Analysing Binary Associations

Arno J. Knobbe, Pieter W. Adriaans
Syllogic
P.O. Box 26, 3990 DA Houten, The Netherlands
Email: {ajknobbe, pieter}@syllogic.nl

Abstract

This paper describes how binary associations in databases of items can be organised and clustered. Two similarity measures are presented that can be used to generate a weighted graph of associations. Each measure focuses on different kinds of regularities in the database. By calculating a Minimum Spanning Tree on the graph of associations, the most significant associations can be discovered and easily visualised, allowing easy understanding of existing relations. By deleting the least interesting associations from the computed tree, the attributes can be clustered.

Introduction

In this paper we investigate new ways of discovering and presenting associations discovered in databases of items. The discovered knowledge will be represented in the form of clusters of items and graphs of relationships between items, and not in the form of rules, which is the traditional form of knowledge representation used for associations (Agrawal et al. 1996, Toivonen et al. 1995). Our particular focus will facilitate easy understanding and interpretation of the discovered associations.

A general problem of discovery algorithms and in particular those that discover association rules, is the great amount of rules that are discovered. By examining rules between sets of items, the emphasis is on completeness rather than on readability. Some solution to this problem have been proposed, such as allowing the user to specify rules of interest, or clustering rules into groups of related structures (Toivonen et al. 1995). Still the user was required to examine lists of rules by hand.

In this paper we focus on associations between single attributes, thus reducing the number of hypothetical associations. We claim that this restricted analysis will discover most of the interesting knowledge contained in the database, while greatly increasing the readability and usefulness of the resulting structure. Experiments show that our focus on simple associations produces acceptable results, and that knowledge expressed in complex rules is, at least to some extent, represented by simple associations, which is a result of the transitivity of simple rules.

An analysis of simple associations produces an association matrix that can easily be visualised in a bar diagram, or in a graph with an association measure with each edge. Still with larger amounts of attributes, such a bar diagram or graph will become too complex and cluttered. We solve this problem by simplifying the association graph by calculating a minimum spanning tree (Cormen & Leiserson 1989, Preparata & Shamos 1985, Prim 1957, Tarjan 1983). The resulting tree can be used to cluster the attributes (Gower & Ross 1969, Preparata & Shamos 1985).

This paper is organised as follows. The following section describes two similarity measures that can be used to generate an association matrix. We compare the properties of each of these similarity measures. The next section describes how minimum spanning trees can be used to reduce the set of associations. This graph can be used to cluster the attributes. Finally we present some experimental results that demonstrate the usefulness of our approach, followed by some conclusions.

Simple association

In this section we focus on finding associations between pairs of attributes. Measures of similarity (amount of association) are calculated by making passes of the database and counting occurrences of 1's for the pair of attributes. Different similarity measures can be thought of, each expressing a particular type of association. We will present two similarity measures, one based on Shannon's information theory (Shannon & Weaver 1949, Li & Vitányi 1993), and one based on conditional probabilities. Other measures can be thought of, but in order to calculate a minimum spanning tree in a later stage, we require all similarity measures to be symmetric in the two associated attributes.

Our similarity measure will be a symmetric function $S(x, y)$ of two attributes x and y, that is calculated from values $p_x(i)$, $p_y(i)$ and $p_{xy}(i, j)$, where i and j may take on the values 0 and 1. $p_x(i)$ is taken to be

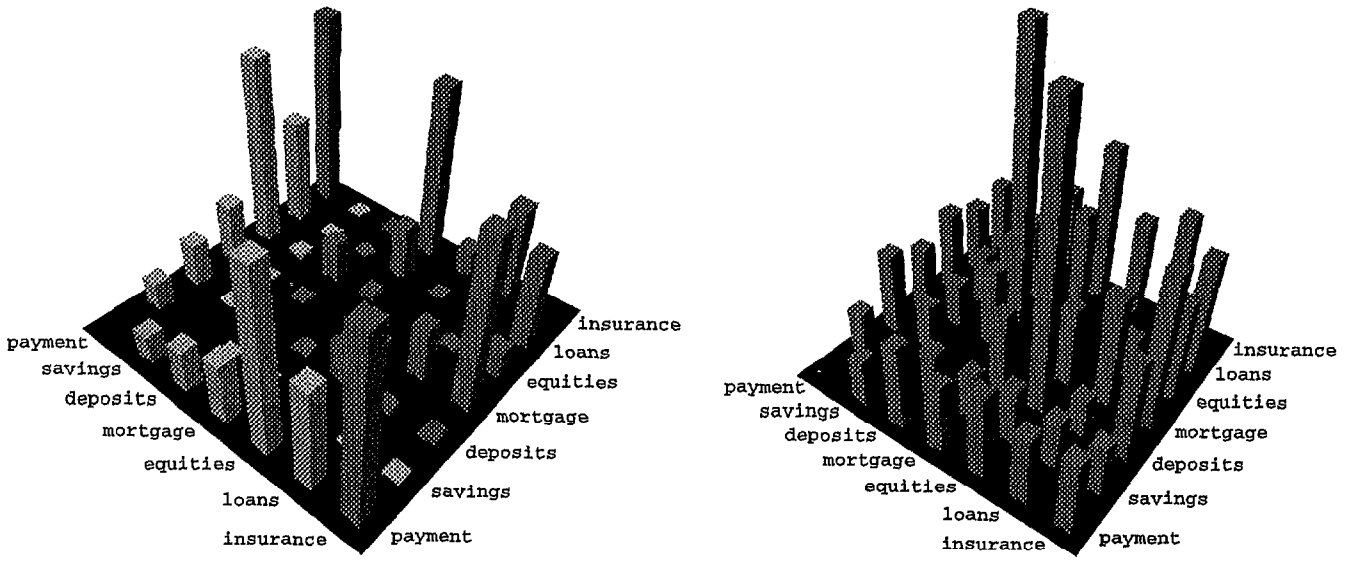

Figure 1: Association matrix for the bank database using $I(x, y)$, and $P(x, y)$.

an estimation of the probability of attribute x being i, and is defined as the number of times x has the value i, divided by the total number of records in the database. Similarly for $p_y(i)$ and $p_{xy}(i, j)$.

The first similarity measure is based on information theory, and is commonly known as *mutual information*. It is defined as

$$I(x, y) = \sum_{i=0}^{1} \sum_{j=0}^{1} p_{xy}(i, j) \lg \frac{p_{xy}(i, j)}{p_x(i) p_y(j)}$$

Clearly $I(x, y)$ is symmetric. For a rationale behind this definition see (Li & Vitányi 1993). The mutual information between two attributes describes the amount of information that one attributes gives about the other. The definition of mutual information describes the amount of information but does not determine the type of relation between two attributes. Two attributes that always have reversed values will be similar according to this measure.

The second measure considered in this paper is based on probability theory. It is defined as

$$P(x, y) = \frac{p_{xy}(1, 1)}{p_x(1) p_y(1)}$$

This measure is closely related to the definition of confidence for association rules (Agrawal et al. 1996). It can be thought of as the ratio between the estimation of the conditional probability $\frac{p_{xy}(1,1)}{p_x(1)}$, and the estimation of the apriori probability $p_y(1)$. The conditional probability coincides with the confidence for an association rule $x \rightarrow y$.

example 1. Fig 1 shows the association matrix for a database of customers of a bank using the two different similarity measures. The database contains 8844 records having seven attributes that describe the seven different classes of products provided by the bank. Fig 1 on the left shows the results of using mutual information as a similarity measure. High bars correspond to pairs of similar attributes. Dark bars are positive relations, light bars are negative. Clearly there is a strong positive relation between payments and insurances, insurances and mortgages, etc., indicated by several dark bars. Apparently there is a negative relation between payments and equities.

Fig 1 on the right shows the results of the similarity measure based on conditional probabilities. The most significant relations are now between equities and deposits, and between insurances and mortgages.

The two measures are biased towards different types of association. $I(x, y)$ will reveal both positive and negative relations, but has a bias towards attributes of which $p_x(1)$ are close to $p_x(0)$. Two attributes that are rarely 1 (or 0) but always at the same time will not be recognised as a significant relation. $P(x, y)$ will reveal relations between attributes that are rarely 1 (see for example equities and deposits) but will only show positive relations. Thus different measures can be used depending on the type of association that is searched for.

Clustering

The association matrix calculated in the previous section can be used to report all association above a certain level. However the end-user would still be required to examine lists of (simple) rules. In this section we consider the graph defined by the association matrix and show how this fully connected graph can be simplified by calculating a *minimum spanning tree* (Cormen & Leiserson 1989, Preparata & Shamos 1985,

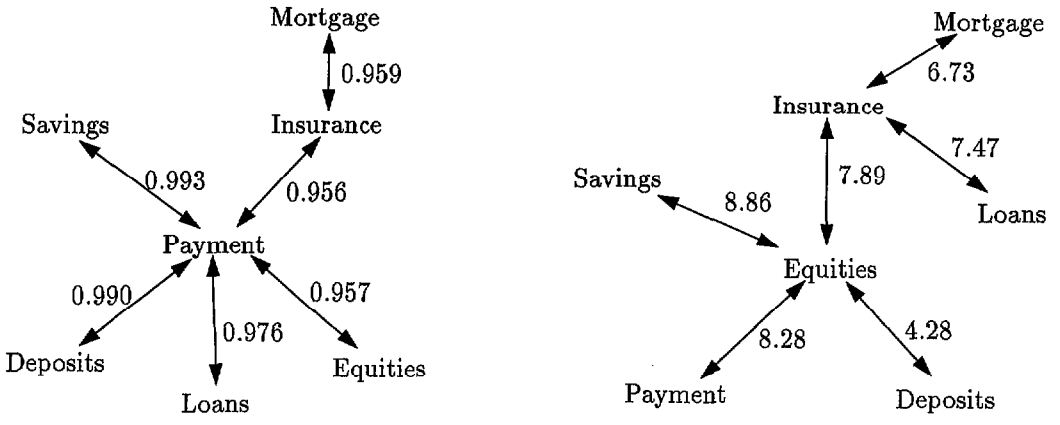

Figure 2: Minimum spanning trees for the bank database using $I(x, y)$, and $P(x, y)$.

Prim 1957, Tarjan 1983). This minimum spanning tree (MST) can then be used to cluster attributes (Cormen & Leiserson 1989).

We define the association graph of a database of items as a fully connect undirected graph $G = (V, E)$, where V is the set of vertices each of which represents an attribute and E the set of edges connecting the attributes. For each edge (x, y) we have a value $d(x, y) = -S(x, y)$ specifying the distance between two attributes, which we define as the negation of the similarity. Note that $d(x, y)$ may be negative.

An acyclic subset $T \subseteq E$ is a MST if it connects all vertices, and minimises the sum

$$\sum_{(x,y)\in T} d(x, y)$$

Two greedy algorithms for computing MSTs, Kruskal's and Prim's algorithm, are described in (Cormen & Leiserson 1989). They both work by growing a tree from a single vertex, adding one edge at a time. They differ in what edge is added to the subset of edges that form the tree at each iteration. Kruskal's algorithm maintains a forest, starting with every vertex being a single tree. At each step two trees are joined by choosing an edge with a minimal weight. Prim's algorithm works by adding edges to a single tree that is a subset of the final MST.

Kruskal's algorithm can be implemented to run in $\mathcal{O}(|E| \lg |V|)$. Prim's algorithm runs in $\mathcal{O}(|E| \lg |V|)$ using ordinary heaps or $\mathcal{O}(|E| + |V| \lg |V|)$ using Fibonacci heaps for finding new edges efficiently. Because $|E| = \mathcal{O}(|V|^2)$ Prim's algorithm will run in $\mathcal{O}(|V|^2 + |V| \lg |V|) = \mathcal{O}(|V|^2)$ which is clearly optimal because an association matrix of complexity $\mathcal{O}(|V|^2)$ needs to be fully examined, if no assumptions on the type of similarity measure are made.

example 2. Fig 2 shows the effect of calculating the MST for the two matrices from example 1. It conveys the most interesting associations in a readable fashion. Again we see the properties of the two similarity measures in the organisation of the two trees. Payment and equities are connected in the graph on the left because there is a reverse relation between the two attributes. The relation between equities and deposits is only revealed in the right graph because the associated similarity measure does focus on such infrequent associations.

Computation of a MST can be thought of as reducing the complexity of the graph while respecting the connectivity of the graph. We can push the balance between these two goals towards reduction of complexity by repeatedly removing those associations that are least important in the MST. Every removal will cause a subtree to be split into two separate groups. We will thus end up with clusters of attributes that have high internal similarity. Because every association between attributes of two different clusters are less then the single connection that was cut (see (Cormen & Leiserson 1989)), dissimilarity between seperate clusters is guaranteed.

Experiments

We analyse an enrollment database of courses in computer science, using the approach presented in the previous sections. The database consists of 2836 records each describing the courses taken by a single student. On average between six and seven courses were taken by each student, from a total of 127 courses.

An MST containing the 127 courses is computed using the mutual information-based measure $I(x, y)$. Fig 3 show some details of the MST. Clearly the structure of the subtrees seems to coincide with our common knowledge of relations between the subjects that are covered in each course. Fig 3 seems to deal with courses about databases and user interfaces. Apparently the link between these two subjects is made by

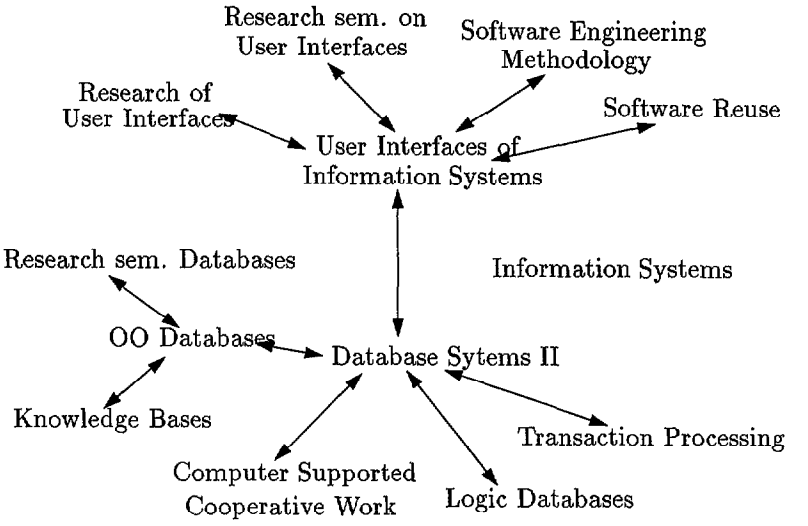

Figure 3: Part of Finnish courses.

'User Interfaces of Information Systems', a course that combines subjects from the two areas.

Clustering on this MST has the effect of putting outlyers in clusters of a single course, such as *'Sem. on Scientific Visualisation'*, *'Computational Geometry'* or *'Principles of Programming Languages (Ada)'*. These courses appear to be taken independently of other available courses. Continuing the clustering process will then split the remaining tree into particular subgroups, such as the area of databases, system programming, etc. A good criterion for continuing the clustering process seems to be hard to define.

Conclusion

This paper describes how information contained in binary associations can be exploited to the fullest. Our approach analyses a subset of the possible associations considered in traditional association rule discovery algorithms, but from the experiments it is clear that it does not suffer from this restriction, and even allows more effective ways of presenting the discovered knowledge.

Two similarity measures for attributes have been presented, each emphasising particular characteristics of associations between attributes. New similarity measures should be examined with the aim of combining useful properties from the presented measures.

By computing the minimum spanning tree of the association graph, we focus on the particular subset of large associations that is sufficient to include all attributes. We intend to examine the effect of reducing or extending this set of associations, especially in the context of clustering.

Acknowledgement

We thank Hannu Toivonen for providing the student enrollment database (in Finnish).

References

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A. 1996. *Fast Discovery of Association Rules.* in Advance in Knowledge Discovery and Data Mining.

Cormen, T.H., Leiserson, C.E., Rivest, R.L. 1989. *Introduction to Algorithms.* MIT Press/McGraw-Hill.

Gower, J.C., Ross, G.J.S. 1969. *Minimum spanning trees and single linkage cluster analysis.* Appl. Stat. 18(1), 54-64.

Li, M., Vitányi, P.M.B. 1993.*An Introduction to Kolmogorov Complexity and its Applications.* Springer-Verlag.

Preparata, F.P., Shamos, M.I. 1985. *Computational Geometry: An Introduction.* Springer-Verlag.

Prim, R.C. 1957. *Shortest connection networks and some generalizations.* Bell System Technical Journal, 36:1389-1401.

Shannon, C.E., Weaver, W. 1949. *The Mathematical Theory of Communication.* University of Illinois Press, Urbana, IL.

Tarjan, R.E. 1983. *Data structures and Network Algorithms.* Society for Industrial and Applied Mathematics.

Toivonen, H., Klemettinen, M., Ronkainen, P., Hätönen, K., Mannila, H. 1995. *Pruning and Grouping Discovered Association Rules.* ECML-95 workshop on Statistics, Machine Learning and Knowledge Discovery in Databases.