AutoClass: A Bayesian Classification System

PETER CHEESEMAN* (CHEESEMAN@PLUTO.ARC.NASA.GOV)
JAMES KELLY† (KELLY@PLUTO.ARC.NASA.GOV)
MATTHEW SELF† (SELF@PLUTO.ARC.NASA.GOV)
JOHN STUTZ‡ (STUTZ@PLUTO.ARC.NASA.GOV)
WILL TAYLOR† (TAYLOR@PLUTO.ARC.NASA.GOV)
DON FREEMAN†

NASA Ames Research Center
Mail Stop 244-17, Moffett Field, CA 94035 U.S.A.

Abstract

This paper describes AutoClass II, a program for automatically discovering (inducing) classes from a database, based on a Bayesian statistical technique which automatically determines the most probable number of classes, their probabilistic descriptions, and the probability that each object is a member of each class. AutoClass has been tested on several large, real databases and has discovered previously unsuspected classes. There is no doubt that these classes represent new phenomena.

1 Introduction

The standard approach in much of AI and statistical pattern recognition research is that a classification consists of a partitioning of the data into separate subsets, and that these subsets *are* the classes. In the Bayesian approach classes are described by probability distributions over the attributes of the objects, specified by a model function and its parameters. To define a class is to describe (not list) the objects which belong to it. This approach appears in the statistical literature as the theory of finite mixtures [5].

The Bayesian approach has several advantages over other methods:

- **The number of classes is determined automatically.**

  Deciding when to stop forming classes is a fundamental problem in classification. More classes can always explain the data better, so what should limit the number of classes the program finds? Many systems rely on an *ad hoc* stopping criterion. The Bayesian solution to the problem lies in the use of prior knowledge. We believe simpler class hypotheses (e.g., those with fewer classes) to be more likely

*RIACS. This work partially supported by NASA grant NCC2–428.
†Sterling Software (Don Freeman is now at the University of Pittsburgh)
‡NASA Ames Research Center

Originally published in *Proceedings of the Fifth International Workshop on Machine Learning*, ©1988, Morgan Kaufmann, San Mateo, CA.

From: Readings in Machine Learning

than complex ones, in advance of seeing any data, and the *prior probability* of the hypothesis reflects this preference. The prior probability term prefers fewer classes, the likelihood of the data prefers more, and the two effects balance at the most probable number of classes. As a result, AutoClass finds only one class in random data.

- **Objects are not assigned to classes absolutely.**

  AutoClass calculates the probability of each object's membership in each class, providing a more intuitive classification than absolute partitioning techniques. An object described equally well by two class descriptions should not be assigned to either class with certainty, because the evidence cannot support such an assertion.

- **All attributes are potentially significant.**

  Classification can be based on any or all attributes simultaneously, not on just the most important one. This represents an advantage of the Bayesian method over human classification. In many applications, classes are distinguished not by one or even by several attributes, but by small differences in many. Humans often have difficulty taking more than a few attributes into account. The Bayesian approach utilizes all attributes simultaneously, permitting uniform consideration of all the data.

Data can be real or discrete.

Many previous methods have difficulty analyzing mixed data. Some methods insist on real valued data [2], while others accept only discrete data [6]. There have been attempts to reconcile the two types of data by coercing real data into discrete form [13] or by incorporating flexible thresholds into categorical classification [11]. Coercion of heterogeneous data to a single type destroys information and is done purely to meet the needs of the particular classification procedure. The Bayesian approach can utilize the data exactly as they are given.

2 Overview of Bayesian Classification

AutoClass is based on Bayes's theorem, a formula for combining probabilities. Given observed data D and a hypothesis H, it states that the probability that the hypothesis explains the data $p(H \mid D)$, (called the *posterior* probability of the hypothesis given The data) is proportional to the probability of observing the data if the hypothesis were known to be true $p(D \mid H)$ (the *likelihood* of the data) times the inherent probability of the hypothesis regardless of the data ($p(H)$, the *prior* probability of the hypothesis). Bayes's theorem is commonly expressed

$$p(H \mid D) = \frac{p(H)\, p(D \mid H)}{p(D)}. \tag{1}$$

For our purposes, the hypothesis H is the number and descriptions of the classes from which we believe the data D to have been drawn. Given D, we must select H to maximize the posterior $p(H \mid D)$.

For a specific classification hypothesis, calculation of the likelihood of the data involves a straightforward application of statistics. The prior probability of the hypothesis is less transparent and is taken up in section 2.3. Finally, the prior probability of the data, $p(D)$ in the denominator above, need not be calculated directly. It can be derived as a normalizing constant or ignored so long as we seek only the relative probabilities of hypotheses.

2.1 Application to Classification

In the theory of finite mixtures (the mathematical foundation of AutoClass) each datum in a database containing I objects is assumed to be drawn from one of J classes. Each class is described by a *class distribution* function, $p(x_i \mid x_i \in C_j, \vec{\theta}_j)$, which gives the probability distribution of the attributes of a datum if it were known to belong to class C_j. These class distributions are described by a *class parameter vector*, $\vec{\theta}_j$, which for a single attribute normal distribution would consist of the class mean, μ_j, and variance, σ_j^2.

The probability of an object being drawn from class j is called the *class probability* π_j. Thus, the probability of a given datum coming from a set of classes is the sum of the probabilities that it came from each class separately, weighted by the class probabilities.

$$p(x_i \mid \vec{\theta}, \vec{\pi}, J) = \sum_{j=1}^{J} \pi_j \, p(x_i \mid x_i \in C_j, \vec{\theta}_j). \tag{2}$$

We assume that the data are drawn from an exchangeable (static) process—that is, the data are unordered and independent of each other given the model. Thus the *likelihood* of measuring an entire database is the product of the probabilities of measuring each object.

$$p(\vec{x} \mid \vec{\theta}, \vec{\pi}, J) = \prod_{i=1}^{I} p(x_i \mid \vec{\theta}, \vec{\pi}, J) \tag{3}$$

For a given value of the class parameters, we can calculate the probability that object i belongs to class j using Bayes's theorem.

$$p(x_i \in C_j \mid x_i, \vec{\theta}, \vec{\pi}, J) = \frac{\pi_j \, p(x_i \mid x_i \in C_j, \vec{\theta}_j)}{p(x_i \mid \vec{\theta}, \vec{\pi}, J)} \tag{4}$$

These classes are "fuzzy" in the sense that even with perfect knowledge of an object's attributes, it will be possible to determine only the probability that it is a member of a given class.

We break the problem of identifying a finite mixture into two parts: determining the classification parameters for a given number of classes, and determining the number of classes. Rather than seeking an *estimator* of the classification parameters (the class parameter vectors, $\vec{\theta}$, and the class probabilities, $\vec{\pi}$), we seek their full *posterior* probability distribution. The posterior distribution is proportional to the product of the prior distribution of the parameters $p(\vec{\theta}, \vec{\pi} \mid J)$ and the likelihood function $p(\vec{x} \mid \vec{\theta}, \vec{\pi}, J)$.

$$p(\vec{\theta}, \vec{\pi} \mid \vec{x}, J) = \frac{p(\vec{\theta}, \vec{\pi} \mid J)\, p(\vec{x} \mid \vec{\theta}, \vec{\pi}, J)}{p(\vec{x} \mid J)} \tag{5}$$

The pseudo-likelihood $p(\vec{x} \mid J)$ is simply the normalizing constant of the posterior distribution, obtained by marginalizing (integrating) out the classification parameters—in effect, treating them as "nuisance" parameters:

$$p(\vec{x} \mid J) = \int\!\!\int p(\vec{\theta}, \vec{\pi} \mid J)\, p(\vec{x} \mid \vec{\theta}, \vec{\pi}, J)\, d\vec{\theta}\, d\vec{\pi}. \tag{6}$$

To solve the second half of the classification problem (determining the number of classes) we calculate the posterior distribution of the number of classes J. This is proportional to the product of the prior distribution $p(J)$ and the pseudo-likelihood function $p(\vec{x} \mid J)$.

$$p(J \mid \vec{x}) = \frac{p(J)\, p(\vec{x} \mid J)}{p(\vec{x})} \tag{7}$$

In principle we can determine the most probable number of classes by evaluating $p(J \mid \vec{x})$ over the range of J for which our prior $p(J)$ is significant. In practice, the multi-dimensional integrals of equation 6 are computationally intractable, and we must search for the maximum of the function and approximate it about that point. Details of the AutoClass algorithm appear in section 3.

2.2 Assumptions

We cannot attempt classification without making some assumptions. Our mathematical formulation of the problem permits us to state our assumptions precisely and to assess their validity. The derivation above incorporates two:

1. The data are independent given the model. That is, the data are unordered. This is a fundamental assumption intrinsic to all classification systems.

2. The model functions—*class distributions* of page 3—are appropriate descriptors of the classes. The model functions themselves may incorporate additional assumptions, such as independence of attribute values (as AutoClass currently does).

2.3 Prior Probabilities

The prior probability term $p(\vec{\theta}, \vec{\pi} \mid J)$ in equation 5 constitutes the fundamental difference between Bayesian and classical statistics and still fuels debate. We will not attempt to defend the use of priors herein but refer the skeptical reader to Jaynes [8] for a full explanation of the Bayesian approach.

Introduction of the prior probability solves two problems. First, it permits mathematical determination of the number of classes by introducing a preference for fewer classes. We believe *a priori* that complex hypotheses are less likely than simple ones, those with fewer classes, for example, and any reasonable prior implants this belief in the equations. A more complex hypothesis will incur a penalty in the prior and will be disfavored unless it explains the data significantly better. Second, the likelihood contains singularities that would complicate analysis if it were used alone. The prior tames the likelihood by damping out the singularities, and the resulting posterior is much better behaved.

However, the prior probability distribution is not completely arbitrary. There are two basic approaches to prior distributions. A prior may be sought which captures some prior knowledge which is available, or an *uninformative* prior distribution may be sought. Uninformative priors which are invariant to changes of scale or origin are available, but the AutoClass II program does not as yet use these priors. Rather, AutoClass uses a weak informative prior. The actual values used do not appreciably affect the classifications found by AutoClass. The actual priors used are discussed in Section 3.2

3 The AutoClass II Program

Section 2 described the theory behind the Bayesian approach to classification. We now outline its implementation, AutoClass II.

3.1 The AutoClass II Class Model

In AutoClass II, we assume the data are in an attribute-value vector form. That is, the database contains I objects x_i, each described by K attribute values x_{ik}, $k \in \{1 \ldots K\}$. Attributes may be either real or discrete variables. In AutoClass II we currently make the further strong assumption that the attributes are independent in each class. This permits an extremely simple form for the class distributions used in equation 2.

$$p(x_i \mid x_i \in C_j, \vec{\theta}_j) = \prod_{k=1}^{K} p(x_{ik} \mid x_i \in C_j, \vec{\theta}_{jk}) \tag{8}$$

where $\vec{\theta}_{jk}$ is the parameter vector describing the kth attribute in the jth class C_j. We plan to extend AutoClass to model covariance of the attributes within a class in the near future.

AutoClass models real valued attributes with a Gaussian normal distribution, parameterized by a mean and a standard deviation, and thus $\vec{\theta}_{jk}$ takes the form

$$\vec{\theta}_{jk} = \begin{bmatrix} \mu_{jk} \\ \sigma_{jk} \end{bmatrix}.$$

The class distribution is thus

$$p(x_{ik} \mid x_i \in C_j,\ \mu_{jk}, \sigma_{jk}) = \frac{1}{\sqrt{2\pi}\sigma_{jk}} \exp\left[-\frac{1}{2}\left(\frac{x_{ik}-\mu_{jk}}{\sigma_{jk}}\right)^2\right]. \tag{9}$$

For discrete attributes the class distribution is specified by the probability ρ_{jkl} of getting each possible value l. The elements of $\vec{\theta}_{jk}$ are the probabilities themselves. If there are L possible values, labeled 1 to L, then the class distribution is

$$p(x_{ik} = l \mid x_i \in C_j,\ \vec{\rho}_{jk}) = \rho_{jkl}. \tag{10}$$

3.2 Informative Priors

Although uninformative priors can be derived for use in Bayesian classification, AutoClass II currently uses informative priors. This is due mostly to the history of the program, and the fact that we have obtained excellent results using these priors. The prior information we use has only a small effect on the classification estimates.

AutoClass II employs *conjugate* priors, that is, prior information in the same form as the data. In effect, the prior information for attribute k in class j consists of a set of w' fictitious data points, described by the same summary statistics as are used for the actual data. The larger the value of w', the stronger the influence of the prior relative to the data. AutoClass II currently treats all classes symmetrically, using the same conjugate points for every class and for each attribute of the same type.

3.3 Search Algorithm

As mentioned in section 2, AutoClass breaks the classification problem into two parts: determining the number of classes and determining the parameters defining them. Equation 7 gives the probability distribution over the number of classes. For each possible number of classes the multi-dimensional integral of equation 6 must be performed. Rather than attempt the integration the posterior distribution of the classification parameters directly, AutoClass performs a search over $\vec{\theta}_j$ and π_j to find the maximum of the posterior distribution (equation 5), and then approximates the integral around that point.

The complete problem involves starting with more classes than are beleieved to be present (as specified by the user), searching to find the best class parameters for that number of classes, approximating the integral to find the relative probability of that number of classes, and then decreasing the number of classes and repeating the procedure.

AutoClass uses a Bayesian variant of Dempster and Laird's EM algorithm [1] to find the best class parameters for a given number of classes (the maximum of equation 5). To derive the algorithm, we differentiate the posterior distribution with respect to the class parameters and equate with zero. This yields a system of nonlinear equations which hold at the maximum of the posterior:

$$\hat{\pi}_j = \frac{W_j \; + \; w' - 1}{I \; + \; J(w' - 1)} \tag{11}$$

$$\frac{\partial}{\partial \theta_j} \ln p(\hat{\theta}_j) + \sum_{i=1}^{I} w_{ij} \frac{\partial}{\partial \theta_j} \ln p(x_i \mid \hat{\theta}_j) = 0, \tag{12}$$

where w_{ij} is the probability that the datum x_i was drawn from class j (previously given in equation 4), and W_j is the total weight in class j:

$$\begin{aligned} w_{ij} &= p(x_i \in C_j \mid x_i, \hat{\theta}, \hat{\pi}) \\ W_j &= \sum_{i=1}^{I} w_{ij}. \end{aligned}$$

To find a solution to this system of equations, we iterate between equations 11 and 12 (treating $\vec{w}$ as a constant) and equation 4 (treating $\vec{\pi}$ and $\vec{\theta}$ as constants).

On any given iteration, the membership probabilities are constant, so equation 12 can be simplified by bringing w_{ij} through the derivative, giving

$$\frac{\partial}{\partial \theta_j} \left[p(\hat{\theta}_j) \prod_{i=1}^{I} p(x_i \mid \hat{\theta}_j, x_i \in C_j)^{w_{ij}} \right] = 0. \tag{13}$$

Thus far, our discussion of the search algorithm has related to a general class model with an arbitrary $\vec{\theta}_{jk}$. We now apply equation 13 to the specific AutoClass II model of equations 8 through 10. For discrete attributes, the values of the updated parameters $\hat{\theta}_{jkl}$ derived from the class probabilities w_{ij} and prior weight w' are

$$\hat{\rho}_{jkl} \;=\; \frac{\sum_{i=1}^{I} w_{ij} \delta(l,\, x_{ik}) + w' - 1}{W_j + L(w' - 1)} \qquad \delta(l,\, x_{ik}) \;\equiv\; \begin{cases} 1, & x_{ik} = l \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

This is simply the weighted proportion of objects in class j for which attribute k had value l.

For real valued attributes, the equations for the updated $\hat{\mu}_{jk}$ and $\hat{\sigma}_{jk}$ are a function of the prior information and the empirical mean, $\bar{x}_{jk}$, and variance, s^2_{jk}, of the kth attribute in class j, weighted by w_{ij}:

$$\begin{aligned} \bar{x}_{jk} &= \frac{\sum_{i=1}^{I} w_{ij} x_{ik}}{W_j}, \\ s^2_{jk} &= \frac{\sum_{i=1}^{I} w_{ij} x^2_{ik}}{W_j} - \bar{x}^2_{jk}. \end{aligned}$$

The update formulas are then:

$$\hat{\mu}_{jk} \;=\; \frac{w' \bar{x}'_k + W_j \bar{x}_{jk}}{w' + W_j} \tag{15}$$

$$\hat{\sigma}^2_{jk} \;=\; \frac{w'(s'_k)^2 + W_j s^2_{jk}}{w' + W_j + 1} + \frac{w' W_j}{(w' + W_j)(w' + W_j + 1)} (\mu'_k - \bar{x}_{jk})^2 . \tag{16}$$

The computational cost of a single iteration is of order $I \cdot J \cdot K$. Typically the procedure converges in about twenty iterations, depending upon the strength of the actual classes. A search in data having many weak classes takes longer than one having few strong classes. The search may be speeded up by over-relaxation techniques. Of course, the procedure may converge to local maxima depending on the starting point chosen for the iteration, so we employ heuristic methods to jump away from local maxima. Even so, many searches may be necessary to establish the global maximum.

3.4 Determining the Number of Classes

We previously outlined the theory for the determination of the number of classes present, but integration over the full parameter space is clearly infeasible. AutoClass II uses a crude but very effective approximation based solely on the results of the iteration algorithm. If a class has negligible posterior probability π_j, then including that class in the model cannot improve the likelihood of the data at all. At the same time, the prior probability of one class probability being near zero is very low. Thus models in which a class has negligible probability will always be less probable than models which simply omit that class. The user runs AutoClass with J larger than the expected number of classes. If all resulting classes have significant probability then the user increases J until some classes are empty. AutoClass then ignores the empty classes, and the populated classes represent an optimal classification of the data given the assumed class model function.

The utility of this approach has been experimentally confirmed on a number of prepared data bases. Specifically, when AutoClass runs with J greater than the actual number of classes present, the iteration converges with negligible probability for the extra classes. This behavior differs qualitatively from the behavior of maximum likelihood methods, which will continue to partition the classes until eventually there is only one object in each class.

4 Extensions to the Model

4.1 Hierarchical Classification

After a database has been analyzed, many classes frequently have many attributes in common. For instance, in a database of mammals, the "dog" class and the "cat" class will be described by the same values for many attributes (fur, four legs, etc.). In this case, a description of all the attributes of all the classes separately is not as useful as a *hierarchical* classification scheme which identifies the common attributes and the distinguishing attributes between classes.

The Bayesian method can accomodate hierarchical classification by considering a model in which some attributes are common to a group of sub-classes. This amounts to a significance test of the equality of two attribute's parameter vectors in the non-hierarchical classification. If there is no significant difference between some attributes of a group of classes, then these attributes may be estimated jointly.

4.2 Supervised Classification

Although AutoClass was designed for automatic unsupervised classification, the prior information in equation 5 permits supervised classification as well. If the user wishes to assert that certain objects are in certain classes, a prior probability can be used which favors class descriptions reflecting this. See Duda and Hart [4] for a discussion of supervised Bayesian inference.

4.3 Missing Data

Consistent probability calculations require that 'unknown' be treated as a valid data value. This is not merely a computational convenience. Failure to determine a value is just as valid an observation as any other, and must be allowed for in any predictive model. There may be physical reasons that a value is unknown, and discarding that fact (by interpolating a value or, even worse, discarding the object completely) destroys potentially valuable information.

A straightforward extension of the class model allows AutoClass to accept objects with unknown values. For discrete attributes it can be shown that the correct procedure for treating an unknown value is equivalent to adding an 'unknown' category to the value set. For real-valued attributes we condition our Gaussian normal model with discrete 'unknown' and 'known' categories:

$$\vec{\theta}_{jk} = [\rho_{jk}, \mu_{jk}, \sigma_{jk}] \tag{17}$$

$$p(x_{ik} \mid x_i \in C_j,\ \rho_{jk}, \mu_{jk}, \sigma_{jk}) = \begin{cases} \rho_{jk} \frac{1}{\sqrt{2\pi}\,\sigma_{jk}} \exp\left[-\frac{1}{2}\left(\frac{x_{ik}-\mu_{jk}}{\sigma_{jk}}\right)^2\right], & x_{ik} \text{ known} \\ 1-\rho_{jk}, & x_{ik} \text{ unknown} \end{cases} \tag{18}$$

The mean and variance are updated as before, but the proportion of data for which x_k is known is also updated just like any other discrete variable.

5 Results

AutoClass has classified data supplied by researchers active in various domains and has yielded some new and intriguing results:

- **Iris Database** Fisher's data on three species of iris [7] are a classic test for classification systems. AutoClass discovers the three classes present in the data with very high confidence, despite the fact that not all of the cases can be assigned to their classes with certainty. Wolfe's NORMIX and NORMAP [12] both incorrectly found four classes, and Dubes's MH index [3] offers only weak evidence for three clusters.

- **Soybean Disease Database** AutoClass found the four known classes in Stepp's soybean disease data, providing a comparison with Michalski's CLUSTER/2 system [10]. AutoClass's class assignments exactly matched Michalski's—each object belonged overwhelmingly to one class, indicating exceptionally well separated classes for so small a database (47 cases, 35 attributes).

- **Horse Colic Database** AutoClass analyzed the results of 50 veterinary tests on 259 horses and extracted classes which provided reliable disease diagnoses, despite the fact that almost 40% of the data were missing.

- **Infrared Astronomy Database** The Infrared Astronomical Satellite tabulation of stellar spectra is not only the largest database Autoclass has assayed (5,425 cases, 94 attributes) but the least thoroughly understood by domain experts. AutoClass discovered classes which differed significantly from NASA's previous analysis but clearly reflect physical phenomena in the data.

  Note that AutoClass knows nothing about spectra—the current model treats the intensity at each wavelength as an independent quantity. As a result AutoClass would find exactly the same classes if the order of the wavelengths were scrambled. The AutoClass infrared source classification is the basis of a new star catalog to appear shortly.

We are actively collecting and analyzing other databases which seem appropriate for classification, including an AIDS database and a second infrared spectral database.

6 Conclusion

This paper has described the Bayesian approach to the problem of classification and AutoClass, a simple implementation of it. Bayesian probability theory provides a simple and extensible approach to problems such as classification and general mixture separation. Its theoretical basis is free of *ad hoc* quantities, and in particular free of any measures which alter the data to suit the needs of the program. As a result, the elementary classification model we have described lends itself easily to extensions.

References

[1] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[2] W. Dillon and M. Goldstein. *Multivariate Analysis: Methods and Applications*, chapter 3. Wiley, 1984.

[3] Richard C. Dubes. How many clusters are best? — an experiment. *Pattern Recognition*, 20(6):645–663, 1987.

[4] Richard O. Duda and Peter E. Hart. *Pattern Recognition and Scene Analysis.* Wiley-Interscience, 1973.

[5] B. S. Everitt and D. J. Hand. *Finite Mixture Distributions. Monographs on Applied Probability and Statistics*, Chapman and Hall, London, England, 1981. Extensive Bibliography.

[6] D. H. Fisher. Conceptual clustering, learning from examples, and inference. In *Proceedings of the Fourth International Workshop on Machine Learning*, pages 38–49, Morgan Kaufmann, 1987.

[7] R. A. Fisher. Multiple measurments in taxonomic problems. *Annals of Eugenics*, VII:179–188, 1936.

[8] Edwin T. Jaynes. Bayesian methods: general background. In James H. Justice, editor, *Maximum Entropy and Bayesian Methods in Applied Statistics*, pages 1–25, Cambridge University Press, Cambridge, Massachusetts, 1986.

[9] Edwin T. Jaynes. *Papers on Probability, Statistics and Statistical Physics.* Volume 158 of *Synthese Library*, D. Reidel, Boston, 1983.

[10] Ryszard S. Michalski and Robert. E. Stepp. Automated construction of classifications: conceptual clustering versus numerical taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5:396–410, 1983.

[11] J. R. Quinlan. Decision trees as probabilistic classifiers. *Proceedings of the Fourth International Workshop on Machine Learning*, 31–37, 1987.

[12] John H. Wolfe. Pattern clustering by multivariate mixture analysis. *Multivariate Behavioural Research*, 5:329–350, July 1970.

[13] A. Wong and D. Chiu. Synthesizing statistical knowledge from incomplete mixed-mode data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9:796–805, 1987.