An Experiment in Discovering Association Rules in the Legal Domain

Trevor Bench-Capon, Frans Coenen and Paul Leng
LIAL - Legal Informatics at Liverpool
Department of Computer Science
The University of Liverpool
Liverpool
UK

Abstract

In this paper we explore the applicability of an algorithm designed for finding association rules in large databases to the discovery of relevant associations from a large case base.

1. Introduction

In this paper we describe an experiment intended to explore the applicability of data mining techniques to legal databases. In many areas of law - especially administrative law - many thousands of cases are decided. This data presents a significant resource. We would generally wish to assume that some rule is being followed so that like cases are decided in a like manner. Is there a way of deciding what the rule being followed is from an automated consideration of the data?

Such a question has relevance to a number of interesting and important issues:

- If there is well defined legislation which defines what the rule should be we may wish to ensure that the rule is being followed;
- If the domain is a discretionary one, we may wish to discover the rule itself;
- Some people have argued that the rule followed in practice is different from the rule which exists in theory (or which might be elicited from experts). For example Edwards (1995) suggests that some areas of law exhibit a systematic bias. Such conjectures could be informed and justified were we able to discover the "real" rule from a database recording the actual practice.

If therefore we had a reliable technique to extract a rule explaining the data in a field of law, it would have many interesting uses. Extraction of such knowledge from large databases is currently the subject of considerable research, particularly in such application areas as the targeting of mail-shots. We will discuss one such technique, designed to produce *association rules*, in section 2.

One problem with many experiments to explore the efficacy of techniques designed to extract knowledge from data is that they use data for which the relationships present are not known at the outset. As a result, what has been discovered and what has been missed, cannot be established definitively. In the work reported here we will use a specially constructed data set, the properties of which are known, and which are thus able to serve as a measurable test of the technique.

The data we use has been used in a previous AI and Law experiment, reported in Bench-Capon (1993), and intended to evaluate the application of artificial neural networks to this problem. The use of this same data set is interesting because it allows for comparison between what can be derived using the association rules technique with what can be discovered using a neural network.

1.1. The Data Set

As previously mentioned the data set used in these experiments is that used in Bench-Capon (1993). The data concerns a fictional welfare benefit. The benefit is supposed to be payable on six conditions. These conditions were chosen to represent different kinds of condition that are found in the legal domain, so that we can see whether the different form of conditions affects their discoverability.

0-7695-0680-1/00 $10.00 © 2000 IEEE

The notional benefit was a fictional welfare benefit paid to pensioners to defray expenses for visiting a spouse in hospital. The conditions were:

1) The person should be of pensionable age (60 for a woman, 65 for a man);
2) The person should have paid contributions in four out of the last five relevant contribution years;
3) The person should be a spouse of the patient;
4) The person should not be absent from the UK;
5) The person should have capital resources not amounting to more than 3,000;
6) If the relative is an in-patient the hospital should be within a certain distance: if an out-patient, beyond that distance.

These conditions represent a range of typical condition types: 3 and 4 are Boolean necessary conditions, one which should be true and one false; 5 is a threshold on a continuous variable representing a necessary condition, and 2 relates five variables, not all of which need be satisfied. 1 and 6 are more interesting since the relevance of a variable depends on the value of another: in 1 sex is relevant only for ages between 60 and 65, and in 6 the effect of the distance variable depends on the Boolean saying whether the patient is an in-patient or an out-patient.

The data was generated using a program written in Common LISP. For this experiment a data set of 1200 records was used: 600 satisfying all of the conditions, and equal numbers of the remainder being designed to fail each of the others. For records designed to fail one of the conditions, satisfaction or otherwise of the remaining conditions was decided randomly for each condition separately.

1.2 The experiment

Using the data described above we applied our algorithm for the extraction of association rules. We deliberately applied the algorithm with as little adaptation as possible, and trying to do no more that the minimum pre-processing. Section 2 gives an overview of association rules; section 3 a description of our algorithm; section 4 describes the experiment and its results and section 5 provides some discussion.

2. Knowledge Discovery by Association Rules

Knowledge Discovery in Databases (KDD) is concerned with the identification of hitherto unrecognised and 'interesting' information in (usually large) databases. Almost always, what is being sought is some relationship which can be observed between categories of information in the data. A particular way to describe such a relationship is in the form of an *association rule* which relates attributes of the data. Methods of deriving association rules were developed, in particular, to analyse data collected in supermarket shopping-basket transactions. Here, the aim is to discover rules of the form "shoppers who purchase items *A*, *B* and *C* are also likely to purchase items *X* and *Y* at the same time".

In the simplest case, assume a set *I* of *n* boolean attributes, $\{a_1,\ldots,a_n\}$, and a database of *m* records, each of which records the presence or absence of each attribute in a single instance, or *itemset.* An association rule *R* is of the form $A \rightarrow B$, where *A, B* are disjoint subsets of the attribute set *I*. The *support* for *R* is the number (or proportion) of itemsets in which the association between *A* and *B* can be observed. The *confidence* in the rule is the ratio of the support for *R* to the support for its antecedent, *A*. These two properties provide a measure of 'interest' in a rule. Usually, a rule will be thought to be interesting if its support exceeds some threshold value (implying a sufficient number of instances of the association to enable conclusions to be drawn) and its confidence is sufficiently high for us to conclude that *A* 'usually' implies *B*. Although the method is described in respect of a database of binary attributes, it will often be possible to convert more complex data into a suitable form for this analysis to be conducted.

Because the confidence in a rule can be determined immediately once the support for a rule and its antecedent are known, methods for discovery of association rules have concentrated particularly on determination of support. The best known algorithm, "Apriori" (Agrawal et al, 1993), proceeds by repeated passes of the database. In the first pass, the support for all single attributes is counted, and from this is derived a "candidate set" of *pairs* of attributes which may be of interest because each attribute of the pair exceeds the required threshold of support. Subsequent passes apply the same process to compute support for pairs, triples, etc., until no further supported sets can be found.

In work described elsewhere (Goulbourne, Coenen and Leng, 1999), we have developed a novel technique which requires only a single pass of the source database. In this pass, a tree is constructed within which are stored *incomplete* counts of support for sets encountered in the database. Further passes applied to this tree complete the computation of support for the sets identified as being of interest.

In the work described here, we apply this method to the identification of rules within the dataset outlined above. Two aspects of this are worthy of note. Firstly, because we are only interested in potential rules which lead to our target attributes, we can restrict our search to sets which include at least one of these. The second point is of greater general interest. As we have explained, association rule algorithms were developed especially to deal with supermarket shopping-basket analysis, in which the rules inferred are essentially *probabilistic*. Here we are trying to apply the method to a domain in which, at least initially, it appears that rules must be definitive: that is, they describe necessary and sufficient conditions for their consequents. An aim of our work is to understand whether association rule methods can have an application in this context.

3. The Association Rule Algorithm

The association rule generation algorithm used comprises three phases. During the first phase the data set is read into a tree structure (the P-tree) using one pass of the database. Each node in the P-tree represents either a row in the data set or a "dummy" node, which has been included to preserve the structure of the tree. The P-tree is generated dynamically as each row in the data set is read. In the worst case each row (record) in the data set will create a new node in the tree, and possibly a dummy node also. In practice, however, most data sets contain a significant number of duplicate records and consequently the size of the P-tree is usually less than *m*.

Once the P-tree is complete we progress to the generation of total supports. These are stored in a second tree, the T-tree, which is produced level by level by a method similar to that of the Apriori algorithm outlined above. We commence with the top level of the tree, which comprises nodes each representing a one element set, and calculate the total support for each node using one pass of the P-tree. If at this stage any element is not adequately supported it can be removed from the T-tree. The remaining elements are then combined to produce a second level of "candidate nodes" (each comprising 2 elements), and the process is repeated until all supported sets have been identified and stored in the T-tree structure. The process can be summarised as follows.

1) Generate next level in T-tree from previous level such that for each node in the new level all the subsets of its elements are already in the T-tree.
2) Determine total supports for new level.
3) Prune new level by removing any nodes that are not adequately supported.
4) If all the nodes in the new level are not supported exit, otherwise repeat.

In the last phase of the algorithm, the association rules are generated. For each node in the T-tree, other than the top level nodes, we produce the set of all possible antecedent/consequent pairings using all the elements represented by the node. Each pairing represents a candidate association rule, for which the confidence level is calculated, and if this is above the confidence threshold the association rule is generated.

4. Set Up of the Experiment

The data used in Bench-Capon (1993) contains both real and Boolean data. The algorithm, on the other hand accepts only Boolean data. It was therefore necessary to begin by pre-processing the data to express it uniformly as Boolean. This was achieved by mapping the reals into Booleans. Where a threshold was known this was simply done by mapping numbers above the threshold to 1 and numbers below to 0. In the case of *age*, however, two values are significant: it is important to identify those between 60 and 65 also. Age was, therefore, mapped into three attributes: under 60, between 60 and 65, and over 65.

To achieve such a mapping we need to use our knowledge of the domain. In some of the motivating applications, where we have a conjecture which we wish to establish, we can assume that such knowledge is present. In the absence of such knowledge, where we were trying to discover such thresholds, it would be necessary to map reals into a larger number of Boolean attributes, intended to capture any threshold that might exist in the data. For age, for example we could choose a separate attribute for all possible five year age bands.

This is the technique employed in the Split-UP system (Zeleznikow and Stranieri, 1997).

After preprocessing, the test data set used comprised 1200 rows (*m*) and 17 columns (*n*), for the 17 attributes to be considered, the first two of which were the target attributes, representing qualification or non-qualification for the benefit. The schema for the data set was as follows:

1) Is qualified
2) Is not qualified
3) Has age < 60
4) Has age >= 60 <= 65
5) Has age > 65
6) Has sex male
7) Has sex female
8) Has contribution 1
9) Has contribution 2
10) Has contribution 3
11) Has contribution 4
12) Has contribution 5
13) Has spouse
14) Is absent
15) Has capital < 0.3
16) Lives distance >.5
17) Is in-patient

The resulting P-tree comprised 448 nodes (significantly less than *m* because of the number of duplicate records contained in the data set). The support threshold was set to 0.75%, and the confidence threshold to 75%. Space precludes discussion of all the rules generated, but we will discuss some of the more interesting ones below.

We have six conditions for the receipt of the benefit, all of which must be satisfied. Therefore we would hope to find rules associating failure to satisfy each of the conditions with attribute 2, as well as some associations of satisfied conditions with attribute 1. There is a problem, however, that the algorithm is designed to associate the *presence* of attributes. Thus where an attribute being 1 is a necessary condition for receipt, and hence an attribute being 0 is a sufficient condition for non-receipt, the association will not be found.

In fact the following necessary conditions (i.e. with attribute 2 being 1) were found:

```
(3) -> (2) confidence = (91/91*100) =
100%
 (16,17) -> (2) confidence =
(46/46*100) = 100%
 (4,6,8,16) -> (2) confidence =
(10/13*100) = 76%
 (4,6,8,10,16) -> (2) confidence =
(10/13*100) = 76%
 (4,6,8,14,16) -> (2) confidence =
(10/13*100) = 76%
 (4,6,8,10,14,16) -> (2) confidence =
(10/13*100) = 76%
 (4,6,8,13,16) -> (2) confidence =
(9/12*100) = 75%
 (4,6,8,10,13,16) -> (2) confidence =
(9/12*100) = 75%
 (4,6,8,13,14,16) -> (2) confidence =
(9/12*100) = 75%
 (4,6,8,10,13,14,16) -> (2) confidence
= (9/12*100) = 75%
 (4,6,8,15,16) -> (2) confidence =
(9/12*100) = 75%
 (4,6,8,10,15,16) -> (2) confidence =
(9/12*100) = 75%
 (4,6,8,14,15,16) -> (2) confidence =
(9/12*100) = 75%
 (4,6,8,10,14,15,16) -> (2) confidence
= (9/12*100) = 75%
```

Attribute 3, age < 60 is shown to be sufficient to determine that the benefit should not be received, as is one combination of attributes 16 and 17, which relate to the sixth condition. All other rules are at a lower confidence level, and contain the attributes 4 and 6 (men between 60 and 65). The lower confidence here is explained by the deliberate mischoice of the threshold for attribute 4: men of exactly 65 do qualify, and there are exactly 3 such examples in the data set. The other conditions are not identified, since the sufficient condition for non-receipt is that some attribute or attributes are 0.

The sufficient conditions for receipt are:

- Either 5 or 4 and 7;
- Four out of 8,9,10,11, and 12;
- 13, 14, 15,
- No more that one of 16 and 17.

The associations discovered, with confidence of 100%) were

```
 (4,7,8,9,11,12,13,14,15) -> (1)
confidence = (25/25*100) = 100%
 (4,7,8,9,11,13,14,15,16) -> (1)
confidence = (20/20*100) = 100%
 (4,8,9,11,12,13,14,15,16) -> (1)
confidence = (13/13*100) = 100%
 (4,7,8,9,11,12,13,14,15,16) -> (1)
confidence = (13/13*100) = 100%
 (4,7,8,9,11,12,13,14,15,17) -> (1)
confidence = (12/12*100) = 100%
```

All of these are valid, but surprisingly all relate to women between 60 and 65. The association with age over 65, appears only in rules of a lower confidence. This is an explicable consequence of the difficulty in dealing with the sixth condition. Consider the following six rules (relating to one way of satisfying the second condition) discovered by the system:

```
(5,8,9,10,12,13,14,15,16) ->  (1)
confidence = (43/45*100) = 95%
 (5,6,8,9,10,12,13,14,15,16) ->  (1)
confidence = (21/22*100) = 95%
 (5,7,8,9,10,12,13,14,15,16) ->  (1)
confidence = (22/23*100) = 95%
(5,8,9,10,12,13,14,15,17) ->  (1)
confidence = (46/48*100) = 95%
 (5,6,8,9,10,12,13,14,15,17) ->  (1)
confidence = (23/24*100) = 95%
 (5,7,8,9,10,12,13,14,15,17) ->  (1)
confidence = (23/24*100) = 95%
```

To qualify for the benefit just one of 16 and 17 must be 1. In 2 of the 45 cases (one male and one female) satisfying a particular set of contribution conditions, the claimant fails solely because both 16 and 17 are true. We thus get this set of rules with confidence 95%. Again, were it possible to create a rule including a 0 attribute as part of the condition, this rule would have 100% support .

5. Discussion

The experiment raised a number of points concerning the application of the technique.

- First consider the use of real numbers. The association rules produced depend on this, but one encouraging result is that the choice of threshold need not be exact. In the experiment we chose >= 65 rather than the correct >65. The result was that the algorithm generated a number of rules, with lower confidence, showing the existence of exceptions centred on the combination of attributes defining men between 60 and 65. Examination of the original data relating to these specific exceptional cases revealed the true threshold.
- That the algorithm attempts to associate only co-presence is a problem. This could be solved by a different method of pre-processing by which Boolean attributes in the data are re-represented as pair of attributes, one indicating satisfaction of the original attribute, and other indicating non-satisfaction of that attribute. This would have allowed the other necessary conditions to have been identified.
- The sixth condition in the example presented particular problems, in that it effectively represents an X-or of two attributes. Again this is indicated by the generation of large number of high confidence associations grouping similar attributes. Again inspection of the failing cases can reveal the true association.

Overall we believe that the experiment is indicative of the potential to find associations in legal-type data, although the data does require some structuring and pre-processing. If the aim is to confirm particular hypotheses about the data, the information necessary to provide this structure is available.

Compared with the original experiment using neural networks, the advantage of association rules is that the output comes in the form of readily understandable rules, which can be examined, evaluated and refined, whereas the extraction of such useful regularities from a neural net is more problematic. On the other hand, the neural net handles reals and 0 valued attributes, and so trained more accurately with less pre-processing.

In conclusion we would say that the extraction of association rules from large data sets can provide a useful addition to the tools that can be used by those who wish to extract knowledge from large sets of legal data, and that the legal field should continue to monitor developments in what is currently a very active area in computer science.

Acknowledgement

We gratefully acknowledge the significant contribution of Graham Goulbourne to the work on the association rule algorithm.

References

Agrawal, R., and Srikant, R., (1994). *Fast Algorithms for Mining Association Rules,* in Proceedings of 20th VLDB Conference, Santiago, pp487-499

Bench-Capon, T.J.M., (1993). *Neural Networks and Open Texture.* In Proceedings of the Fourth International Conference on AI and Law; ACM Press, New York.

Edwards, L., (1995). Modelling Law Using a Feminist Theoretical Perspective. *Law, Computers and Artificial Intelligence,* Vol 4, pp95-110

Goulbourne, G., Coenen, F., and Leng, P., (1999). *Algorithms for Computing Association Rules using a Partial-Support Tree,* in M Bramer, A MacIntosh and F Coenen (eds) *Research and Development in Intelligent Systems XVI,* Springer, London, pp132-147.

Zeleznikow, J., and Stranieri, A., *Knowledge Discovery in the Split Up Project,* in Proceedings of the Sixth International Conference on AI and Law, ACM Press, New York. pp 89-97.-+