Decision Support Systems 51 (2011) 141–154

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models

Johan Huysmans [a], Karel Dejaeger [a], Christophe Mues [b], Jan Vanthienen [a], Bart Baesens [a,b,c,*]

[a] Department of Decision Sciences and Information Management, Katholieke Universiteit Leuven, Naamsestraat 69, B-3000 Leuven, Belgium
[b] School of Management, University of Southampton, Southampton, SO17 1BJ, United Kingdom
[c] Vlerick Leuven Gent Management School, Leuven, Belgium

ARTICLE INFO

Article history:
Received 26 November 2008
Received in revised form 29 October 2010
Accepted 5 December 2010
Available online 10 December 2010

Keywords:
Data mining
Classification
Knowledge representation
Comprehensibility
Decision tables

ABSTRACT

An important objective of data mining is the development of predictive models. Based on a number of observations, a model is constructed that allows the analysts to provide classifications or predictions for new observations. Currently, most research focuses on improving the accuracy or precision of these models and comparatively little research has been undertaken to increase their comprehensibility to the analyst or end-user. This is mainly due to the subjective nature of 'comprehensibility', which depends on many factors outside the model, such as the user's experience and his/her prior knowledge. Despite this influence of the observer, some representation formats are generally considered to be more easily interpretable than others. In this paper, an empirical study is presented which investigates the suitability of a number of alternative representation formats for classification when interpretability is a key requirement. The formats under consideration are decision tables, (binary) decision trees, propositional rules, and oblique rules. An end-user experiment was designed to test the accuracy, response time, and answer confidence for a set of problem-solving tasks involving the former representations. Analysis of the results reveals that decision tables perform significantly better on all three criteria, while post-test voting also reveals a clear preference of users for decision tables in terms of ease of use.

1. Introduction

Predictive models are widely used in both research and business applications. For example, based on past applications, financial institutions construct credit scoring models to predict whether an applicant for a loan will be able to pay back the loan or will default on his/her loan obligations. The model is then used to decide which new credit applications should be granted or denied [2]. Similarly, insurance companies develop predictive models to identify the claims that are likely to be fraudulent [60], and in the medical sciences predictive techniques may be used to decide whether a particular medical condition is malign or benign.

In some of these applications, selection of the best predictive model is based solely on the ability to provide correct predictions for previously unseen examples. In other situations though, the interpretability of the model is equally important, i.e. one must be able to understand how the model reaches a particular decision. For example, in the domain of credit scoring, financial institutions often face the legal obligation of being able to motivate why a certain customer was denied credit [17]. Also, in the medical sciences, understanding how the model comes to its conclusions is often crucial, as it provides information about the variables that influence the disease and can therefore point to a potential cure or prevention strategy. An important criterion for selecting a model from a series of candidate models with similar performance is that it is in line with previous domain knowledge (see e.g. [45] for an overview of the literature on incorporating domain knowledge into data mining). This is especially true in many business settings, in particular where models support decision making or form the basis of policy development, and when there is a risk that sampling bias might cause the introduction of counter-intuitive relationships (e.g. [18]). Hence, there is often a trade-off between the predictive accuracy of powerful but essentially black box models such as neural networks or support vector machines and the good interpretability of other types of representations that may facilitate validation by an analyst or domain expert.

For these reasons, rule induction algorithms, which return a set of 'if–then' rules, or decision tree learners are often the preferred choice as they should offer the required level of interpretability. Alternatively, rule extraction [1,24,33] can be performed on black box models, such as neural networks and support vector machines, to extract a set of rules that approximate the black box as closely as

* Corresponding author. Department of Decision Sciences and Information Management, Katholieke Universiteit Leuven, Naamsestraat 69, B-3000 Leuven, Belgium. Tel.: +32 16 32 68 84; fax: +32 16 32 66 24.
E-mail addresses: Karel.Dejaeger@econ.kuleuven.be (K. Dejaeger), C.Mues@soton.ac.uk (C. Mues), Jan.Vanthienen@econ.kuleuven.be (J. Vanthienen), Bart.Baesens@econ.kuleuven.be (B. Baesens).

0167-9236/$ – see front matter
doi:10.1016/j.dss.2010.12.003

possible and at the same time provide a more understandable representation to the users. However, previous research concerning rule extraction techniques [25] indicated that some algorithms return models that closely approximate the underlying black box model, but at the cost of being very complex. For example, as reported in [27], the algorithms G-REX and CART return trees with an average of respectively 22 and 91 nodes when applied to some real-life data sets. Zhou et al. [63] applied the REFNE rule extraction algorithm and found the average number of rules to be 31.[1] It can be feared that such models might fail in their primary task of providing insight in the black box model from which they are extracted, as the black box is only replaced by a myriad of rules.

It was observed by Pazzani [35] that few papers actually aim to empirically assess comprehensibility beyond simply reporting the size of the resulting representations. Moreover, Pazzani notes that there is little understanding of the factors influencing comprehensibility (with even the effect of size remaining unexplored), and that there have been no attempts to show e.g. that users actually prefer certain visualizations over mere textual representations. Hence, he argues that data mining can benefit considerably from the interaction with cognitive science to increase the usefulness of knowledge discovery and the user's acceptance of the models obtained. Similarly, Freitas [19] reviews some recent concepts and approaches for discovering not just accurate but also comprehensible and/or 'interesting' (i.e. novel or surprising) knowledge from data. However, he echoes Pazzani's observation that there is no agreement on which of these representation types (e.g. if–then rules, decision trees, etc.) is the most comprehensible in general, and that there seems to be no study actually comparing their comprehensibility from the point of view of human users.

In this paper, we will present the results of an experiment that compares the impact of several representation formats on the aspect of comprehensibility. More specifically, an end-user experiment was set up to test the accuracy, response time, and answer confidence for a set of problem-solving tasks involving these representations. The experiment is run in a credit scoring context, which involves the use of predictive models for the task of assessing whether a loan applicant is likely to pay back or default on this loan and should therefore be accepted or rejected. The following formats are considered: decision tables, (binary) decision trees, propositional rules and oblique rules. Besides comparing these different representation formats, our study also investigates the influence of the size of each of these representations on their interpretability. A better understanding of this relation would provide an indication whether a model could be useful in practice as an explanation or validation aid, or should be avoided because it is too confusing for end-users or analysts and has little added value over a black box model.

In the next section, we describe in detail the four representation formats covered in the experiment. In Section 3, the theory underpinning the experiment is discussed and based on this theory, a series of research propositions are formulated. Section 4 discusses the empirical setup of the experiment while Section 5 presents the findings. Section 6 provides a discussion of the obtained results. Finally, the paper concludes with the key findings and interesting topics for future research.

2. Rule representation

In this section, an overview of the representation formats selected is provided. Given the wide range of representation schemes proposed in the literature, and their numerous variations, inevitably a selection of representation formats was required. Based on the ease of use for novice users, inclusion in prior studies, and prominence in the machine learning/data mining literature, we opted for the following representation formats.

Firstly, the most common type of rules is without any doubt *propositional if–then rules*. The condition part of a propositional rule consists of a combination of conditions on the input variables. While the condition part can contain conjunctions, disjunctions, and negations, most algorithms will return rules that only contain conjunctions.

Most algorithms will ensure that the condition parts of each rule demarcate separate areas in the input space: i.e., the rules are mutually exclusive. Therefore, only one rule is satisfied when a new observation is presented and that rule will be the only one used for making the classification decision. Other algorithms allow multiple rules to fire for the same observation. This requires an additional mechanism to combine the predictions of individual rules, such as assigning a confidence factor to each rule [9] or sorting the rules and allowing only the first firing rule to decide [50]. For this paper, it is assumed that all rules are mutually exclusive and therefore these mechanisms are not required.

Various formats can be used to represent propositional rules. The most straightforward approach is to simply write the rules down, as in the following example:

IF (INCOME > 400 AND GOAL = CAR) THEN ACCEPT
IF (INCOME > 900 AND GOAL = HOUSE) THEN ACCEPT
DEFAULT:REJECT

This fictitious example shows the credit policy of a financial institution which may be used to decide on loan applications. Based on this policy, the credit manager would accept all applications where the applicant has an income above 900 and the goal is the purchase of a house or where the income is above 400 and the goal is the purchase of a car. If these conditions are not satisfied, the default rule specifies that applications are to be rejected.

Other more graphical-oriented representations that are frequently used to depict conditional logic are decision tables and decision trees. A *decision table* [55] is a tabular representation that consists of four quadrants separated by horizontal and vertical double lines (see Fig. 1).

The horizontal line divides the table into a condition part (top) and an action part (bottom), whereas the vertical line separates subjects (left) from entries (right). Every column in the entry part corresponds to a rule, combining condition states with the appropriate action(s) to take. A dash symbol (-) in the condition part of the table indicates that the value is irrelevant in that condition and an "X" in the action part represents the correct conclusion to make if the conditions leading to that column are satisfied.

(a) Single-hit table

<table>
<tr><td>INCOME</td><td colspan="2">< 1000</td><td>≥ 1000</td></tr>
<tr><td>AGE</td><td>< 25</td><td>≥ 25</td><td>-</td></tr>
<tr><td>ACCEPT</td><td>X</td><td></td><td></td></tr>
<tr><td>REJECT</td><td></td><td>X</td><td>X</td></tr>
</table>

(b) Multiple-hit table

INCOME	≥ 1000	-	< 1000
AGE	-	< 25	< 25
ACCEPT			X
REJECT	X	X	

Fig. 1. Example decision tables.

[1] These techniques are only mentioned because the authors included complexity information in their papers; the above should not be considered as a criticism of these specific algorithms.