KNOWLEDGE MEDIA INSTITUTE

An Introduction to the Robust Bayesian Classifier

Marco Ramoni
Knowledge Media Institute
The Open University

Paola Sebastiani
Statistics Department
The Open University

KMi-TR-79

March 9, 1999

An Introduction to the Robust Bayesian Classifier

Marco Ramoni
Knowledge Media Institute
The Open University

Paola Sebastiani
Statistics Department
The Open University

Abstract

Bayesian supervised classifiers are one of the most promising data mining techniques for large scale applications. When the database is complete, they provide an efficient and scalable solution to classification problems. When some data are missing in the training set, methods exist to learn these classifiers, albeit less efficiently, under the assumption that data are missing at random. This paper describes the implementation of RoC, a Bayesian classifier able handle incomplete databases with no assumption about the pattern of missing data.

Keywords: Bayesian learning, supervised classification, missing data.

Reference: KMi Technical Report KMi-TR-79, Knowledge Media Institute, The Open University, Milton Keynes, United Kingdom, March 9, 1999.

Address: Marco Ramoni, Knowledge Media Institute, The Open University, Milton Keynes, United Kingdom MK7 6AA. PHONE: +44 (1908) 655721, FAX: +44 (1908) 653169, EMAIL: `m.ramoni@open.ac.uk`, URL: `http://kmi.open.ac.uk/people/marco`.

Contents

1. Introduction

Classification is a common task performed by knowledge discovery systems and consists of assigning a class label to a set of unclassified cases. Classification typically involves two steps: first the system is trained on a set of data and then it is used to classify a new set of unclassified cases. When the possible classes are known in advance and the system is trained on a set of classified cases, the task is termed *supervised* classification.

Bayesian methods provide one of oldest methods to perform supervised classification. A Bayesian classifier [4, 11] is trained by estimating the conditional probability distributions of each attribute, given the class label, from a database. The classification of a case, represented by a set of values for each attribute, is accomplished by computing the posterior probability of each class label, given the attributes values, by using Bayes' Theorem. The case is then assigned to the class with the highest posterior probability. This classification rule is optimal under a 0-1 loss [3] and assumes that mis-classification costs are equal. Other classification rules that account for different mis-classification costs can be adopted but the underlying approach relies on the ability to compute the conditional probability of each class value.

The simplifying assumptions underpinning the Bayesian classifier are that the classes are mutually exclusive and exhaustive and that the attributes are conditionally independent once the class is known. These assumptions have been responsible for a form of ostracism suffered by the Bayesian classifier that has been dismissed as naive, useful at most as "straw man" to evaluate more sophisticated techniques [3]. Notwithstanding these theoretical limitations, recent empirical evaluations have found the Bayesian classifier surprisingly accurate [9, 5]. Furthermore, the growing interest in large databases, mainly motivated by the efforts to scale up machine learning methods to data mining tasks, has led to a new appreciation of the computational advantages offered by the Bayesian classifier and, together with the new empirical findings about their actual accuracy, has induced a process of rehabilitation. As a matter of facts, the deprecated assumptions make both learning and classification very efficient and therefore feasible to tackle large databases.

Unfortunately the learning efficiency, so precious to their success, is lost when the database is not complete, that is, it records some entry as unknown. In this case, the exact estimate of each conditional probability distribution, required to define a classifier, is a mixture of the estimates that can be computed in each database generated by the combination of all possible values of the missing entry, and the computational cost of this operation grows exponentially in the number of missing data. This limitation could turn out to be even more dangerous to the deployment of a Bayesian classifier in real-world applications than its — almost forgiven — representational limitations, because missing data are a frequent problem of observational databases, like those used in data mining and knowledge discovery tasks.

Simple solutions to handle missing data are either to discard the cases with missing entries or to ascribe the missing entries to an *ad hoc* dummy value. These solutions are extremely vulnerable to bias, because they assume that the missing data are not relevant to the estimation of the probability distributions. More sophisticated techniques involve the use of iterative or stochastic methods, such as the EM algorithm [2] or Gibbs Sampling [6]. Both methods rely on the assumption that data are *Missing at Random* [12], that is, the probability that an entry is missing is independent of the unknown values in the database. This assumption is not verifiable and, when violated, the resulting estimates suffer of a dramatic decrease in accuracy [16]. However, even if the Missing at Random assumption holds, the deficiencies of the two methods make them inappropriate for estimation in large databases. The EM algorithm is notoriously slow and can easily end up in a sub-optimal solution when the number of missing data is large. Gibbs sampling, compared to the EM algorithm, provides measures of the estimation accuracy but the computational cost of this method heavily depends on the absolute number of missing data, thus preventing its scalability to large databases. A partial solution is provided by a deterministic method, called *Bound* and *Collapse* (BC) [14]. BC does not rely, *per se*, on the Missing at Random assumption but allows the user to specify any pattern of missing data, including the Missing at Random assumption, if this is the case. Unfortunately, BC still requires an assumption about the process generating missing data, and this information may not be readily available.

This situation motivated the development of the *robust Bayesian estimator* (RBE) [13], which does not rely on any assumption about the pattern of missing data. The RBE is based on a entirely new view of incomplete data: with no information on the pattern of missing data, an incomplete data set can only constrain the set of estimates that can be induced from its completions. The RBE produces probability estimates that are robust with respect to the pattern of missing data by providing probability intervals that contain the estimates learned from all possible completions of the incomplete databases. Using this theory, we have developed a Robust Bayesian classifier (RBC), which is trained on an incomplete data set using the RBE and performs classification by propagating probability intervals. The particular statistical model underlying the Bayesian classifier allows a closed-form solution of both these operations and the definition of efficient algorithms to perform both of them.

This paper describes RoC, a computer program implementing the robust Bayesian classifier. The next section summarizes the principles of the robust Bayesian classifier and identifies the main design issues of the implementation. Section 3 describes the implementation in some details and Section 4 describes the application of RoC to a relatively large database, to describe behavior and performance of the program.

2. Bayesian Classifiers

A Bayesian classifier is defined by a set C of classes and by a set $\mathcal{A}$ of attributes. We denote a generic class by $c_j \in C$ and a generic attribute by $A_i \in \mathcal{A}$. We assume that the classes

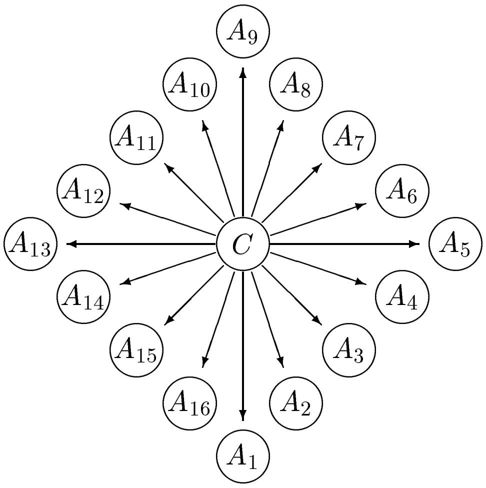

Figure 1: Structure of a Bayesian classifier.

are mutually exclusive and exhaustive, that is, one and only one class label can be true in a given situation. In this way, we can regard the set C of classes as a stochastic variable, taking on one of the values c_i with a probability that represents the unknown "state of the world".

Suppose our task is to classify a set of citizens in two classes: the class of citizens making more than 50,000$ a year and the class of citizens earning no more than 50,000$ a year. The set of classes will be defined, in this case, by two elements $\{$`>50,000`,`<=50,000`$\}$ and the variable C will be in one of these states: c_1 =`>50,000` or c_2 =`<=50,000`. The probability $p(C = c_1)$ represents the chance that a citizen chosen at random earns more than 50,000$. Also attributes can be regarded as stochastic variables and we assume that these variables are discrete. Each attribute A_i, like `gender` or `education`, will therefore take on one of the values in a set $\{a_{ij}\}$. For instance, `gender` will take on one of the two values `male` or `female`, while `education` may take on one of the values `primary`, `high school`, `college`, `postgraduate`. We denote the assignment $A_i = a_{ik}$ and $C = c_j$ respectively by a_{ik} and c_j.

The assumption that attributes are conditional independent given the class can be graphically represented as a Bayesian Belief Network [8]. Figure 1 shows a graphical representation of a Bayesian classifier with 16 attributes. Each arrow represents a conditional dependency between the classes and each attribute. Every (undirected) path between two attributes passes through the class C and this graphical structure encodes the assumption that each attribute is conditionally independence of the others, given the class. Intuitively, this assumption means that, once we know the class label, knowledge of other attributes values are irrelevant to A_i.

Suppose we are now given a database $\mathcal{D}$. Each case in the database reports a set of attributes values (also called instances), together with the class label of the case. The training of a Bayesian classifier consists of the estimation of the conditional probability distribution of each attribute, given the class, from the database $\mathcal{D}$.

Let $n(a_{ik}|c_j)$ be the number of cases in which the attribute A_i appears with value a_{ik} and the class is c_j. The frequentist estimate of the conditional probability $p(a_ik|c_j)$ is the relative frequency of relevant cases in the database

$$p(a_{ik}|c_j) = \frac{n(a_{ik}|c_j)}{\sum\limits_k n(a_{ik}|c_j)}$$

Similarly, we can estimate the probability $p(c_j)$ as

$$p(c_j) = \frac{n(c_j)}{n}$$

where $n(c_j)$ is the frequency of cases in the database with class c_j, and $n = \sum_h n(c_h)$ is the size of the database. These estimates, however, are only a function of the data while, in some situations, we may also want to take into account external information, such as an expert's opinion, in the estimation process. The Bayesian approach encodes this external information as a set of *prior probability distribution* for each attribute, given the class, and the class. These distributions, in turn, can be regarded as the relative frequency of *imaginary cases* upon which this prior assessment has been formulated. If we denote by α_{jk} the number of imaginary cases encoding our prior probability $p(a_{ik}|c_j)$, we can write the Bayesian estimate of $p(a_{ik}|c_j)$ as

$$p(a_{ik}|c_j) = \frac{\alpha_{jk} + n(a_{ik}|c_j)}{\alpha_j + n(c_j)} \tag{1}$$

where $\alpha_j = \sum_k \alpha_{jk}$ is the sample size of the imaginary cases with class c_j and encodes, in some way, the confidence in our prior probability: the higher the number of imaginary cases, the higher the confidence in our prior probability. Because of this, α_j is called *local prior precision* [7]. Similarly, the Bayesian estimate of $p(c_j)$ is given by

$$p(c_j) = \frac{\alpha_j + n(c_j)}{\alpha + n}$$

where $\alpha = \sum_j \alpha_j$ is the prior *global precision* and $n = \sum_j n(c_j)$ is the number of instances in the data set. Lack of prior knowledge about the problem at hand is usually represented by letting $\alpha_j = \alpha/c$, where c is the number of classes, and $\alpha_{jk} = \alpha_j/a_i$, where a_i is the number of values of the attribute A_i. In this way, we have that, *a priori*, $p(c_j) = 1/c$ and $p(a_{ik}|c_j) = 1/a_i$. See [15] for further reference.

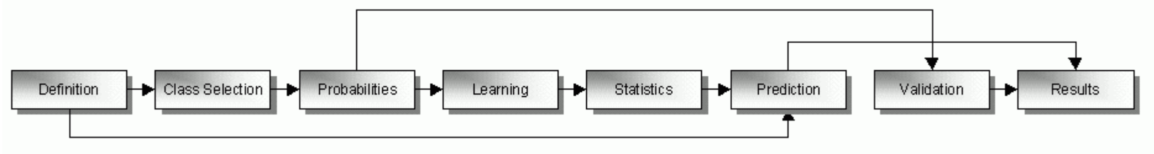

Figure 2: The flow of tasks performed by RoC.

Once the classifier has been trained, we can use it for the classification of new cases. A standard application of Bayes' theorem let us calculate the probability of the class value $C = c_j$ given the set of attribute values $e_k = \{A_1 = a_{1k}, \ldots, A_m = a_{mk}\}$ as

$$p(c_j|e_k) = \frac{\prod_{k=1}^{m} p(a_{ik}|c_j)p(c_j)}{\sum_{h=1}^{c} \prod_{k=1}^{m} p(a_{ik}|c_h)p(c_h)} \tag{2}$$

and the case is assigned to the class with the largest posterior probability. An advantage of the Bayesian classifier is that the assumption of conditional independence of the attributes given the class allows us to iteratively compute $p(c_j|e_k)$. We begin by computing

$$p(c_j|a_{1k}) = \frac{p(a_{1k}|c_j)p(c_j)}{\sum_{h=1}^{c} p(a_{1k}|c_h)p(c_h)} \tag{3}$$

and this value is used in the next iteration by letting

$$p(c_j|a_{1k}, a_{2k}) = \frac{p(a_{2k}|c_j)p(c_j|a_{1k})}{\sum_{h=1}^{c} p(a_{2k}|c_h)p(c_h|a_{1k})} \tag{4}$$

and so on. The iterative application of this formula to all entries in each case will compute the posterior probability distribution of the classes.

All these properties hold until the database is complete. When the database reports some data as unknown, the computational efficiency of the Bayesian classifier is lost. In this case, the exact estimate of each conditional probability distribution must be computed as the mixture of the estimates for each database generated by the combination of all possible values for each missing entry, and the computational cost of this operation grows exponentially in the number of missing data. Current methods to cope with this problem use approximate solutions based on the assumption that data are Missing at Random or, at least, that the pattern of missing data is known. Both assumptions can be unreasonable in data mining applications as databases are often collected over time and little information may be available about the collection procedure. Since an incorrect assumption about the pattern of missing data can introduce a dramatic bias in the estimates, we would like to be able to train a classifier without making any assumption about the mechanism generating

Figure 3: Definition screen.

missing data. To this aim, we have devised a new robust Bayesian estimator [13] that does not assume any pattern of missing data. The next section describes the integration of this estimator in a complete program for supervised classification, called RoC.

3. A Robust Classifier

The aim of RoC is to extend the capabilities of the Bayesian classifier described in the previous section to situations in which the database reports some entries as unknown. Thus, RoC can train a Bayesian classifier from an incomplete database, and uses the classifier for classification of new cases. To make RoC feasible for data mining applications, both training and classification steps maintain the computational efficiency of the standard Bayesian classifier. RoC interface is implemented as a set of sequential steps (called Wizard interface), and we will follow these steps to describe the behaviour of the program. Figure 2 shows the flow of the program steps.

3.1 Definition

RoC can either define a Bayesian classifier from a database or load a previously defined classifier. In the former case, RoC needs first to define the components of the classifier, that is identify the class variable and define the classes as well as the attributes values. This step can be done automatically as RoC can define classes and attributes from a database of cases.

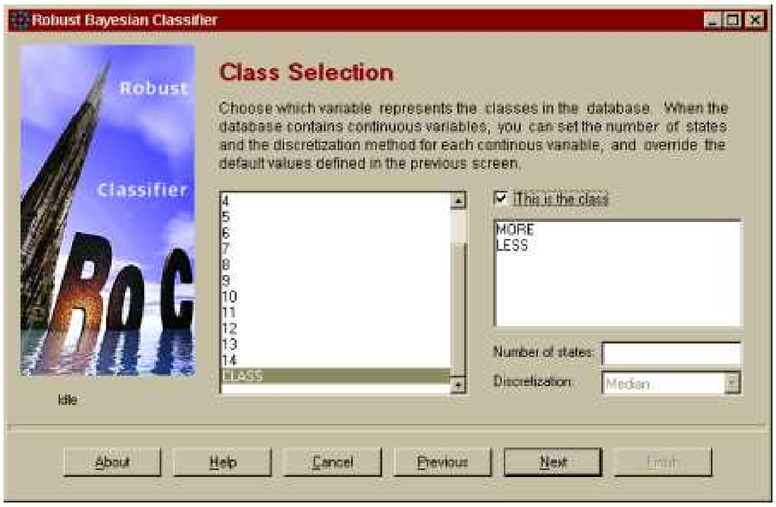

Figure 4: Class selection and discretization screen.

The format of the database has cases as rows and variables — either attributes or class — as columns. The first row contains the names of the variables that are the names used then by RoC to call the variables. RoC will then scan the cases to identify the possible values of all variables in the database. The user is asked to set the maximum number of states for a numeric variable to be considered an ordinal variable, that is, a variable with discrete numeric states. When a variables passes this threshold, RoC will detect this variable to be continuous.

RoC handles discrete variables only, so variables detected as continuous are discretized. The user is allowed to define the number of discretized states for continuous variables and it is allowed to choose between two unsupervised discretization methods. For each continuous variable, the first method takes the minimum and the maximum value detected in the database and divide the range between minimum and maximum in a set of equally spaced intervals. The second discretization method splits the range between minimum and maximum into intervals containing the same number of cases.

3.2 Class selection and Discretization

The second step of the process consists of choosing the variable in the database representing the class. For each continuous attribute, the user can override both the default number of discretized states and the discretization method set in the previous screen.

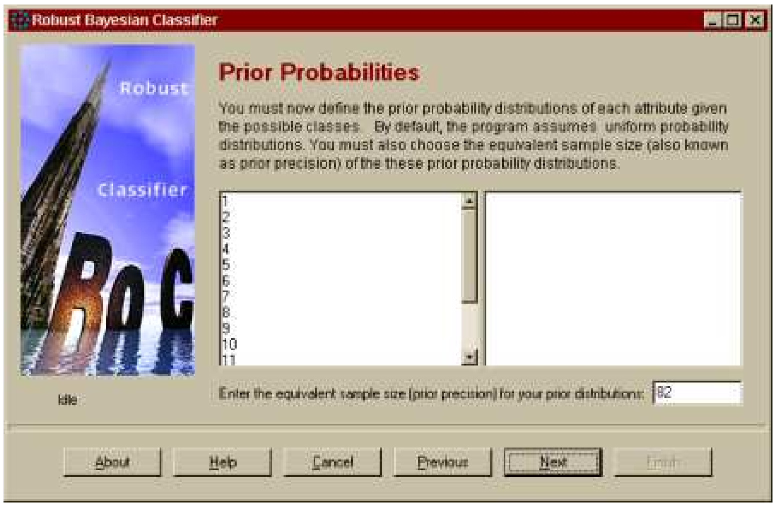

Figure 5: Prior probability distribution assessment screen

3.3 Prior Probability Distributions

The core of a Bayesian approach to probability estimation is the possibility of setting prior probability distributions representing information about the class distribution and the dependence of each attribute on the class. In this screen, the user can set the conditional probability distribution $p(a_{ik}|c_j)$ of each attribute given the class, and the prior probability $p(c_j)$ of each class. By default, RoC assumes these distributions to be uniform.

The user has also to set the *prior precision* α, described in the previous section. For each attribute value a_{ik}, the parameters α_{jk} used in the estimation process will be computed as $p(a_{ik}|c_j) \times p(c_j) \times \alpha$, where $p(a_{ik}|c_j)$ and $p(c_j)$ represent the prior conditional distributions of the attribute A_i given the class and the probability distribution over the class C, respectively. The default value suggested by RoC is computed as $|C| \times \max_i A_i$, so that each α_{jk} in the classifier is at least 1. This value can be overridden.

3.4 Learning

The system is now ready to start the training session that consists of learning the conditional probabilities $\{p(a_{ik}|c_j)\}$ and $\{p(c_j)\}$ from a data set $\mathcal{D}$ in which some attributes and class values are unknown. RoC does not make any assumption about the mechanism underlying missing data and it therefore uses the Robust Bayesian Estimator (RBE) introduced by [13]. For each conditional probability $p(a_{ik}|c_j)$ and each probability $p(c_j)$, the RBE returns intervals containing all the *consistent estimates* of $p(a_{ik}|c_j)$ and $p(c_j)$. A consistent estimate

Figure 6: Learning screen

is an estimate that can be computed in a *consistent completion* of $\mathcal{D}$ which, in turns, is any complete database $\mathcal{D}_c$ from which we can obtain $\mathcal{D}$ using some deletion process. The set of consistent completions $\{\mathcal{D}_c\}$ is given by all data sets in which the unknown entries are replaced with one of the possible values of the unobserved variables. This section describes the estimation procedure implemented in RoC to compute these probability intervals from an incomplete data set.

The first step of the analysis is the definition of *virtual frequencies* that are used to find the extreme points of the probability intervals. The unknown entries in the database give rise to three types of incomplete cases that are relevant to the estimation of $p(a_{ik}|c_j)$:

- The attribute A_i takes value a_{ik} and the value of C is not observed, so that it can take any of the c_j value.
- C takes value c_j, but the value of A_i is missing, and can be any of the a_{ik} values.
- Both values of A_i and C are unknown.

We denote the frequency of these cases, respectively, by $n(a_{ik}|?)$, $n(?|c_j)$ and $n(?|?)$. We use these frequencies of incomplete cases to define the virtual frequencies $n_{max}(a_{ik}|c_j)$ and $n_{min}(a_{ik}|c_j)$. The quantity $n_{max}(a_{ik}|c_j)$ is the maximum number of incomplete cases (A_i, C) that can be completed as (a_{ik}, c_j) and is given by

$$n_{max}(a_{ik}|c_j) = n(?|c_j) + n(a_{ik}|?) + n(?|?) \tag{5}$$

On the other hand, the virtual frequency $n_{min}(a_{ik}|c_j)$ is the maximum number of incomplete cases (A_i, C) that can be ascribed to c_j without increasing the frequency $n(a_{ik}|c_j)$ and is

$$n_{min}(a_{ik}|c_j) = n(?|c_j) + \sum_{h \neq k} n(a_{ih}|?) + n(?|?) \tag{6}$$

These virtual frequencies are used by the RBE to compute bounds on the set of consistent estimates of $p(a_{ik}|c_j)$. The RBE returns the probability interval

$$[p_{min}(a_{ik}|c_j)\ p_{max}(a_{ik}|c_j)]$$

containing the set of consistent estimates of $p(a_{ik}|c_j)$. The values $p_{min}(a_{ik}|c_j)$ and $p_{max}(a_{ik}|c_j)$ are, respectively, the minimum and the maximum estimate of $p(a_{ik}|c_j)$ that can be found in the consistent completions of $\mathcal{D}$ and are respectively

$$p_{min}(a_{ik}|c_j) = \frac{\alpha_{jk} + n(a_{ik}|c_j)}{\alpha_j + n(c_j) + n_{min}(a_{ik}|c_j)} \tag{7}$$

$$p_{max}(a_{ik}|c_j) = \frac{\alpha_{jk} + n(a_{ik}|c_j) + n_{max}(a_{ik}|c_j)}{\alpha_j + n(c_j) + n_{max}(a_{ik}|c_j)} \tag{8}$$

where $n(a_{ik}|c_j)$ is the frequency of instances with $A_i = a_{ik}$ and $C = c_j$ and $n(c_j) = \sum_k n(a_{ik}|c_j)$. The proof is in [13]. We can similarly use the RBE to estimate the marginal distribution of C. We note that the virtual frequencies $n_{min}(c_j)$ and $n_{max}(c_j)$ are both equal to the number of instances in which the class is not observed, say $n(?)$. Hence, we obtain

$$p_{min}(c_j) = \frac{\alpha_j + n(c_j)}{\alpha + n + n(?)} \tag{9}$$

$$p_{max}(c_j) = \frac{\alpha_j + n(c_j) + n(?)}{\alpha + n + n(?)} \tag{10}$$

where $n = \sum_j n(c_j)$. When the data set is complete, then Equations 7 and 8 reduce to Equations 1, and similarly Equations 9 and 10 return the estimate of $p(c_j)$.

Note that the use of the RBE to estimate probability intervals maintains the *learning locality* of the standard Bayesian classifier trained on complete data, as each conditional probability distribution is estimated independently of the others. This locality is usually lost in methods like Gibbs sampling to estimate conditional probabilities from incomplete data.

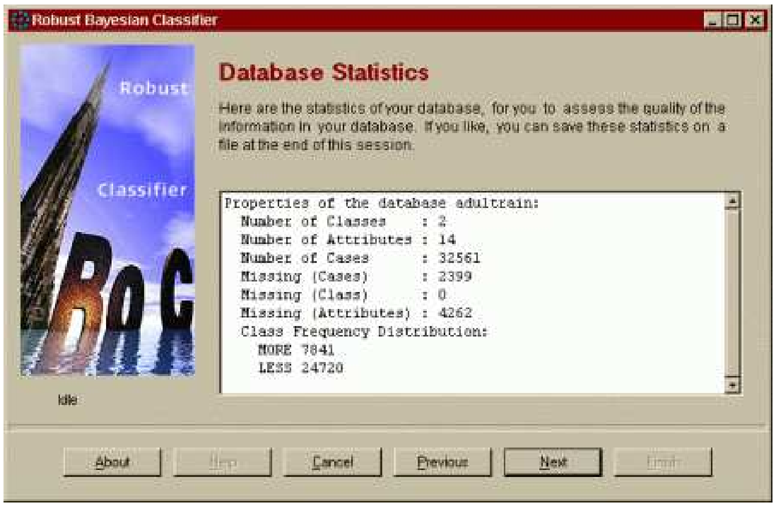

Figure 7: Database statistics screen.

3.5 Database Statistics

When the learning process is complete, RoC returns some information about the database in terms of summary statistics, as shown in Figure 7. Namely, the number of cases, classes and attributes in the database; the number of incomplete class observations, incomplete cases and missing entries; and the distribution of the classes.

3.6 Predictions

Once the classifier has been trained, it can be used to predict the class of new cases. Unfortunately, we cannot directly apply Equation 2 to calculate the posterior probability of each class given a set of attributes values to select the class with highest posterior probability, because we are now handling probability intervals. We must devise two new methods i) to compute the posterior probability distribution of the class and ii) to select the class label. RoC implements an iterative procedure to compute posterior probability intervals of each class given the set of attributes values, and two classification rules to assign the class label on the basis of probability intervals.

Posterior Probability Intervals Let $e = \{a_{1k}, ..., a_{a_i k}\}$ be attributes values of a new case that we wish to assign to one class. We compute probability intervals for $c_j|e$, iteratively, as follows. The initialization step sets, for all $j = 1, ..., c$

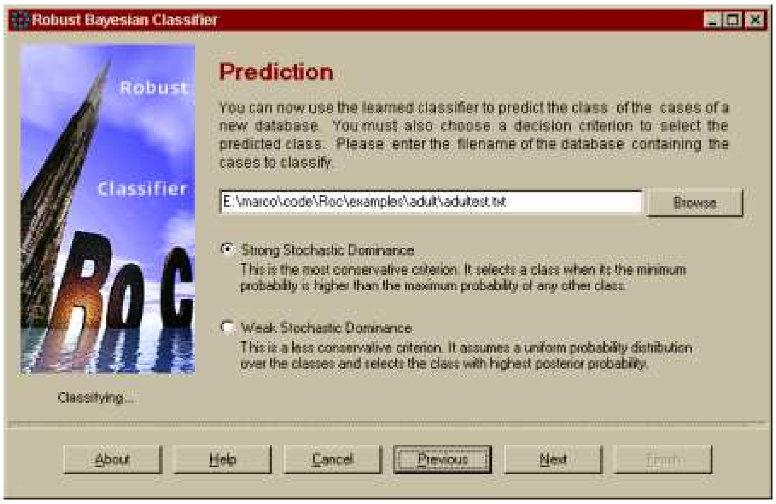

Figure 8: Class prediction screen.

$$p_{max}(c_j|a_{1k}) = \frac{p_{max}(a_{1k}|c_j)p_{max}(c_j)}{p_{max}(a_{1k}|c_j)p_{max}(c_j) + \sum_{l\neq j} p_{min}(a_{1k}|c_l)p_{min}(c_l)}$$

and

$$p_{min}(c_j|a_{1k}) = \frac{p_{min}(a_{1k}|c_j)p_{min}(c_j)}{p_{min}(a_{1k}|c_j)p_{min}(c_j) + p_{max}(\sum_{l\neq j} p(a_{1k}|c_l)p(c_l))}$$

where $p_{max}(\sum_{l\neq j} p(a_{1k}|c_l)p(c_l))$ is the maximum value in the set of $c-1$ quantities

$$\{p_{max}(a_{1k}|c_g)p_{max}(c_g) + \sum_{l\neq j,g}^{c} p_{max}(a_{1k}|c_l)p_{min}(c_l)\}$$

for $g \neq j$. The first iteration step ends by replacing the original interval distribution of C, $[p_{min}(c_j)\ p_{max}(c_j)]$, by $[p_{min}(c_j|a_{1k})\ p_{max}(c_j|a_{1k})]$. The next iteration step repeats the same procedure to compute $[p_{min}(c_j|a_{1k}, a_{2k})\ p_{max}(c_j|a_{1k}, a_{2k})]$ by replacing $p_{max}(a_{1k}|c_j)$ and $p_{min}(x_{1k}|c_j)$ with $p_{max}(a_{2k}|c_j)$ and $p_{min}(x_{2k}|c_j)$ and using the updated distribution of $C|a_{1k}$. By repeating the procedure for all a_{ik}, we find the bounds $[p_{min}(c_j|e)\ p_{max}(c_j|e)]$ on the distribution of $C|e$. Ramoni and Sebastiani [13] show that the probability intervals for $C|e$ so found are sound, as they contain all possible conditional probabilities $p(c_j|e)$ that we would compute by training the classifier on the set of consistent completions of $\mathcal{D}$. When the data set is complete, the RBE intervals reduce to the point estimates given in Equation

2, and the procedure described in this section to compute conditional probability intervals is equivalent to the procedure used by the standard Bayesian classifier. The procedure maintains the same complexity of the standard Bayesian classification, as each interval propagation — to compute the interval distribution of the class variable given a set of attribute values — needs to make $c(c-1)$ evaluations of linear forms.

Class Assignment Once we have computed the set of posterior probability intervals for $c_j|e$, we are left with the problem of selecting the class label. The standard criterion of selecting the class label with the highest posterior probability cannot be applied as is, because our posterior probabilities are intervals rather than single values. The user is given here the possibility of selecting one of two different criteria: the *Stochastic dominance* criterion and the *Weak dominance* criterion.

The *Stochastic dominance* criterion [10] assigns the new case to the class c_j if the minimum probability $p_{min}(c_j|e)$ is larger than the maximum probability $p_{max}(c_h|e)$, for any $h \neq j$. Stochastic dominance is the safest and most conservative criterion since the classification is independent of the distribution of missing data. When the probability intervals are overlapping, the Stochastic dominance criterion is not applicable and we face a situation of undecidability.

We have therefore introduced a weaker version of the Stochastic dominance criterion, called *Weak dominance*, that works as follows. Each probability interval $[p_{min}(c_j|e)\ p_{max}(c_j|e)]$ is summarized into a point, representing a *score* assigned to $p(c_j|e)$. We then rank intervals according to their scores and assign the new case to the class with the largest score. Since we do not want to commit ourselves with a particular model for the missing-data mechanism, we can assume that all mechanisms are equally likely, so that the probability $q(c_j)$ that an unknown value of C must be assigned to c_j is $1/c$ where c is the number of classes. We then define the *robust classification score* of $c_j|e$ as

$$s_u(c_j|e) = \frac{p_{min}(c_j|e)(c-1)}{c} + \frac{p_{max}(c_j|e)}{c}$$

and assign the new case to the class having the largest score. The score is in the interval $[p_{min}(c_j|e)\ p_{max}(c_j|e)]$.

3.7 Cross Validation

RoC gives the user the opportunity to evaluate the classifier trained on a data set using cross validation. Cross validation is the standard procedure adopted to evaluate classifiers, although its validity is now under serious scrutiny. Using cross validation, the user can evaluate the classifier on the same database by splitting it into n parts of roughly the same size and by classifying each part after learning on the remaining $n-1$ parts. The process can be iterated for several times and the average number of correct answers will be taken as a measure of the accuracy of the classifier. As the classification process with missing

Figure 9: Final screen.

data can leave some cases unclassified, RoC returns also a measure of the *coverage* of the classifier, that is, the number of cases for which a class was assigned.

The performance metrics returned are the same as the ones computed for a class prediction session with reported classes, as show in Figure 8.

3.8 Congratulations!

The last screen of a session gives the user the opportunity to save the results of the session (database statistics and either prediction results or cross validation results) on a log file.

3.9 Help

At any time, the user can hit the Help button an obtain a on-screen description of the actions and options available in the current screen.

4. An Example

This section describes the application of RoC to one of the largest publicly available incomplete databases. The database is known under the name of *Adult* database, is described in [9], and is available from the UCI Machine Learning repository [1]. The database reports the values of 14 (8 continuous and 6 nominal) attributes over 48841 cases from the 1994 US Census. The database cases are classified into two classes: those who earn more than

50,000$ a year and those who don't. About the 7% of the cases is incomplete. The database is distributed in two parts, a training set of 32561 cases and a test set of 16281 cases. The task is to train the classifier on the first database and classify the cases of the second one.

We are interested in two performance measures: classification *accuracy* and *coverage*. The classification accuracy measures the predictive quality of the classifier and it is given by the percentage of cases in which the class label predicted by the classifier is equal to the class label observed in the test set. The coverage is the proportion of cases that the classifier is able to assign to a class and it is computed as the ratio between the number of classified cases and the total number of cases in the database.

Current analysis of Bayesian classifiers remove incomplete cases from the database [9, 3, 5]. If the incomplete cases are removed, the coverage of the resulting classifier is simply the proportion of complete cases in the database and amounts to 93%. We discretized the continuous attributes in ten equally spaced intervals and we run a Bayesian classifier on the database with the incomplete casesremoved, both in the training and in the test set. The resulting accuracy was 81.74%[1].

We then trained RoC on the whole training set and we used the resulting classifier to predict the classes of the whole test set, using first the Stochastic dominance criterion and then the Weak Stochastic dominance criterion. Under the first criterion, RoC achieved an accuracy of 86.51%. This increase in accuracy was paid by a decrease of coverage to 87%. The use of the Weak Stochastic dominance criterion brings the coverage to 100% and the accuracy to 82.5%, still slightly better than the classifier with the cases with missing entries removed.

We can combine accuracy and coverage in a single performance measure. We define such a combined measure as the difference between the number n_s of cases correctly classified under the Stochastic dominance criterion — or the number n_w of cases correctly classified under the the Weak dominance — and the number n_c of cases correctly classified by the Bayesian classifier using the reduced database. This measure may be regarded as a *relative gain in accuracy*. To make relative this measure, the differences $n_s - n_c$ and $n_w - n_c$ are divided by the number of instances n in each data set. Under the Stochastic dominance criterion, this combined measure amounts to -0.002. This means that even the remarkable increase in accuracy of RoC fail to compensate for the loss in coverage by a 0.2%. Needless to say, this measure can be meaningless in applications in which the accuracy plays a critical role: in these cases, a loss of coverage means that a quota of the data will have to be carefully reconsidered but we do not not have to pay the cost of classification errors. Under the Weak Stochastic dominance criterion, this combined measure amounts to 0.7. This value means that RoC achieves a relative gain of accuracy of 7%. This gain may appear small until we

[1] The note accompanying the *Adult* database in the distribution reports a 83.88% accuracy for the classifier, probably due to a more sophisticated discretization method used by the author. Note that this increase of accuracy does not change significantly the findings of our analysis.

realize that it is achieved on a database with only a 7% of incomplete cases.

We must also note that the training phases took 29.3 seconds on a 300Mhz PC with 128MB RAM, showing that the computational cost of any increase in accuracy is still reasonable and does not prevent the scalability of the Bayesian classifier.

5. Conclusions

This paper presents RoC, a Bayesian classifier able to handle incomplete databases with no assumptions about the pattern of missing data. A simple application illustrates some situations in which this ability can be useful to achieve higher accuracy with an associated computational cost that does not impair the scalability of the technology to large databases. RoC is available as a research prototype from the web site of the Bayesian Knowledge Discovery project of The Open University at http://kmi.open.ac.uk/projects/bkd.

Acknowledgments

This research was supported, in part, by the ESPRIT programme of the Commission of the European Communities under contract EP29105 and by equipment grants from Apple Computer and Sun Microsystems.

References

[1] C. Blake, E. Keogh, and C.J. Merz. *UCI Repository of machine learning databases.* University of California, Irvine, Dept. of Information and Computer Sciences, 1998.

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

[3] P. Domingos and M. Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130, 1997.

[4] R. O. Duda and P.E Hart. *Pattern classification and scene analysis.* Wiley, New York, NY, 1973.

[5] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 1997.

[6] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[7] I.J. Good. *The Estimation of Probability: An Essay on Modern Bayesian Methods.* MIT Press, Cambridge, MA, 1968.

[8] D. Heckerman. Bayesian networks for data mining. *Data Mining and Knowledge Discovery*, 1:79–119, 1997.

[9] Ron Kohavi. Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, San Mateo CA, 1996. Morgan Kaufman.

[10] H.E. Kyburg. Rational belief. *Behavioral and Brain Sciences*, 6:231–273, 1983.

[11] P. Langley, W. Iba, and K. Thompson. An analysis of Bayesian classifiers. In *Proceedings of the National Conference on Artificial Intelligence*, pages 223–228, San Mateo, CA, 1992. Morgan Kaufman.

[12] R.J.A. Little and D.B. Rubin. *Statistical Analysis with Missing Data.* Wiley, New York, 1987.

[13] M. Ramoni and P. Sebastiani. Robust learning with missing data. Technical Report KMi-TR-28, Knowledge Media Institute, The Open University, 1996. Available at http://kmi.open.ac.uk/techreports/KMi-TR-28.

[14] M. Ramoni and P. Sebastiani. Parameter estimation in Bayesian networks from incomplete databases. *Intelligent Data Analysis Journal*, 2, 1998.

[15] M. Ramoni and P. Sebastiani. Bayesian methods for intelligent data analysis. In M. Berthold and D.J. Hand, editors, *An Introduction to Intelligent Data Analysis*, New York, 1999. Springer.

[16] D.J. Spiegelhalter and R.G. Cowell. Learning in probabilistic expert systems. In *Bayesian Statistics 4*, pages 447–466, Oxford, 1992. Clarendon Press.