In our work, we have found that accumulating scores from various parts to verify a whole object hypothesis was useful in boosting the detection performance. However, we gave equal weight to all the part scores irrespective of their relative discriminative powers. Clearly we should be able to do better by weighting a part score in proportion to its discriminative power.

We have only addressed a few issues above that we thought to be important. Since the main focus of the thesis was only on developing good distance measures for nearest neighbor search, there is clearly more room for improvement in almost every aspect of the rest of the object detection scheme presented in this thesis.

Bibliography

Abbott, A. and Zheng, B. (1995). Active fixation using attentional shifts, affine resampling, and multiresolution search. In *ICCV95*, pages 1002–1008.

Arman, F. and Aggarwal, J. (1993a). Cad-based vision: Object recognition in cluttered range images using recognition strategies. *CVGIP*, 58(1):33–48.

Arman, F. and Aggarwal, J. (1993b). Model-based object recognition in dense range images. *Surveys*, 25(1):5–43.

Baluja, S. and Pomerleau, D. (1997). Dynamic relevance: Vision-based focus of attention using artificial neural networks. *AI*, 97(1-2):381–395.

Barros, J. E., French, J. C., Martin, W. N., Kelly, P. M., and Cannon, T. M. (1996). Using the triangle inequality to reduce the number of comparisons required for similarity-based retrieval. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 392–403.

Baxter, J. and Bartlett, P. (1998). The canonical distortion measure in feature space and 1-NN classification. In Jordan, M. I., Kearns, M. J., and Solla, S. A., editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press.

Beis, J. and Lowe, D. (1997). Shape indexing using approximate nearest-neighbor search in highdimensional spaces. In *CVPR*, pages 1000–1006.

Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522.

Berman, A. and Shapiro, L. G. (1997). Efficient image retrieval with multiple distance measures. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 12–21.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford University Press.

Blanzieri, E. and Ricci, F. (1999). A minimum risk metric for nearest neighbor classification. In *Proc. 16th International Conf. on Machine Learning*, pages 22–31. Morgan Kaufmann, San Francisco, CA.

Buntine, W. (1993). Learning classification trees. In Hand, D. J., editor, *Artificial Intelligence frontiers in statistics*, pages 182–201. Chapman & Hall,London.

Burl, M., Weber, M., and Perona, P. (1998). Slippre: Face localization by shape likelihood plus part responses. In *ECCV98*.

Burt, P. (1988). Attention mechanisms for vision in a dynamic world. In *ICPR88*, pages 977–987.

Canny, J. (1986). A computational approach to edge detection. *PAMI*, 8(6):679–698.

Chen, S. and Rosenfeld, R. (2000). A survey of smoothing techniques for me models. *IEEE Transactions on Speech and Audio Processing*, 8(1).

Chin, R. and Dyer, C. (1986). Model-based recognition in robot vision. *Surveys*, 18(1):67–108.

Comaniciu, D., Ramesh, V., and Meer, P. (2000). Real-time tracking of non-rigid objects using mean shift. In *CVPR00*, pages II:142–149.

Cover, T. (1991). *Elements of information theory*. Wiley.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13:21–27.

Culhane, S. and Tsotsos, J. (1992). An attentional prototype for early vision. In *ECCV92*, pages 551–560.

Dasarathy, B. (1991). *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. Computer Society Press.

de Bonet, J., Viola, P., and Fisher, III, J. (1998). Flexible histograms: A multiresolution target discrimination model. In *SPIE*.

Della Pietra, S., Della Pietra, V., and Lafferty, J. (1997). Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4).

Deriche, R. (1992). Recursively Implementing the Gaussian and Its Derivatives. In *Proc. Second International Conference On Image Processing*, pages 263–267, Singapore.

Duda, R., Hart, P., and Stork, D. (2001). *Pattern Classification*. Wiley.

Egan, J. (1975). *Signal Detection Theory and ROC Analysis*. Academic Press.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis. *Communications of the ACM*, 24:381–395.

Freund, Y. and Mason, L. (1999). The alternating decision tree algorithm. In *Intl. Conf. on Machine Learning*, pages 124–133.

Freund, Y. and Shapire, R. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. of Computer and System Sciences*, 55(1):119–139.

Friedman, J. (1994). Flexible metric nearest neighbor classification. Technical Report Technical Report 113, Stanford University Statistics Department.

Fukanaga, K. and Flick, T. (1984). An optimal global nearest neighbor metric. *PAMI*, 6(3):314–318.

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*. Academic Press.

Gersho, A. and Gray, R. M. (1992). *Vector Quantization and Signal Compression*. Kluwer Academic Publishers.

Green, D. and Swets, J. (1966). *Signal Detection Theory and Psychophysics*. Wiley.

Greenspan, H., Belongie, S., Perona, P., Goodman, R., Rakshit, S., and Anderson, C. (1994). Overcomplete steerable pyramid filters and rotation invariance. In *CVPR94*, pages 222–228.

Grimson, W., Klanderman, G., O'Donnell, P., and Ratan, A. (1994). An active visual attention system to play where's waldo. In *ARPA94*, pages II:1059–1065.

Grove, T. and Fisher, R. (1996). Attention in iconic object matching. In *BMVC96*, pages Model Fitting, Matching, Recognition.

Hastie, T. and Tibshirani, R. (1990). *Generalized Additive Models*. Chapman and Hall.

Hastie, T. and Tibshirani, R. (1996). Discriminant adaptive nearest neighbor classification and regression. In Touretzky, D. S., Mozer, M. C., and Hasselmo, M. E., editors, *Advances in Neural Information Processing Systems*, volume 8, pages 409–415. The MIT Press.

Huang, C., Camps, O., and Kanungo, T. (1999). Object representation using appearance-based parts and relations. In *UMD*.

Huttenlocher, D., Klanderman, G., and Rucklidge, W. (1993). Comparing images using the hausdorff distance. *PAMI*, 15(9):850–863.

Huttenlocher, D. and Ullman, S. (1990). Recognizing solid objects by alignment with an image. *IJCV*, 5(2):195–212.

Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *PAMI*, 20(11):1254–1259.

Jacobs, D., Weinshall, D., and Gdalyahu, Y. (2000). Classification with nonmetric distances: Image retrieval and class representation. *PAMI*, 22(6):583–600.

Jaynes, E. (1957). Information theory and statistical mechanics. *Physical Review*, 106:620–630.

Kane, T., McAndrew, P., and Wallace, A. (1991). Model-based object recognition using probabilistic logic and maximum entropy. *PRAI*, 5:425–437.

Lebanon, G. and Lafferty, J. (2001). Boosting and maximum likelihood for exponential models. In *Advances in Neural Information Processing Systems*, volume 14.

Leung, T., Burl, M., and Perona, P. (1995). Finding faces in cluttered scenes using labelled random graph matching. In *ICCV95*, pages 637–644.

Mason, L., Baxter, J., Bartlett, P., and Frean, M. (2000). Boosting algorithms as gradient descent. In Solla, S., Leen, T., and Muller, K.-R., editors, *Advances in Neural Information Processing Systems*, volume 12, pages 512–518. The MIT Press.

McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models*. Chapman and Hall.

Mel, B. (1997). Seemore: Combining color, shape, and texture histogramming in a neurally inspired approach to visual object recognition. *NeurComp*, 9(4):777–804.

Mikolajczyk, K. and Schmid, C. (2002). An affine invariant interest point detector. In *ECCV02*, page I: 128 ff.

Minka, T. (2000). Distance measures as prior probabilities. Technical report, http://www.stat.cmu.edu/ minka/papers/learning.html.

Moghaddam, B. and Pentland, A. (1998). Beyond eigenfaces: Probabilistic matching for face recognition. In *Intl. Conf. on Automatic Face and Gensture Recogntition*.

Murase, H. and Nayar, S. (1997). Detection of 3d objects in cluttered scenes using hierarchical eigenspace. *PRL*, 18(4):375–384.

Nayar, S., Nene, S., and Murase, H. (1996). Real-time 100 object recognition system. In *ARPA96*, pages 1223–1228.

Nelson, R. and Selinger, A. (1998). A cubist approach to object recognition. In *ICCV98*, pages 614–621.

P. Indyk, R. M. (1998). Approximate nearest neighbors: Towards removing the curse of dimensionality. In *STOC*, pages 604–613.

Phillips, P., Moon, H., Rauss, P., and Rizvi, S. (1997). The feret evaluation methodology for face-recognition algorithms. In *CVPR*, pages 137–143.

Phillips, P. J. (1999). Support vector machines applied to face recognition. In *Neural Information Processing Systems 11*, pages 803–809.

Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (1992). *Numerical Recipes in C*. Cambridge University Press.

Rao, R. and Ballard, D. (1995). Object indexing using an iconic sparse distributed memory. In *ICCV95*, pages 24–31.

Roberts, L. (1965). Machine perception of 3-d solids. In *OE-OIP65*, pages 159–197.

Rowley, H., Baluja, S., and Kanade, T. (1998). Neural network-based face detection. *PAMI*, 20(1):23–38.

Ruzon, M. and Tomasi, C. (1999). Corner detection in textured color images. In *ICCV99*, pages 1039–1045.

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

Schiele, B. (1997). *Object Recognition using Multidimensional Receptive Field Histograms*. PhD thesis, I.N.P. Grenoble.

Schneiderman, H. (2000). *A Statistical Approach to 3D Object Detection Applied to Faces and Cars*. PhD thesis, Robotics Institute, Carnegie Mellon University.

Selinger, A. and Nelson, R. (2001). Appearance-based object recognition using multiple views. In *CVPR01*, pages I:905–911.

Shapiro, L. and Costa, M. (1995). Appearance-based 3d object recognition. In *ORCV95*, pages 51–64.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *PAMI*, 22(8):888–905.

Short, R. and Fukanaga, K. (1981). The optimal distance measure for nearest neighbor classification. *IT*, 27:622–627.

Stough, T. and Brodley, C. (2001). Focusing attention on objects of interest using multiple matched filters. *IP*, 10(3):419–426.

Swain, M. and Ballard, D. (1991). Color indexing. *IJCV*, 7(1):11–32.

Tomasi, C. and Shi, J. (1994). Good features to track. In *CVPR94*, pages 593–600.

Turk, M. and Pentland, A. (1991). Eigenfaces for recognition. *CogNeuro*, 3(1):71–96.

Ullman, S. and Basri, R. (1991). Recognition by linear combinations of models. *PAMI*, 13(10):992–1005.

Vapnik, V. N. (1999). *The Nature of Statistical Learning Theory*. Springer.

Viola, P. (1995). Complex feature recognition: A bayesian approach for learning to recognize objects. In *MIT AI Memo*.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *CVPR01*, pages I:511–518.

Weber, M., Welling, M., and Perona, P. (2000). Towards automatic discovery of object categories. In *CVPR00*, pages II:101–108.

Westlius, C., Westin, C., and Knutsson, H. (1996). Attention control for robot vision. In *CVPR96*, pages 726–733.

Worthington, P. and Hancock, E. (2000). Histogram-based object recognition using shape-from-shading. In *CVPR00*, pages I:643–648.

Zhu, S. C., Wu, Y., and Mumford, D. (1998). Filters, random fields and maximum entropy (frame). *IJCV*, 27(2):1–20.