Decision Support Systems for Sustainable Development

A Resource Book of Methods and Applications

EDITED BY
GREGORY E. KERSTEN
ZBIGNIEW MIKOLAJUK
ANTHONY GAR-ON YEH

International Development Research Centre (IDRC)
and
Kluwer Academic Publishers
DECISION SUPPORT SYSTEMS FOR SUSTAINABLE DEVELOPMENT

A Resource Book of Methods and Applications
For
Margaret Kersten,
Anna Mikolajuk and
Brenda Yeh
with thanks for
their inspiration
and encouragement.
Thank you!
# Contents

## Preface

Decision Support Systems for Sustainable Development: An Overview  
Tung X. Bui  

### I. SUSTAINABLE DEVELOPMENT AND DECISION MAKING

1. Sustainable Development and Decision Support Systems  
Zbigniew Mikolajuk and Anthony Gar-On Yeh  

2. Decision Making and Decision Support  
Gregory E. Kersten  

3. Decision Support with Geographic Information Systems  
Anthony Gar-On Yeh  

### II. APPLICATIONS AND CASE STUDIES

4. Decision Support for Sustainable Land Development: A Case Study of Dongguan  
Anthony Gar-On Yeh and Xia Li  

5. Water Resource Management: A Case Study for EcoKnowMICS  
Eduardo D. Gamboa  

6. Decision Support for Incentive Strategies: A Rural Development Application in Central Africa  
Benoit Gailly and Michel Installé  

Alexander V. Lotov, Lioubov V. Bourmistrova, and Vladimir A. Bushenkov  

8. Integrated Rural Energy Decision Support Systems  
Shaligram Pokharel and Muthu Chandrashekar
9. DSS for Sustainable Land Management: A South-East Asia Case
Mohammad Rais, Samuel Gameda, Eric T. Craswell, Adisak Sajjapongse, and Hans-Dieter Bechstedt

10. Mapping Critical Factors for the Survival of Firms: A Case Study in the Brazilian Textile Industry
Carlos A. Bana e Costa, Emerson C. Corrêa, Leonardo Ensslin and Jean-Claude Vansnick

11. Learning Negotiations with Web-based Systems: The Base of lIMB
T. R. Madanmohan, Gregory E. Kersten, Sunil J. Noronha, Margaret Kersten and David Cray

Ahmed Rafea

III. RESEARCH ISSUES

13. Rule Induction in Constructing Knowledge-Based Decision Support
Tu Bao Ho

14. Organizational Memory Information Systems: A Case-based Approach to Decision Support
Helen G. Smith, Frada V. Burstein, Ramita Sharma and Dayo Sowunmi

15. Software Internationalization Architectures for DSS
Patrick A.V. Hall

16. Software Integration for Environmental Management
Victor S. Chabanyuk and Alexandr V. Obvintsev

17. Design of Decision Support Systems as Federated Information Systems
David J. Abel, Kerry Taylor, Gavin Walker and Graham Williams

18. Knowledge Discovery in Databases and Decision Support
Anantha Mahadevan, Kumudini Ponnudurai, Gregory E. Kersten and Roland Thomas
IV. DECISION SUPPORT SYSTEMS FOR SUSTAINABLE DEVELOPMENT

19. Experience and Potential 369
Patrick A.V. Hall et al.

20. DSS Application Areas 391
Gregory E. Kersten and Gordon Lo

Glossary 409

Contributors 415
13 RULE INDUCTION IN CONSTRUCTING KNOWLEDGE-BASED DECISION SUPPORT
Tu Bao Ho

1. Introduction

Decision support systems (DSSs) are computer-based systems that can help decision-makers to solve semistructured problems by allowing them to access and use data and analytic models interactively. Knowledge-based systems (KBSs) are systems that generate quality solutions to problems requiring computer-based reasoning knowledge. There are two main ways of using KBS techniques in computer-based decision-making. The first is to use KBSs directly as kinds of DSSs. Recent progress in techniques for coupling databases and knowledge bases allows the user to exploit KBSs more effectively in various decision problems such as diagnosis, design, and planning (Sprague and Watson, 1993; Blanning and King, 1993). The other is to integrate KBS technologies with conventional DSSs (El Najdawi and Stylianou, 1993; Sullivan and Fordyce, 1994; Turban and Aronson, 1998). We call knowledge-based decision support systems (KBDSSs) "software in both directions."

By assisting with the simultaneous use of data, knowledge, analytic models, and dialogue, KBDSSs have the potential to be useful in the field of sustainable development, particularly in helping developing countries meet their own unique development needs. Knowledge and advanced information technology from industrialized countries could be used to solve complex problems in fields like public administration, urban and rural development, transportation, health care, special needs, education, environment, and agriculture (Mansell and Wehn, 1998). As knowledge is central to intelligence systems, how to obtain that knowledge for decision-making is a crucial issue, and it remains the bottleneck in the development of KBDSSs. Automatic access to knowledge from external resources would be particularly beneficial to developing countries where there is a lack of expertise in providing knowledge for KBDSSs, be-
cause many large databases in the fields listed above already exist. It is clear that decision-makers in these countries need to understand and use their data to find useful knowledge that can help in their country's development.

Knowledge acquisition (KA) has been seen as a problem of transferring knowledge — in other words, extracting discourse and translating it into the implementation language constructs. Recently, however, KA is not merely being seen as an exercise in 'expertise transfer' but as a modeling process. Much of the current work focuses on systems that follow the KA paradigm of expert or external resources \(\rightarrow\) automatic acquisition \(\rightarrow\) knowledge base instead of the traditional paradigm of expert to knowledge base by way of a knowledge engineer.

Automatic KA shares the same goal as the rapidly growing interdisciplinary field of knowledge discovery and data mining — extracting useful knowledge from large databases (Fayyad et al., 1996). Automatic KA is based mainly on applications of machine learning and rule induction. Machine learning (ML) — the study of computational algorithms that improve automatically through experience — can provide increasing levels of automation in the construction of knowledge bases. Rule induction is one paradigm of ML that has achieved many recent successes in real-world applications.

Section 2 deals with the problem and the main steps of the rule induction process. Section 3 presents the key ideas of two rule induction methods, CABRO and OSHAM, from supervised and unsupervised data, respectively, and their implementation as interactive-graphic systems (Ho, 1995; Ho, 1997; Ho et al., 1998; Nguyen and Ho, 1999). To evaluate these systems, we carry out experimental comparative studies for CABRO, OSHAM, and some widely used systems such as C4.5 (Quinlan, 1993), CART (Breiman et al., 1984), AUTOCLASS (Cheeseman and Stutz, 1996), and others. Section 4 describes the mutual use of CABRO and OSHAM in the construction of KBDSSs within the framework of the knowledge-based system generator TESOR (Ho et al., 1992). Section 5 presents a case study to illustrate how diagnostic knowledge was induced from a clinical database on meningoencephalitis.

2. Problem and steps of rule induction

Two main components of a KBS are an inference engine and a knowledge base. Most available KBSs provide an inference engine and knowledge editor but not the capability to acquire knowledge automatically. Langley and Simon (1995) pointed out that "rule induction may never entirely replace the knowledge engineer in constructing KBS, but significant progress toward automation in knowledge engineering has already been made." They examined various successes in applying rule induction to real-world problems, including increasing yield in chemical process control, making credit decisions, forecasting severe thunderstorms, diagnosis of mechanical devices, improving separation of gas from oil, and preventing breakdowns in electrical transformers.

Langley and Simon called rule induction the paradigm of ML that "employs condition-action rules, decision trees, or similar logical knowledge structures. Here the performance element sorts instances down the branches of the decision trees or finds
the first rule whose conditions match the instance, typically using a logical matching process. Information about classes or predictions is stored in the action sides of the rules or the leaves of the trees. Learning algorithms in the rule-induction framework usually carry out a greedy search through the space of decision trees or rule sets, typically using a statistical evaluation function to select attributes for incorporation into the knowledge structure. Most methods partition the training data recursively into disjoint sets, attempting to summarize each set as a conjunction of logical conditions.”

There are five main steps in the rule-induction process which the DSS user and developer should know while applying rule induction programs.

The first step is to formulate the decision problem so that part of it can be handled by an induction method. An important distinction in ML has to do with whether one uses the learned knowledge for one-step classification or prediction, or for some form of multistep inference or problem solving. Decision tasks such as process control, design, diagnosis, planning, and scheduling are often complex, yet one can identify components that involve simple classification, a task for which there exist robust induction algorithms.

The second step is to determine the representation of knowledge to be learned. Knowledge can be represented in different ways on a computer — rules, decision lists, inference networks, decision trees, and concept hierarchies. A decision tree is a classifier in the form of a tree structure that is either a leaf that indicates a class of instances, or a decision node that specifies some test to be carried out on a single attribute value, with one branch and subtree for each possible outcome of the test. A decision tree can be used to classify an instance by starting at the root of the tree and moving up through it until a leaf is encountered. A decision tree or concept hierarchy can be converted easily into a set of decision rules.

The third step is to collect the training data. Rule induction methods start from data sets representing cases in application domains. The simplest situation concerns Boolean or binary features, with each case specifying the presence or absence of a feature. A slightly more sophisticated situation concerns symbolic attributes that consist of nominal (categorical) or ordinal ones. Numeric attributes that take on real values are also possible. Data are often viewed in the instance space, which may consist of a mix of attributes. Inductive learning methods may be classified into two groups on the basis of the degree of supervision in the training data. Early works in concept learning would focus on the simpler supervised task in which the learner is asked to characterize concepts from a given set of classified instances (members of the concept). This task continues to receive much attention in ML, where most of the research is concerned with influential methods like decision tree induction.

The unsupervised task, though less developed than the supervised, can arise just as often in real-world problems; as a result, it has attracted a considerable amount of interest. In this task, the learner is asked to determine concepts and knowledge structures from a given set of unclassified instances. In general, rule induction programs work on data sets represented in data tables where each line stands for the description of an instance and each column stands for an attribute. A special column, which exists only in supervised data, contains the label of instances. Training data are also provided offline (all instances are presented simultaneously) or online (instances are presented once at a time). This leads to nonincremental or incremental learning, respec-
tively. After collecting the training data, the user needs to distinguish between these different methods to choose which one is most suitable.

The fourth step is to evaluate the learned knowledge. Experiments show that rules induced from training data are not always of high quality. A standard way to evaluate these rules is to divide the data into two sets, training and testing. One can repeat this process a number of times with different splits, then average the results to estimate the rules performance. This procedure is known as cross-validation.

The fifth and final step is to field the knowledge base. In some cases, discovered knowledge can be used without embedding it into a computer system. Otherwise, users might expect that induced rules could be put on computer and exploited by some programs. In particular, the results of a rule program could be considered decision knowledge for a KBDSS. It is most important to determine whether the learned knowledge in fact can be used in decision-making.

3. Two rule induction methods

3.1 Rule induction from supervised data

CABRO is a method of constructing decision trees from supervised (classified) data. The basic task of supervised learning is to extract knowledge that can be used to predict classes of classified instances correctly from a given set of classified instances. Here is an example of discovering rules from data for diagnosing two kinds of heart disease. CABRO started from a database (Cleveland Clinic Foundation) containing records of patients who belong to one of two heart disease classes, ‘H’ or ‘S’ (the last column), and discovered diagnosis rules with associated (estimated) prediction accuracy. Each record consists of information observed on 14 numeric and categorical attributes: age, sex, chest pain type, resting blood pressure, vessels colored, etc. Some patient records extracted from the database follow:

63, male, angina, 145.0, 233.0, true, hyp, 150.0, fal, 2.3, down, 0.0, fix, H
67, male, asympt, 120.0, 229.0, fal, hyp, 129.0, true, 2.6, flat, 2.0, rev, S
41, female, abnang, 130.0, 204.0, fal, hyp, 172.0, fal, 1.4, up, 0.0, norm, H
56, male, abnang, 120.0, 236.0, fal, norm, 178.0, fal, 0.8, up, 0.0, norm, H
62, female, asympt, 140.0, 268.0, fal, hyp, 160.0, fal, 3.6, down, 2.0, norm, S
57, female, asympt, 120.0, 354.0, fal, norm, 163.0, true, 0.6, up, 0.0, norm, H

The diagnosis of unpredicted patients can be obtained by matching their symptoms (records) against conditions of discovered rules such as

Rule 7: IF sex = male
        age <= 67
        resting blood press > 108
        max heart rate <= 171
        number of vessels colored > 0
    THEN diagnosis of heart disease S [accuracy = 88.8%]; and

Rule 35: IF sex = female
        number of vessels colored <= 0
        thal = norm
resting blood press \( \leq 140 \)

THEN diagnosis of heart disease H [accuracy = 93.6%].

Among approaches to supervised concept learning, decision-tree induction (DTI) is certainly the most active and applicable one. DTI systems differ from each other in the way they deal with the problems of attribute selection (or choosing the ‘best’ attribute to split a decision node) and pruning (or avoiding overfitting). This chapter presents our solution to the attribute selection problem, among others, including model selection, visualization, and interactive learning (Nguyen and Ho, 1999).

It is known that the most widely used measures for attribute selection are information-theory based, such as information gain or gain-ratio (Quinlan, 1993), or statistics-based, such as Chi-square or Gini-index (Breiman et al., 1984). For attribute selection, CABRO employs R-measure (Ho et al., 1998), which is inspired by the notion of dependency degree in the rough set theory (Pawlak, 1991). Rough set theory is a mathematical tool to deal with vagueness and uncertainty. The basic idea in this theory is to ‘view’ approximately each subset \( X \) of an object set \( O \) by its lower and upper approximations, \( E \) and \( E^* \) w.r.t., an equivalence relation \( E \subseteq O \times O \). These approximations of \( X \) are defined, respectively, by and \( E(x) = \{ o \in O : [o]_E \subseteq X \} \) and \( E(x) = \{ o \in O : [o]_E \cap X \neq \emptyset \} \) where \([o]_E\) denotes the equivalence class of an object \( o \) in \( E \). A key concept in the rough set theory is the degree of dependency, \( \mu_p(Q) \), of an attribute set \( Q \) on an attribute set \( P \):

\[
\mu_p(Q) = \frac{\text{card}(\bigcup_{[o]_P} P([o]_Q))}{\text{card}(O)} = \frac{\text{card}(\{ o \in O : [o]_E \subseteq [o]_Q \})}{\text{card}(O)}.
\]

The measure \( \mu_p(Q) \) can be used directly in decision-tree induction for selecting attributes, where \( Q \) stands for the class attribute and \( P \) stands for each descriptive attribute. Our analysis and experiments have shown, however, that it is not robust enough with noisy data or sensitive enough when partitions of \( O \) generated by \( Q \) and \( P \) are nearly identified. Inspired by \( \mu_p(Q) \), we proposed a new measure, called R-measure, for the dependency of \( Q \) on \( P \):

\[
\tilde{\mu}_p(Q) = \sum_{[o]_P} \frac{1}{\text{card}(O)} \max_{[o]_Q} \frac{\text{card}([o]_P \cap [o]_Q)^2}{\text{card}([o]_P)}.
\]

We show that R-measure can overcome the limitations of \( \mu_p(Q) \) in different situations (Nguyen and Ho, 1999). Using this measure for selecting attributes, we developed the CABRO system in the following common framework of decision-tree induction:

1. choose the ‘best’ attribute that maximizes the chosen attribute selection measure;
2. extend the tree by adding a new branch for each attribute value;
3. sort training examples to leaf nodes; and
4. if examples are unambiguously classified, then stop, or else repeat steps 1-4 for leaf nodes.

There are three main criteria for evaluating discovered decision trees: predictive accuracy, size, and understandability. Predictive accuracy refers to the ability of the decision tree to predict unknown instances into learned classes. The predictive accuracy is measured also in terms of error rates (the proportion of incorrect predictions that a tree makes on the test data). Size relates to the generally accepted Occam’s Razor principle — the fewer nodes in the tree, the better. Understandability relates to the knowledge representation.

In order to obtain a reliable evaluation of R-measure, we have carried out an experimental comparative study. Recently, k-fold stratified cross validation has been recommended as an appropriate method for evaluating techniques with real-world datasets similar to those of the University of California, Irvine (UCI) repository of ML databases. We used this benchmark to evaluate three other well-known measures — gain-ratio (Quinlan, 1993), Gini-index (Breiman et al., 1984), Chi-square in statistics and R-measure — on 18 datasets from the UCI repository. Table 1 includes information on the datasets such as their name, dimension (number of attributes X number of instances), and type ([s] for symbolic, [n] for numeric and [m] for mixture), as well as the experimental comparative results of the error rates of pruned trees for the four measures on these datasets. The lowest error rate attained at each dataset is in bold.

Table 1. Experimental comparative error rates of four measures.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Gain-ratio</th>
<th>Gini-index</th>
<th>Chi-square</th>
<th>R-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vote, 16X300 (s)</td>
<td>5.0 ± 2.8</td>
<td>5.9 ± 2.7</td>
<td>5.9 ± 2.7</td>
<td>5.7 ± 2.7</td>
</tr>
<tr>
<td>Cancer, 9X700 (s)</td>
<td>7.4 ± 2.9</td>
<td>7.4 ± 3.5</td>
<td>7.4 ± 3.5</td>
<td>7.1 ± 3.4</td>
</tr>
<tr>
<td>Promoters, 45X105 (s)</td>
<td>24.5 ± 7.5</td>
<td>22.7 ± 10.0</td>
<td>22.7 ± 10.0</td>
<td>22.7 ± 10.0</td>
</tr>
<tr>
<td>Shuttle, 9X956 (s)</td>
<td>0.2 ± 0.1</td>
<td>0.2 ± 0.1</td>
<td>0.3 ± 0.1</td>
<td>0.3 ± 0.1</td>
</tr>
<tr>
<td>Solar flare, 9X956 (s)</td>
<td>25.3 ± 1.5</td>
<td>27.8 ± 1.3</td>
<td>26.6 ± 2.0</td>
<td>25.5 ± 1.0</td>
</tr>
<tr>
<td>Diabetes, 8X768 (n)</td>
<td>25.3 ± 2.6</td>
<td>25.6 ± 2.5</td>
<td>25.5 ± 2.5</td>
<td>25.3 ± 2.6</td>
</tr>
<tr>
<td>Spice, 45X3189 (n)</td>
<td>8.0 ± 1.7</td>
<td>8.4 ± 1.8</td>
<td>8.8 ± 1.7</td>
<td>8.6 ± 1.9</td>
</tr>
<tr>
<td>Glass, 9X214 (n)</td>
<td>34.5 ± 8.2</td>
<td>36.8 ± 6.8</td>
<td>37.3 ± 6.4</td>
<td>35.9 ± 6.9</td>
</tr>
<tr>
<td>Waveform, 36X3195 (s)</td>
<td>25.7 ± 1.1</td>
<td>24.4 ± 1.8</td>
<td>26.8 ± 1.3</td>
<td>25.1 ± 1.1</td>
</tr>
<tr>
<td>Heart disease, 13X270 (m)</td>
<td>25.6 ± 4.1</td>
<td>25.6 ± 5.6</td>
<td>26.3 ± 4.9</td>
<td>25.2 ± 4.6</td>
</tr>
<tr>
<td>Vehicle, 18X846 (n)</td>
<td>32.7 ± 5.1</td>
<td>32.0 ± 3.7</td>
<td>31.9 ± 3.2</td>
<td>31.8 ± 3.5</td>
</tr>
<tr>
<td>Hypothyroid, 25X3163 (n)</td>
<td>0.9 ± 0.4</td>
<td>0.9 ± 0.4</td>
<td>0.9 ± 0.4</td>
<td>0.9 ± 0.4</td>
</tr>
<tr>
<td>Audiology, 70X226 (s)</td>
<td>30.9 ± 11.0</td>
<td>30.9 ± 11.9</td>
<td>45.2 ± 8.7</td>
<td>29.1 ± 11.7</td>
</tr>
<tr>
<td>Cars, 8X392 (n)</td>
<td>26.0 ± 2.0</td>
<td>26.8 ± 5.2</td>
<td>26.5 ± 5.2</td>
<td>25.2 ± 4.8</td>
</tr>
<tr>
<td>Horse colic, 28X368 (n)</td>
<td>14.3 ± 5.1</td>
<td>16.8 ± 3.5</td>
<td>17.0 ± 3.3</td>
<td>15.9 ± 4.2</td>
</tr>
<tr>
<td>Pima diabetes, 8X768 (n)</td>
<td>23.4 ± 3.6</td>
<td>23.5 ± 3.5</td>
<td>23.5 ± 3.5</td>
<td>23.9 ± 3.2</td>
</tr>
<tr>
<td>Segmentation, 19X2310 (n)</td>
<td>6.2 ± 1.6</td>
<td>6.1 ± 2.0</td>
<td>7.6 ± 2.0</td>
<td>6.1 ± 2.1</td>
</tr>
<tr>
<td>Iris, 4X150 (n)</td>
<td>3.3 ± 3.3</td>
<td>2.7 ± 3.2</td>
<td>4.0 ± 4.0</td>
<td>4.0 ± 4.0</td>
</tr>
</tbody>
</table>

Two observations can be drawn from these experiments:

- The gain-ratio and R-measure are somehow comparable. They had the lowest values of error rates in nine datasets. The Gini-index and Chi-square have average and high error rates and attained those in five and two datasets, respectively.
Moreover, CABRO offers various advantages over other systems on model selection, visualization, and interactive learning (Nguyen and Ho, 1999).

- As for tree size, the gain-ratio demonstrated its advantage of smaller trees when datasets were small or mid-size, but not when datasets were large. However, for datasets with big trees, Chi-square often had noticeably bigger ones.

Figure 1. Generation and interpretation of decision trees from soybean data by CABRO.

3.2 Rule induction from unsupervised data

Here, the basic KA task is to find simultaneously, from a given set of unsupervised (unclassified) instances, a hierarchical clustering of subsets of instances and intensional descriptions of these subsets that satisfy certain conditions. Essentially, unsupervised induction methods differ from each other in two ways: views on concepts and constraints of categorization. Among views on concepts, the classical, prototype and exemplar views are widely known and used. Among categorization constraints, the similarity, feature correlation, and structure of the concept hierarchy are widely known and used. OSHAM employs the classical view on concepts and is able to form an effective concept hierarchy from unclassified data (Ho, 1995). Essentially, OSHAM searches to extract a good concept hierarchy by exploiting the lattice structure of Ga-
lois concepts as the hypothesis space. Recently, OSHAM has been extended to a hybrid system that allows for a higher performance by combining its original view on concepts with the prototype and exemplar views (Ho, 1996; Ho, 1997). The essential ideas of the hybrid OSHAM are

1. to extend the classical representation of a concept from a pair of extent, intent and its hierarchical structure information to a hybrid representation of a 10-tuple with additional components (these include the probability of the concept occurrence, homogeneity, typical instances, conditional probability for occurrence and dispersion of typical instances, and the quality of the concept evaluated in terms of the concept hierarchy structure); and

2. to find sufficiently general and discriminant concepts at each level of the concept hierarchy while determining their typical instances on the basis of a quality criterion that combines the similarity and structural constraints.

OSHAM has been implemented in the X Window on a Sparcstation with the direct manipulation style of interaction that allows the user to participate actively in the discovery process. The user can initialize parameters to cluster data, visualize the concept hierarchy gradually, observe the results and the quality estimation, manually modify the parameters when necessary before the system continues to go further to cluster subsequent data, or backtrack to regrow branches of the concept hierarchy with respect to the categorization scheme. Figure 2 shows a main screen of the interactive OSHAM with a hierarchy of overlapping concepts learned from breast cancer data collected at a Wisconsin hospital. A full description of a concept in Figure 2 is given below:

```
CONCEPT 43
Level = 5
Super_Concepts = {29}, Sub_Concepts = {52, 53}
Features = (Uniformity of Cell Size, 1) AND (Bare Nuclei, 1) AND
           (Bland Chromatin, 1) AND (Uniformity of Cell Shape, 2)
Local_instances/Covered_instances = 6/25
Local_instances = {8, 127, 221, 236, 415, 661}
Concept_probability = 0.041666
Local_instance_conditional_probability = 0.240000
Concept_dispersion = 0.258848
Local_instance_dispersion = 0.055556
Subconcept_partition_quality = 0.519719.
```

The interpretation of induced results is commonly understood as the process of matching an unknown case \( e \) to discovered classes (concepts). As the generality decreases along branches of a hierarchical structure, we say that a concept \( C_k \) matches the unknown instance \( e \) if \( C_k \) is the most specific concept in a branch that matches \( e \) intensionally (though all superconcepts of \( C_k \) match \( e \)). Naturally, there are three types of outcomes when one is logically matching an unknown instance \( e \) with the learned concepts: only one concept that matches \( e \) (single-match), many concepts that match \( e \) (multiple-match), and no concept that matches \( e \) (no-match). We developed an interpretation procedure for concept hierarchies that uses the concept intent, the hierarchical structure information, the probabilistic estimations, and the nearest neighbors of
unknown instances (Ho and Luong, 1997). This interpretation procedure consists of two stages: (1) find all concepts on the concept hierarchy that match e intensionally, and (2) decide which one among these concepts matches e best.

Comparative experiments were carried out for the well-known system AUTOCLASS (Cheeseman, 1996) and OSHAM on 10 datasets from the UCI repository of ML databases.

Table 2. Experimental comparative results of OSHAM.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>num.att.</th>
<th>sym.att.</th>
<th>instances</th>
<th>classes</th>
<th>AUTOC</th>
<th>OSHAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wisconsin breast cancer</td>
<td>9 -</td>
<td>699</td>
<td>2</td>
<td>96.6</td>
<td>92.6</td>
<td></td>
</tr>
<tr>
<td>Congressional voting</td>
<td>17</td>
<td>-</td>
<td>435</td>
<td>2</td>
<td>91.2</td>
<td>93.7</td>
</tr>
<tr>
<td>Mushroom</td>
<td>23</td>
<td>-</td>
<td>8125</td>
<td>2</td>
<td>86.5</td>
<td>88.2</td>
</tr>
<tr>
<td>Tic-tac-toe</td>
<td>9</td>
<td>-</td>
<td>862</td>
<td>9</td>
<td>82.3</td>
<td>92.6</td>
</tr>
<tr>
<td>Glass identification</td>
<td>-</td>
<td>9</td>
<td>214</td>
<td>6</td>
<td>55.7</td>
<td>65.3</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>-</td>
<td>35</td>
<td>351</td>
<td>2</td>
<td>91.5</td>
<td>84.6</td>
</tr>
<tr>
<td>Waveform</td>
<td>-</td>
<td>21</td>
<td>300</td>
<td>3</td>
<td>59.2</td>
<td>73.0</td>
</tr>
<tr>
<td>Pima diabetes</td>
<td>-</td>
<td>8</td>
<td>768</td>
<td>2</td>
<td>68.2</td>
<td>72.7</td>
</tr>
<tr>
<td>Thyroid (new) disease</td>
<td>-</td>
<td>6</td>
<td>215</td>
<td>3</td>
<td>89.3</td>
<td>84.6</td>
</tr>
<tr>
<td>Cleveland heart disease</td>
<td>8</td>
<td>5</td>
<td>303</td>
<td>2</td>
<td>49.2</td>
<td>60.8</td>
</tr>
</tbody>
</table>
The number of attributes, instances, and 'natural' classes of these datasets are given in Table 2. All experiments on these datasets were carried out with 10-fold cross validation by AUTOCLASS (AUTOC) and OSHAM in the same condition (the same datasets randomly divided into subsets). For AUTOC, we used the public version AUTOCLASS-C implemented in C and ran three steps of search, report, and predict with the default parameters. We obtained the predicted name and predictive accuracy of AUTOCLASS and OSHAM. In order to avoid a biased evaluation of OSHAM (although with each dataset, parameters can be adjusted to obtain the most suitable concept hierarchy), we fixed the values $\alpha = 1\%$ of the size of the training set, $\beta = 15\%$ and $\sigma = 10\%$ of the number of attributes, and the beam size $\eta = 3$ in common to all datasets.

The predictive accuracies of OSHAM and AUTOCLASS in these first trials were only slightly different. Each system performs better in different datasets; therefore, these two systems can be considered to have comparable performance. The main advantage of OSHAM is that its concept hierarchies can be more easily understood by its extended classical view on concepts and its graphic support.

4. **Fielding knowledge bases in decision-making**

Among knowledge representation models, the object/rule model, developed by the knowledge-based system generator TESOR (Ho *et al.*, 1992), provides an understandable structure of knowledge that allows final decisions from an initial state to be reached efficiently in orderly stages.

The process of constructing a knowledge base in TESOR is shown in Figure 3 (Ho, 1996). The Graphic Knowledge Editor (GKE) is preferred if the knowledge is already available; for example, in the form of a set of decision rules or an object hierarchy. Otherwise, CABRO can generate decision rules/decision trees from supervised data. It can handle large data tables and noisy data effectively.

OSHAM can generate hierarchy/decision rules from unsupervised data. The output of OSHAM is compiled directly by the TESOR GKE into the internal representation form. The object hierarchy is displayed graphically to enable an expert to give his judgment until a satisfactory hierarchical object knowledge base is obtained.

The visualization of knowledge bases by the TESOR GKE offers the user an intuitive understanding about the structure of the domain knowledge. It helps the expert/knowledge engineer in clustering data, observing the results and quality...
evaluations, modifying parameters, and adjusting results at each level before the system continues further.

Figure 3. Mutual use of CABRO and OSHAM in constructing knowledge bases.

5. A Case Study

In this section we illustrate discovered knowledge from a medical database by two DTI systems, CABRO and C4.5. This is a database on meningoencephalitis collected at the Medical Research Institute, Tokyo Medical and Dental University (Tsumoto, 1999). Each record contains 35 attributes. Their names and values are given in Table 3. Part of the database is shown below. The third field represents the class attribute DIAG with nine values (VIRUS, ABSCESS, BACTERIA, BACTE (E), BACTERI, VIRUS (E), TB(E), CRYPT, and Rubella), which are classified into two groups: 'VIRUS' with the values VIRUS and VIRUS (E), and 'BACTERIA' with the other values.

Table 3. Attributes in the database.

<table>
<thead>
<tr>
<th>Category</th>
<th>Type</th>
<th>Number of attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present history</td>
<td>Numerical and categorical</td>
<td>7</td>
</tr>
<tr>
<td>Physical examination</td>
<td>Numerical and categorical</td>
<td>8</td>
</tr>
<tr>
<td>Laboratory examination</td>
<td>Numerical</td>
<td>11</td>
</tr>
<tr>
<td>Diagnosis</td>
<td>Categorical</td>
<td>1</td>
</tr>
<tr>
<td>Therapy</td>
<td>Categorical</td>
<td>1</td>
</tr>
<tr>
<td>Clinical course</td>
<td>Categorical</td>
<td>4</td>
</tr>
<tr>
<td>Final status</td>
<td>Categorical</td>
<td>1</td>
</tr>
<tr>
<td>Risk factor</td>
<td>Categorical</td>
<td>1</td>
</tr>
</tbody>
</table>

58,M,VIRUS,0,12,11,0,0,0,acute,37.6,2,0,0,0,--,6000,0,0,7,n.p.,a,--,660,86,44,F,--,0,0,244,--
29,F,VIRUS,0,2,4,0,0,0,acute,38.5,4,0,1,15,--,8300,0,0,14,n.p.,a,--,464,88,59,F,--,0,0,423,--
40,M,VIRUS,0,5,3,3,0,0,ACUTE,37.5,0,0,0,0,--,8200,0,0,0,n,p,a,--,208,348,0,F,--,0,0,353,--
We did an experimental comparative evaluation of C4.5 and CABRO. The evaluation was carried out as follows:

- all instances and 35 attributes in the original data set where DIAG is the class attribute were used in order to discover diagnostic knowledge for two classes, 'BACTERIA' and 'VIRUS';
- an automatic random shuffle of the dataset then divided the dataset into 10 stratified subsets; and
- an automatic cross-validation was done for both systems with the same division of the dataset into 10 folds.

This process was repeated many times and the results were found to be stable (varying only slightly with each shuffle of the dataset). Our experiments showed that on the testing data, the accuracy of CABRO is higher than that of C4.5, but the size of pruned trees from C4.5 is smaller than that from CABRO. From small datasets, C4.5 was able to produce very small trees. From many experiments with different datasets, we observed that this difference in tree size does not exist when learning from large datasets (Ho et al., 1998). Below are some simplified diagnostic rules discovered by CABRO:

If THERAPY2 = ope Then BACTERIA (2.0/1.0)
If THERAPY2 = Zobirax Then VIRUS (17.0/1.3)
If THERAPY2 = PIPC+CTX Then BACTERIA (1.0/0.8)
If THERAPY2 = ABPC+LMOX Then BACTERIA (2.0/1.0)
If THERAPY2 = change Then BACTERIA (8.0/2.4)
If THERAPY2 = ABPC+FMOX Then BACTERIA (3.0/1.1)
If THERAPY2 = ARA-A Then VIRUS (8.0/1.3)
If THERAPY2 = Dara-P Then BACTERIA (1.0/0.8)
If THERAPY2 = ABPC+CZX Then BACTERIA (8.0/1.3),

... If THERAPY2 = globulin Then VIRUS (1.0/0.8)
If THERAPY2 = ALA-A Then VIRUS (1.0/0.8)
If THERAPY2 = ABPC+CTX Then BACTERIA (2.0/1.0)
If THERAPY2 = ABPC+CEX Then BACTERIA (1.0/0.8)
If THERAPY2 = :-
| CRP = 0 Then VIRUS (51.0/2.6)
| CRP = 1 Then BACTERIA (4.0/2.2)
| CRP = 2 Then VIRUS (0.0)
| CRP = 3 Then VIRUS (0.0).

Conclusion

As decision-making is a process of choosing among alternative courses of action for the purpose of attaining a goal, knowledge about these choices plays a crucial role in
Rule Induction and Knowledge-based DSS

any decision. Finding knowledge in data is the ultimate goal of rule induction, a sub-field of ML and KA that can be used effectively in decision-making theory and practice.

In this chapter, we introduced the main stages of the rule-induction process that the DSS user/developer needs to know when finding automatic decision knowledge in data. We also presented two rule-induction methods for supervised and unsupervised data. These methods have been implemented successfully in the CABRO and OSHAM systems and have several significant advantages in comparison with other systems doing the same tasks. We have demonstrated how to use CABRO and OSHAM to construct knowledge bases within the knowledge-based system generator TESOR. It is possible that the user may consider rules/concept hierarchies induced by CABRO and OSHAM as available knowledge in many other KBS/expert system tools. Since finding decision rules is one of the most difficult and expensive tasks in the development of KBDSSs, we believe that rule induction techniques in general, and CABRO and OSHAM in particular, will make a significant contribution to DSSs for sustainable development.

Acknowledgements

The author would like to thank professors Masayuki Kimura and Setsuo Ohsuga for their support, and Nguyen Trong Dung, the codeveloper of CABRO. This work has been supported by Kokusai Electric Co., Ltd., Japan Advanced Institute of Science and Technology (Japan), and the National Research Programme on Information Technology KC01-RDO8 (Vietnam).

References


