Available online at www.sciencedirect.com
SCIENCE DIRECT®
European Journal of Operational Research 160 (2005) 638–662
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/dsw

A list-based compact representation for large decision tables management

J.A. Fernández del Pozo [a,*], C. Bielza [a], M. Gómez [b]

[a] *Decision Analysis Group, Artificial Intelligence Department, Technical University of Madrid, Campus de Montegancedo, Boadilla del Monte, 28660 Madrid, Spain*

[b] *Computer Science and Artificial Intelligence Department, University of Granada, Daniel Saucedo Aranda, 18071 Granada, Spain*

Received 23 January 2002; accepted 15 June 2003
Available online 15 December 2003

Abstract

Due to the huge size of the tables we manage when dealing with real decision-making problems under uncertainty, we propose turning them into minimum storage space multidimensional matrices. The process involves searching for the best order of the matrix dimensions, which is a NP-hard problem. Moreover, during the search, the computation of the new storage space that each order requires and copying the table with respect to the new order may be too time consuming or even intractable if we want a process to work in a reasonable time on an ordinary PC. In this paper, we provide efficient heuristics to solve all these problems. The optimal table includes the same knowledge as the original table, but it is compacted, which is very valuable for knowledge retrieval, learning and expert reasoning explanation purposes.

Keywords: Combinatorial optimisation; Decision analysis; Decision support systems; Heuristics; Learning and explanation

1. Introduction

The Decision Support Systems (DSS) now in demand are very complex knowledge-based systems. Based on the Decision Analysis discipline, see, e.g., Raiffa (1968), their construction involves structuring the decision-making problem (using modern graphical models like influence diagrams, see Shachter, 1986), eliciting uncertainty and preferences (using probability and utility models, see Keeney and Raiffa, 1993), and solving the problem. Although all these tasks are difficult, we shall go one step further.

Once we have solved the problem, we have one decision table per decision variable, containing its optimal alternatives, i.e. the alternatives of maximum expected utility. In general, a decision table can be considered as a set of attributes or variables that determine an action, alternative or policy. The table grows exponentially with the number of attributes. In real problems, each table may have up

* Corresponding author. Tel.: +34-91-3367433; fax: +34-91-3524819.
E-mail addresses: jafernandez@fi.upm.es (J.A. Fernández del Pozo), mcbielza@fi.upm.es (C. Bielza), mgomez@decsai.ugr.es (M. Gómez).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2003.10.005

to millions of rows (each attribute configuration) and typically more than twenty columns (attributes), the results of the problem thereby relying on a combinatorial knowledge representation space.

Storing and managing so much information is not the only problem that arises. Decision-makers use the decision tables to query which is the best recommendation for a certain case or attribute configuration. Indeed, these tables are very important for this purpose. However, decision-makers demand DSSs that provide clear, concise, consistent and complete explanations that translate the reasoning mechanism (underlying the table content) into their domain to justify the decisions proposed. The explanation should give a description of why the proposed decision is optimal and new insights into the problem solution. This kind of knowledge synthesis will also serve for validating the system.

In fact, a system providing good explanations is very hard to build (Henrion et al., 1991). The main reasons are: explanations should be presented from all the possible points of view in a structured and hierarchical way, with different levels for users and analysts; they should only employ knowledge from the user domain; they should be as general as possible and emphasise the evidence of the presence (absence) of arguments in favour of (against) the proposal.

Despite these difficulties, in this paper we show how they can be addressed, resulting in useful systems for real problems.

So-called *learning from data* is a general goal pursued by a number of disciplines for extracting important patterns and trends and understanding what the data say. Therefore, it is easy to imagine that our proposal in this paper may bear some resemblance to some such techniques. For example, the aim of (supervised) classification from Machine Learning is to learn a mapping from a vector of attributes to a class variable. In our case, this variable is the optimal alternative from the decision-making problem. Tree-based classifiers, such as CART (Breiman et al., 1993), ID3 (Quinlan, 1986), C4.5 (Quinlan, 1993), have proved to be particularly useful with non-metric data and without prior information about the appropriate form of classifier (Duda et al., 2001).

However, our approach is quite different. Our method is based on a list-based structure rather than on a tree-based structure. The search for good candidates is global, involving the whole attribute set, while trees use a greedy local search over structures. Avoiding the hierarchism of the tree-building process, we overcome the typical instability found in the trees with regard to small changes (see Hastie et al., 2001). On the other hand, tree-classifiers are very flexible and can be used with every data type (metric, non-metric, or in combination). Our methodology will be limited to finite data.

In this paper, the central idea is that the table content is not knowledge unless it is organised somehow, like a torn book is knowledge only when it has been properly stuck together and repaired. Unlike classification trees, our list does not have to be built, it has to be reorganised. Trees aim at maximising a score of class purity, which does not make any sense for our "classifier", because it does not yield misclassifications. All the cases are already correctly classified, and we want to explain why they are classified like this. We translate this problem into finding the shortest list. In a sense, we work with full trees, which are later "pruned" when we find sets of cases that share certain information values (see Section 2.3 below), providing just the sought-after explanations.

Also, the tables were originally collected for a purpose other than the explanations we seek, i.e. they are the influence diagram evaluation output. Thus, data were not collected using efficient strategies to answer specific questions; they are observational data as opposed to experimental data. Moreover, the rows of our tables range over all the attribute configurations. It means that they cannot be repeated, obviously not being the classical carefully selected laboratory training sample found in Machine Learning and Statistics. Hence, missing values arise only in the class variable, unlike tree-based methods which have missing values in the attributes. This occurs when the decision tables only include a subset of the whole problem solution due to computational problems, leading to unknown policies. We will see below how to deal with these values.

This discussion suggests setting our framework in the Data Mining field, see, e.g., Hand et al. (2001).