Towards Association Rules with Hidden Variables

Ricardo Silva[1] and Richard Scheines[2]

[1] Gatsby Unit, University College London, WC1N 3AR London, UK
[2] Machine Learning Department, Carnegie Mellon, Pittsburgh PA, 15213, USA

Abstract. The mining of association rules can provide relevant and novel information to the data analyst. However, current techniques do not take into account that the observed associations may arise from variables that are unrecorded in the database. For instance, the pattern of answers in a large marketing survey might be better explained by a few latent traits of the population than by direct association among measured items. Techniques for mining association rules with hidden variables are still largely unexplored. This paper provides a sound methodology for finding association rules of the type $H \Rightarrow A_1, \ldots, A_k$, where H is a hidden variable inferred to exist by making suitable assumptions and $A_1, \ldots, A_k$ are discrete binary or ordinal variables in the database.

1 Contribution

Consider the problem of discovering association rules of the type

$$H \Rightarrow A_1, A_2, \ldots, A_k \tag{1}$$

where H is a variable that is not present in the database (a *hidden*, or *latent* variable) but that explains the association among recorded discrete variables $\{A_1, \ldots, A_k\} \subseteq \{X_1, X_2, \ldots, X_N\}$. This paper provides a novel algorithm for mining such rules.

The motivation is two-fold: first, the outcome of such an analysis can aid the discovery of plausible and novel hidden variables that may be used to characterize the population of interest. Second, it might provide a more concise set of rules.

For instance, suppose that our data was generated by the graphical model shown in Figure 1, where H is hidden and $X_1, \ldots, X_4$ are observable. A typical association rule algorithm might find all sorts of rules such as $X_1 \Rightarrow X_2, X_3$, $X_2 \Rightarrow X_1, X_3, X_4$, etc. A hidden variable approach could in principle output a single rule subsumming all of such rules.

This paper is organized as follows: in Section 2, we introduce the particular class of hidden variable association rules we use, making the link to related work in latent variable graphical models. Section 3 is the main section of the paper, describing the detailed approach. Experiments are discussed in Section 4.

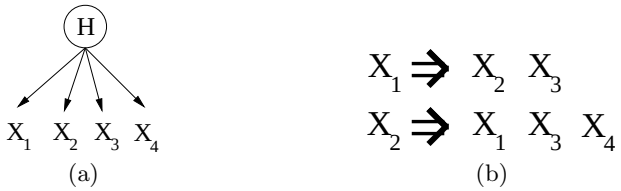

Fig. 1. Assume the data was generated according to the latent variable model in (a), where H is not recorded in the database and not known to exist. A potential set of association rules that ignore hidden variables is given in (b).

2 Probabilistic formulation

Following the framework of [7], we assume that our data was generated by a causal *directed acyclic graph*, where an edge $A \rightarrow B$ has the meaning that "A is a direct cause of B". There are several advantages on trying to extract *subgraphs* of the original graph as a type of association rule, instead of discovering a full graph [7], as further discussed in Section 2.1. Assuming there is a true graph G that generates our data, the semantics of a latent association rule $H \Rightarrow A_1, \ldots, A_k$, as given in this paper, are:

- H is a hidden node and a common ancestor of $A_1, \ldots, A_k$ in G, i.e., H is a hidden common cause of all elements in $A_1, \ldots, A_k$
- all variables $A_1, \ldots, A_k$ are probabilistically dependent, but become independent when conditioning on H[1];

The problem of discovering hidden variables is ill-defined without making extra assumptions. That is, if H is allowed to assume any distribution, then any distribution over observed variables can be generated by a hidden variable. This can be accomplished, for instance, by making H be a discrete variable with as many states as the entries of the contingency table of $A_1 \times \ldots \times A_k$. Such a rule can never be falsified. Instead, we will assume that our data was generated by some model from the family of *latent trait models* [1].

A model of this class is more easily understood through a graphical representation, as illustrated in Figure 2(a): each directed edge $H \rightarrow X_i$ from hidden node H to observed node X_i can be interpreted by having some intermediate hidden variable X_i^* on the path, as in $H \rightarrow X_i^* \rightarrow X_i$. The underlying X_i^* with latent parents $\{H_1^{X_i}, \ldots, H_k^{X_i}\}$ is given by

$$X_i^* = \sum_{j=1}^{k} \lambda_{ij} H_j^{X_i} + \epsilon_i; \epsilon_i \sim \mathrm{N}(0, \sigma_i^2);$$

and each λ_{ij} corresponds to the linear effect of parent $H_j^{X_i}$ on X_i^*. Latent variables are assumed to follow a multivariate Gaussian distribution, centered at

[1] That is, unlike traditional association rules, the right-hand side of the rule is not meant to assume any particular value (e.g., $A_1 = true$). Instead, the interpretation is that $A_1, \ldots, A_k$ are associated, but independent conditioned on H.

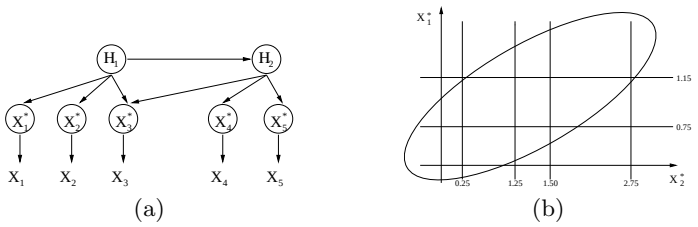

Fig. 2. (a) A graphical representation of a latent trait model with 5 ordinal observed variables. (b) Two ordinal variables X_1 and X_2 can be seen as discretizations of two continuous variables X_1^* and X_2^*. The lines in the graph above represent thresholds that define the discretization. The ellipse represents a contourplot of the joint Gaussian distribution of the two underlying continuous variables X_1^*, X_2^*.

zero. The observed variables X_i are then just discretizations of the respective X_i^*, as illustrated in Figure 2(b). More details on this model are given by [1] and [5].

This model imposes *constraints* on the observed joint distribution of the ordinal variables. Different graphical structures imply different correlations, and this can be used to test plausible association rules, as discussed in Section 3.

Even though this family of models rely on strong parametric assumptions, it has been successfully used to model binary and ordinal data, particularly *survey data* such as marketing questionnaires, social sciences and public opinion polls. It is also the basis of several psychological and educational testing studies [1].

2.1 Related work

What do we gain by extracting association rules from a graphical model instead of trying to learn the graphical structure directly? One major reason is *scalability*, as motivated by [7]: the data might have been generated by a directed graph that is too large to be efficiently learned from data. This is even more problematic in latent trait models, which requires the computation of high-dimensional Gaussian integrals. This scalability problem is also connected to the *statistical* problem of trying to learn large structures: different substructures of the graph might be more strongly supported by the data, and it would be of interest to report only on those substructures (i.e., association rules) of high confidence. Another major motivation is *identifiability*. As discussed at length by [6], there might be many different graphical structures that equally explain the data, but that agree on particular substructures. Rule mining focuses directly on those substructures that are uniquely identifiable from the assumptions.

Although there are other approaches for discovering latent variable models for discrete data (e.g., [3]), they do not address the issues raised above. The goal is usually density estimation, not knowledge discovery. Morever, they often assume that latent variables are marginally independent, an unnecessary assumption made mostly for the sake of simplicity.

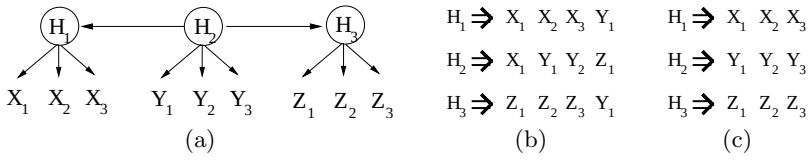

Fig. 3. If the data is generated by the structure in (a), Lemma 1 alone could be used to generate the rules in (b), with some incorrect information concerning the right-hand sides. However, the rules shown in (c) can be obtained by application of Lemma 2.

3 Methodology

We now describe an algorithm that generates rules corresponding to subgraphs of the (unknown) true graph G, which is assumed to have generated the data. Traditionally, conditional independency constraints are used to discover graphical structures. However, in a latent trait model, few, if any, of such constraints are observable [6]. Other types of constraints should be used. Consider a set of four variables, $\{W, X, Y, Z\}$ such that $\sigma_{WX}\sigma_{YZ} = \sigma_{WY}\sigma_{XZ} = \sigma_{WZ}\sigma_{XY}$, where σ_{XY} is the covariance of random variables X and Y. Under assumptions common in structure learning algorithms, the following holds in linear models [6]:

Lemma 1. *Let G be a linear latent variable model, and let $\{X_1, X_2, X_3, X_4\}$ be such that $\sigma_{X_1X_2}\sigma_{X_3X_4} = \sigma_{X_1X_3}\sigma_{X_2X_4} = \sigma_{X_1X_4}\sigma_{X_2X_3}$. If $\sigma_{AB} \neq 0$ for all $\{A, B\} \subset \{X_1, X_2, X_3, X_4\}$, then there is a node P conditioned on which all elements in $\{X_1, X_2, X_3, X_4\}$ are independent.*

This holds even if P is not observed, which means we can detect the existence of latent variables by using the covariance matrix of the given observed variables[2]. Lemma 1, however, does not provide enough information, since it does not indicate if such variables are descendants of P or not. To solve this issue, we rely on the following result (also in [6]):

Lemma 2. *If constraints $\sigma_{X_1Y_1}\sigma_{X_2X_3} = \sigma_{X_1X_2}\sigma_{X_3Y_1} = \sigma_{X_1X_3}\sigma_{X_2Y_1}$, $\sigma_{X_1Y_1}\sigma_{Y_2Y_3} = \sigma_{X_1Y_2}\sigma_{Y_1Y_3} = \sigma_{X_1Y_3}\sigma_{Y_1Y_2}$, $\sigma_{X_1X_2}\sigma_{Y_1Y_2} \neq \sigma_{X_1Y_1}\sigma_{X_2Y_2}$ all hold, then X_1 and Y_1 do not have a common parent in G.*

Notice that this result could be used to correct the rules in the example of Figure 3: one can verify that the above result holds for the pairs $\{X_1, X_2, X_3\} \times \{Y_1, Y_2, Y_3\}$, $\{Y_1, Y_2, Y_3\} \times \{Z_1, Z_2, Z_3\}$ and $\{X_1, X_2, X_3\} \times \{Z_1, Z_2, Z_3\}$.

What follows is an adaption of the algorithm in [6] to generate association rules. The main algorithm, BuildLatentRules (Table 1), starts by generating sets of variables (cliques) that could not be judged to measure different latents

[2] In our case variables are ordinal or binary, not continuous. However, there is an equivalent notion of covariance matrix for ordinal and binary variables, and tests of statistical significance for such constraints [5]. If there is enough memory to cache all second moments of the data, then this requires a single pass through the data.

Algorithm BuildLatentRules
Input: dataset $\mathcal{D}$ with observed variables $\mathbf{X}$

1. Let C be a fully connected undirected graph with nodes in $\mathbf{X}$
2. Remove edge $X_i - X_j$ from C if X_i and X_j are statistically marginally independent
3. Remove $X_i - X_j$ if X_i and X_j can be statistically separated as in Lemma 2
4. Let $\mathbf{M}$ be the set of maximal cliques in C.
5. $\mathbf{R_C} \leftarrow$ PurifyIndividualSets($\mathcal{D}, \mathbf{M}$).
6. Return FilterRedundant($\mathbf{R_C}$).

Table 1. An algorithm for learning association rules with hidden variables.

Algorithm PurifyIndividualSets
Inputs: dataset $\mathcal{D}$ with observed variables $\mathbf{X}$
Sets, a set of subsets of $\mathbf{X}$;

1. **Output** $\leftarrow \emptyset$
2. Repeat Step 3 below for all $Set \in$ **Sets**
3. If there is some $\{W, X, Y, Z\} \subset Set$ such that constraint $\sigma_{XY}\sigma_{WZ} = \sigma_{XW}\sigma_{YZ} = \sigma_{XZ}\sigma_{WY}$ is not true according to a statistical test, remove the node in Set that participates in the largest number of violated constraints. Repeat until all constraints are satisfied.
4. If Set has at least three variables, add it to **Output**.
5. Return **Output**.

Table 2. Identifying association rules from potential clusters of variables.

(using Lemma 2). However, failure to be separated by Lemma 2 does not imply such nodes indeed have latent parents in common. A second pass through such sets has to be performed to "purify" each set, resulting in a desirable association rule. This is performed by algorithm PurifyIndividualSets (Table 2): it ensures that Lemma 1 holds for any foursome in a selected set[3].

Because there might be several ways of "purifying" each candidate set, there might be many rules that are a consequence of the same hidden variable. Optionally, we might want to present just one rule for each hidden variable. This is performed by algorithm FilterRedundant defined as follows: if two rules overlap in three or more observed variables, then by Lemma 1 the hidden variable responsable for this pattern should be the same. FilterRedundant will allow only one rule for each hidden variable and also remove any rule whose right-hand size is contained in the union of other rules. This helps to minimize the number of spurious rules that are included by statistical mistakes.

[3] This algorithm requires a rule to have at least three variables on its right-hand side. For rules with fewer than three variables, see the complete algorithm in [5]. Moreover, for technical reasons omitted for lack of space, due to identifiability limitations of latent trait models it is possible that one (and at most one) of the elements on the right-hand side might actually not be a child of the latent (see [5, 6]).

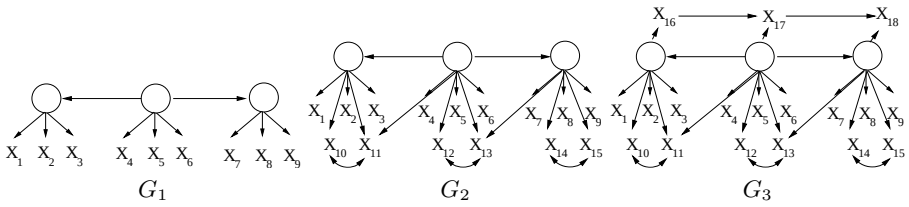

Fig. 4. The graphs used in our simulation studies.

Table 3. Results obtained with BUILDLATENTRULES and APRIORI for the problem of learning latent rules. For BUILDLATENTRULES, each number is an average over 10 trials, with the standard deviation over these trials in parenthesis.

BUILDLATENTRULES **statistics**				
	Sample	**Precision**	**Recall**	**#Rules**
G_1	*1000*	1.00(.0)	0.97(.1)	3.2(.4)
	5000	0.98(.05)	0.97(.1)	2.9(.3)
G_2	*1000*	0.94(.04)	1.00(.0)	3.2(1.03)
	5000	0.94(.05)	1.00(.0)	3.4(0.70)
G_3	*1000*	0.90(.06)	0.90(.16)	4.2(.91)
	5000	0.90(.08)	0.90(.22)	3.5(.52)

APRIORI **statistics**					
	Sample	**MIN**	**MAX**	**AVG**	**STD**
G_1	*1000*	15	159	81	59.4
	5000	9	546	116	163.9
G_2	*1000*	243	2134	1070.4	681.2
	5000	336	3565	1554.7	1072.2
G_3	*1000*	363	6036	2916.7	1968.7
	5000	158	4434	2608.3	1214.6

4 Experiments

In the following sections we evaluate our algorithm in a series of simulated experiments, and exploratory results on a real data set. In the simulated cases, we report statistics about the number of association rules that the standard algorithm APRIORI (using the implementation of [2]) returns on the same data. The goal is to provide evidence that standard algorithms might produce thousands of rules, despite the simple underlying latent variable model.

For the simulation study, let G be our true graph, from which we want to extract association rules. The graph is known to us by simulation, but it is not known to the algorithm. The goal of experiments with synthetic data is to objectively measure the performance of BUILDLATENTRULES[4] in finding correct and informative latent rules. Correctness is measured by a *Precision* statistic: the average precision of each rule. The precision of a rule is the proportion of items on the right-hand size that are in fact independent given the latent on the left. Completeness is measured by a *Recall* statistic: the proportion of latents $\{H_i\}$ in G such that there is at least one corresponding rule in our output. In our study we use the three graphs depicted in Figure 4, where all latents are potentially identifiable. Given each graph, we generated 10 parametric models and a sample of size 1,000 from each. Other 10 models were generated to sample datasets of 5,000 cases. The sampling scheme is given in [5]. Results are shown

[4] We use a slightly different variation of the algorithm to preprocess feasible candidate rules. Details in [5], Chapter 5.

Table 4. Examples of rules obtained by BUILDLATENTRULES on Deck 6 of the Freedom and Tolerance data set (question number and respective textual description).

Rule 1	
X27	I feel it is more important to be sympathetic and understanding of other people than to be practical and tough-minded
X3	I like to discuss my experiences and feelings openly with friends instead of keeping them to myself
X31	People find it easy to come to me for help, sympathy, and warm understanding
X67	When I have to meet a group of strangers, I am more shy than most people
X7	I would like to have warm and close friends with me most of the time
Rule 2	
X28	I lose my temper more quickly than most people
X30	I often react so strongly to unexpected news that I say or do things I regret
X41	I often push myself to the point of exhaustion or try to do more than I can
X61	I find it upsetting when other people don't give me the support that I expect
Rule 3	
X9	I usually demand very good practical reasons before I am willing to change my old ways of doing things
X53	I see no point in continuing to work on something unless there is a good chance of success
X46	I like to think about things for a long time before I make a decision
Rule 4	
X3	I like to discuss my experiences and feelings openly with friends instead of keeping them to myself
X40	I am slower than most people to get excited about new ideas and activities
X12	My friends find it hard to know my feelings because I seldom tell them about my private thoughts

in Table 3. We also display the number of rules that are generated. Ideally, in all cases we should generate exactly 3 rules. Due to statistical mistakes, more or less than 3 rules can be generated. It is noticeable that there is a tendency to produce more rules than necessary as the graph increases in complexity. It is also worthy to point out that without the FILTERREDUNDANT algorithm, we obtain around around 5 to 8 rules in most of the experiments. As a comparison, we report the distribution of rules generated by APRIORI in Table 3. We report the maximum and minimum number of rules for each model and sample size across the 10 trials, as well as average and standard deviation. The outcome is that not only APRIORI generates a very large number of rules, but the actual number per trial varies enormously (see, e.g., G_1 at sample size 5000).

We also applied BUILDLATENTRULES to the data collected in a 1987 study[5] on freedom and tolerance in the United States [4]. This is a large study comprising 381 questions targeting political tolerance and perceptions of personal freedom in the United States. 1267 respondents completed the interview. Each

[5] Available at http://webapp.icpsr.umich.edu/cocoon/ICPSR-STUDY/09454.xml

question is an ordinal variable with 2 to 5 levels, often with an extra non-ordinal value corresponding to a "Don't know/No answer" reply. However, several questions are explicitly dependent on answers given to previous questions. To simplify the task, in this empirical evaluation we will focus on a particular section of this questionnaire, the Deck 6. This deck of questions is composed of a self-administred questionnaire of 69 items concerning an individual's attitude with respect to other people. We obtained 15 rules, where 40 out of the 69 questions appear on at least on rule. Some of such rules are depicted in Table 4. There is a clear relation among items within most rules. For instance, items on Rule 1 correspond to measures of a latent trait of empathy and easiness of communication. Rule 2 has three items (X28, X30, X61) that clearly correspond to measures of a tendency of impulsive reaction. The fourth item (X41) is not clearly related to this trait, but the data supports the idea that this latent trait explains the associations between pushing oneself too much and reacting strongly to other people. See [5] for a more extensive discussion.

5 Conclusion

Our approach should be seen as a complement, not a substitute, to traditional association rule mining. There are three clear limitations: scalability; the limitation of a single hidden variable in the antecedent of each rule; and the adherence to a particular linear parametric family. However, it does provide extra information that typical association rule mining methods cannot replicate. As future work, we want to investigate how dynamic programming can be used to scale up the method, and how to better pre-process the data (e.g., by finding which marginally independent variables can be efficiently separated). Moreover, there are different sets of assumptions one can make in order to identify hidden variables [5]. To conclude, we hope that the ideas discussed in this paper can spark a new generation of algorithms for rule mining with hidden variables.

References

1. D. Bartholomew, F. Steele, I. Moustaki, and J. Galbraith. *The Analysis and Interpretation of Multivariate Data for Social Scientists*. Arnold Publishers, 2002.
2. C. Borgelt and R. Kruse. Induction of association rules: Apriori implementation. *15th Conference on Computational Statistics*, 2002.
3. W. Buntine and A. Jakulin. Applying discrete PCA in data analysis. *Proceedings of 20th Conference on Uncertainty in Artificial Intelligence*, 2004.
4. J. Gibson. *Freedom and Tolerance in the United States*. Chicago, IL: University of Chicago, National Opinion Research Center [producer], 1987. Ann Arbor, MI: Inter-university Consortium for Political and Social Research [distributor], 1991.
5. R. Silva. Automatic discovery of latent variable models. *PhD Thesis, Machine Learning Department, Carnegie Mellon University*, 2005.
6. R. Silva, R. Scheines, C. Glymour, and P. Spirtes. Learning the structure of linear latent variable models. *Journal of Machine Learning Research*, 7:191–246, 2006.
7. C. Silverstein, S. Brin, R. Motwani, and J. Ullman. Scalable techniques for mining causal structures. *Data Mining and Knowledge Discovery*, 2000.