Machine Learning, 9, 275-308 (1992)

Explorations of an Incremental, Bayesian Algorithm for Categorization

JOHN R. ANDERSON AND MICHAEL MATESSA JA0S@ANDREW.CMU.EDU
Department of Psychology, Carnegie Mellon University, Pittsburgh, PA 15213

Editor: Dennis Kibler

Abstract. An incremental categorization algorithm is described which, at each step, assigns the next instance to the most probable category. Probabilities are estimated by a Bayesian inference scheme which assumes that instances are partitioned into categories and that within categories features are displayed independently and probabilistically. This algorithm can be shown to be an optimization of an ideal Bayesian algorithm in which predictive accuracy is traded for computational efficiency. The algorithm can deliver predictions about any dimension of a category and does not treat specially the prediction of category labels. The algorithm has successfully modeled much of the empirical literature on human categorization. This paper describes its application to a number of data sets from the machine learning literature. The algorithm performs reasonably well, having its only serious difficulty because the assumption of independent features is not always satisfied. Bayesian extensions to deal with nonindependent features are described and evaluated.

Keywords. Bayesian inference, concept learning, human learning, incremental algorithms

1. Introduction

We have been engaged in a project to understand human categorization which has led us to develop a machine learning algorithm. Our research began as an exploration of the issue of whether human categorization can be considered optimal. We were interested in this both as a philosophical issue and as a practical means for predicting human behavior. As to the philosophical score, if human categorization can be shown to be optimal this would be further evidence for the view that human cognition in general is strongly adapted to its environment. As a practical matter, if optimal, one can predict human categorization by investigating what is optimal in a particular categorization situation, thus bypassing the traditional path of proposing specific cognitive mechanisms and all the murky issues of identifiability that come with a mechanistic approach (Anderson, 1990).

To pursue the issue of whether human cognition is optimal requires specifying two things. First, we need a definition of optimality. Second, we need a specification of the structure of the environment so we can determine what behavior is optimal in that environment. These are the first two issues that we will address in this paper.

1.1. Preliminary definition of optimization

Our assumption has been that the goal of categorization is to predict unknown features of various objects that we encounter. For instance, when one sees a creature on a path one would like to predict whether it is dangerous or not. One can gain accuracy in prediction of certain features by identifying the category (e.g., tiger) from which the object comes. Optimal prediction behavior is behavior that achieves a maximal tradeoff between accuracy of prediction and cost of computing the prediction. It is clear we need this trade-off. An exquisitely accurate estimate of the danger of this object would do no good if it took hours to compute. It is the constraint of minimizing computation that leads to a concern with the efficiency of the algorithm for computing the prediction.

Thus, there are the issues of how to measure accuracy, computational cost, and how to combine them. With respect to accuracy, we have adopted in this paper the goal of minimizing absolute error. In the case of predicting discrete features, this comes down to predicting the most probable value. In the case of predicting features with continuous normal distributions, this comes down to predicting the mean value. In terms of a Bayesian decision framework (e.g., DeGroot, 1970), one might not always want to minimize accuracy in terms of absolute error. For instance, one might not always want to predict the most probable discrete value. A possible example of this is treating an animal as dangerous even if it is more likely friendly because the cost of misclassifying a dangerous animal as friendly is greater than the cost of misclassifying a friendly animal as dangerous. However, since we do not have such complex utility metrics available in our applications, we have opted for minimizing absolute error.

With respect to computational cost we have chosen to focus on minimizing time. This ignores potentially relevant considerations such as space but time is generally viewed as a more precious commodity in the human case. It is also the case that the steps we will take to minimize time will also substantially reduce storage costs. Minimizing time is a somewhat underspecified goal and will require further statement of the constraint under which the minimization takes place. We will develop these later in the paper.

To have a precise definition of optimization, we need a rule for combining error and time to come up with a total cost. Assuming each unit of time has a cost *a* and each unit of error has a cost *b*, the total cost should be cast as a weighted sum of time and error—i.e., a function of the form aT + bE where T is time and E is absolute error. Before we can more precisely specify time or error, we need to discuss the structure of the environment.

1.2. *The structure of the environment*

Our theory of the structure of the environment has been focused on the structure of living things (arguably, the largest portion of the objects in the world) because of the aid biology gives in objectively specifying the organization of these objects. In particular the theory developed rests on the structure of living objects produced by the phenomenon of species. Species form a nearly disjoint partitioning of the living things because of the inability to interbreed between species. Within a species there is a common genetic pool which means that individual members of the species will display particular feature values with probabilities that reflect the proportion of that phenotype in the population. Another useful feature of species structure is that the display of features within a freely-interbreeding species is largely independent. For instance, there is little relationship between size and color in freely-interbreeding species where those two dimensions vary. Thus, the critical aspects of speciation is the disjoint partitioning of the object set and the independent probabilistic display of features within a species.

An interesting question is whether other types of objects display these same properties. The other common type of object is the artifact. Artifacts approximate a disjoint partitioning but there are occasional exceptions—for instance, mobile homes which are both homes and vehicles. Other types of objects (stones, geological formations, heavenly bodies, etc.) seem to approximate a disjoint partitioning but here it is hard to know whether this is just a matter of our perceptions or whether there is any objective sense in which they do. One can use the understanding of speciation for living creatures and understanding of the intended function in manufacture in the case of artifacts to objectively test the hypothesis of disjoint partitioning.

In the case of our psychological applications we try to argue that this characterization of the universe is approximately correct for most domains humans face. However, in the context of a machine learning paper we rather take the stance that we are describing a learning algorithm which is optimal in the case that these assumptions about the structure of the domain are satisfied. Certainly, there will be domains (real or made-up) where categories exist that violate the assumption of independent display of features. It is also the case that there will be relationships among features that cannot be captured by a disjoint partitioning. So, for instance, what happens to an object when it is thrown is to be predicted by the physics of the throwing and not by its category membership. As with all learning algorithms, the current one works for certain domains. In our case we start with a specification of what these domains are.

One thing to stress about this characterization of the universe is that it sees nothing special about category labels. Category labels are just another feature one might want to predict about an object. There is nothing logically different about predicting an object is called a tiger than predicting it is dangerous. Because of the arbitrariness of labels and their large possible number, one cannot have strong priors about what an object will be called in contrast to some dimensions. As we will see this means that category labels are distinguished from some other dimensions in terms of the parameters of their treatment but it does not mean that they are logically any different.

2. Algorithms for prediction

Recall that we want an algorithm that minimizes a weighted sum of time and error. We will start with what we call the ideal algorithm which produces an absolute minimum in error. Then we will describe an incremental algorithm which locally trades off accuracy in prediction for computation time. Finally, we will describe what we call a radical incremental algorithm which is the extreme of the iterative algorithm which is achieved by placing a very large premium on minimizing computational time. This last is the algorithm that was used to obtain most of the results reported in this paper.

2.1. The ideal algorithm

Given this specification of the goal of categorization and the structure of the environment we can proceed to sketch the ideal prediction algorithm if computational considerations were not an issue. This would be to consider all the different ways the objects seen so far could be broken up into categories, determine the probability of each such partitioning, and use this to weight the probability that the object will display a particular feature if that were the partition. Formally, this amounts to calculating:

$$g_i(y|\mathrm{F_n}) = \sum_x P(x|\mathrm{F_n})f_i(y|x) \qquad (0)$$

where $g_i(y|F_n)$ is the function specifying the probability an object will display a value y on a dimension i given F_n the observed feature structure of all the objects. The summation is across all possible partitionings x of the n objects into disjoint sets, $P(x|F_n)$ is the probability of partitioning x given the objects display feature structure F_n, and $f_i(y|x)$ is the function giving the probability the object in question would display value y on dimension i if x were the partition. Anderson (1990) describes a Bayesian scheme for estimating $P(x|F_n)$ and $f_i(y|x)$ such that equation (0) can be used to estimate the true posterior distribution of $g_i(y|F_n)$. Given the information of the distribution $g_i(y|F_n)$, one can then select a value of y that will minimize expected absolute error. However, the scheme described in Anderson (1990) is not of much interest because the cost of calculating equation (0) is unacceptably high. The problem with this algorithm is that the number of partitions of n objects grows exponentially as the Bell exponential number (Berge, 1971). Thus, the computational cost grows exponentially with the number of objects to be classified.

There is another aspect of this algorithm which is unacceptable for its intended applications. This is that it does not make commitment to any specific hypothesis about the categorical structure of the experienced objects. This contradicts our common experience of seeing objects as belonging to a specific category and is in conflict with the goal of most machine learning programs for categorization which are also trying to come up with a specific categorization.

2.2. Incremental algorithms

At the other extreme from this ideal algorithm are the radical incremental algorithms such as those of Fisher (1987) and Lebowitz (1987). Rather than maintaining all possible partitionings of the object set just a single partitioning is maintained. When a new object comes in, different extensions of that partitioning are considered to accommodate that object. One extension is selected as best by some criterion and becomes the new partitioning. This will be the kind of algorithm that we will use but it is hard to see directly how such an algorithm relates to the optimality criterion that we sketched out earlier. However, one can see under what assumptions these algorithms might be ideal if one first considers a class of incremental algorithms which are intermediate between the ideal algorithm and these radical incremental algorithms. Rather than keeping all partitions or just one, these algorithms maintain some number of partitionings, constantly pruning away partitions which are judged not to be worth their computational cost.

Consider a system which is looking at a set of partitions and is considering whether it is worth deleting one of its partitions. This judgment should be made by comparing the cost of keeping the partition with the contribution of the partition to the overall accuracy of prediction. The cost of a partition is linearly related to the number of categories in the partition. Therefore, the system should delete the partition from the set when

$$n \cdot ccost > L$$

where n is the number of categories in the partition, ccost is the cost per category, and L is the loss in accuracy of prediction. The parameter ccost would be defined as aT/b where a is the weighting of time, b the weighting of accuracy, and T the time per category. Thus, it reflects the ratio of the cost of processing a category to the cost associated with a unit loss in accuracy.

Let us assume that the loss, L, is proportionate to the reduction in the accuracy of the prediction caused by removal of partition t. With the partition in, the prediction is given by equation (0). When we remove the partition t the prediction becomes:

$$g_i'(y|F_n) = \frac{\sum_{x \neq t} P(x|F_n) f_i(y|x)}{1 - P(t|F_n)}$$

The loss in accuracy can be shown to be

$$L = g_i(y|F_n) - g_i'(y|F_n) = \frac{P(t|F_n)}{1 - P(t|F_n)} (f_i(y|t) - g_i(y|F_n))$$

The value of $g_i(y|t) - g'(y|F_n)$ will depend on the object and feature being predicted. In choosing to delete a partition, one typically does not know what predictions one will be asked to make. Indeed, one is really interested in an expected value of L over all predictions weighted by the probability of having to make that prediction. It hardly seems reasonable to make all these calculations before deleting a partition for the purpose of making these calculations more efficient. Therefore, it seems reasonable to replace $f_i(y|t) - g_i(y|F_n)$ by a single value which is a measure of how far off the prediction for a partition is an average from the true value. Replacing this by a constant pcost gives us the following criterion for deleting a partition:

$$n \cdot cost > \frac{P(t|F_n)}{1 - P(t|F_n)}$$

where cost = ccost/pcost. Thus, it reflects the cost of processing a category, relative to the cost of the average deviation in prediction. This is a criterion which tends to reject low probability partitions with many categories.

One can then replace the ideal algorithm with the following iterative algorithm.

1. Before seeing any objects, the set of category partitionings has a single partitioning and this is the partitioning that contains the empty set of no categories.
2. Given a set of partitionings for the first $n - 1$ objects, a set of partitionings for the first n objects is created as follows:
   (a) For partitionings with m categories, create $m + 1$ new partitionings, each formed by assigning the new object to a different category or by assigning the new object to its own category.
   (b) Calculate for each resulting partitioning x, the probability $P(x|F_n)$.
3. Filter these partitionings as follows:
   (a) Find a partitioning t with the largest value of

$$n{\cdot}cost - \frac{P(t|F_n)}{1 - P(t|F_n)}$$

   and delete it if that value is positive
   (b) Recalculate the probabilities by the formula: $P'(x|F_n) = P(x|F_n)/1 - P(t|F_n)$.
   (c) Go back to step (a) as long as there are partitionings to delete.
4. To predict value y on an unobserved dimension i for the nth object with feature structure F_n use equation (0) with the remaining partitionings.

This essentially describes a beam search in which the algorithm chooses to expand the most probable branches. Each expansion to accommodate a new object is optimal by the criterion given. However, as is true of beam search, this local optimality brings no guarantee of global optimality. That is to say, it is possible that some of the branches that are pruned off would have grown to become the most probable interpretations.

Of course, one cannot pursue all possible partitionings to insure against the danger of missing global optima by maximizing local criteria. The probability that a branch will prove to be a global optima is related to its local probability. Thus, we have already built into our pruning criterion a measure of how likely a partitioning is to prove a global optima. One can view this as part of the *pcost* associated with deleting a partitioning.

2.3. A radical incremental algorithm

If cost is set to 1 or greater in the pruning criterion, then the incremental algorithm will reject all but the most probable partition.[1] Thus, by setting cost to 1, we turn the general incremental algorithm into a radical one that considers only single partitioning at a time. This is an algorithm that places a high premium on accuracy relative to time—that the gain in prediction from including an extra partition is not worth the cost associated with processing even one category in that partition. The radical iterative algorithm can be implemented more simply because there is only one partitioning at a time.

1. Before seeing any objects, the category partitioning of the objects is initialized to be the empty set of no categories.
2. Given a partitioning of the first $n - 1$ objects into categories, calculate for each category k the probability $P(k|F)$ that the nth object comes from category k given that the object has features F. Let $P(0|F)$ be the probability that the object comes from a completely new category.
3. Create a partitioning of the n objects with the nth object assigned to the category with maximum probability.
4. To predict value y on an unobserved dimension i for the nth object with observed features F calculate

$$g_i(y|F) = \sum_k P(k|F)\, f_i(y|k) \tag{1}$$

where $P(k|F)$ is the probability the nth object comes from category k and $f_i(y|k)$ is the probability of an object from category k displaying value y on dimension i.

The basic algorithm is one in which the category structure is grown by assigning each incoming object to the category it is most likely to come from. Thus, a specific partitioning of the objects is produced. Note, however, that the prediction for the new nth object is *not* calculated by determining its most likely category and the probability of y given that category. This calculation is performed over all categories. This gives a much more accurate approximation to the ideal $g_i(y|F_n)$ because it handles situations where the new object is ambiguous between multiple categories. It will give approximately equal weight to these competing categories. Note also equation (1) calculates a probability distribution although for most purposes we will be predicting a single value from that distribution. The full distribution is available should it be needed for some application.

This algorithm has the property that if the partitioning of the first n − 1 objects is the true partitioning, the partitioning of the first n objects will be the maximum likelihood partitioning and the partitioning that will minimize squared error. However, neither guarantee is true for the partitionings assigned after further objects. Of course, the goal of the algorithm is not to identify the maximum likelihood partitioning but the partitioning that will yield the closest approximation to the predictions of the ideal algorithm. This is frequently not the maximum likelihood partitioning but usually one of considerably greater than average probability. In the fourth section of this paper, we will consider the consequences of choosing one high probability partitioning rather than another.

3. Probability calculations

It remains to come up with formula for calculating $P(k|F)$ and $f_i(y|k)$ in equation (1). Since $f_i(y|k)$ turns out to be involved in the definition of $P(k|F)$, we will start with $P(k|F)$. In Bayesian terminology $P(k|F)$ is a posterior probability that the object belongs to category k given that it has feature structure F. Bayes' formula can be used to express this in terms of a prior probability $P(k)$ of coming from category k before the feature structure is inspected and a conditional probability $P(F|k)$ of displaying the feature structure F given that it comes from category k.

$$P(k|F) = \frac{P(k)P(F|k)}{\sum_j P(j)P(F|j)} \tag{2}$$

where the summation in the denominator is over all the categories j currently in the partitioning and a potential new one. This then focuses our analysis on the derivation of a prior probability $P(k)$ and a conditional probability $P(F|k)$.

3.1. Prior probability

With respect to prior probabilities, the critical assumption is that there is a fixed probability c that two objects come from the same category and this probability does not depend on the number of objects seen so far or the position of these objects in the sequence. This is called the coupling probability. If one takes this assumption about the coupling probability between two objects being independent of the other objects and generalizes it, one can derive a simple form for $P(k)$ (See Anderson, 1990, for the derivation):

$$P(k) = \frac{cn_k}{(1 - c) + cn} \tag{3}$$

where c is the coupling probability, n_k is the number of objects assigned to category k so far, and n is the total number of objects seen so far. Note for large n this closely approximates n_k/n which means that we have a strong base rate effect in these calculations with a bias to put new objects into large categories. Presumably the rational basis for this is apparent.

We also need a formula for $P(0)$ which is the probability that the new object comes from an entirely new category. This is

$$P(0) = \frac{(1 - c)}{(1 - c) + cn} \tag{4}$$

For large n this closely approximates $(1 - c)/cn$ which is again a reasonable form—i.e., the probability of a new category depends on the coupling probability and number of objects seen. The greater the coupling probability and the more objects, the less likely it is that the new object comes from a new category.

3.2. Conditional probability

Within a category we will consider the probability of displaying features on various dimensions to be independent of the probabilities on other dimensions. Then we can write

$$P(F|k) = \prod_i f_i(y|k) \tag{5}$$

The reader will recognize $f_i(y|k)$ from equation (1) which is the probability of displaying value y on dimension i for an object which comes from category k. The independence assumption in equation (5) is reasonably justified for freely interbreeding species. It is less clear how well justified it is for other categories.

This independence assumption does not prevent us from recognizing categories with correlated features. Thus, we may know that being black and retrieving sticks are highly correlated for Labradors. This would be represented by high probabilities of the stick-retrieving and the black features in the Labrador category.[2] What the independence assumption prevents us from doing is representing categories where values on two dimensions are either both one way or both the opposite. Thus, it would prevent us from recognizing a single category of animals which were either large and fierce or small and gentle, for instance. The algorithm would create two categories in the face of such a structure.

The effect of equation (5) is to focus us down on an analysis of the individual $f_i(y|k)$. Derivation of this quantity is itself an exercise in Bayesian analysis. A special case derivation for a discrete dimension is described in Anderson (1990). Here we will describe a more general derivation. We will indicate the major mathematical steps in this derivation for the discrete case to show how the Bayesian analysis works. We will not give the mathematical detail for the derivation of the continuous case which is more complex. There we will just state the final result.

3.3. Discrete dimensions

It is assumed that there is some prior probability density for the probabilities p_j of displaying value j. Note $\Sigma\ p_j = 1$. The typical prior density for this is the Dirichlet density (Berger, 1985):

$$f_D(p_1, p_2, \ldots, p_m|\alpha_1, \alpha_2, \ldots, \alpha_m) = \frac{\Gamma(\alpha_0)}{\prod_{j=1}^{m} \Gamma(\alpha_j)} \prod_{j=1}^{m} p_j^{\alpha_j - 1} \tag{6}$$

where $\alpha_0 = \Sigma\ \alpha_j$ and the gamma function, Γ, is defined as in Beyer (1987).[3] In this distribution the mean expected value for p_j is α_j/α_0.

The next step in a Bayesian analysis is to specify the conditional probability of the observed distribution of values on dimension i given a set of probabilities p_j. Let c_1, c_2, $\ldots$, c_m be frequency counts for the number of objects showing each of the m values on dimension i. What we have observed is n multinomial trials corresponding to the objects and the probability of this sequence is described by

$$f_M(c_1, c_2, \ldots, c_m | p_1, p_2, \ldots, p_m) = \binom{n}{c_1 c_2 \ldots c_m} \prod_{j=1}^{m} p_j^{c_j} \tag{7}$$

What we next need to do is to calculate the posterior distribution of the p_j given the observed c_j. This is calculated by the standard Bayesian formula for posterior densities:

$$f(p_1, p_2, \ldots, p_m | c_1, c_2, \ldots, c_m) =$$

$$\frac{f_M(c_1, c_2, \ldots, c_m | p_1, p_2, \ldots, p_m)\, \mathrm{f_D}(p_1, p_2, \ldots, p_m | \alpha_1, \alpha_2, \ldots, \alpha_m)}{\int_0^1 \int_0^{1-p_1} \cdots \int_0^{1-p_1-\ldots-p_{m-2}} f_M(c_1, c_2, \ldots, c_m | p_1, p_2, \ldots, p_m) f_D(p_1, p_2, \ldots, p_m | \alpha_1, \alpha_2, \ldots, \alpha_m) dp_1, dp_2 \ldots dp_{m-1}}$$

$$= f_D(p_1, p_2, \ldots, p_m | \alpha_1 + c_1, \alpha_2 + c_2, \ldots, \alpha_m + c_m) \tag{8}$$

The posterior distribution of probabilities is also a Dirichlet distribution but with parameters $\alpha_j + c_j$ (Berger, 1985).[4] This implies that the mean expected value of displaying value j on dimension i is $(\alpha_j + c_j)/\Sigma_j(\alpha_j + c_j)$. This is $f_i(j|k)$ for equation (5):

$$f_i(j|k) = \frac{c_j + \alpha_j}{n_k + \alpha_0} \tag{9}$$

where n_k is the number of objects in category k which have a value on dimension i and c_j is the number of objects in category k with the same value as the object to be classified. For large n_k this approximates c_j/n_k which one frequently sees promoted as the rational probability. However, it has to have this more complicated form to deal with problems of small samples. For instance, if one has just seen one object in a category and it has had the color red, one would not want to guess that all objects are red. If there were seven colors equally probable on prior grounds and the α_j were 1, the above formula would give 1/4 as the posterior probability of red and 1/8 for the other six colors unseen as yet. It is an interesting question how to set the α_j. In most cases there is no basis entertaining a strong belief as to possible values of these parameters. There are a number of conventions in the Bayesian literature for setting non-informative priors in such cases. In this paper, we have chosen to set the α_j at 1.0 which is probably the most commonly practiced convention (see Berger, 1985; Lee, 1989). The one exception in the paper concerns category labels. Here we have set α_j to be equal but at a much lower value of .01 to reflect the very large number of possible labels and very weak prior basis for believing any would be used. Setting of $\alpha_j = 1.0$ means our priors are given the same weighting as a single observation while setting $\alpha_j = .01$ means our priors are given 1/100 the weighting of an empirical observation.

3.4. Continuous dimensions

Below we will briefly state what is probably the most useful Bayesian analysis for continuous distributions (for details, see Lee, 1989). The natural assumption is that the variable is distributed normally and the induction problem is to infer the mean and variance of that distribution. In standard Bayesian inference methodology, we must begin with some prior assumptions about what the mean and variance of this distribution is. It is unreasonable to suppose we can know in advance precisely what either the mean and variance will be. Our prior knowledge must take the form of probability densities over possible means and variances. This is basically the same idea as in the discrete case where we had a Dirichlet distribution giving priors about probabilities of various values. The major complication is the need to separately state prior distributions for mean and variance.

One suggestion for the prior distributions is that the variance Σ^2 is distributed according to an inverse chi-square distribution or more specifically,

$$\Sigma^2 \sim a_0\sigma_0^2\chi_{a_0}^{-2}$$

where σ_0^2 reflects the mean prior variance and a_0 reflects the confidence in that prior variance. The obvious suggestion for the prior distribution of the mean, M, is that it has a normal distribution. One manifestation of this is the following assumption:

$$M \sim N\left(\mu_0, \frac{\sigma_0}{\sqrt{\lambda_0}}\right)$$

where μ_0 is the prior mean and λ_0 reflects confidence in this prior.

Given these prior distributions, the probability of displaying value y on dimension i in category k, after n observations, has the following t distribution:

$$f_i(y|k) = t_{a_i}(\mu_i, \sigma_i\sqrt{1 + 1/\lambda_i}) \tag{10}$$

where a_i are the degrees of freedom, μ_i is the mean, and $a_i\sigma_i^2(1 + 1/\lambda_i)/(a_i - 2)$ is the variance. The parameters a_i, μ_i, σ_i, and λ_i are defined as follows:

$$\lambda_i = \lambda_0 + n \tag{11}$$

$$a_i = a_0 + n \tag{12}$$

$$\mu_i = \frac{\lambda_0\mu_0 + n\bar{y}}{\lambda_0 + n} \tag{13}$$

$$\sigma_i^2 = \frac{a_0\sigma_0^2 + (n - 1)s^2 + \frac{\lambda_0 n}{\lambda_0 + n}(\mu_0 - \bar{y})^2}{a_0 + n} \tag{14}$$

where $\bar{y}$ is the mean of the n observations and s^2 is the variance. These equations basically provide us with a formula for merging the prior mean and variance, μ_0 and σ_0^2, with the empirical mean and variance, $\bar{y}$ and s^2, in a manner that is weighted by our confidences in these priors, λ_0 and a_0.

As with the case of discrete dimensions there arises the issue of how to set the parameters of the model. We have used the following reasonable conventions for setting noninformative priors (see Berger, 1985; Lee, 1989). We set the prior means of the continuous distributions to be the halfway point of the range of all instances and we set the prior variance so that it was equal to the square of a quarter of the range. We set the strengths of belief in the prior mean and variance, a_0 and λ_0, to both be 1. This setting means that we weight our priors as much as we would one empirical observation.

Equation 10 for the continuous case describes a probability density in contrast to equation (9) for the discrete case which gives a probability. The product of conditional probabilities in equation (5) can then be a mixture of probabilities and density values. Basically, equations (5), (9) and (10) give us a basis for judging how similar an object is to the category's central tendency.

4. Properties of the algorithm

Before looking at the application of this algorithm to other data sets, it is worthwhile to consider some of its important properties and their potential consequences. Before doing that it is worthwhile to have an example of the algorithm applying to a data set.

4.1. Illustration of the algorithm

The first experiment in Medin and Schaffer (1978) is a nice one for illustrating the detailed calculations of the algorithm. They had subjects study the following six instances each with binary features:

1 1 1 1 1
1 0 1 0 1
0 1 0 1 1
0 0 0 0 0
0 1 0 0 0
1 0 1 1 0

The first four binary values were choices in visual dimensions of size, shape, color, and number. The fifth dimension reflects the category label. They then presented these 6 objects without their category label plus six new objects also without a label: 0111__, 1101__, 1110__, 1000__, 0010__, and 0001__. Subjects were to predict the missing category label. The two advantages of the Medin and Schaffer data set are that the number of objects is relatively small and so we can do exhaustive analyses and that there is experimental data on humans against which we can compare our algorithm. It is also representative of the kinds of materials used in psychological experiments.

We derived simulations of this experiment by running the program across various random orderings of the stimuli and averaging the results. Figure 1 shows one simulation run where we used the order 11111, 10101, 10110, 00000, 01011, 01000 and had the coupling probability $c = .5$ (see equations (3) and (4)) and set all $\alpha_j = 1$ (see equation (9)).[5]

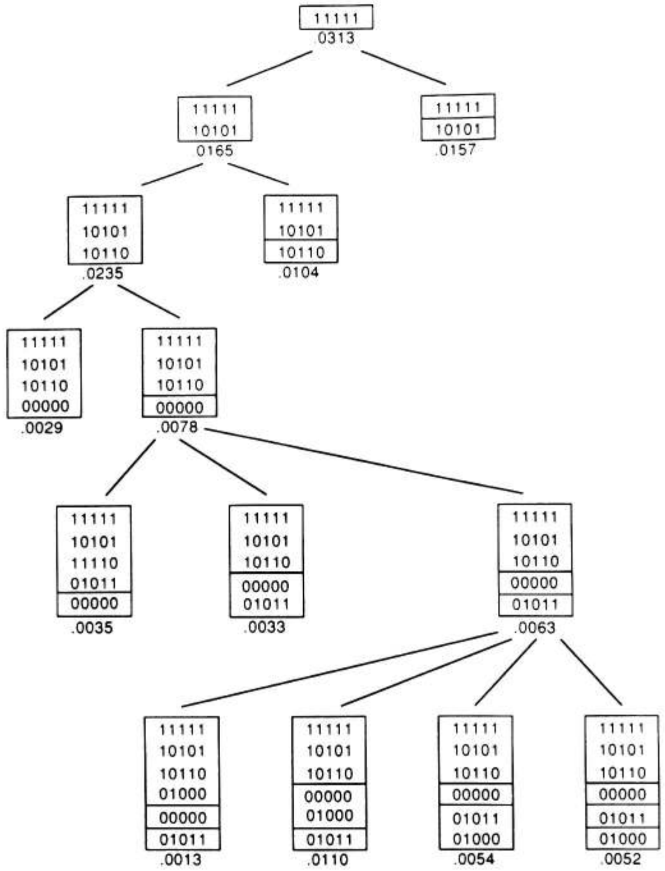

Figure 1. An illustration of the operation of the iterative algorithm in the material from the first experiment of Medin and Schaffer (1978).

What is illustrated in figure 1 is the search behavior of the algorithm as it considers various possible partitionings. The numbers associated with each partition are measures of how probable the new item is given the category to which it is assigned in that partition. These are the values P(k)P(F|k) calculated by equations (3) through (9). Thus, we start out with categorizing 11111 in the only possible way—that is, assigning it to its own category. The probability of this is the prior probability of a 1 on each dimension or $(.5)^5 = .0313$. Then we consider the two ways to expand this to include 10101 and choose the categorization that has both objects in the same category because that is more likely. Each new object is incorporated by considering the possible extensions of the best partition so far. We end up choosing the partition {11111, 10101, 10110}, {00000, 01000}, {01011} which has three categories. Note the systems' categorization does not respect the categorization of Medin and Schaffer. The Medin and Schaffer categorization does not maximize the overall probability of the examples given our parameter values.

Having come up with a particular categorization, we then tested the algorithm by presenting it with the 12 test stimuli and assessing the probabilities it would assign to the two possible values for the fifth dimension which is the label. Figure 2 relates our algorithm to their data. Plotted along the abscissa are the 12 test stimuli of Medin and Schaffer in their rank order determined by subjects' confidence that the category label was a 1. The ordinate is the algorithm's probability that the missing value was a 1. Figure 2 illustrates three functions for different ranges of the coupling probability. The best rank order correlation was gotten for coupling probabilities in the range .2 to .3. In Anderson (1990, 1991) we consistently get best fits to human data setting $c = .3$. This setting of the coupling probability seems rather high. We suspect it is as high as it is to reflect a bias to assign new objects to existing categories to avoid the computational expense associated with a large number of categories.

The reader will note that the actual probabilities of category labels estimated by the model in figure 2 only deviate weakly above and below .5. This reflects the very poor category structure of these objects. With better structured material much higher prediction probabilities are obtained as we will see in the applications to follow.

4.2. *Order sensitivity*

As noted earlier, the algorithm is order sensitive in terms of what partition it will select. This shows up more strongly in cases of relatively unclear category structure such as the Medin and Schaffer material. There are 720 permutations of their six training stimuli and 203 possible partitionings. We ran the radical incremental algorithm over all 720 orders

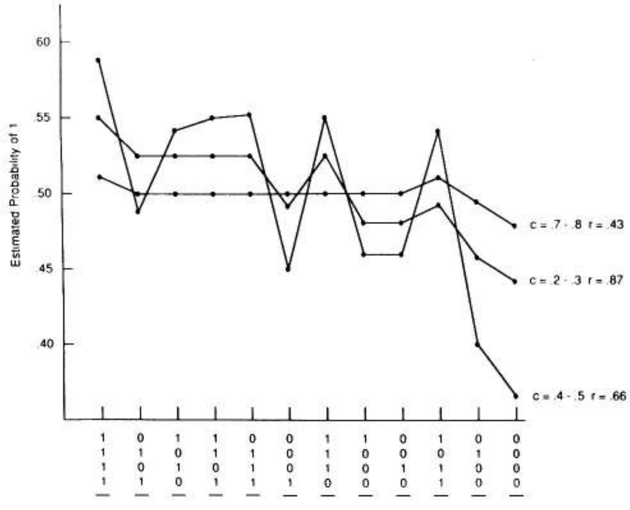

Figure 2. Estimated probability of category 1 for the 16 stimuli in the first experiment of Medin and Schaffer (1978). Different functions are for different ranges of the coupling probability.

using the parameter settings of C = .50 and $a_j = 1$. Depending on the permutations, the incremental algorithm came up with one of the following partitionings:

A. (10101, 10110, 11111) (01011, 00000, 01000)
B. (01011) (00000, 01000) (10101, 10110, 11111)
C. (11111, 01011) (00000, 01000) (10101, 10110)

Partitioning A is the most probable (.046) and is selected for 61% of the presentation orders; B is the second most probable (.041) and is selected 22% of the time; C is eleventh most probable (.019) and is selected 6% of the time. By way of comparison, a partitioning that merges all items into one category is eighth most probable (.020), one that splits them up into six categories is 36th most probable (.008), and the absurd partitionings (11111, 00000), (01000, 10101), (10110, 01011) and (11111, 00000), (01000, 10110), (10101, 01010) are tied for least probable with a probability of .00008.

As this example illustrates, the algorithm tends to uncover the more probable partitionings. The question of interest is how well does the algorithm do in prediction relative to the ideal prediction. We compared predictions for the 12 stimuli in figure 1 using six bases: the predictions of the ideal quantity which can be computed since there are only 203 partitions for 6 stimuli, the predictions from each of the three chosen partitions (A–C), prediction from one of the worst partitions, and the average prediction obtained by weighting each of the three partitions by the frequency with which it is chosen. Table 1 presents the correlations among all these quantities.

These results show that each of the selected partitions correlates reasonably highly with the ideal and that the worst partitions do not correlate well. Interestingly, the most probable does not have the highest correlation with the ideal. This is because it represents less than 5% of the probability. This situation frequently arises in which there are multiple reasonable partitionings and little basis for selecting among them. The model will do well in prediction if it selects any of these.

Thus, it is our conclusion that the order sensitivity of this algorithm is not a problem with respect to its stated goal which is maximizing predictive accuracy while minimizing computational cost. We are not bothered by the fact that there is not always a clearly most probable interpretation nor that we fail to always select the most probable partitioning. This could be distressing if one's goal were to find the best interpretation of the data as is the case with some categorization programs. However, this is not our goal.

Table 1. Correlations among the various basis for predicting the stimuli for Figure 2.

	Ideal	A	B	C	Worst
Partition A	.89	X	X	X	X
Partition B	.89	.49	X	X	X
Partition C	.98	.81	.86	X	X
Worst	.00	.00	.00	.00	X
Average	.96	.92	.78	.96	.00

4.3. Computational performance

The computational time is proportional to the number of categories (p), number of dimensions per category (m), and number of instances (n). Thus, the time to process n instances is proportional to n × m × p. This is the minimum computational function for an algorithm that is going to relate each feature of each object to each category.[6] The major cost is in calculating $f_i(y|k)$ which assigns a probability to a feature-category correspondence. By replacing the *t* distribution (equation (10)) by the more easily calculated normal approximation we are able to reduce computation time by a factor of at least 3.

4.4. Bias in estimation

One of the consequences of maintaining only a single partitioning is that one's categories and predictions can become biased towards extreme values. A simple case to see this is the following: Suppose the instances come from two categories and only have one feature, a continuous dimension. Both categories have normally distributed values with variances 1. One category has mean 0 and the other mean 2. Then, because it only keeps the maximum likelihood interpretation, our algorithm is going to assign all instances with values less than 1 to one category and greater than 1 to another category. This will lead to a systematic bias in the mean of the two distributions (to −.17 and 2.17) and reduction in their variances (to .64) because of the systematic misclassification of all observations close to mean of the other distribution.

However, it does not seem that this has significant consequences. It is a strange situation to be classifying stimuli that only have a single dimension. This situation can not have any predictive consequence since prediction involves using the values on some dimensions to predict the values on other dimensions. The simplest case where this bias might have some effect on prediction is where there are two dimensions and one can use a value on one dimension to predict another.

We explored the situation where there were two categories each with two dimensions. The values for two dimensions for one category had means 0 and variances 1 while for the other category they had means 2 and variances 1. To see asymptotic performance, we exposed our algorithm to a mixture of 10,000 randomly general instances from each category. Then we tested it with another 10,000 instances from each category with the first value present and the second to be predicted.

About 99% of the instances were sorted to be one of two categories with a few odd-ball instances finding their ways to other categories. The means and variances of the two major categories were somewhat biased but not as much as in the one dimensional case. The means were biased .05 rather than .17 in the one-dimensional case and the variances were reduced to .85 rather than .64. Then we explored the consequences for prediction. Figure 3 shows the predictions of the algorithm for various values in the other dimension and compare these values to the true values. It can be seen that the amount of overestimation or underestimation is less than .10.

As more dimensions are added, the tendency to misclassify instances will decrease further and the resulting bias in prediction will decrease further yet. Thus, while this is a problem in principle we do not regard it as a problem in practice.

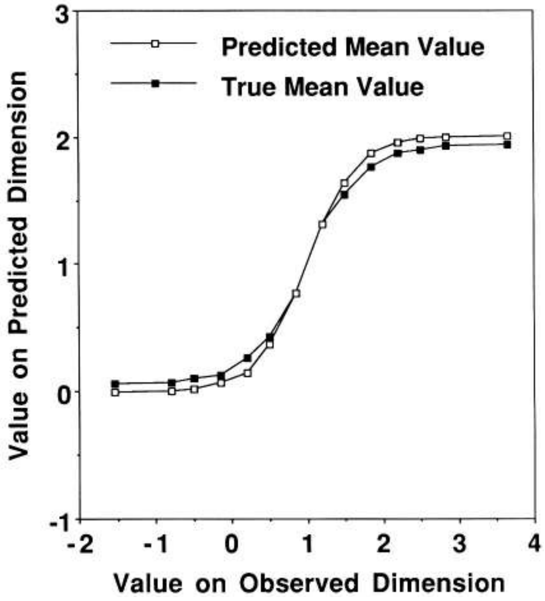

Figure 3. Bias in prediction because of misclassification of two-dimensional stimuli.

5. Psychological modeling

We have enjoyed great success in applying this algorithm to the psychological data and have successfully reproduced every major empirical trend that we have noted. These applications are reviewed in detail in Anderson (1990) and Anderson (1991). Here we will just briefly overview the major empirical trends that we have captured.

Our algorithm makes categorization a function of the distance of an instance from the central tendency of the category just as human subjects do. This holds both for categories defined by discrete dimensions where distance is measured by number of non-majority features an instance displays and for categories defined by continuous dimensions where the measure is Euclidean distance from the mean. In addition to sensitivity to central tendency, human subjects and our model are sensitive to the existence of individual instances or clusters of instances different from the overall central tendency of a category. Both subjects and the model will more reliably classify test instances close to these deviant instances than they do other instances equally distant from the overall central tendency. In the model this is because a separate category is grown to accommodate the deviant instances.

The model often forms multiple internal categories to correspond to items that are assigned a single categorical label by an experimenter. This enables it to capture correlations that exist within an official category. The experiment by Medin, Altom, Edelson, and Freko (1982) is a nice one for illustrating this. They had subjects study the 9 cases in table 2 which were all supposed to represent instances from one disease category, burlosis. This

Table 2. Symptoms of burlosis (from Medin, et al. (1982)).

Case Study	Blood Pressure	Skin Condition	Muscle Condition	Condition of Eyes	Weight Condition
1. R.L.	0	1	0	1	1
2. L.F.	1	1	0	1	1
3. J.J.	0	0	1	1	1
4. R.M.	1	0	1	1	1
5. A.M.	1	1	1	1	1
6. J.S.	1	1	1	1	1
7. S.T.	1	0	0	0	0
8. S.E.	0	1	1	0	0
9. E.M.	1	1	1	0	0

(A 0 denotes absence of the symptom and a 1 denotes presence.)

was simulated by presenting these 9 cases to the model with a sixth dimension, a disease label which was always burlosis. This was arbitrarily treated as a binary dimension. Note that each of the five symptoms show a majority of ones associated with the disease.

The critical feature of these materials from the perspective of correlated features concerns the fourth dimension of condition of eyes and the fifth dimension of weight. Values are either both 1 or both 0. The first six items in table 2 have two 1's in these dimensions; the last three have two 0's. The question is how should one go about representing such correlated features. When these stimuli were fed into the algorithm with $c = .3$, it typically extracted 3 categories—one to represent the first six items, one for the seventh, and one for the last two. Thus, the way it dealt with correlated discrete features was to break out separate categories for the different possible values of the correlation. It is not so easy to deal with correlated continuous features as we will see in some of the applications to be reported in this paper.

There is more evidence that the model is correct in treating category labels as just another feature to be predicted rather than something intrinsically tied to category membership. Most experiments have looked at subjects trying to predict category labels given features, but recently Heit (1990) has looked at what happens when we have subjects predict other features as well. In his research he gives subjects various subsets of attributes associated with an item and asks them to predict other subsets. He finds nothing special about a category label whether it is a feature to be predicted or feature to predict from. Research has also been done looking at the effect of category labels during training. Some subjects are trained without the feedback of category labels and some subjects are given labels. Labels prove just to be another feature that can make the category structure apparent when it is correlated with the other features. Our model can simulate this array of results (Anderson, 1991).

The model also is able to simulate the array of evidence showing sensitivity to statistical properties of the stimuli. It is influenced by the base rates of various categories such that it is more likely to assign a stimulus to a category with a higher base rate as subjects do. In situations where the categorization rule is probabilistic it will assign instances to categories with probabilities that match the objective probabilities as subjects do. When the instances it learns form a hierarchy of categories, it chooses as its categories those that optimize the predictive structure of the stimuli. These are the basic-level categories which Rosch (e.g., Rosch, Mervis, Gray, Johnson, & Boyes-Braem, 1976) studied at great length. All of these phenomena are discussed in detail in Anderson (1990).

6. Application to machine learning data sets

In this section we will discuss the application of the algorithm to a number of standard data sets in the machine learning literature.[7] Our model does well sometimes and not so well other times. We will examine the characteristics of each data set to understand the pattern of the performance. This is in keeping with our intention, announced at the beginning of the paper, which is to understand the algorithm in terms of what domains it is optimized to. We think the point has passed where it is informative to engage in horse races between learning algorithms. Different algorithms will work optimally given different data sets and it should be our first task to understand the characteristics of the domains to which the algorithms are adapted. If one wants to engage in competition among algorithms that should be done by arguing which domain characteristics are more typical of real problems.

One can only understand the performance of our model by understanding the relationship between the domain characteristics it assumes and the characteristics of a particular data set. Thus, part of our effort in these analyses has been to come to a better understanding of the data sets to which we apply the model. We feel that one of the accomplishments of our applications is this deeper understanding of these data sets which have become part of the machine learning literature.

In subsequent subsections we will give detailed discussions of our applications to various machine learning data sets. Table 3 is a summary, describing the size of the training set, size of the test set, number of attributes excluding category on which the objects varied, number of categories, and performance of the radical iterative algorithm in predicting category membership. While our algorithm is concerned with much more than predicting category membership, this is the performance measure used to evaluate most other algorithms. Throughout our performance measures are averaged over 10 randomized runs.

Table 3. Summary of application to machine learning data sets.

	Training Size	Test Set Size	No. Attributes	No. Classes	Performance
LED	50	50	7	10	74%
Iris	75	75	4	3	91%
Soybean	290	340	34	15	92%
Congressional Voting	217	218	16	2	95%
Breast Cancer	143	143	9	2	72%
Waveform	100	100	21	3	78%

6.1. LED display domain

One of the more straightforward data sets for illustration of the algorithm is that concerned with the classification of noisy LED displays (Breiman, Friedman, Olshen, & Stone, 1984).

The LED contains 7 light-emitting diodes for displaying digits. There are 3 vertical bars and 4 horizontal bars (see figure 4). Each diode is in one of two states, on or off. There is a 10% probability of having the value of any diode inverted. It turns out that the optimal classification rate for these stimuli is 74% since some test items can randomly be transformed into others. For instance, if one vertical bar is randomly turned off and another randomly turned on, a 2 can be transformed into a 3. Other classification programs (Quinlan, 1987; Tan & Eshelman, 1988) approach this theoretical maximum although they require more training trials than does our program.

This is an ideal domain for illustrating our algorithm because the stimuli correspond precisely to the assumptions of our model. There are in fact 10 disjoint categories corresponding to the 10 digits. Each category has 7 binary features which it displays probabilistically (p = .90). In training there is an eighth 10-valued feature present which is the category label. In test, the seven binary features are presented and the task is to predict the eighth. Since these were all discrete features we had to set the α parameters for equation (9). These were set to 1.0 for the binary features and .01 for the 10-valued label dimension in keeping with the policy discussed earlier for labels. We were able to achieve an optimal classification rate of 74% with 50 training instances. Ten categories emerged which corresponded to the 10 external categories. 50 training instances (5 per category) was enough to reliably identify the majority value for all dimensions for all categories. The odds that all of the 70 dimensions (10 categories by 7 features) would be the majority value is about .8.

So with little surprise our algorithm works optimally in this domain with respect to the classification task. However, it is worthwhile to look beyond the classification task. Recall that the algorithm is not especially designed to predict category labels. It is also intended to predict the other dimensions. Therefore, we explored how well this algorithm could do in predicting the values of any of the 7 binary dimensions given the values of the remaining dimensions. What we have calculated for current purposes is proportion errors. The results

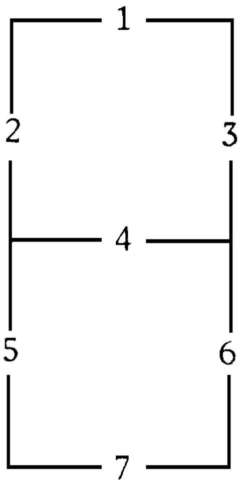

Figure 4. The seven diodes for the LED display.

Table 4. Proportional error of prediction in the LED data base.

	Incremental Algorithm	Majority Value	Linear Regression
Top Bar	.08	.23	.15
Top Left	.09	.44	.20
Top Right	.07	.21	.18
Middle	.10	.30	.16
Bottom Left	.12	.39	.14
Bottom Right	.11	.17	.20
Bottom	.07	.28	.13

are shown in table 4. We have given a number of possible different sets of numbers. One is what is gotten with our categorization algorithm. A second is what is gotten with an algorithm which simply chooses the majority value over the instances. The third is what is gotten with a linear regression model that tries to predict one value by regressing it on the rest. The considerable improvement in fit represents what is gained by using the 10 categories over just one (majority model) or linear regression. This improvement in prediction is the raison d'etre for the rational categorization algorithm.

6.2. Iris data base

We also applied the algorithm to the Iris data base described by Fisher (1936). Three species of Iris, Iris setosa, Iris versicolor, and Iris virginica are described in terms of four continuous dimensions which are sepal length, sepal width, petal length, and petal width. There are 50 instances of each type described. As it turns out this considerably underrepresents the dimensional complexity of irises which vary in other dimensions including color, shape, overall size, and texture of the various parts. Still the data base has been used as a target for many classification efforts including the Autoclass program of Cheeseman, Kelly, Self, Stutz, Taylor and Freeman (1988) which is one of the very few attempts that claims success in identifying all three of the underlying categories. We will discuss this program later since it represents a Bayesian approach to classification.

In attempting to apply our model to this data, we split the set randomly into a training set of 75 instances and a test set of 75 instances. Applying equations (10) through (14) to the continuous dimensions we set the parameters. $\lambda_0 = 1$, $\alpha_0 = 1$, μ to all the mean of all 150, and σ_0^2 to the square of one-quarter of the range. Again, we set $\alpha = .01$ for the one discrete dimension of category membership.

There are two ways to apply the model in training. One is to give it access to the category labels or, as is the more typical practice with this data set, only give the continuous dimensions and see how well the categories it induces correspond to the categories associated with the labels. Either way we took the categories formed during training and used these to predict the labels of the 75 test instances. In the case where category labels were not present during training we nonetheless attached labels to the categories formed in proportion to their frequency during training. Thus, if a category was formed with 25 Iris versicolor and 10 Iris Virginica (as sometimes happened) at training we would credit the category with 25 counts of one label and 10 counts of the other for application of equation (9).

With the guide of category labels the program was able to extract out the three categories. It classified correctly anywhere from 88% to 98% of the test instances depending on the randomization. The average classification performance (over 10 runs) was 91%.[8] It appears that an occasional iris from one category appears indiscriminable from irises from another category, perhaps reflecting the rather impoverished stimulus description. On the other hand, when trained without labels the algorithm vacillated between identifying two categories or three depending on randomization. In the case of two categories there was always one category that corresponded to the Iris setosa, and another that corresponded to the two remaining species. In the case of three categories, there was again one that corresponded to the Iris setosa, one that included most or all of the Iris versicolor and some Iris virginica, and one that contained the larger Iris virginica. Using the categories without label feedback for prediction, the algorithm varies from 67% correct to 90% in predicting category labels with an average of 75%. The three species varied in size of sepal length, petal length, and petal width with the Iris setosa being smallest and quite discriminable from the other two and the Iris virginica being slightly larger than the Iris versicolor. In fact, there is no consensus in the botanical world about whether Iris virginica and Iris versicolor should actually be considered separate species (Mathew, 1981). Interestingly, when we present human subjects with computer-drawn flowers that vary in just these four dimensions they tend to reproduce the behavior of this algorithm—that is, they either extract two categories merging the Iris versicolor and the Iris virginica or they produce a category containing the Iris versicolor and about half of the virginica and a separate category for the other half of the larger virginica.

From the perspective of our model, it is a rather unnatural task to pose to a system to ask it to be able to predict the labels of categories when it has had no training on these labels. The algorithm is forming categories in order to be able to extrapolate from experience on dimensions of old instances to possible values on those dimensions for new instances. Therefore, we decided to compare the performance of various versions of this program with respect to predicting the four continuously varying dimensions. We presented test instances with three of the dimensions present and looked at accuracy in predicting the fourth dimension. We looked at this for a number of cases—when three categories were formed with the use of category labels, when two or three categories were formed in the absence of category labels, when all the instances were merged into a single category, when separate categories were formed for each training instance (75 categories in all), and a linear regression model. The results averaged over 10 runs in each case are displayed in table 5.

Table 5. Mean-squared error in prediction for the iris data set.

	3 Perfect Categories	2 or 3 Imperfect Categories	Just 1 Category	75 Singleton Categories	Linear Regression	Incremental Algorithm with Correlation
Sepal length	.39	.39	.82	.79	.11	.11
Sepal width	.13	.15	.17	.16	.09	.08
Petal length	.21	.41	3.21	3.08	.09	.08
Petal width	.05	.10	.59	.56	.04	.04

The prediction is somewhat better given the perfect three-category structure than the imperfect categorization derived from the iterative algorithm. However, both are much better than just one or 75 categories. Interestingly, prediction based on any categorization is quite clearly outperformed by a linear regression model. The reason is that there are strong correlations among the three dimensions of sepal length, petal length, and petal width. The three categories just occupy three overlapping positions along what amounts to an overall dimension of size. If one has two or three categories one is in position to estimate the overall linear relationship by extrapolating from the two or three points along the dimension (the problem with one category is we only have one point, and with 75 is that the points are not accurately measured). However, any attempt at categorization obscures the fact that this linear relationship exists as much within as between categories. Thus, a linear regression model does best.

The basic problem is that these dimensions are not independent. It is curious that we find nonindependence in a domain of living things since the structure of living things was used to motivate the independence assumption in the first place. The problem is that we are looking at dimensions which are all reflections of one underlying genetic trait which is size. Just as one would not be surprised by a correlation between length of the left arm and length of the right arm, one should not be surprised by these correlations. This points out an important constraint on the application of this model to the biological domain—the dimensions chosen have to reflect separate genetic traits. We will return at the end of this section to the issue of how to deal with such nonindependence within the framework of our algorithm. We report there an extension of our model. The data from this extension is in the last column of table 5.

6.3. Soybean data base

A data base that has become a classic for the testing of categorization programs is the soybean disease data base of Michalski and Chilausky (1980). This consists of 290 training instances and 340 test instances where the instances are descriptions of soybean diseases. There are some 15 disease categories and each instance is described by up to 35 attributes with potentially missing attributes. The best categorization applications (e.g., Michalski & Chilausky; Tan & Eshelman, 1988) result in approximately 95% ability to predict the disease category of test instances from the underlying features. When we run our categorization algorithm on this data base using $\alpha = .01$ for category label, it fails to separate all the categories and only extracts 11 categories and gives 79% performance in the final classification cost. We need to set $\alpha = .0001$ to reliably separate the categories. At this level it yields 92% correct classification which is comparable to past programs. On the other hand there is no indication that it does any worse at predicting any of the other 35 features with $\alpha = .01$ than $\alpha = .0001$. Table 6 gives the performance of various methods on predicting the category label and the other features. We give separate statistics for the 33 discrete dimensions (percent accuracy) and for the 2 interval dimensions (mean square error). Many of the discrete dimensions are pretty arbitrary which explains why the average statistics are quite low. It is worth noting that our model can sort through this noise to identify the disease categories given a sufficiently low value of α for category label.

Table 6. Accuracy of prediciton and mean-squared error for the soybean data set.

	One Category	290 Categories	11 Categories with $\alpha = .01$	15 Categories with $\alpha = .0001$
Prediction of disease label	14%	82%	79%	92%
Prediction of other 33 discrete features	67%	72%	76%	76%
Prediction of 2 interval features	1.84 M.S.E.	1.26 M.S.E.	1.18 M.S.E.	1.14 M.S.E.

The performance of the program when it is forced to merge all categories into one gives us a definition of chance guessing performance. As can be seen, compared to this, the other three approaches perform quite well. The case where 290 categories are extracted represents an instance-based model which has been quite common in psychology (Medin & Schaffer, 1978; Nosofsky, 1988). Performance is not quite as good with instance-based prediction replicating results of table 5. The problem is that one observation does not allow reliable statistics to emerge. Prediction of the other discrete dimensions is not that much improved over the chance one-category level in any of the multi-category conditions. This indicates that for this data base the category label is more predictable than any other dimension.

6.4. Congressional voting

Another data set that we have applied the algorithm to is the congressional voting records that have been used by Schlimmer (1987) and Fisher (1987). This data base consists of 435 members of Congress and their votes on 16 key uses in 1984. A typical use of this program has been to predict party membership from voting record. Schlimmer reports about 90% to 95% success for his Stagger program. We applied our program to this using our split-half method of having it learn on a random half of the set and predict the other half. We compared the accuracy of (a) an instance scheme that forced each member into its own class; (b) a scheme that formed exactly two classes corresponding to the parties; and (c) the standard incremental program free to form its own categories. As before we looked at prediction of each attribute by the remainder. The incremental program averaged 8 categories. There are two large categories recognizable as liberal Democrats and conservative Republicans, two smaller categories recognizable as conservative Democrats and liberal Republicans, and smaller, more esoteric categories. Performance on predicting party affiliation is 91% using individual categories, 90% using two categories, and 95% using the incremental categories. Prediction of the vote on the other 16 issues is 76% using individual categories, 71% using two categories, and 77% using the incremental categories. Thus, this is a set where the iterative model appears to fare quite well. Forcing all the congressmen into their party categories misses predictable variance. On the other hand, keeping all the congressmen separate misses the opportunity to separate systematic trends from accidental. It is a bit of a curiosity that congressional voting patterns should correspond so well to the assumptions derived from a genetic model of species.

6.5. Breast cancer

Another frequently used data set is a set of 286 cases of breast cancer[9] where the goal has been to predict recurrence of breast cancer given nine attributes which describe the characteristics of the patient, the original cancer, and the treatment. Typical applications (Michalski, Mozetic, Hong, & Lavrac, 1986; Clark & Niblett, 1987; Tan & Eshelman, 1988; Cestnik, Konenenko & Bratko, 1987) have reported accuracies from 65% to 73.5% with Cestnik et al. getting 78%. Using the split-half methodology, we compared the performance of our algorithm when it extracted just one category, singleton categories for each individual, extracted a number of categories according to the iterative algorithm (it extracted on average 4.4 categories) and non-recurrence cases. Table 7 compares the performance of the algorithm under these various conditions. In each case these are the averages of 10 runs based on different random split halves.

Chance level of performance in prediction of recurrence is 70% as indicated by the accuracy in the one-category condition. This is gotten by predicting the majority outcome (no recurrence) all the time. The performance in the multiple-category conditions is only marginally better at 72%. These numbers are also in the range of other applications which suggests that this is not a particularly predictable data set. We also looked at predictions of the other features. Except for the dimensions of age and menopause, all categorical structures were performing at about chance level. There is an obvious correlation between age and menopause which is being partially identified in the case of singleton categories. This is another case where there is a correlation among dimensions that is not being captured by our independence assumption.

The low level of performance of all algorithms in this data base compared to chance level (70%) suggests that this is not a particularly predictable domain. In fact, recurrence of breast is notoriously hard to predict.

6.6. Waveform data base

The final data set that we applied our algorithm to was the waveform data base (Breiman, Friedman, Olshen, & Stone, 1984). Three underlying waveforms were created as illustrated in figure 5. Each waveform is represented by 21 measurements. Three classes are generated

Table 7. Accuracy of predictions for the breast cancer data set.

	One Category	Singleton Categories	Incremental Categories	Two Categories
Recurrence	70%	72%	72%	72%
Age and menopause	39%	56%	40%	39%
Other features	53%	55%	55%	54%

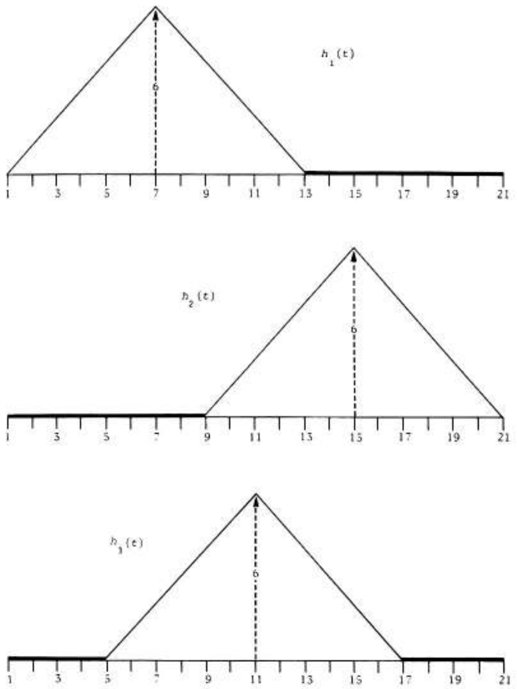

Figure 5. The three underlying waveforms being mixed.

as random mixtures of forms 1 and 2, forms 1 and 3, and forms 2 and 3. A random mixture of two waves is created by taking P% of the first wave and 100 − P% of the second where P is uniformly chosen from the interval 0 to 100. In addition a substantial random error is added to each of the 21 points so that the resulting pattern is quite noisy. Breiman, et al. report 72% accuracy in classification for their CART algorithm and 78% for a nearest neighbor algorithm. This is given 300 training instances.

We trained the program on a random 100 of such stimuli and then tested it another random 100. Table 8 shows the performance of the algorithm in cases where we extracted just one category, where we extracted a separate category for each instance, where we allowed our iterative algorithm to run which extracted 2 to 5 categories per run (averaging 3), and where we forced all instances with the same category level into the same category. Compared to the chance level defined by the one-category condition, the other three conditions were all performing as well. With respect to predicting category label, the intervention of forcing the category structure to mirror the labels led to best performance. This is as it should be because in this case category label was a perfect indicator of underlying category membership. We get with 100 instances classification as good as Breiman et al. were able to get with 300 instances.

Table 8 also reports performance at predicting the 21 points. We have aggregated sets of three adjacent points as well as reporting average error over all 21 points. Worse performance is obtained with using the three correct categories than using either singleton or

Table 8. Accuracy of prediction and mean square error for the waveform data set.

	One Category	100 Singleton Categories	Incremental Categories	Three-Perfect Categories
Category Level	35%	69%	66%	78%
Point Prediction:				
First three	1.49	1.64	1.42	1.42
Second three	2.73	1.80	1.81	2.07
Third three	3.37	1.80	1.85	2.28
Fourth three	2.73	1.90	2.21	2.12
Fifth three	3.50	1.81	1.95	2.37
Sixth three	3.27	1.97	2.14	2.40
Seventh three	1.47	1.71	1.39	1.36
Overall error in point prediction	2.65	1.80	1.82	2.01

iterative categories. This is because there are correlations among adjacent points of a waveform that are not captured by the category averages. The singleton categories and iterative categories formed were able to reflect this. The problem is not that one application is better for predicting category labels and another for predicting points. The problem is that the within-category independence assumption on which both models depend is violated.

6.7. Extension to correlated dimensions

In principle it is not that difficult to extend our Bayesian approach to include correlations among continuous dimensions and we have tried a couple of applications using this extension which involved one revision in our starting assumptions.

The revision in our starting assumption is that, rather than assuming a pair of continuous dimensions *i* and *j* are independent, we assume the more general multivariate normal model which allows for a correlation r_{ij} between each pair of dimensions *i* and *j*. One can essentially proceed as before except that one needs a basis for estimating r_{ij}. The basic solution is to calculate an empirical correlation between two dimensions within a category and merge these with a prior correlation to come up with a posterior correlation estimate. We use the approach recommended in Box and Tiao (1973) and Lee (1989) for combining the empirical correlation *r* and the prior π to get an estimated $\hat{r}$ according to the following formula:

$$\hat{r} = tanh \left[\frac{n_1\, tanh^{-1}\, r + n_2\, tanh^{-1}\, \pi}{n_1 + n_2} \right]$$

where n_1 is the number of observations going into the empirical sample (i.e., the number of items in the category) and n_2 is the strength of belief in π. We felt, since correlations can vary from −1 to +1, the obvious prior to have in absence of any expectation was $\pi = 0$ which is what we used. We also hold a considerable belief that dimensions will be independent and so we set $n_2 = 10$. This means that it will take 10 empirical observations before we will weight the empirical correlation as strongly as the prior. Also empirical correlations become non-trivial only when $n_1 \geq 3$ and so $\hat{r}$ was always 0.0 when category size was less than 3.

With these specifications we then applied the algorithm to the Iris data base which is one case where there was considerable evidence for a correlation among dimensions. The iterative algorithm still extracts 2 or 3 categories which still tend to merge Iris versicolor and Iris virginica. Its performance in predicting the four continuous dimensions is reported in the last column of table 5. Perhaps not surprising, the rational algorithm is now doing as well as the multiple regression approach.

However, there is not always improvement in cases where one might have expected it. We also applied this algorithm to the waveform data in table 8. Since neighboring points are correlated, one might have expected an improvement in prediction but none was observed over the iterative algorithm. This is because of the noise associated with the manifestation of any of the points. Apparently, the category structure identified by the iterative algorithm captured all of the predictable structure that could be captured.

While this algorithm can lead to improved performance in some cases, it is not without a considerable computational cost. The amount of computation that is required to calculate the correlations and regression coefficient for *m* dimensions is on the order of m^3 (we have to process on the order of a $m \times m$ matrix to make predictions about each of the *m* dimensions). Moreover, these calculations are substantial. This makes the complexity of the overall algorithm on the order of $n \times m^3 \times p$ compared to $n \times m \times p$ for the original algorithm. The cost of the m^3 was particularly apparent when we applied it to the waveform data set with its 21 continuous dimensions.

Thus, we do not believe this reflects a reasonable approach in full generality. Rather, we think people and the algorithm should be sensitive to correlations among a few select dimensions which possibly might be related to one another. It remains a future research issue how we might identify such candidate dimensions for correlational monitoring.

7. Comparisons to other systems

There are a good many learning systems that we might compare our system to. In doing this it is important to keep clear on what the goals of the systems are and what assumptions they make about the nature of the environment. It is a fair generalization to say that most systems take as their primary goal to produce a categorization of the object set. They are interested in prediction, if at all, as a side benefit of the categorization they produce. In contrast, our system is primarily concerned with prediction and only does categorization as a means to an end.

7.1. Deterministic models

The majority of machine learning systems (e.g., Aha, Kibler, & Albert, 1991; Clark & Niblett, 1989; Lebowitz, 1987; Michalski, Mozetic, Hong, & Lavrac, 1986; Quinlan, 1986;

Salzberg, 1991) take a fundamentally deterministic view of the environment in contrast to ours which is fundamentally probabilistic. This is seen in the frequent reference to dealing with "noise." A basic assumption in these systems is that there is a correct category assignment for a particular object description and if the data set contains objects identical in description but mapped to different categories, this is a sure sign of noise. Rather our view is that this is to be expected and all we can do is estimate a probability of a category label or any other feature.

A natural assumption would be that our model would do better in domains fitting our assumptions while the other models would do better in domains fitting their assumptions. To see whether this was true, we created two artificial environments—one satisfying the deterministic assumptions and one satisfying the probabilistic assumptions. Both domains involved generating instances for four categories. The categories, in addition to a category label, had four continuous dimensions with values concentrated in the interval 0–4.

In the case of the deterministic domain, two hyperrectangles were associated with each category. The hyperrectangles were generated by applying the following rule for each dimension:

1. Choose a random width for the dimension uniformly between 0 and 2.
2. Choose a random starting point for the dimension between 0 and 2.

Thus, if .8 were selected in step 1 and 1.5 in step 2, the hyperrectangle would range from 1.5 to 2.3 on that dimension. Since these hyperrectangles potentially overlapped, they were ordered from first hyperrectangle for first category to first hyperrectangle for last category and then second hyperrectangle for first category to second hyperrectangle for the last category. A point was judged to be in the category corresponding to the first hyperrectangle that included it in the ordering. In essence we had an ordered set of eight rules for classifying points. This was done to produce a complex but deterministic partition of the space. For training and test, 25 instances were generated randomly by selecting uniformly from each hyperrectangle. Since there were two hyperrectangles for each of four categories, there were 200 training and test instances.

The probabilistic domain was created by associating two four-dimensional normal distributions with each category. The mean of the distribution on each dimension was randomly selected on the interval 0–4 and the standard deviation on the interval 0–1. For study and test, 25 instances were randomly generated from each distribution and assigned to that category. Note that the same instance in principle could be generated from more than one category in contrast to the deterministic case. Since there were two distributions for each of the four categories, there were 200 training and test instances.

We applied our algorithm to both of these domains and, as a representative of a deterministic algorithm, the NGE algorithm of Salzberg (1991). It was chosen both because it was tailor-made for this deterministic domain and because it was very simple to implement given the clear specifications of Salzberg. Our algorithm delivers a probability of each category. To make it correspond to NGE, we had it choose the most probable category. We ran 25 experimental runs of both algorithms in both domains. The results are reported in table 9.

Table 9. Comparison of the rational algorithm on Salzberg's NGE for two artificial domains.

	NGE	Rational Algorithm	
Deterministic domain	93.5%	88.6%	91.1%
Random domain	81.1%	84.3%	82.7%
	87.3%	86.5%	

An analysis of variance was run on these data. It revealed a significant effect of domain ($F_{1,96} = 41.4$; $p < .001$) no significant effect of algorithm ($F_{1,96} = .4$) and a significant interaction between algorithm and domain ($F_{1,96} = 10.0$; $p < .005$).

It is interesting that the larger effect is due to domain with both algorithms doing better in the deterministic domain. However, there is also evidence that these algorithms do better, relatively speaking, when the domain matches their assumptions. There is no guarantee that NGE is optimally tuned to the deterministic domain and we know the incremental Bayesian algorithm only yields an approximation to the ideal quantity (equation (0)). Still this exercise does illustrate the point about match between assumptions of the learning program and the domain.

7.2. Comparisons to hierarchical models

There are a number of models which try to retrieve a hierarchical organization of the instance set. Lebowitz (1987) and Fisher (1987) are interesting contrasts to our system because they also use incremental learning algorithms. Fisher's COBWEB is particularly close to ours because it is also trying to optimize a probability measure. COBWEB tries to find a categorization of the objects which will maximize the following quantity (taken from Gluck & Corter, 1985):

$$\sum_k P(k) \sum_i \sum_y f_i(y|k)^2$$

which has obvious similarities to equations (2) and (5) which determine our partitioning in that it emphasizes priors and conditional probabilities of the features.[10] Our model in effect replaces the summations by products and so does not need a squaring of the conditional probabilities to get non-linearity. The motivation for the COBWEB equation is largely intuitive but it will correlate highly in many cases with our metric.

Being an incremental model, COBWEB faces the same problem as our model of finding the partition that maximizes the quantity without searching the whole space. In our case, it is not a serious matter whether we find a partitioning that optimizes our probability measure. What is critical is that we get a partitioning with predictions that closely approximate the true predictions computed over all partitionings. However, this is a serious problem in COBWEB since its goal is to uncover a partitioning that will be informative in itself.

COBWEB places objects in existing categories or creates new categories according to what will optimize its probability measure, just as does our model. However, it maintains a hierarchical structure of these categories so it can consider merging categories together or splitting up a category into subcategories.

Anderson and Matessa (1991) report experiments on use of such a hierarchical system within our system. While it did retrieve category structures which were marginally more probable and intuitively more appealing, the hierarchical algorithm failed to produce any improvement in the accuracy of the predictions. The cost of the merging and splitting (in particular computing whether a merge or split is justified) can be quite high.

7.3. Comparisons to Autoclass

Our model is probably closest in spirit to the Autoclass model of Cheeseman, Kelly, Self, Taylor, and Freeman (1988). Like our model they take a fundamentally probabilistic view of the nature of categories. They try to optimize the same conditional probabilities that we do except that they appear to use the normal approximations to the t-distribution. There are two major points of contrast, however.

First, their model buys into the notion of categorization as the primary goal rather than the means to an end of prediction. This leads them as others to place a premium on finding the most probable interpretations of the data even though, as we saw, this leads to no apparent advantages in achieving prediction. Their model is not designed to deliver predictions although it could be easily extended to do so.

Second, they do not use incremental algorithms. Rather they use the EM algorithm for optimization (Dempster, Laird, & Rubin, 1977). They start out with more categories than expected and their task is to find the assignment of objects to categories. This model infers a probability that an object is in each category rather than assigns an object to a category. They observe that the algorithm often emerges with a high probability assignment to a single category. Any categories which result with negligible assignment of objects are deleted. The basic assumption of the model is that the likelihood of the data given a category structure will overwhelm any considerations of the prior probability of that category structure.

The basic iteration in the algorithm involves a computation that is a function of the product of number of categories, number of objects, and number of attributes which is the same computational complexity as faced by our model. The number of iterations required is unclear. The algorithm also has to take special measures to try to avoid being trapped in local optima.

This algorithm is designed to run once and is poorly adapted to the situation we are interested in where one must have a basis for prediction after each object. To run the algorithm after each object would mean its cost would grow with the square of the number of objects.

With respect to prediction it would have to work with a partitioning in which objects are fractionally assigned to categories rather than our case where all objects (except the one to be predicted) are assigned to one category. This provides no better or worse a basis for approximating the ideal values which would have to be calculated over all ways of assigning objects to categories.

8. Conclusions

The outcome of the application of the algorithm to various domains, real and artificial, has been mixed. Basically, the algorithm is successful to the extent that the structure of the domains satisfies the basic assumptions of its model of the environment. This is not altogether an obvious outcome because the iterative algorithm gives only an approximation to the true quantities prescribed by that statistical model. However, as advertised it appears that nothing is seriously lost by this approximation.

The success of the overall algorithm is predicated on three key features. The first is the use of the efficient iterative algorithm to deliver approximations to the ideal model. The second feature of this approach is that its goal is to deliver accurate predictions and not to find the "true" categorical structure. This performance-oriented focus means that we can tolerate situations where the underlying categorical structure may seem a bit peculiar and intuitively not appealing. The third feature is also related and this is the denial of any special status for category labels. By treating category labels as just another feature to be predicted we often gain overall predictive power. A noteworthy feature of this approach is that it does predict the values on all dimensions without any extra work.

Finally, we close by stressing once again the observation that the success of the model does prove to be a function of the structure of the domain to which it is being applied. We think the direction to go in extending our work is one in which we inquire as to what other types of structure exist, develop models for these structures, and perhaps develop ways of detecting one structure versus another. We want to emphasize starting with the structure of the domain and not the learning algorithm. From the observations of the performance of our algorithm, we feel that the most important kind of structure which is found in the real world and which our model does not capture concerns correlated features. We have shown that we can extend the algorithm to deal with such features but that in general the computational cost would be unacceptably high.

Acknowledgments

We would like to thank Ken Koedinger and Ching-Fon Sheu for their comments on this paper. The research was supported by BNS-87-05881 from the National Science Foundation and contract N00014-J-1489 from the Office of Naval Research.

Notes

1. If cost is set to 0, the algorithm becomes the ideal algorithm and calculates all partitionings. We have implemented this adjustable algorithm although we will only report results from the radical incremental version.
2. As this example makes clear, human intervention has created the breed (e.g., Labrador), a specialization within the species (i.e., dog). It is the breed and not the species that defines the freely interbreeding unit and for our purposes the category.
3. $\Gamma(X) = (X - 1)!$ for integer X.
4. Because the posterior distribution is of the same form as the prior distribution, the Dirichlet distribution is referred to as the *conjugate prior* for the multinomial.

5. For purposes of this illustration only, the α_j for the category label is 1.0. In the applications to be reported, it will be set at 0.1 which creates a considerable reluctance to merge items with different labels into the same category.
6. It is possible to imagine an algorithm which would only select some features for some categories.
7. These data sets were obtained from UCI ML database maintained by David W. Aha.
8. Weiss and Kapouleas (1989) reported a special crafted rule that classifies 97% of Irises in the Fisher data base. This is the best result that we know of but it is probably a case of overfitting the data and would not do as well on a new data set.
9. This data is provided by M. Zwitter and M. Soklic of the University Medical Center, Institute of Oncology, Ljubljana, Yugoslavia.
10. We have recast the Gluck and Corter quantity in our notation.

References

Aha, D.W., Kibler, D., & Albert, M.K. (1991). Instance-based learning algorithm. *Machine Learning, 6*, 37-66.

Anderson, J.R. (1990). *The adaptive character of thought.* Hillsdale, NJ: Erlbaum.

Anderson, J.R. (1990b). *Cognitive psychology and its implications.* Third Edition. New York: W.H. Freeman.

Anderson, J.R. (1991). The adaptive nature of human categorization. *Psycholocial Review, 98*, 409-429.

Anderson, J.R. & Matessa, M. (1991). In D.H. Fisher, M.J. Pazzani, & P. Langley (Eds.), *Concept formation: Knowledge and experience in unsupervised learning.* Palo Alto, CA: Morgan Kaufman, 45-70.

Berge, C. (1971). *Principles of combinatorics.* New York: Academic Press.

Berger, J.O. (1985). *Statistical decision theory and Bayesian analyses.* New York: Springer-Verlag.

Beyer, W.H. (1987). *CRC standard mathematical tables.* Boca Raton, FL: CRC Press.

Box, G.E.P. & Tiao, G.C. (1973). *Bayesian inference in statistical analysis.* Reading, MA: Addison-Wesley.

Breiman, L., Friedman, J.H., Olshen, R.A., & Stone, C.J. (1984). *Classification and regression trees.* Belmont, CA: Wadsworth International.

Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W., & Freeman, D. (1988). A Bayesian classification system. *Proceedings of the Fifth International Conference on Machine Learning* (pp. 54-64). San Mateo, CA.

Clark, P.E. & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning, 3*, 261-284.

DeGroot, M.H. (1970). *Optimal statistical decisions.* New York, NY: McGraw-Hill.

Dempster, A.P., Laird, N.M., & Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39.

Fisher, R.A. (1936). Multiple measurements in taxonomic problems. *Annals of Eugenics, 7*, 179-188.

Fisher, D.H. (1987). *Knowledge acquisition via incremental conceptual clustering.* Doctoral dissertation, Department of Information and Computer Science, University of California, Irving, CA.

Fisher, D.H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning, 2*, 139-172.

Gluck, M.A. & Corter, J.E. (1985). *Information and category utility.* Unpublished Manuscript. Stanford University.

Lebowitz, M. (1987). Experiments with incremental concept formation: UNIMEM. *Machine Learning, 2*, 103-138.

Lee, P.M. (1989). *Bayesian statistics.* New York: Oxford.

Mathew, B. (1981). *The iris.* New York: Universe Books.

Medin, D.L., Altom, M.W., Edelson, S.M., & Freko, D. (1982). Correlated symptoms and simulated medical classification. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 8*, 37-50.

Medin, D.L. & Schaffer, M.M. (1978). Context theory of classification learning. *Psychological Review, 85*, 207-238.

Michalski, R.S. & Chilausky, R.L. (1980). Learning by being told and learning from examples: An experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis. *International Journal of Policy Analysis and Information Systems, 4*, 125-161.

Michalski, R.S., Mozetic, I., Hong, J., & Lavrac, N. (1986). The multi-purpose incremental learning system AQ15 and its testing application to three medial domains. *Proceedings of the Fifth National Conference on Artificial Intelligence.* (pp. 1041-1045). Philadelphia, PA: Morgan Kaufmann.

Nosofsky, R.M. (1988). Similarity, frequency, and category representation. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 11*, 54-65.

Quinlan, J.R. (1987). Simplifying decision trees. *International Journal of Man-Machine Studies.* To appear.

Quinlan, J.R. (1986). Induction of decision trees. *Machine Learning*, *1*, 81-106.

Rosch, E., Mervis, C.B., Gray, W., Johnson, D., & Boyes-Braem, P. (1976). Basic objects in natural categories. *Cognitive Psychology*, *7*, 573-605.

Salzberg, S. (1991). A nearest hyperrectangle learning method. *Machine Learning*, *6*, 251-276.

Schlimmer, J.C. (1987). *Concept acquisition through representational adjustment*. Doctoral dissertation, Department of Information and Computer Science, University of California, Irvine, CA.

Tan, M. & Eshelman, L. (1988). Using weighted networks to represent classification knowledge in noisy domains. *Proceedings of the Fifth International Conference on Machine Learning*, (pp. 121-134). Ann Arbor, MI.

Weiss, S.M. & Kapouleas, I. (1989). An empirical comparison of pattern recognition, neural nets, and machine learning classification methods. *Proceedings of the Eleventh International Conference on Artificial Intelligence*, (pp. 781-787).

Zwitter, M. & Soklic, M. (1988). Breast cancer data. University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia.