Appeared in Proceedings of the ARPA Image Understanging Workshop, Palm Springs, 1996.

A Theory of Pattern Rejection *

Simon Baker and Shree K. Nayar
Department of Computer Science, Columbia University
New York, N.Y. 10027

Abstract

The efficiency of pattern recognition is critical when a large number of classes are to be discriminated, or when the recognition algorithm needs to be applied a large number of times. We propose and analyze a general technique, namely pattern rejection, that results in efficient pattern recognition. Rejectors are introduced as algorithms that can very quickly eliminate from further consideration most classes or inputs (depending on the setting). Rejectors may be combined to form composite rejectors, which are more effective than any single rejector. Composite rejectors are analyzed and conditions derived which guarantee both efficiency and practicality. A general technique is proposed for the construction of composite rejectors, based on a single assumption about the classes. The generality of this assumption is shown through its connection with the Karhunen-Loéve expansion. A relation of pattern rejection with Fisher's discriminant analysis is also shown. Composite rejectors were constructed for two applications, namely, object recognition and local feature detection. In both cases, a substantial improvement in efficiency over existing techniques was found.

1 Introduction

We address the efficiency of pattern recognition, which is known to be vital when the number of classes involved is large. An example application in computational vision is object recognition, which in many cases can be reduced to a classical pattern recognition problem [Murase and Nayar 95]. Of particular importance in this context, is the growth rate of recognition time as a function of the number of classes (objects). High efficiency also proves critical when the recognition algorithm must be applied a large number of times. This is the case in local feature detection [Nayar et al. 95], where the detector needs to be applied at every pixel in the image.

We propose a general theory that results in substantial efficiency improvements in both of the above scenarios. We introduce the notion of a *rejector*, that can be used to efficiently eliminate from further consideration most of a large number of classes (e.g. objects in recognition) or inputs (e.g. local image brightness values in feature detection). The following are the main results that constitute the proposed theory:

1. The definition of correctness for a rejector is less constraining than that for a recognizer. As a result, rejectors can be constructed that are more efficient than recognizers.

2. Although, in general, a single rejector does not provide the final solution to the pattern recognition problem, it can significantly reduce the number of possible classes or inputs. Consequently, the recognizer can dedicate its computational resources to a small number of candidates.

3. A collection of rejectors may be combined in a tree-like structure to form a more effective one, which we term a *composite rejector*. At each node is a single rejector, that is tuned to the small set of classes which were not eliminated by a previous rejector.

4. It is possible to analyze the performance of composite rejectors. For instance, we derive conditions that guarantee logarithmic time complexity in terms of the total number of classes involved.

5. Although the exact structure of a composite rejector is determined by the application at hand, the proposed rejection technique is very general and is based on a single assumption, namely, *the class assumption*. The generality of the class assumption is argued by establishing a connection with the Karhunen-Loéve (K-L) expansion [Fukunaga 90].

We demonstrate the utility of rejection via experiments on appearance matching based object recognition [Murase and Nayar 95] and feature detection [Nayar et al. 95]. First, we constructed a composite rejector for a widely used image database of 20 objects (each in a large number of poses). A composite rejector was able to completely (and without error) discriminate between all 20 objects with an efficiency that is a significant improvement over existing techniques. We empirically illustrate logarithmic growth in the time complexity with the number of objects. Next, we constructed a composite rejector for the task of feature detection. This results in a very efficient method of preprocessing an image to identify pixels that truly deserve the application of a full-fledged feature detector, such as the one proposed in [Nayar et al. 95].

*This research was supported in part by ARPA Contract DACA-76-92-C-007, in part by DOD/ONR MURI Grant N00014-95-1-0601, and in part by an NSF National Young Investigator Award.

2 Related Work

The recursive structure of the composite rejector constitutes a decision tree (or more generally a directed acyclic graph.) A complete list of references that use such a structure is beyond the scope of this paper, but a small selection is [Henrichon and Fu 69, Payne and Meisel 77, Weng 94]. Also, a relationship can be established between our technique for rejector construction and Fisher's discriminant analysis [Fisher 36, Fukunaga 90]. Whereas our ideas are geared towards the computational efficiency of recognition, discriminant analysis is concerned with representational compactness. Further, there is little work that investigates when discriminant analysis will work, and if so, how much computation will be required. Our results provide insights into these issues.

Connections can also be made between the present work and the large body of work on computationally motivated nearest neighbor classifiers [Friedman et al. 77, Bentley 80, Yianilos 93]. Though the problem we address is somewhat similar, namely, efficient classification, our setting is shown to be more general. The present work attempts to unify ideas from the nearest neighbor literature which is concerned with complexity issues, and the pattern recognition literature which is mainly concerned with representational issues.

3 Theory

3.1 Assumptions and Definitions

A classification decision is based on a finite set of measurements of an underlying physical process. Hence, we assume the existence of a *classification space,* $S = \Re^d$, where d is the number of measurements. Elements, $x \in S$, will be referred to as *measurement vectors*, or for convenience, *vectors.* Next, we assume the existence of a finite collection, $W_1, W_2, \ldots, W_n$, of *classes*; that is subsets of S. The classes themselves are defined by the application in question. We will therefore assume that the classes are given to us. Now we can define a classifier:

Definition 1 *A* classifier *is an algorithm, ϕ, that given an input, $x \in S$, returns the class label i for which $x \in W_i$.*

A *rejector* is a generalization of a classifier in the sense that it returns a set of classes. This set must contain the correct class, but may also contain others:

Definition 2 *A* rejector *is an algorithm, ψ, that given an input, $x \in S$, returns a set of class labels, $\psi(x)$, such that $x \in W_i \Rightarrow i \in \psi(x)$ or equivalently $i \notin \psi(x) \Rightarrow x \notin W_i$.*

The *rejection domain* for W_i, is the set of all $x \in S$ for which i does not appear in the rejector output:

Definition 3 *If ψ is a rejector, and W_i is a class, then the* rejection domain, R_i^ψ, *of ψ, for class W_i is the set of $x \in S$ for which $i \notin \psi(x)$.*

Then, the following important properties hold:

1. From Definitions 2 & 3, the rejector, ψ, eliminates W_i from further consideration if $x \in R_i^\psi$.
2. Subject to $R_i^\psi \subseteq \overline{W_i}$, we are free to choose the rejection domains and still conform with the correct definition of a rejector. This freedom to choose rejection domains with "simple" decision boundaries, is what allows rejectors to be efficient.

3.2 Rejection Based Classifiers

Applying a rejector does not guarantee that we will be able to uniquely answer the classification question, since there may be more than one class in the output of the rejector. We deal with this potential ambiguity by adding a verification stage:

Definition 4 *A* verifier *for a class W_i is a boolean algorithm which, given an input, $x \in S$, returns the result, 1, if x is a member of W_i, and 0 otherwise.*

We form a *rejection-based classifier* by first applying a rejector, ψ, and then applying a verifier for each class, $i \in \psi(x)$. Combining the results, we can classify the input, $x \in S$. The efficiency of our rejection-based classifier, ϕ^{rb}, can be shown [Baker and Nayar 95] to be:

$$T_{av}(\phi^{rb}) = T_{av}(\psi) + E_{x \in S}(|\psi(x)|) \cdot T_{ver} \qquad (1)$$

where, $T_{av}(\phi^{rb})$ is the average run time of the rejection-based classifier, $T_{av}(\psi)$, is the average run time of the rejector, $E_{x \in S}(|\psi(x)|)$ is the expected cardinality of the rejector output, and T_{ver} is the run time of each of the verifiers (assumed to be the same for all verifiers.) We now introduce a further definition:

Definition 5 *If ψ is a rejector, we define the* effectiveness *of ψ by:* $\mathrm{Eff}(\psi) = \frac{E_{x \in S}(|\psi(x)|)}{n}$

Note that a small numeric value of $\mathrm{Eff}(\psi)$ corresponds to an "effective" rejector. Then, Equation (1) shows that our rejection-based classifier is efficient when, (a) rejection is *efficient,* and (b) rejection is *effective.*

3.3 Composite Rejectors

Applying a rejector results in a subset of classes, and so a smaller instance of the original classification problem. Recursively applying a rejector, tuned to the smaller subset of classes, may enable us to further narrow down the set of classes under consideration. This leads to the notion of a *composite rejector:*

Definition 6 *A* composite rejector, Ψ, *is a collection of rejectors, $\Psi = \{\psi_\imath : \imath \in \Im\}$, where $\Im$ is an index set for Ψ, and such that, (a) there is a rejector in Ψ designed for the complete set of classes, and (b) for any rejector, $\psi_\imath \in \Psi$, and any $x \in S$, either $\psi_\imath(x) = 1$ or there is a rejector in Ψ designed for $\psi_\imath(x)$.*

Intuitively, the recursive structure of the composite rejector leads us to expect logarithmic complexity. Sufficient conditions to prove such a result are as follows:

1. For all $\psi_\imath \in \Psi$, and for all $x \in S$, either $|\psi_\imath(x)| = 1$, or at least one class is eliminated by $\psi_\imath$.
2. With respect to the underlying *a priori* probability density function from which the measurement vectors are drawn, the events, $\{x : \ x \notin R_j^{\psi_\imath}\}$, are mutually independent.
3. The effectiveness of all of the rejectors is the same: $\forall \imath \in \Im$, $\mathrm{Eff}(\psi_\imath) = \mathrm{E}$, say.

Then, we can show (see [Baker and Nayar 95]) that a rejection-based classifier using a composite rejector runs in time:

$$T_{av}(\phi^{rb}) \leq \lceil \log_{\mathrm{E}^{-1}} n \rceil \cdot T_{rej} + 2 \cdot T_{ver} \qquad (2)$$

where, T_{rej} is the run time of each of the rejectors (assumed constant), and n is the number of classes.

One potential problem with the composite rejector is that the number of rejectors within Ψ may be very large, possibly as large as 2^n. To avoid this exponential growth, we impose constraints on each $\psi_\imath$. We require that: (a) for each $\psi_\imath \in \Psi$, the number of different possible outputs is two, (b) the two possible output subsets of classes are of equal cardinality, and (c) the intersection between the two outputs consists of at most a fraction, $\epsilon \in [0, 1)$, of the classes for which the rejector was constructed. Then, if we denote by $M(n)$, the maximum number of rejectors in Ψ that may be reached after, and including, the rejector constructed for a collection of n classes, then it can be shown that:

$$M(n) \leq n^{1/(1-\log_2(1+\epsilon))} \Leftrightarrow 1 \qquad (3)$$

In practice, it may not be straightforward to completely satisfy the three requirements stated above. However, the following three design criteria may be used as guidelines while implementing each rejector in the composite rejector Ψ: (a) avoid rejectors that produce a large number of outputs, (b) attempt to balance the output cardinalities, and (c) minimize the overlap between the outputs.

3.4 Construction of Rejectors

In this section, we assume that the norm of a measurement vector is unimportant for classification purposes, so we restrict attention to the surface of the unit ball, $B = \{x \in S : \ \|x\|_2 = 1\}$.

The design of a rejector is equivalent to deciding on the rejection domains. Since, we require $R_i^{\psi} \subseteq \overline{W_i}$, this choice depends on the nature of the underlying classes. Hence, we make the following assumption about the classes, illustrated in Figure 1:

The class assumption *For each W_i, there exists a $c_i \in S$, a linear subspace, $L_i \subseteq S$, and a threshold, $\delta_i \geq 0$, such that $\forall x \in W_i$, $dist(x, c_i + L_i) \leq \delta_i$. Further we assume: (a) $dim(L_i) \ll d$, and (b) $\delta_i \ll 1$.*

The class assumption[1] is approximately equivalent to assuming that the application of the K-L expansion results in a compact and accurate representation of the class. Suppose that M_i^k is the subspace spanned by the k most important K-L eigenvectors, and λ_i are the decaying K-L eigenvalues, then we have:

$$E_{x \in W_i}(\mathrm{dist}(x, E_{y \in W_i}(y) + M_i^k)^2) = \sum_{s=k+1}^{d} \lambda_s \approx 0. \qquad (4)$$

Setting c_i to be $E_{x \in W_i}(x)$, L_i to be M_i^k, we see that the difference between the class assumption and the K-L expansion is one of expected versus maximum value. The widespread use of the K-L expansion allows us to argue that the class assumption can be expected to hold extensively.

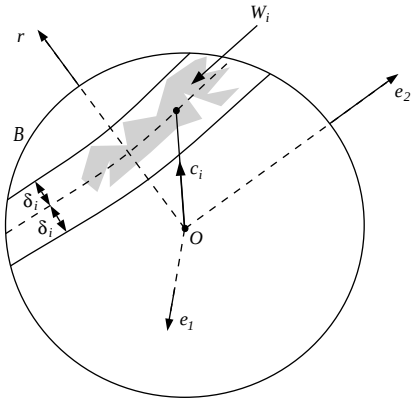

Figure 1: An illustration of the class assumption for a low dimensional example, $S = \Re^3$. The subspace, L_i, is the 2 dimensional subspace spanned by the vectors, $\{e_1, e_2\}$. Every vector in W_i can be approximated to within error, δ_i, by the linear combination of c_i and a vector in L_i.

Starting from the class assumption, we now derive a general form for a rejector. We begin by defining the notion of a *rejection vector*:

Definition 7 *Suppose the class assumption holds for the classes, $W_1, W_2, \ldots, W_n$. Then a* rejection vector *is a unit vector, $r \in B$, for which $r \perp \bigoplus_{i=1}^{n} L_i$.*

If r is a rejection vector it follows immediately from orthogonality and the Cauchy-Schwarz inequality, that:

$$x \in W_i \ \Rightarrow \ |\langle r, x\rangle \Leftrightarrow \langle r, c_i\rangle| \leq \delta_i \qquad (5)$$

Equation (5) means that the rejection vector projects each class onto approximately a point. So long as the points, $\langle r, c_i\rangle$, are well separated, the intervals $[\langle r, c_i\rangle \Leftrightarrow \delta_i, \langle r, c_i\rangle + \delta_i]$ will not intersect. So, we can use this equation to discriminate[2] between the classes. Based on this fact, we define a *derived rejector*:

Definition 8 *Given that the class assumption holds for the classes, $W_1, W_2, \ldots, W_n$, and that $r \in B$ is a rejection vector, then we define the* derived rejector, *ψ_r by: $i \in \psi_r(x) \ \Leftrightarrow \ |\langle r, x\rangle \Leftrightarrow \langle r, c_i\rangle| \leq \delta_i$*

[1] The class assumption is very general and allows various "shapes," including disconnected multi-cluster distributions.

[2] There is no guarantee that we will be able to find a rejection vector that will completely distinguish between a given pair of classes, for example when the two classes' convex hulls overlap. This fact need not effect the usefulness of a derived rejector, since the goal of a rejector is to eliminate most of the classes, not necessarily all.

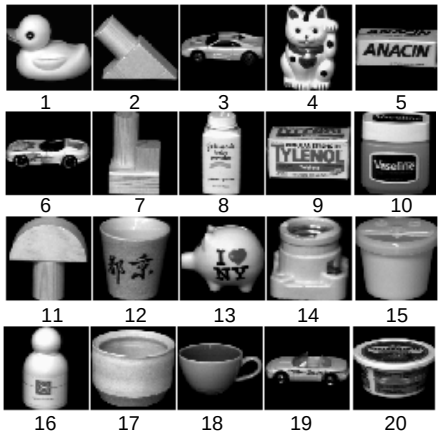

Figure 2: The 20 objects used in the recognition experiment. There are 72 images of each object, with each pair of images separated by 5° of pose. The data set was provided by H. Murase and S.K. Nayar and is that used in [Murase and Nayar 95].

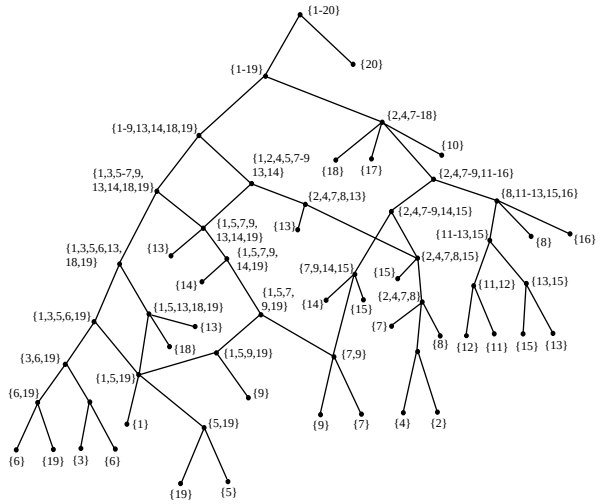

Figure 3: A representation of the composite rejector. Each interior node denotes a single rejector, and is labeled with the set of objects that it is designed to act on. At each node, only one dot product and a couple of comparisons need to be performed. Each leaf denotes a possible output of the composite rejector.

We have some freedom in choosing the direction of the rejection vector, since it is simply constrained to lie orthogonally to $\bigoplus_{i=1}^{n} L_i$. We should choose r to optimize the effectiveness of the derived rejector:

Definition 9 *An* optimal rejection vector,[3] r_{opt}, *is a rejection vector which minimizes* Eff (ψ_r)

We are unable to exactly calculate the optimal rejection vector, however [Baker and Nayar 95] contains the details of an algorithm to approximate one.

4 Practical Issues and Implementation

While implementing the proposed ideas, several practical issues must be addressed, including verifying the class assumption, approximating the optimal rejection vector, estimating the δ_i, and avoiding the exponential explosion in the size of the composite rejector. For lack of space, we refer the interested reader to [Baker and Nayar 95] for the details.

5 Example Applications

5.1 3D Object Recognition

We follow the appearance matching approach, first described in [Murase and Nayar 95]. Object recognition is reduced to pattern recognition by first segmenting the object and then scale normalizing it to an image of size 128×128 pixels. The image is then treated directly as a 16,384 dimensional measurement vector in the classification space, S, reading the pixels in a raster scan fashion. The vector is finally intensity normalized to lie on the unit sphere, B.

The data set that we used (see Figure 2) consists of 20 objects (classes). It contains 72 images of each object separated by 5^o intervals of pose. The images were divided into a training and a test set each comprising 36 images of each object. The training set for each object is then treated as samples of the corresponding class and used to implement the composite rejector, a representation of which is presented in Figure 3. As it happens, every leaf of the composite rejector contains a single class, hence the composite rejector can fully discriminate between the 20 objects. (We would have regarded the rejector as successful even if each leaf had contained 2-3 objects.)

We found the composite rejector responded 100% correctly for both the training and test sets. We calculated the average number of rejectors which we need to apply to be just 6.43, based on the assumption that each image in the data set is equally likely to appear. Since the cost of making the decision at each node is essentially the cost of one inner product (convolution), the efficiency compares very favorably with the results obtained by Murase and Nayar [Murase and Nayar 95]. Their implementation required 20 inner products, followed by a sophisticated search procedure.

We investigated the growth rate of the number of rejectors required as a function of the number of classes. The results, in Figure 4, validate the hypothesized logarithmic growth in time complexity.

5.2 Local Feature Detection

We constructed a composite rejector for implementing a feature detectors of the type proposed in [Nayar et al. 95]. (The details of the implementation may be found in [Baker and Nayar 95].) The out-

[3] The optimal rejection vector may be related to Fisher's discriminant analysis [Fisher 36]. By working in a space orthogonal to each L_i, we are limiting the within-class scatter. Further, an optimal rejection vector will tend to spread out the points $\langle r, c_i \rangle$, and so will be the one that tends to maximize the between-class scatter.

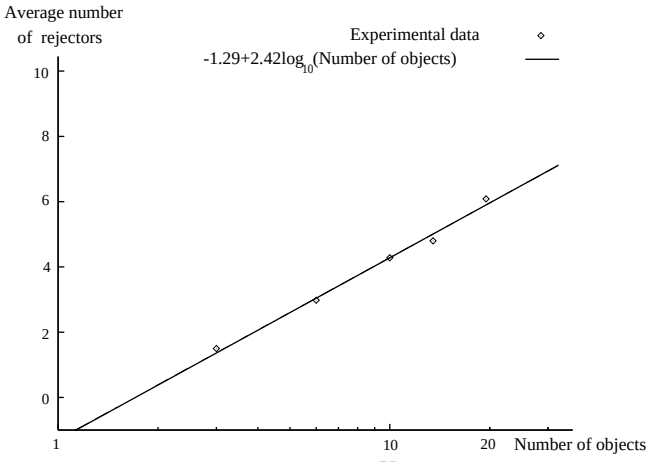

Figure 4: A graph of the number of objects against the average number of simple rejectors required to completely discriminate between those objects. The graph is plotted using a log scale on the abscissa, implying a logarithmic growth rate in the time complexity.

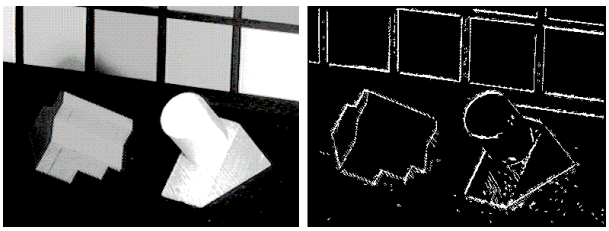

Figure 5: The edge rejector applied to a real image. The output on the right consists of those pixels which our rejection algorithm has quickly decided as worthy of further consideration. On average 1.81 rejectors (each corresponding to a convolution) were applied at each pixel.

put of the composite rejector is used as input to the feature detector, and consists of pixels at which further consideration is worthwhile. Although the technique is applicable to general parametric features, we only have space to display our results (see Figure 5) for edge detection.

6 Discussion

Our primary goal has been to introduce a computational theory of pattern recognition. In this respect we have made considerable progress:

1. We have provided conditions for logarithmic growth in time complexity as a function of the number of classes, and validated the performance empirically. However, further investigation of these conditions is needed to enhance our understanding of when they apply.

2. We analyzed the growth in the number of rejectors required to construct a composite rejector. The key is the number of possible outputs of the rejectors, and the amount of intersection between them. This growth, rather than the time complexity, may well turn out to the limiting factor in the scalability of our approach. A comparison with the much less conservative k-d trees [Friedman et al. 77] would probably enlighten what is essentially a time-space tradeoff.

3. The class assumption is the heart of our technique for constructing rejectors. As expected, it holds for some objects far more than for others, however further study of when and why it holds is required.

References

[Baker and Nayar 95] S. Baker and S.K. Nayar, "A Theory of Pattern Rejection," *Columbia University Technical Report,* CUCS-013-95, 1995.

[Bentley 80] J.L. Bentley, "Multidimensional divide-and-conquer," *Communications of the ACM,* 23:214–229, 1980.

[Fisher 36] R.A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics,* 7:179–188, 1939.

[Friedman et al. 77] J.H. Friedman, J.L. Bentley, and R.A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Transactions on Mathematical Software,* 3:209-226, 1977.

[Fukunaga 90] K. Fukunaga, *Statistical Pattern Recognition*, Academic Press, 1990.

[Henrichon and Fu 69] E.G. Henrichon and K-S. Fu, "A Nonparametric Partitioning Procedure for Pattern Classification," *IEEE Transactions on Computers,* 18:614–624, 1969.

[Murase and Nayar 95] H. Murase and S.K. Nayar, "Visual Learning and Recognition of 3D Objects from Appearance," *International Journal of Computer Vision,* 14:5–24, 1995.

[Nayar et al. 95] S.K. Nayar, S. Baker, and H. Murase, "Parametric Feature Detection," *Columbia University Technical Report,* CUCS-028-95, 1995.

[Payne and Meisel 77] H.J. Payne and W.S. Meisel, "An Algorithm for Constructing Optimal Binary Decision Trees," *IEEE Transactions on Computers,* 26:905–916, 1977.

[Weng 94] J. Weng, "SHOSLIF: The Hierarchical Optimal Subspace Learning and Inference Framework," *Michigan State University Technical Report*, CPS 94-15, 1994.

[Yianilos 93] P.N. Yianilos, "Data Structures and Algorithms for Nearest Neighbor Search in General Metric Spaces," *Proc. of ACM-SIAM Symposium on Discrete Algorithms,* pp. 311–321, 1993.