NASA Disk 3 ICT 505 8th. Ave., NY, NY 10018 **ARC-12797**
BL 11/27/90

Bayesian Classification Scheme

A statistical approach leads to automatic identification of classes.

Ames Research Center, Moffett Field, California

A scheme derived via a statistical approach automatically identifies classes in a set of data. The scheme is applicable to real-valued, discrete (e.g., binary) or continuous data in the form of parameter vectors that represent the attributes of objects. It determines the most probable number of classes, the probabilistic descriptions of the classes, and the probability that each object is a member of each class. It incorporates partial prior knowledge of some classes that may be present, striking a balance between the tendency to put each datum in a class by itself and the tendency to put all data in the same class.

The scheme is based on the theory of finite mixtures, in which each datum in a set of I data is assumed to be drawn from one of J classes. The jth class, C_j, is described by a class distribution function, $p(x_i \mid x_i \in C_j, \boldsymbol{\theta}_j)$, which gives the probability distribution of the parameters of a datum (components of a vector) that belongs to that class. The jth class distribution is described by a class parameter vector, $\boldsymbol{\theta}_j$. For a single-attribute normal distribution, $\boldsymbol{\theta}_j$ consists 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 $\boldsymbol{\pi}_j$. 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.

$$p(x_i \mid \boldsymbol{\theta},\boldsymbol{\pi},J) = \sum_{j=1}^{J} \boldsymbol{\pi}_j \, p(x_i \mid x_i \in C_j, \boldsymbol{\theta}_j) \qquad \text{(a)}$$

Provided that the data are unordered and independent of each other, the likelihood of measuring an entire data base is the product of the probabilities of measuring each datum.

one ?

$$p(\mathbf{x} \mid \boldsymbol{\theta},\boldsymbol{\pi},J) = \prod_{i=1}^{I} p(x_i \mid \boldsymbol{\theta},\boldsymbol{\pi},J) \qquad \text{(b)}$$

For given values of the class parameters, one can calculate the probability that object i belongs to class j by use of Bayes's theorem:

add

$$p(x_i \in C_j \mid x_i,\boldsymbol{\theta},\boldsymbol{\pi},J) = \frac{\boldsymbol{\pi}_j \, p(x_i \mid x_i \in C_j, \boldsymbol{\theta}_j)}{p(x_i \mid \boldsymbol{\theta},\boldsymbol{\pi},J)} \qquad \text{(c)}$$

The problem of identifying a finite mixture is broken into two parts: determining the classification parameters for a given number of classes and determining the number of classes. Rather than seek the class parameter vectors, $\boldsymbol{\theta}$, and the class probabilities, $\boldsymbol{\pi}$, one seeks the full posterior probability distribution of them. The posterior distribution is proportional to the product of the prior distribution of the parameters $p(\boldsymbol{\theta},\boldsymbol{\pi} \mid J)$ and the likelihood function $p(\mathbf{x} \mid \boldsymbol{\theta},\boldsymbol{\pi},J)$.

$$p(\boldsymbol{\theta},\boldsymbol{\pi} \mid \mathbf{x},J) = \frac{p(\boldsymbol{\theta},\boldsymbol{\pi} \mid J)\,p(\mathbf{x} \mid \boldsymbol{\theta},\boldsymbol{\pi},J)}{p(\mathbf{x} \mid J)} \qquad \text{(d)}$$

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

$$p(\mathbf{x} \mid J) = \iint p(\boldsymbol{\theta},\boldsymbol{\pi} \mid J)\,p(\mathbf{x} \mid \boldsymbol{\theta},\boldsymbol{\pi},J)\,d\boldsymbol{\theta}\,d\boldsymbol{\pi} \qquad \text{(e)}$$

To solve the second half of the classification problem (determining the number of classes), one calculates the posterior distribution of the number of classes J. This is proportional to the product of the prior distribution $p(J)$ and the pseudolikelihood function $p(\mathbf{x} \mid J)$.

$$p(J \mid \mathbf{x}) = \frac{p(J)\,p(\mathbf{x} \mid J)}{p(\mathbf{x})} \qquad \text{(f)}$$

In principle, one can determine the most probable number of classes by evaluating $p(J \mid \mathbf{x})$ over the range of J for which the prior $p(J)$ is significant. In practice, the multidimensional integrals of equation (e) are computationally intractable, and one must search for the maximum of the function and approximate it about that point. The search and approximate integration are performed by the AutoClass algorithm. The complete problem involves starting with more classes than are believed 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's and Laird's EM algorithm to find the best class parameters for a given number of classes [the maximum of equation (d)]. To derive the algorithm, one differentiates the posterior distribution with respect to the class parameters and equates the derivative with zero. This yields a system of nonlinear equations that are solved iteratively.

The number of classes is determined as follows. If a class has negligible posterior probability $\boldsymbol{\pi}_j$, 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 are always less probable than are models that 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 and repeats 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.

This work was done by John Stutz of **Ames Research Center**, *Peter Cheeseman of RIACS, Will Taylor and (prename unknown) Hanson of Sterling Software, and Matthew Self of NEXT. For further information, Circle on the TSP Request Card.*
ARC-12797

?

To ARC —
Source author should double check just in case vector and scalar quantities were not mixed. We seem to have it OK
Thanks

Type fonts on eqs b and c need corrections. Also, Σ and Π should be rechecked. Need different fonts.