IN-64-TM
146702
17P.

Automatic Discovery of Optimal Classes

Peter Cheeseman, John Stutz, Don Freeman and Matthew Self

NASA Ames Research Center
Mail Stop 244-7
Moffett Field, CA 94035

Abstract

This paper describes a criterion, based on Bayes' theorem, that defines the optimal set of classes (a classification) for a given set of examples. This criterion is transformed into an equivalent minimum message length criterion with an intuitive information interpretation. This criterion does not require that the number of classes be specified in advance, this is determined by the data. The minimum message length criterion includes the message length required to describe the classes, so there is a built in bias against adding new classes unless they lead to a reduction in the message length required to describe the data. Unfortunately, the search space of possible classifications is too large to search exhaustively, so heuristic search methods, such as simulated annealing, are applied. Tutored learning and probabilistic prediction in particular cases are an important indirect result of optimal class discovery. Extensions to the basic class induction program include the ability to combine category and real valued data, hierarchical classes, independent classifications and deciding for each class which attributes are relevent.

1 Introduction

This paper describes a method for discovering (inducing) optimal classes from a given data base. These classes can then be used to make predictions in particular cases or give insight into the patterns that occur in the particular domain. Many previous authors have published approaches and results in the area of automatic class discovery [7], but these approaches have been generally disappointing when applied to real data. These previous approaches are usually based on some sort of "similarity" measure, and they give different results depending on the similarity measure chosen. Even more disturbing is that these methods usually require the user to specify the number of classes to be discovered, or rely on *ad hoc* methods for choosing an appropriate number of classes. They produce classes when given random data, indicating a serious problem with the classification criterion. The experiments described here can be viewed as a first step toward intelligent systems that learn about their environment, with or without human help.

(NASA-TM-101174) AUTOMATIC DISCOVERY OF OPTIMAL CLASSES (NASA) 17 p CSCL 12A
N88-25238
Unclas
G3/64 0146702

The approach described in this paper does not use a "similarity" measure—it is actually finding the *most probable* classification given the data (and prior expectations). The most probable classification occurs when the members of a class are most predictive of each other—a domain independent form of "similarity". The most probable classification also decides the optimal number of classes, as well as the class definitions. The criterion for deciding which is the most probable classification is not new, but was successfully applied to the classification problem nearly 20 years ago [15], and is shown here to be a direct consequence of Bayes' theorem. This Bayesian *most probable* criterion is shown (in section 4) to be equivalent to finding the classification with the shortest possible total message length. a special case of the Kolomogrov-Chaitin complexity criterion [11] for determining the best theory given a set of data. The message length involves the information required to encode the data given the class assignments, as well as that required to describe the classes. The decrease in message length obtained by more specific classes is balanced aginast the cost of additional class descriptions. This trade-off can be viewed as a formal implementation of Occam's razor.

The class induction procedure described in this paper has many new features. For example, section 5.2.1 presents a method for combining both category-valued information (e.g. Sex) and real-valued information (e.g. Blood-pressure). Another extension (section 5.2.2) allows the system to optimally decide for each attribute whether it is informative for the particular class description. The classification procedure can also be extended to include hierarchical classes (section 5.2.3), and even independent (non exclusive) classes (section 5.2.4).

Even though we are searching for the most probable classification, the cost of performing a complete search is computationally extreme. Consequently, we use a heuristic search; always looking for a *better* classification than the current best. Experiments with different search techniques give locally minimal results, but are not guaranteed to have found the global minimum. These results are reported in section 3. The derivation of the minimum message length criterion is given in section 4, along with the necessary assumptions. Relaxations of these assumptions are given in section 5.2—particularly the extension to allow a mixture of category and real data. The current statistical techniques for multivariate analysis (e.g. factor analysis) do not allow mixed data to be combined, even though most data is of this form. The use of the discovered classification for prediction purposes is discussed in section 6. This method of making predictions via a mapping onto classes is a common pattern of human reasoning, and it has excellent computational properties. The use of the class induction procedure for discovery of classes and their subsequent use for prediction can be viewed as automatic discovery of expert systems—without the expert.

2 Basic Learning Procedure

This section describes the method for automatic discovery of the most probable classification

Cases	Blood Group	Sex	···	Class
Zaphrod Beeblebrox	O-	M	···	1
Peter Cheeseman	A+	M	···	1
David Letterman	AB+	M	···	4
Mickey Mouse	*	M	···	1
Minnie Mouse	*	F	···	1
⋮	⋮	⋮	⋮	⋮

Figure 1: Patient Data Base

(i.e. the partition of a set of cases into classes), given the available data. The "most probable" criterion is shown in Section 4.2 to be equivalent to finding the particular classification with the Minimum Message Length (MML) for encoding the class descriptions and the data. This MML criterion has the properties intuitively expected for an inductive procedure. The MML criterion gives a bias toward classifications with fewer classes (i.e. "simpler" classifications), unless there is sufficient information in the data to justify more classes. It is not necessary to state initially how many classes are present—this is determined by the data. Unfortunately, in practice it is computationally infeasible to find *the* classification that satisfies MML criterion; instead heuristic search techniques for finding near optimal MMLs are discussed in the next section.

A typical data base of the form required in this analysis is shown in Fig. 1. Assume that all the data is in the form of an attribute list and fixed attribute ordering with known possible values. Attributes are assumed to be categories. For example, it is possible to record if a patient is married or not, but not whether two particular patients are married to each other. This limitation can be overcome by considering pairs of patients, but such extensions are not discussed here. If the data base has missing data (indicated by *), this is treated as another possible value of the given attribute. The possible values of the "Sex" attribute are then: [Male, Female, *]. The last column of Fig. 1 represents a particular class hypothesis (classification), where every case is assigned to some class. The classes are required to be mutually exclusive and exhaustive. We use the MML criterion to find the optimal (or near optimal) class assignment.

For a *given* classification, the total MML must be calculated. The total MML is the minimum message length required to optimally encode *all* the data, as well as the class description (i.e. the last column of Fig. 1). This total MML is compared with the MML for other possible classifications, with the aim of finding the smallest classification. According to Information theory, the minimum possible message length (MML)to encode the next (ith) outcome in a sequence of trials, where the *ith* outcome has probability P_i, is given by:

$$\text{MML} = -\log P_i \tag{1}$$

Here, the outcomes are the particular attribute values that actually occurred. The $-$ sign ensures that the MML is positive, and that the base of the log can be chosen to be anything convenient, as long as the same units are used throughout. For example, if the current probability of having blood-group "A+" in a particular class is .25, the MML required to encode an "A+" outcome for a new case (in this class) is $-\log_2(.25) = 2$(bits). If we consider encoding these outcomes serially, the probability of the next outcome is conditioned on all those that went before. Typically, before the data is known, the user only knows how many possible values a given attribute can have (I). After seeing N cases in a particular class, the user will know $(n_1, n_2, \cdots, n_i, \cdots, n_I)$ where $n_1 + \cdots + n_I = N$, and n_i,is the number of occurrences of the ith value of an attribute. Given such information, the minimally informative probability (see below) that the ith value will be the *next* outcome for the next case in the particular class, is given by:

$$P_i = \frac{n_i + \frac{1}{I}}{N+1} \tag{2}$$

Equations (1) and (2) allow the calculation of the MML for the next observed attribute value. That is, we can serially encode the values that actually occurred, updating the statistics for each attribute value (the n_is) for subsequent use in Eqn. (2). For example, the MML required to encode the Blood-group attribute ($I = 5$) for the cases in class 1 (Fig. 1) is given by:

$$\text{MML for Blood-group} = -\ln\frac{1}{5} - \ln\frac{1}{10} - \ln\frac{1}{15} - \ln\frac{3}{10}\cdots = 1.609 + 2.303 + 2.708 + 1.204 + \cdots \tag{3}$$

An equivalent MML calculation must be performed for every attribute to give the total MML for encoding the next case (based on its current class assignment). Assuming the attributes within a class are mutually independent, the total MML for all the attributes for all the cases in the class, is just the sum of the MMLs for each attribute separately. Note that we are not actually encoding the cases—we are calculating what the total MML would be if they were to be optimally encoded (e.g. by using a Huff encoding scheme).

Note that the probability to encode the first value, by Eqn. (2) is $(1/I)$. This is assigning equal (prior) probability to each possible outcome, and so does not favour any particular outcome over another. This assignment may need to be modified if missing data are possible—there may be a priori information on the probability of missing data distinct from the probability of possible attribute values. With Eqn. (2), there is a reasonable probability assignment even when there are no data (i.e. $N = 0$), but the probability rapidly approaches the classical definition (n_i/N) for large N. That is, the prior probability $(1/I)$ is rapidly overwhelmed by the data. The prior term in Eqn. (2) can be regarded as distributing a single prior outcome over the set of possibilities [13]. If stronger prior information is available, it can easily be accommodated using the extended formula:

$$P_i = \frac{n_i + a_i}{N + A}; \qquad \text{where} \qquad a_1 + \cdots + a_I = A \tag{4}$$

where A is the total prior size (it determines the prior strength of belief), and the a_is are the prior number of occurrences of each possible outcome. The prior sample is obtained from all cases known to the user, but not in the data base. If these prior numbers are subjective estimates, then approximate A and a_is can be extracted from estimates of the uncertainty of the subjective probabilities [9]. Provided $N \gg A$, the data will eventually override the prior probabilities, even if the priors are very misleading. [1]

The serial encoding method for calculating the MML—Eqn (1)—starts with the prior probability $\frac{1}{I}$ and uses the accumulated n_is to compute the probability of the current outcome. The n_is are then updated based on the last outcome—i.e. the serial method adapts the probabilities for the next outcome based on on the numbers that have been already been seen. This means that there is typically a high information cost (MML) to encoding the first few cases, until sufficient data has been seen to give a good estimates of the underlying probabilities. This high initial cost is the prior information "penalty" paid for initially describing class probabilities. This penalty is what prevents an excessive number of classes from being accepted. This effect occurs because of the additional penalty must be paid for every attribute value if a new class is added. This additional penalty is only overcome if the extra class has sufficiently different underlying probabilities from the other classes.

Eqn. (2) gives the probability of a *particular* outcome based on the already known outcomes with N cases within the same class already seen. The combined MML for *all* of the observed outcomes of a particular attribute for a particular case (represented by the subtotals $n_1, \cdots, n_I$) within the same class is:

$$\text{MML} = \sum_j -\ln P_j = \ln\Gamma(N+1) + I\ln\Gamma(1/I) - \sum_{i=1}^{I}\ln\Gamma(n_i + \frac{1}{I}); \quad \text{where} \quad \Gamma(x+1) = x! \tag{5}$$

This equation is found from the sum of the individual serial MML for each particular outcome of an attribute (within a particular class), and rearranging the terms. Note that this total MML equation is independent of the order of encoding of outcomes (since it only depends only on the n_is), as required. This equation can be accurately approximated with the help of Stirlings approximation, and it provides an incremental form for calculating the *change* in MML when when a single case is moved from one class to another.

[1] Eqn. (2) is found by using Bayes' theorem to invert the multinomial distribution, using a conjugate prior [13]. This analysis was originally performed by Laplace over 150 years ago, but the classical statistics literature continues to use the asymptotic result ($P_i = n_i/N$), even though it is meaningless in the small sample case.

In summary, this section describes how to compute the MML for a *given* assignment of the cases to classes, as in Fig. 1. For each class, (i.e. the set of cases with the same class label), calculate MMLs for all attributes using Eqn. (5), and the number of occurrences of each value of that attribute (the n_is). The total MML is the sum of all the attribute MMLs for each case in a class, and the MML required to give the class information. This class description MML is found by regarding the last column of Fig. 1 as if it is just another attribute, and using Eqn. (5) to calculate the corresponding MML (with N here being the total number of cases, and the n_is being the numbers in each class). The total MML is the information required to encode the entire data base, including the information required to to describe the classes. The MML criterion works because cases that tend to have the same attribute values are put into the same class, where their similarity allows the latter cases to be more efficiently encoded.

3 Search Methods

The previous section describes how to compute the total MML for a given class assignment. However, we want *the* class assignment that gives the smallest total MML for all possible class assignments. The space of all possible class assignments is too large in a real problem to use a brute force search. Instead, we calculate the *change* in MML caused by moving a case from one class to another. If this change lowers the MML, a *better* classification has been found. By moving an individual case from one class to another only if it lowers the total MML, a minimum total MML can be found. Unfortunately, such minima are only *local* minima, preventing a simple convergence on the global MML. To partially overcome this problem we have experimented with strategies for searching for the best local minima within the time alloted. This best local minima may also be the global minimum, but this is not likely unless a significant search effort is invested. The search for the best local MML is the main cost of the class discovery process.

3.1 Local Minima

In a series of experiments, we tried starting from different initial random class assignments (number of classes fixed), and moved cases from one class to another if it lowered the MML, until a minimum was reached. That is, we started with a particular data base where each case was given an initial class assignment and cycled through all the cases, one at a time, testing for each case which other classes (if any) would lower the total MML if this case were to be reassigned. If found, the case was moved to the class that lowered the MML the most. This procedure was repeated until no case could be moved to a better (i.e. lower MML) class. Different total MMLs were found (see Fig. 2). This unfortunate result means that a simple minimization procedure will not produce *the* MML—from the variances of the results, it is clear

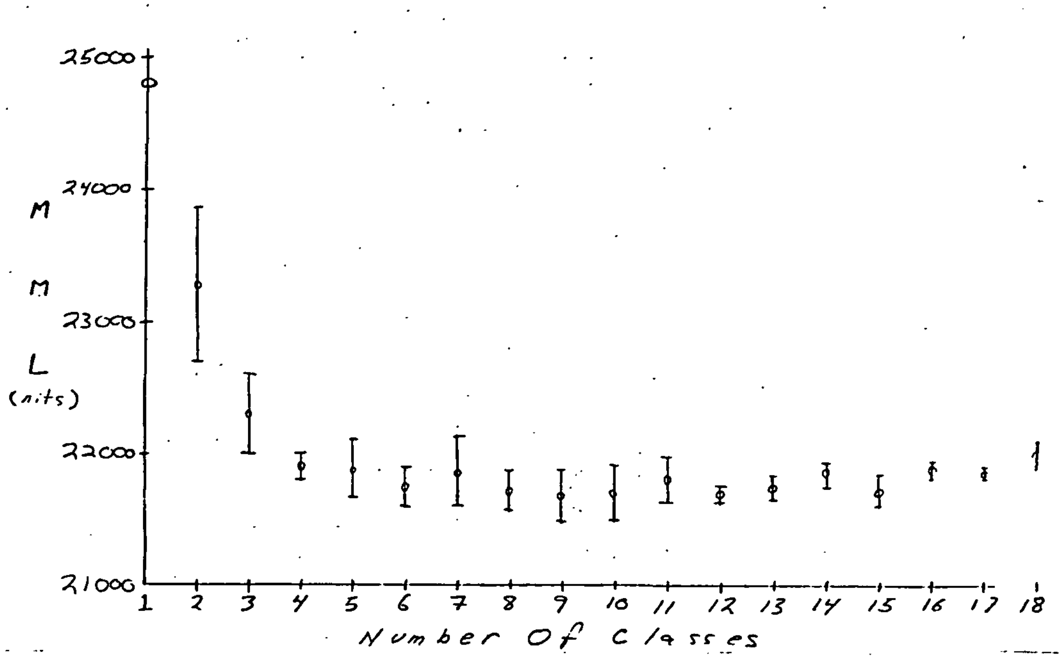

Figure 2: Total MML verses number of classes

that the local minima it produces are not necessarily even close to the global minimum.

When these experiments were performed with a large number of initial classes, the simple minimization procedure correctly reduced the number of classes (by producing classes with no members), but different runs not only produced different MMLs, but also different numbers of classes (see Fig 2).

3.2 Heuristic Search

Because of the local minima problem, we experimented with several heuristics for searching for better local minima. These heuristics apply when the simple minimization procedure has reached a local minima, and include:

1. **Class Splitting**—This heuristic selects a class (with a preference for classes with high average MML per case—i.e. low intra-class cohesion), and randomly splits it into two (or more) classes.
2. **Dispersal**—This heuristic empties the contents of a class randomly over all the other (non-empty) classes.

When either of these heuristics is applied, the cases are then moved to drive the system back into a locally minimal MML. If the new minima MML is greater than the that for the previous state, the procedure backtracks to the previous state and tries another random split or dispersal. This heuristic search produces lower MMLs than just repeatedly generating local minima from many random starts, for similar search time.

3.3 Simulated Annealing

Because we are searching for a global minimum MML in a space with many local minima, a method of search called Simulated Annealing [10] is appropriate. Simulated annealing requires a method of perturbing the current state and evaluating the resulting change in "energy". If this energy is lower than for the previous state then the new state becomes the current state and the cycle continues. If the new energy is greater than the previous (by amount ΔE), then the new state *may* be accepted as the current state with a probability:

$$P = e^{-\frac{\Delta E}{T}} \tag{6}$$

where T is the current "temperature". There is a finite probability that *worse* states will be accepted, with the probability of acceptance decreasing with high ΔE and low T. As a result, simulated annealing may escape from local minima. When simulated annealing is applied to finding the global MML, the current MML is the equivalent of the "energy", and a perturbation consists of altering a class assignment. When simulated annealing was applied to the classification problem, it generally produced classifications with significantly lower MMLs than did heuristic search (for approximately equivalent search effort). Observation of the annealing process showed interesting properties. As the temperature is lowered, the classes with the highest mutual similarity (low average MML) emerge first and are stable. While the MML decreases on average with temperature as classes begin to form, the total MML fluctuates strongly, and the onset of stable class formation greatly reduces the number of classes. There is a strong analogy here between the crystallization of substances from a molten mixture and the emergence of classes at a particular temperature—hence the name simulated annealing.

4 Derivation

The total MML criterion for choosing between alternative classification hypotheses, described in section 3, has the following properties:

4.1 Required Inductive Properties

- The data determines the number of classes—this number does not need to be specified in advance.

- Cases with the most "similar" attribute values end up in the same class.

- Very small classes are discriminated against—unless a small set of cases happen to be almost identical, the "penalty" in discovering the underlying probabilities is not paid for by a reduced MML for the whole classification.

- Very similar classes can be distinguished if there are enough cases in each to expose the (small) differences in their underlying probabilities.

- In many situations, there is a "left-overs" class with a high information (MML) cost per case containing all those cases that do not fit in other "strong" classes.

These are the properties that one would expect of a good inductive class criterion, and they have been amply confirmed in practice—both on real data and artificial data designed to test these properties. This excellent behavior of the MML criterion is not a coincidence, since the MML criterion is just the Bayesian maximum posterior probability criterion. That is, the classification with the smallest MML is also the most probable by Bayes' theorem, as shown below.

4.2 Derivation

Here, we use Bayes' theorem to assign a relative probability between two class hypotheses H_i and H_j. A class hypothesis H_i is a particular partition of the known objects (cases or examples) into specific classes—each class is implicitly defined by the set of cases assigned to it. The *relative* probability of these hypotheses given the data D, by Bayes' theorem is:

$$\frac{P(H_i \mid D)}{P(H_j \mid D)} = \frac{P(H_i)}{P(H_j)} \frac{P(D \mid H_i)}{P(D \mid H_j)} \tag{7}$$

In the absence of any prior knowledge implying that one class hypothesis is more likely a priori than another, we assign them equal probability, so that $P(H_i)/P(H_j) = 1$. This is just an application of the principle of indifference, although in this case, we are also assigning equal probability to hypotheses with different number of classes. This leaves the problem of determining $P(D \mid H_k)$, for a class partition denoted by H_k. In Eqn. (7), D is the entire data base, which can be further partitioned into subsets (D_l) corresponding to the particular class partition H_k. That is:

$$P(D \mid H_k) = P(D_1 \mid C_1) \cdots P(D_l \mid C_l) \cdots P(D_n \mid C_n) \tag{8}$$

i.e., a product of component probabilities, where C_l is the lth class under the class hypothesis H_k, and D_l is the set of cases that define the class C_l. Eqn. (8) shows the explicit partition of the cases into independent classes under the hypothesis H_k. This expansion assumes that information about the members of one class is non-informative about members of another class (i.e. the classes are independent). This assumption is not true, for example, in hierarchical classes. The connection between Bayes' theorem—Eqn. (7)—and minimum message lengths (MMLs) is given by taking logs of Eqn. (7), to give:

$$\log \frac{P(H_i \mid D)}{P(H_j \mid D)} = \log \frac{P(D \mid H_i)}{P(D \mid H_j)} = -\log P(D \mid H_j) - (-\log P(D \mid H_i)) = \Delta MML \qquad (9)$$

That is, the log of the probability ratio of two different class hypotheses H_i and H_j is the difference in the message lengths required to minimally encode the data under the two different hypotheses—the hypothesis with the shortest message length is the most probable. Eqn. (9) implies that for two hypotheses with a difference in message length ΔMML, the ratio of their posterior probabilities is $exp(\Delta MML)$, in favor of the shorter MML. This means that even a relatively small ΔMML can overwhelmingly favor one hypothesis.

Each case c_i from the set $D_l = \{c_1 \cdots c_m\}$, is described by an ordered set of attribute values, where each attribute value is drawn from a fixed set of possible values associated with each attribute. The probability of a case is dependent on the other cases of the same class, i.e.

$$\begin{aligned} P(D_l \mid C_l) &= P(\{c_1, \cdots, c_m\} \mid C_l) \\ &= P(c_1 \mid C_l) P(c_2 \mid c_1, C_l) P(c_3 \mid c_1, c_2, C_l) \cdots P(c_m \mid c_1, \cdots, c_{m-1}, C_l) \qquad (10) \end{aligned}$$

This equation is just the multiplication theorem of standard probability theory corresponding to a particular order of cases. An important property of probability theory is that the joint probability is the same regardless of order in which cases are evaluated. The problem is now to calculate terms such as:

$$P(c_p \mid c_1, \cdots, c_{p-1}, C_l) \qquad (11)$$

Where each case c_p consists of an ordered set of attribute values, i.e $c_p =< a_1, \cdots, a_n >_p$. That is, the probability of a particular attribute value in a case is conditioned on all the attribute values seen in previous cases for that class. If the previous cases are strongly predictive, the conditional probabilities in Eqn. (11) will be different from the probability that would result from assigning all cases to the same class. If we assume that the attributes are independent within a class, we can calculate the probability of the attribute values separately to obtain the joint probability of all the observed attributes, as described in section 2. That is, for independent attributes, the desired probabilities are only dependent on the total frequency of occurrence of the particular attribute values within each class.

5 Limitations (Assumptions) and Extensions

The assumptions (or limitations) built into the MML criterion described in section 2 and derived above are as follows:

5.1 Assumptions

1. That all attributes are *category* variables (e.g. Sex)—the extension to real variables is discussed in the next section.

2. That all attributes are useful in distinguishing classes—the extension to allowing every attribute to be either relevent or irrelevent to a particular class is discussed in the next section.

3. That all classes are independent of each other—this implies that knowledge of the probabilities of particular attributes in one class give no information about the underlying probabilities in any other class. One method of removing this assumption is to introduce hierarchical classes, discussed below.

4. Classifying all cases into a set of mutually exclusive and exhaustive classes is appropriate. Independent classifications, discussed below, provides an alternative.

5. That the attributes within a class are independent—i.e. the attributes are conditionally independent; conditioned on belonging to the given class. This is not correct when attributes such as Height, Weight, Length etc., are used, since they are all dependent on a common "shape" factor. It is possible to correct for such dependencies using interaction terms in a Log-Linear model, but these correction factors are not discussed here.

6. That all the data can be cast in the form of properties of individual cases—i.e. no relations between cases are permitted (see section 2).

7. That all class hypotheses (including ones with different number of classes) are equally likely a priori.

These assumptions are discussed below and possible relaxations are presented.

5.2 Relaxation of Constraining Assumptions

5.2.1 Real and Category Data

Although some attributes, such as blood-type, can only have discrete values, other attributes (variables), such as Age, Blood-pressure, etc., are real valued. It is possible to force these variables into a category form by arbitrarily imposing intervals on the real scale and finding which interval each particular value falls into. This crude method is commonly recommended in statistics texts in order to force all data into the same format. Clearly, this crude approximation is throwing away information—not only by losing where in the interval the particular values fell, but also by ignoring the order of the intervals. Order is irrelevent for category attributes, such as Blood-group, but can be very important for real variables. It is better to treat real variables using models that fit the properties of reals.

The model we assume here is that the real values associated with a particular class are samples from a normal distribution of initially unknown mean and standard deviation. A normal distribution is the expected distribution if there is an underlying (unknown) value and the measurements have random (unbiased) noise associated with them. Alternatively, if there are only two moments to characterize an unknown probability distribution, then the maximum entropy distribution is the normal distribution [14]. Using the normal model, the following formula [13] gives joint probability of a set of real values $(x_1, x_2, ..., x_n)$.

$$p(x_1, x_2, ..., x_n) = \frac{\Gamma\left(\frac{n+1}{2}\right) |x'_1 - x'_2|}{\left(\sqrt{\pi}\right)^{n+1} \left(\sqrt{n+2}\right)^{n+2} s^{n+1}} \, dx_1 \, dx_2 \cdots dx_n,$$

where x'_1 and x'_2 are two fictitious data points which represent our prior knowledge of the real variable. Loosely, x'_1 and x'_2 may be considered to be min-max bounds on x. These prior points are similar in principle to the initial value used in the discrete case, Eqn. (2). Also s is the empirical standard deviation of the data points (including x'_1 and x'_2):

$$s^2 = \overline{(x - \bar{x})^2} = \bar{x}^2 - \overline{x^2}.$$

Note that these equations are scale and location invariant. This means that the origin and units chosen to record the real values x_i does not effect the probability. If the x_is are the observed real attribute values in the data base, then $- \ln P(x_1, x_2, ..., x_n)$ gives us the total MML to encode this data. The terms $dx_1 \, dx_2 \cdots dx_n$ are approximately equal to $\Delta x_1, \cdots, \Delta x_n$—i.e. the errors associated with the individual measurements. Since we are only interested in the *relative* MML, and the $\Delta x \approx dx$s are common to all encodings, they can be dropped from the MML calculation. This formula is analogous to the total MML discrete case Eqn. (5). Although a normal (Gaussian) model was used here, other models with more (or less) parameters, (e.g. Poisson models) could be used instead. In practice, mixed real and categorical information is often given. With appropriate encoding, the MML criterion does not distinguish between the types.

5.2.2 Attribute Relevency

The basic method presented in section 2 assumes that all the attributes are informative in deciding class membership. This restriction can be removed by specifying for each class which attributes are relevent to that class description. Those attributes that are relevent have their MML calculated as previously described previously. Those attributes that are judged irrelevent to a particular class description have their MML calculated using global statistics. That is, all irrelevent attributes are merged into a global set, and it is the n_is of this set that are used in Eqn. (2), instead of the class statistics.

An attribute is judged to be irrelevent to a class description if the total MML is smaller when encoded globally rather than within the class. This is possible, because the prior penalty for encoding globally is shared across all classes for which that attribute is irrelevent, rather than being paid for within each class. In other words, if the probabilities of a particular attribute in a particular class are similar to those of other classes, then a shorter MML is obtained by pooling their statistics. Unfortunately, this complicates the search problem. The movement of a single case from one class to another not only changes the relative MML, but can also change the status of some of the attributes. We now have an extra degree of freedom to search—the relevent/irrelevent assignments for every attribute in every class.

As always in Bayesian inference, you do not get something for nothing. Here, it requires information to specify the relevence/irrelevence information. This additional information penalty must be paid by a reduced MML to encode the data; the data MML is smaller because we can now chose between two sets of statistics—the global or local class statistics. Also, to specify the relevence/irrelevence information it is necessary to specify a prior probability on the possible assignments. If we assume all attributes are as likely to be relevent as irrelevent a priori, the information required to specify the actual assignment for A attributes and M classes is $A \ln(2^M - M)$. This amount of information should be added to the class specification information to give the full total MML. Note that this additional information adds a stronger bias against a large number of classes M unless justified by the data.

5.2.3 Hierachical Classes

The assumption that all classes are independent of each other (as well as being mutually exclusive and exhaustive) may not be correct in many applications. The independence assumption implies that knowledge of the probability distribution for attribute values in one class is non-informative about the corresponding distribution in another class. This assumption can be relaxed by introducing hierarchical classes, where classes closer together on the (hierarchical) tree are closer to each other. "Closer" here means that the classes tend to share more attributes, rather than encode then separately. Sharing attributes is very similar to the relevence/irrelevence criterion discussed above. The idea is that shared attributes use statistics obtained by pooling all the data from the shared attributes belonging to the classes on the same branch of the tree. The closer classes are to each other on the tree, the higher the expectation that they will have shared attributes. This higher expectation is reflected in longer MMLs to specify which attributes are *not* shared. Intuitively, closeness of classes on the tree (i.e. adjoining sub-classes) indicates strong similarity between the classes. Unfortunately, the search space of possible hierarchical classes is considerably larger than just the "flat" search space in section 3. This means that local minima in possible hierarchies occurs as well as local minima in the number and contents of possible classes.

5.2.4 Independent Classifications

Perhaps the strongest assumption is that a classification is appropriate at all! That is, there are many situations where the assumption of mutually exclusive and exhaustive classes is not appropriate. HIV infected patients are either [Non-symptomatic, Pre-AIDS, ARC or AIDS]—i.e., such patients *must* be one or other of these alternatives. On the other hand, diseases are not mutually exclusive, so it is possible for a patient to both have typhoid *and* cholera simultaneously.

The MML criterion can be extended to deal with situations where multiple (overlapping) classifications are possible. The method is to add more ("independent") classifications, where the optimal number of classes within each classification and the optimal number of different classifications is to be determined by the data. For example, in Fig. 1, we would add extra classification columns, so that every case now belongs to a class within each classification (i.e. within each column). The goal is now to find the optimal number of columns and classes within each column. To compute the smallest total MML with multiple classifications (columns), the problem is to calculate the probability of each observed attribute value given multiple class membership. If we assume that the classifications are independent of each other, and also conditionally independent given each attribute value, the required equation is:

$$P(A_i \mid C_j, D_k, \cdots) = \frac{P(A_i \mid C_j)P(A_i \mid D_k)\cdots}{P(A_i)^{n-1}} \tag{12}$$

where n is the number of classifications, and $P(A_i \mid C_j)$ is the probability of the ith attribute value given that the case is in the jth class under the Cth classification (and similarly for the Dth classification). The terms such as $P(A_i \mid C_j)$ are calculated as previously described in section 2. The presence of extra classifications increases the size of the search space considerably. During incremental search, the class a particular case is assigned to under one classification can be changed (and the resulting ΔMML calculated) without disturbing any other classification. The independence assumptions built in to Eqn. (8) could be relaxed by introducing cross terms (joint probabilities) between the classes under different classifications. This interdependence considerably complicates the analysis, and is not considered further here.

5.2.5 Extended Models

The smallest MML criterion for finding the best classification model (and its various extension discussed in this section) is of much greater generality. The derivation in section 4.2 has been specialized to classification, but other models are possible. For a description of applications of the MML criterion to domains such as learning of grammars, finite state machines, line finding etc. see [8]. A natural extension of the automatic classification approach is to include time dependent data bases—i.e. trend analysis. This extension will require models of how systems evolve with time and priors on these possible models. Temporal models are being investigated.

6 Prediction

The purpose of finding good classifications is usually not clear in the AI and statistical pattern recognition literature. The classification work reported here was motivated by the desire to make (probabilistic) predictions directly from data. A previous approach [4], based on maximum entropy, allowed the direct calculation of the (conditional) probability of any attribute value given any combination of other attribute values (i.e. given particular evidence). The domain information in the previous approach consisted of a set of joint probabilities (constraints) that summarized all the significant information in the domain. These significant constraints were found by comparing the *expected* probabilities of attribute value combinations with the *observed* values in a data base. Unfortunately, the cost of computing the expected probabilities increases exponentially with the number of known constraints, and becomes prohibitive when there are many overlapping probabilistic interactions. The classification approach described here was tried as a method of avoiding this computational bottle-neck.

In prediction, there are always two steps—the first step is to induce the model from the known data (in our case the model is the best classification); and the second step is to use the model and information about a particular case to make a (probabilistic) prediction The first step has been the subject of this paper so far, the second step consists of taking the information about the particular case and finding the corresponding posterior probability distribution over the set of possible classes, then using these new class probabilities to calculate the probability of the attributes of interest. That is, prediction is performed indirectly, by mapping the available information into a probabilistic class membership distribution (for the particular case), then making predictions based on this class membership. From the conditional independence assumption (section 5.1) built into our class induction procedure, the necessary equations are:

$$\begin{aligned} P(A_i \mid P_j, Q_k, \cdots) &= \sum_{\text{classes}(c)} P(A_i \mid C_c, P_j, Q_k, \cdots) \times P(C_c \mid P_j, Q_k, \cdots) \\ &= \sum_{\text{classes}(c)} P(A_i \mid C_c) \times P(C_c \mid P_j, Q_k, \cdots) \qquad \text{Conditional Independence} \end{aligned}$$

where $P(A_i \mid P_j, Q_k, \cdots)$ is the probability of the ith value of attribute A given that the jth value of the Pth attribute occurred, etc. $P(C_c \mid P_j, Q_k, \cdots)$ is the probability of the the cth class of the C classification. This is given by:

$$P(C_c \mid P_j, Q_k, \cdots) = \frac{P(C_c)P(P_j \mid C_c)P(Q_k \mid C_c)\cdots}{P(P_j, Q_k, \cdots)} \tag{13}$$

Note that $P(P_j, Q_k, \cdots)$ is a normalizing constant that can be deduced from the requirement that $\sum_c P(C_c \mid P_j, Q_k, \cdots) = 1$. These equations are all that is necessary for making probabilistic predictions, because terms such as $P(A_i \mid C_c), P(P_j \mid C_c)$ are known from the statistics obtain during the class induction procedure. This method of using the information about the particular case to decide the probability of class membership for that case, then using a weighted sum of this membership information to make a prediction is a familiar pattern of human inference. However, in people there is a strong tendency to decide *which* class the case belongs to and make all predictions based on this identification. Clearly, this is non-optimal behavior, since the information available is usually insufficient to make a clear identification.

This probabilistic class formation and prediction sheds some light on the controversy surrounding fuzzy sets [2]. Zadeh [16], and many philosophers have noted that most common concepts (e.g. cat, tall, chair etc.) do not have sharp definitions that allow all members to be distinguished from non-members—a basic requirement for classical sets and predicates. The approach described in this paper shows that for induced classes, clear class boundaries are *not* required. Instead, we have a probabilistic definition of classes in terms of possible attribute values that allows us to give any case a probabilistic "degree of membership"—i.e. the probability distribution given in Eqn. (11). A special case of this probabilistic analysis is so called "tutored learning", where the class information is given, and the aim is to find the class definition [12]. Because the class information is given, it is unnecessary to search—it is only necessary to calculate the total MML for the given classification. In the process, the necessary statistics will be collected (i.e. the necessary probabilities are computed), so that the probability of class membership can be computed using Eqn. (11).

7 Discussion

The work reported in this paper may be regarded as a first step in the direction of truly intelligent machines. The basic Bayesian theory for (probabilistic) induction provides a sound basis for learning and prediction. The translation of the Bayesian criterion into an information theoretic criterion (the MML criterion) provides a computationally convenient measure. The major computational problem is a search problem, since the search space (the space of possible theories) is far too large to be searched exhaustively. The major research issues for the future are how to best perform such searches—the simulated annealing approach has yielded promising results.

To apply the Bayesian approach to other problems will require the specification of the hypothesis (theory) space and prior probabilities over them. It is this step which has caused so much controversy in the past [3],[5], but the idea that it is possible to perform induction without such a step has yet to produce a useful result. With few exceptions (e.g. ID3 [12]), the majority of work in induction (learning) in AI has implicitly used prior probabilities, but hidden them in general "principles", such as "always prefer the simlpest theory unless there is evidence to the contrary". The operational definitions of "simplicity" and "evidence to the contrary" embedded in these inductive programs are just *ad hoc* attempts to meet correct Bayesian definitions.

References

[1]Boulton, D. M. and Wallace, C. S., "An Information Measure for Hierarchical Classification", **Computer Journal**, 16, (3), pp 57-63, 1973.

[2]Cheeseman, P. C., "Probabilistic verses Fuzzy Reasoning", in "Uncertainty in AI", Nth. Holland, Eds. Kanal and Lemmer, 1986.

[3]Cheeseman, P. C., "In Defense of Probability", Proc. Ninth International Conference on Artificial Intelligence, Los Angeles, Aug. 1985, pp 1002-1009.

[4]Cheeseman, P. C., "A Method of Computing Generalized Bayesian Probability Values for Expert Systems, Proc. Eight International Conference on Artificial Intelligence, Karlruhe, Aug. 1983, pp 198-202.

[5]Cheeseman, P. C., "Learning Expert Systems from Data, Proc. Workshop on Principles of Knowledge-Based Systems, Denver, pp 115-122, Dec. 1984.

[6]Cheeseman, P. C., "Induction of Models under Uncertainty", Int. Methodologies in A.I., Knoxville, Tennessee, 1986.

[7]Duda, R. O. and Hart, P. E., "Pattern Recognition and Scene Analysis", Wiley-Interscience, 1973.

[8]Georgeff, M., "A General Selection Criterion for Inductive Inference", Proc. 6th. European Conf. on AI, Pisa, Italy, Sept., 1984.

[9]Howard, R., (to be inserted).

[10]Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P., "Optimization by Simulated Annealing", **Science**, 220, (4598), pp 671-680, May 1983.

[11]Kolmogorov, A. N., "Three Approaches to the Quantitative Definition of Information", Problems of Information Transmission, 1, 1, pp 1-7, 1965.

[12]Michalski, R. S., Carbonell, J. G. and Mitchell, T. M. (Eds), "Machine Learning: An Artificial Intelligence Approach", Tioga Press, Palo Alto, 1983.

[13]Self, M., and Cheeseman, P., "Probabilistic Model Based Inference: A Bayesian Analysis of Deductive and Inductive Reasoning", NASA tech. note (to appear)

[14]Tribus, M., "Rational Descriptions, Decisions and Designs", Pergamon Press, NY, 1969.

[15]Wallace, C.S. and Boulton, D.M. "An Information Measure for Classification, Computer Journal, 11, 2, pp 185-194, 1968.

[16]Zadeh, L. A., "Possibility Theory and Soft Data Analysis, In Mathematical Frontiers of the Social and Policy Sciences, Ed. L. Cobb and R. M. Thrall, pp 69-129.