Lecture Notes in Artificial Intelligence 1952

Subseries of Lecture Notes in Computer Science
Edited by J. G. Carbonell and J. Siekmann

Lecture Notes in Computer Science
Edited by G. Goos, J. Hartmanis and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milan
Paris
Singapore
Tokyo

Maria Carolina Monard
Jaime Simão Sichman (Eds.)

Advances in Artificial Intelligence

International Joint Conference
7th Ibero-American Conference on AI
15th Brazilian Symposium on AI
IBERAMIA-SBIA 2000
Atibaia, SP, Brazil, November 19-22, 2000
Proceedings

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saabrücken, Germany

Volume Editors

Maria Carolina Monard
University of São Paulo
Department of Computer Science and Statistics
Computational Intelligence Laboratory
Avenida Trabalhador Sãocarlense 400, 13566-590 São Carlos, Brazil
E-mail: mcmonard@icmc.sc.usp.br

Jaime Simão Sichman
University of São Paulo
Computer Engineering Department
Intelligent Techniques Laboratory
Av. Prof. Luciano Gualberto, 158, tv. 3, 05508-900 São Paulo, Brazil
E-mail: jaime@pcs.usp.br

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Advances in artificial intelligence : international joint conference ; proceedings /
7th Ibero-American Conference on AI ; 15th Brazilian Symposium on AI,
IBERAMIA-SBIA 2000, Atibaia, SP, Brazil, November 19 - 22, 2000.
Maria Carolina Monard ; Jaime Simão Sichman (ed.). - Berlin ; Heidelberg ;
New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Singapore ; Tokyo :
Springer, 2000
(Lecture notes in computer science ; Vol. 1952 : Lecture notes in
artificial intelligence)
ISBN 3-540-41276-X

CR Subject Classification (1998): I.2

ISBN 3-540-41276-X Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Stefan Sossna
Printed on acid-free paper SPIN: 10780961 06/3142 5 4 3 2 1 0

Preface

This year, Brazil celebrates its 500 years of discovery. To mark this great event, the Brazilian Artificial Intelligence (AI) community organized a special international joint conference putting together SBIA 2000 (the Brazilian AI Symposium) and IBERAMIA 2000 (the Ibero-American AI Conference).

SBIA 2000 is the 15th conference of the SBIA conference series, which is the leading conference in Brazil for presentation of research and applications in Artificial Intelligence. Since 1995, SBIA has become an international conference, with papers written in English, an international program committee, and proceedings published in Springer-Verlag's *Lecture Notes in Artificial Intelligence* (LNAI) series.

IBERAMIA 2000 is the 7th conference of the IBERAMIA conference series, which has been one of the most suitable forums for ibero-american AI researchers (from South and Central America, Mexico, Spain, and Portugal) to present their results. Following the SBIA and EPIA (Portuguese conference on AI) experiences, from IBERAMIA'98 on, it has also become an international conference, with proceedings published in Springer-Verlag's LNAI series.

The IBERAMIA–SBIA 2000 was held in Atibaia, São Paulo, Brazil, 19–22 November 2000. The call for papers was very sucessful resulting in 156 submissions for the paper track and 69 for the open discussion track, from 18 different countries. In order to mantain the good standard of the conference, each submission was sent to at least three members of the program committee. A total of 48 papers were accepted for paper track presentation and are included in this volume. In addition, 36 papers were accepted for the open discussion track and were published in a local procceedings.

The IBERAMIA–SBIA 2000 program was completed with the presentation of invited talks, some introductory and advanced tutorials, several workshops covering specific topics, and the 2nd Ibero-American Thesis and Dissertation Contest.

We would like to thank all researchers for submitting their papers, and the PC members and additional referees for the work they have done. We are also very grateful to our colleagues who provided invaluable organizational support.

November 2000

Maria Carolina Monard
Jaime Simão Sichman

Organization

IBERAMIA–SBIA 2000 was organized by several AI research groups that belong to the University of São Paulo (USP):

Laboratório de Técnicas Inteligentes (LTI)
Departamento de Engenharia de Computação e Sistemas Digitais (PCS)
Escola Politécnica (EP)

Laboratório de Inteligência Computacional (LABIC)
Departamento de Ciências de Computação e Estatística (SCE)
Instituto de Ciências Matemáticas e de Computação (ICMC)

Laboratório de Lógica, Inteligência Artificial e Métodos Formais (LIAMF)
Departamento de Ciência da Computação (MAC)
Instituto de Matemática e Estatística (IME)

Organizing Committee

General Chair:	Jaime Simão Sichman (LTI/PCS/EP)
Scientific Chair:	Maria Carolina Monard (LABIC/SCE/ICMC)
Publicity Chair:	Fabio Nauras Akhras (LTI/PCS/EP)
Publication Chair:	Maria Carolina Monard (LABIC/SCE/ICMC)
Workshop Chair:	Leliane Nunes de Barros (LIAMF/MAC/IME)
Tutorial Chair:	Anna Helena Reali Costa (LTI/PCS/EP)
Finance Chair:	Edson Satoshi Gomi (LTI/PCS/EP)
Local Arrangements Chair:	Marco Túlio C. de Andrade (LTI/PCS/EP)

Supporting Scientific Societies

SBC	Sociedade Brasileira de Computação
AAIA	Associación Argentina de Inteligencia Artificial
AEPIA	Associación Española para Inteligencia Artificial
APPIA	Associação Portuguesa para Inteligência Artificial
AVINTA	Associación Venezuelana de Inteligencia Artificial
SCCC	Sociedad Chilena de Ciencia de la Computación
SMCC	Sociedad de Matemática y Computación de Cuba
SMIA	Sociedad Mexicana de Inteligencia Artificial

Program Committee

Adolfo Guzmán-Arenas	Instituto Politécnico Nacional (Mexico)
Agnar Aamodt	Norwegian University of Science and Technology (Norway)
Alex Alves de Freitas	Pontifícia Universidade Católica do Paraná (Brazil)
Alexis Drogoul	Université de Paris 6 (France)
Alvaro del Val	Universidad Autonoma de Madrid (Spain)
Ana Paiva	Universidade Técnica de Lisboa (Portugal)
Analía Amandi	Universidad Nacional del Centro (Argentina)
Andre Valente	FasTV Corporation (USA)
Antonio Sanchez	Universidad de las Americas (Mexico)
Bert Bredeweg	University of Amsterdam (The Netherlands)
Christian Lemaitre	Laboratorio Nacional de Informatica Avanzada (Mexico)
Cristiano Castelfranchi	Universitá di Siena (Italy)
Díbio Leandro Borges	Universidade Federal de Goiás (Brazil)
Donia Scott	University of Brighton (United Kingdom)
Eduardo Morales	ITESM (Mexico)
Ernesto Costa	Universidade de Coimbra (Portugal)
Eugênio Costa Oliveira	Universidade do Porto (Portugal)
Evandro de Barros Costa	Universidade Federal de Alagoas (Brazil)
Francisco J. Garijo	Telefónica (Spain)
Gabriel Pereira Lopes	Universidade Nova de Lisboa (Portugal)
Gabriela Henning	Universidad Nacional del Litoral (Argentina)
Geber Ramalho	Universidade Federal de Pernambuco (Brazil)
Hector Geffner	Universidad Simón Bolivar (Venezuela)
Helder Coelho	Universidade de Lisboa (Portugal)
Isabelle Bloch	Ecole Nationale Sup. des Telecommunications (France)
Jacques Wainer	Universidade de Campinas (Brazil)
Javier Pinto	Pontifícia Universidad Católica de Chile (Chile)
John Self	University of Leeds (United Kingdom)
Jose Ramirez	Universidad Simón Bolivar (Venezuela)
Leopoldo Bertossi	Catholic University of Chile (Chile)
Liz Sonenberg	University of Melbourne (Australia)
Lluis Godo Lacasa	Instituto de Investigación en Inteligencia Artificial (Spain)
Luciano Garcia	Universidad de La Habana (Cuba)
Luis Enrique Sucar	ITESM (Mexico)

Maria Carolina Monard	Universidade de São Paulo em São Carlos (Brazil) (*Chair*)
Martin Molina Gonzalez	Universidad Politécnica de Madrid (Spain)
Michael Huhns	University of South Carolina (USA)
Miguel Toro	Universidad de Sevilla (Spain)
Nada Lavrac	Jozef Stefan Institute (Slovenia)
Nitin Indurkhya	Nanyang Technological University (Singapore)
Olivier Boissier	Ecole Nationale Sup. des Mines de Saint-Etienne (France)
Osvaldo Cairo Battistutti	ITAM (Mexico)
Pavel Brazdil	Universidade do Porto (Portugal)
Peter Flach	University of Bristol (United Kingdom)
Ramon Lopes de Mantaras	Instituto de Investigación en Inteligencia Artificial (Spain)
Robert Fisher	University of Edinburgh (United Kingdom)
Rosaria Conte	National Research Council (Italy)
Sheila Regina Murgel	Universidade Federal do Rio de Janeiro (Brazil)
Stan Matwin	University of Ottawa (Canada)
Uma Garimella	SQL Star International (India)
Vera Lucia Strube de Lima	Pontifícia Univ. Católica do Rio Grande do Sul (Brazil)
Vicente Botti	Universidad Politécnica de Valencia (Spain)

Referees

Adam Kilgarriff
Agenor Martins
Alejandro Clausse
Aleks Jakulin
Aleksander Sadikov
Andre Carvalho
Angelo Susi
Anna Helena Reali Costa
Arturo Hernandez-Aguirre
Bogdan Filipic
Bráulio Coelho Ávila
Carlos Soares
Celina de Figueiredo
Celso A. A. Kaestner
Claire Cardie Claire Kennedy
Cristina Riff
Daniela Godoy
David Tugwell
Domingo Gallardo
Edson Emílio Scalabrin
Edward Ross
Emil Petriu
Eugénio Oliveira
Federico Barber
Fernando Moura Pires
Flavio Oliveira
Flávio Soares Corrêa da Silva
Francesc Esteva
Francico Blanes
Francisco B. Pereira
Francisco de Assis Tenório de Carvalho
Frederic Grimaud
Gerardo Acosta
Gonzalo Escalada-Imaz
Graça Gaspar
Gustavo E. A. P. A. Batista

Heloisa Camargo
Hilan Bensusan
Igor Zelic
Irene Loiseau
Irene Pimenta Rodrigues
Jacques Robin
Jean-Francois Delannoy
Joao Gama
Joaquim Costa
John Gallagher
José C. Riquelme
José Pacheco de Almeida Prado
Jose Simo
Julio Cesar Nievola
Katya Rodriguez-Vazquez
Louis Hugues
Luis Antunes
Luis Moniz
Luis Torgo
Lynne Cahill
Marcelo Campo
Maria das Graças Volpe Nunes
Maria Inés Castiñeira
Marko Grobelnik
Miguel Toro Bonilla
Ole André Brevik
Olga Padron
Paola Riccucci
Paul Piwek
Paula Brito
Paulo Quaresma
Philippe Beaune
Rafal Bogacz
Renata Vieira
Renata Wassermann
Renato Ramos
Richard Power
Robert Engels
Roberto Vivo
Rodger Kibble
Roger Evans
Roseli Romero
Rui Camacho
Sabine Broda
Samuel Landau
Sandra Sandri
Sébastien Picault
Sheila Veloso
Silvia Schiaffino
Siva Perraju Tolety
Solange Oliveira Rezende
Stan Szpakowicz
Tallys Hoover Yunes
Torbjorn Dahl
Viktor Jovanoski

Sponsoring Institutions

Until the publication of this volume, IBERAMIA–SBIA 2000 has had the financial support of the following institutions:

FAPESP	Fundação de Amparo à Pesquisa do Estado de São Paulo
CNPq	Conselho Nacional de Pesquisa
CAPES	Coordenação de Aperfeiçoamento do Pessoal de Nível Superior
CTEEP	Companhia de Transmissão de Energia Elétrica Paulista

Table of Contents

AI in Education and Intelligent Tutoring Systems

Knowledge Representation and Reasoning

Machine Learning and Knowledge Acquisition

Knowledge Discovery and Data Mining

Natural Language Processing

Robotics

Computer Vision

Uncertainty and Fuzzy Systems

Genetic Algorithms and Neural Networks

Decision-Rule Solutions for Data Mining with Missing Values

Sholom M. Weiss and Nitin Indurkhya

IBM T.J. Watson Research Center,
P.O. Box 218, Yorktown Heights, NY 10598, USA
sholom@us.ibm.com, nitin@data-miner.com

Abstract. A method is presented to induce decision rules from data with missing values where (a) the format of the rules is no different than rules for data without missing values and (b) no special features are specified to prepare the the original data or to apply the induced rules. This method generates compact Disjunctive Normal Form (DNF) rules. Each class has an equal number of unweighted rules. A new example is classified by applying all rules and assigning the example to the class with the most satisfied rules. Disjuncts in rules are naturally overlapping. When combined with voted solutions, the inherent redundancy is enhanced. We provide experimental evidence that this transparent approach to classification can yield strong results for data mining with missing values.

Keywords: decision rule induction, boosting

1 Introduction

Data warehousing has increased the opportunities for data mining. Unlike the datasets that have often been used in scientific experimentation, transactional databases often contain many missing values. Data with missing values complicates both the learning process and the application of a solution to new data. Depending on the learning method, special data preparation techniques may be necessary. This increases the amount of data preprocessing.

The most common preprocessing techniques involve filling in the missing values. For instance, in [8], several general approaches are described to replace the missing values prior to mining:

- Estimate values using simple measures derived from means and standard deviations
- Estimate values by regression
- Augment each feature with a special value or flag that can be used in the solution as a condition for prediction

While potentially useful, each of these techniques has obvious drawbacks. Estimating the missing value by a simple measure like a class mean is often

M.C. Monard and J.S. Sichman (Eds.): IBERAMIA-SBIA 2000, LNAI 1952, pp. 1–10, 2000.

circular reasoning that is a direct substitute for the class label. Moreover, missing values for new cases remain a problem. Estimating by regression is just as complex a task as the given classification problem. Using the occurrence of a missing value to reach a positive or negative conclusion may not be sensible in many contexts and clearly increases the complexity of the solution.

With the commercial application of data mining methods, increased attention is given to decision trees and rules. These techniques may perform well and have the potential to give insight to the interpretation of data mining results, for example in marketing efforts.

Decision trees methods have a long history of special techniques for processing missing values[4],[9]. They process training data without any transformations, but have surrogates for tree nodes when values are missing. When a true-or-false test can potentially encounter a missing value, a number of alternative tests are also specified that hopefully track the results of the original test. Thus, the original data remain stable, but special methods and representations are needed to process missing data.

Decision rules are closely related to decision trees. The terminal nodes of a tree can be grouped into Disjunctive Normal Form (DNF) rules, only one of which is satisfied for a new case. Decision rules are also DNF rules, but allow rules to overlap, which potentially allows for more compact and interesting rule sets.

Decision tree induction methods are more efficient than those for decision rule induction–some methods for decision rule induction actually start with an induced decision tree. Procedures for pruning and optimization are relatively complex[12][5]. Single decision trees are often dramatically outperformed by voting methods for multiple decision trees. Such methods produce exaggeratedly complex solutions, but they may be the best obtainable with any classifier. In [6], boosting techniques [10] are used by a system called SLIPPER to generate a weighted set of rules that are shown to generally outperform standard rule induction techniques. While these rules can maintain clarity of explanation, they do not match the predictive performance of the strongest learning methods, such as boosted trees. Of particular interest to our work is [7] where very small trees are boosted to high predictive performance by truncated tree induction (TTI). Small trees can be decomposed into a collection of interpretable rules. Some of the boosted collections of tiny trees, even tree stumps, have actually performed best on benchmark applications.

In this paper, we discuss methods for learning and application of decision rules for classification from data with many missing values. The rules generated are Disjunctive Normal Form (DNF) rules. Each class has an equal number of unweighted rules. A new example is classified by applying all rules and assigning the example to the class with the most satisfied rules. Disjuncts in rules are naturally overlapping. When combined with voted solutions, the inherent redundancy is enhanced. The method can induce decision rules from data with missing values where (a) the format of the rules is no different than rules for data without missing values and (b) no special features are specified to prepare the the original data or to apply the induced rules. We provide experimental evidence that this transparent approach to classification can yield strong results for data mining with missing values.

2 Methods and Procedures

The classical approach to rule induction is a two-step process. The first step is to find a single covering solution for all training examples. The covering rule set is found directly by inducing conjunctive rules or indirectly by inducing a decision tree. The direct solution usually involved inducing one rule at a time, removing the cases covered by the rule, and then repeating the process. The second step is to prune the covering rule set or tree into smaller structures, and pick the best one, either by a statistical test or by applying the rule sets to independent test cases.

A pure DNF rule for classification is evaluated as satisfied or not. If satisfied, the rule implies a specific class. The conditions or components of a rule can be tested by applying $\leq$ or $>$ operators to variables and coding categorical values separately as 1 for true and 0 for false.

We can measure the size of a DNF rule with two measurements: (a) the length of a conjunctive term and the number of terms (disjuncts). For example,

$$\{\mathrm{c}_1\ \mathrm{c}_2\ \mathrm{c}_3\} \text{ OR } \{\mathrm{c}_1\ \mathrm{c}_3\ \mathrm{c}_4\} \Rightarrow \text{Class}$$

is a DNF rule for conditions c_i with maximum length of three and two terms (disjuncts). Complexity of rule sets can be controlled by providing an upper bound on these two measurements.

Table 1 describes the standard analysis of results for binary classification. For evaluation purposes, a rule is applied to each case. Classification error is measured as in equation 1. For case i, FP(i) is 1 for a false positive, FN(i) is 1 for a false negative, and 0 otherwise.

Table 1. Analysis of Error for Binary Classification

	Rule-true	Rule-false
Class-true	True positives (TP)	False negatives (FN)
Class-false	False positives (FP)	True negatives (TN)

$$Error = FP + FN;\ \ FP = \sum_i FP(i);\ \ FN = \sum_i FN(i) \tag{1}$$

For almost all applications, more than one rule is needed to achieve good predictive performance. In our lightweight approach, a solution consists of a set of an equal number of unweighted rules for each class. A new example is classified by picking the class having the most votes, the class with the most satisfied rules. We are very democratic; each class has an equal number of rules and votes, and each rule is approximately the same size.

The principal remaining task is to describe a method for inducing rules from data. So far we have given a brief description of binary classification. Yet, this form of binary classification is at the heart of the rule induction algorithm. Let's continue to consider binary classification. The most trivial method for rule induction is to grow a conjunctive term of a rule by the greedy addition of a single condition that minimizes error. To ensure that a term is always added (when error is nonzero) we can define a slightly modified measure, err1 in equation 2. Error is computed over candidate conditions where TP is greater than zero. If no added condition adds a true positive, the cost of a false negative error is doubled and the minimum cost solution is found. The cost of a false positive remains at 1. The minimum err1 is readily computed during sequential search using the bound of the current best err1 value.

$$Err1 = FP + k \cdot FN \; \{where\; k = 1, 2, 4 ... and\; TP > 0\} \tag{2}$$

$$Frq(i) = 1 + e(i)^3 \tag{3}$$

$$FP = \sum_i FP(i) \cdot frq(i); \;\; FN = \sum_i FN(i) \cdot frq(i) \tag{4}$$

The lightweight method is adaptive, and follows the well-known principle embodied in boosting: Give greater representation to erroneously classified cases. The technique for weighting cases during training is greatly simplified from the usual boosting methods. Analogous to [3], no weights are used in the induced solution. Weighting of cases during sampling follows a simple method: Let $e(i)$ be the cumulative number of errors for case i for all rules. It is computed by applying all prior induced rules and summing the errors for a case. The weighting given to a case during induction is an integer value, representing a relative frequency of that case in the new sample. Equation 3 is the frequency that is used. It has good empirical support, having had the best reported results on an important text-mining benchmark [11], and was first described in [13]. Thus if 10 rules have been generated, 4 of them erroneous on case i, then case i is treated as if it appeared in the sample 65 times. Based on prior experience, alternative functions to Equation 3 may also perform well. Unlike the results of [1] for the alternative of [3], Equation 3 performs well with or without random resampling, and the LRI algorithm uses no random resampling. The computation of FP and FN during training is modified slightly to follow Equation 4.

Err1 is computed by simple integer addition. In practice, we use only 33 different values of e(i), for i=0 to 32. Whenever, the number of cumulative errors exceeds 32, all cumulative errors are normalized by an integer division of 2.

The training algorithm for inducing a DNF rule R is given in Figure 1. The algorithm is repeated sequentially for the desired number of rules. Rules are always induced for binary classification, class versus not-class. A m-class classification problem is handled by mapping it to m binary classification problems – one for each class. Each of the binary classification problems can be computed independently and in parallel. As we shall in Section 3, the equality of voting and rule size, makes the predictive performance of rules induced from multiple binary classification problems quite comparable.

1. Grow conjunctive term T until the maximum length (or until $FN = 0$) by greedily adding conditions that minimize err1.
2. Record T as the next disjunct for rule R. If less than the maximum number of disjuncts (and $FN > 0$), remove cases covered by T, and continue with step 1.
3. Evaluate the induced rule R on all training cases i and update $e(i)$, the cumulative number of errors for case i.

Fig. 1. Lightweight Rule Induction Algorithm

A pure DNF rule induction system has strong capabilities for handling missing values. Disjunction can produce overlap and redundancy. If we apply a rule to a case, and a term is not satisfied because one of its conditions has a missing value, the rule may still be satisfied by one of the other disjuncts of the rule. These rules have no special conditions referring to missing values; they look no different than rules induced from data with no missing values. How is this accomplished? For the application of rules, a term is considered not satisfied when a missing value is encountered in a case. During training, the following slight modifications are made to the induction procedures:

- When looping to find the best attribute condition, skip cases with missing values.
- Normalize error to a base relative to the frequency of all cases.

$$Norm_k = \frac{\sum_{n,\, all} frq(n)}{\sum_{i,\, w/o\ missing\ vals} frq(i)} \tag{5}$$

$$FP_k = Norm_k \cdot \sum_i FP(i) \cdot frq(i) \tag{6}$$

$$FN_k = Norm_k \cdot \sum_i FN(i) \cdot frq(i) \tag{7}$$

Each feature may have a variable number of missing values. The normalization factor is computed as in Equation 5 for feature k. The normalization factor is the total number of cases, n, including missing values cases, divided by the frequency of cases without missing values. False positives and negatives are computed as in Equations 6 and 7, a straightforward normalization of Equation 4.

To select the solution with the best predictive performance, decisions must be made about the two key measures of rule size: (a) conjunctive term length and (b) the number of disjuncts. For data mining large samples, the best solution can be found by using an independent test set for estimating true error. If only a single complexity measure is varied during training, such as the number of disjuncts, then the estimated error rates for comparing the different solutions using only one independent test set are nearly unbiased[4].

3 Results

Before we present experimental results for lightweight rule induction, let's consider our real-world experience in an important data mining application: the detection of patterns in survey data. IBM, like many companies, surveys the marketplace trying to gauge customer attitudes. In the case of IBM, thousands of IT professionals are surveyed about their buying intentions and their view of IBM products and the products of competitors. Survey data may be collected every quarter or perhaps as frequently as every week. For some recent period, such as the most recent quarter, the survey data are grouped into a sample. The sample can be mined, and purchasing patterns that are detected can potentially be of great value for marketing. The actual survey questions number in the many hundreds. Not all questions are asked of every respondent; records contain many missing values.

What might be interesting questions? In practice, it's relatively easy to specify critical classification problems, for example "can we distinguish those people who intend to increase purchases of IBM equipment versus those that do not?" With hundreds of features and relatively difficult goals for discrimination, solutions of high complexity are likely when standard methods are used to find a minimum error solution. Such solutions would not be acceptable to the marketers who make recommendations and take actions. In our case, the lightweight approach has an effective means of bounding the complexity of solutions. We can tradeoff complexity and somewhat stronger predictive performance with clarity of interpretation. While it may seem severe, rules induced from one survey were restricted in size to no more than two terms with no disjunction and three rules for each of two classes. These simplified rules perform somewhat weaker than more complex and larger rule sets, but in this application, interpretability far outweighs raw predictive performance. Moreover, although the survey data are riddled with missing values, the solutions, posed in the form of decision rules, extract the essential patterns without ever mentioning missing values.

To evaluate formally the performance of lightweight rule induction, datasets from the UCI repository [2] were processed. Table 2 summarizes the characteristics of these data. The number of features describes numerical features and categorical variables decomposed into binary features. Because the objective is data mining, we selected datasets having relatively large numbers of training ca-