USING SAS FOR MINING INDIRECT ASSOCIATIONS IN DATA

Pang-Ning Tan, University of Minnesota, Minneapolis, MN 55455
Vipin Kumar, University of Minnesota, Minneapolis, MN 55455
Harumi Kuno, Hewlett-Packard Laboratories, Palo Alto, CA 94304

ABSTRACT

Association rules are often used to capture significant dependencies among attributes in a variety of datasets. A typical association rule for market basket data might be of the form, "Purchasers who buy an item x will very likely buy another item y." We have introduced a new type of dependence relationship called indirect association. A pair of items, x and y, are indirectly associated if they rarely occur together in the dataset, and there exists a set Z such that the presence of x or y in a transaction is highly dependent on the occurrence of items in Z. Indirect association has found its applications in retail, textual and stock market domains. In this paper, we describe the implementation of a prototype system for discovering indirect associations using SAS. We have tested the prototype system using masked data from Hewlett-Packard's online purchase transactions to discover interesting product purchase patterns.

INTRODUCTION

In recent years, there has been considerable interest in extracting association rules [1,2] from large databases. Conceptually, an association rule indicates that the presence of a set of items in a transaction would imply (to a certain degree of confidence) the presence of other items in the same transaction. This type of pattern can be used for product placement, target marketing, catalog design, etc.

The problem of mining association rules is often decomposed into two separate tasks: (1) to discover all itemsets (i.e. sets of items) having support above a user-defined threshold, and (2) to generate rules from these frequent itemsets. Under this formulation, any itemsets that do not have sufficient support are considered to be uninteresting. However, we believe that some of the infrequent itemsets may provide useful insight into the data. This has led us to introduce a new type of pattern called indirect association [6]. Consider a pair of items, *a* and *b*, that seldom co-occur together in the same transaction. If both items are highly dependent on the presence of another itemset, *M*, then *a* and *b* are said to be indirectly associated via *M*. Figure 1 illustrates a high-level view of indirect association.

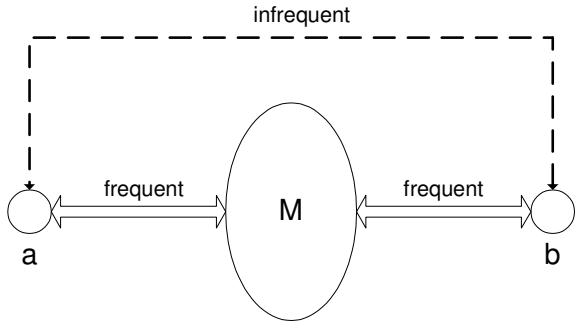

Figure 1: Indirect Association between *a* and *b* via mediator *M*.

There are many potential applications for indirect associations. For market basket data, this method can be used to perform competitive analysis among products. As an example, *a* and *b* may correspond to products of competing brand names, models, price-range, etc. Using the indirect association between competing items, marketing executives can identify prospective candidates for doing up-sale promotions. The mediator set is needed to ensure that customers who buy either one of the competing items share similar buying behavior. This increases their propensity to switch to the competing item, whenever certain incentives are given. Moreover, items in the mediator set can be included in the up-sale promotion to further entice the potential customers. On the other hand, if *a* and *b* are not competing products, this will offer a cross-selling opportunity for the vendors.

For text documents, indirect association often corresponds to pairs of words that are present in the different contexts of the "mediator" word. For example, if a user queries on the word *mining* using some of the popular Internet search engines, the collection of Web pages returned by these search engines contains a mixture of documents about data mining and metallurgic mining. Indirect association provides a technique to explicitly separate the different contexts in which the queried word appears in the corpus of text documents [6].

Similarly, for stock market data, indirect association can be used to identify the different set of events that are influencing the movement of a stock price. For example, the stock price of Microsoft and Intel may move together frequently since they both belong to the same industrial group[1]. If Intel issues a profit warning on its stocks, the price of other leading technology stocks (such as Microsoft) will be dragged down too. On the other hand, Microsoft and Red Hat[2] stocks could move in opposite direction since Red Hat's Linux operating system is a competitor to Microsoft Windows. Moreover, Intel and Red Hat stocks may not move together that often during the same period. Therefore, an indirect association may exist between Red Hat and Intel via Microsoft. One can interpret this pattern as saying that the events that are causing Microsoft shares to drop can be partitioned into at least two other sets of events: one which is associated with the decline of Intel stock prices while the other which is causing the price of Red Hat stocks to rise.

In this paper, we describe the design and implementation of a prototype system for mining indirect associations using SAS. This system has been tested on data from Hewlett-Packard's online purchase transactions to discover potentially interesting patterns.

The rest of this paper is organized in the following way. First, a formal definition of indirect association is presented. Next, an algorithm for mining such patterns is

given, followed by a description of the prototype system. Finally, we conclude with a summary of our preliminary results and suggestions for future research.

PROBLEM FORMULATION

Let $I = \{i_1, i_2, \ldots, i_d\}$ be a set of binary literals, called items, and T denotes the set of all transactions, i.e. $T = \{ T_j \mid \forall j : T_j \subseteq I \}$. Throughout this paper, we will use upper case letters to represent itemsets and lower-case letters for individual items. The support of an itemset X, denoted as sup(X), indicates the fraction of transactions that contain X. For convenience, we denote sup($\{a,b\}$) as sup(a,b) and sup($\{a\} \cup X$) as sup(a,X).

Definition 1: An itempair {a, b} is said to be indirectly associated via a mediator set M if the following conditions hold:

1. sup(a,b) < t_s (Itempair Support Condition)
2. There exists a non-empty set M such that
    a. sup(a,M) ≥ t_f and sup(b,M) ≥ t_f (Mediator Support Condition).
    b. $d(a,M) \geq t_d$ and $d(b,M) \geq t_d$ where $d(p,Q)$ is a measure of the dependence between p and Q (Mediator Dependence Condition).

The above thresholds are called the itempair support threshold (t_s), dependence threshold (t_d) and frequent itemset threshold (t_f), respectively. Each indirect association pattern will be denoted by a triplet, <a, b, M>.

Condition 1 is needed because an indirect association becomes significant only if both items rarely occur together in the same transaction. Otherwise, it makes more sense to characterize the pair in terms of their direct association. Alternatively, condition 1 can be modified to test for independence between items a and b. However, it is often the case that itempairs that have very low support values are either independent or negatively correlated with each other. Thus, condition 1 is sufficient to effectively discover indirect relationships between independent or negatively correlated itempairs.

Condition 2(a) can be used to guarantee the statistical significance of the mediator set. This is especially important for market basket analysis where the support of an itemset may reflect the amount of revenue generated, thus justifying the feasibility of promoting the items together. From an algorithmic perspective, support also has a nice downward closure property that allows us to reduce the exponential number of candidate mediators.

Condition 2(b) ensures that only items that are highly dependent on both a and b are used to form the mediator set. Suppose there exists an item k that appears in every transaction (e.g. shipping and handling charges). Without the mediator dependence condition, all pairs of infrequent items will be indirectly associated via k. Thus, condition 2(b) is necessary to prevent the generation of spurious indirect associations due to uninteresting mediators. Currently, there are many objective interest measures that can be used to represent the degree of dependencies between attributes of a dataset. For example, the confidence of the rule ($a \rightarrow b$) or ($b \rightarrow a$) can be used to measure the dependencies between a and b. However, as shown by Brin et al. [3], this measure may produce counter-intuitive results, especially for negatively correlated itemsets. Instead, Brin et al. [3] proposed the χ^2 test to identify correlated itemsets. However, the χ^2 statistic is not sufficient because it does not measure the strength of dependency between items. Furthermore, for large datasets, almost all frequent itemsets would easily pass the χ^2 test. Other alternative measures that have been proposed in the statistical literature include Pearson's ϕ coefficient, Goodman and Kruskal's λ, Yule's Q and Y coefficients, etc.

Interest factor is an objective measure that has been used quite extensively in the data mining literature [3]. The interest factor between a pair of items x and y is defined as:

$$I(x,y) = \frac{P(x,y)}{P(x)P(y)}$$

For itempairs, the interest factor is equivalent to the lift measure. An important criterion for a good objective interest measure is its ability to capture the notion of statistical correlation (ϕ). For binary variables, Pearson's ϕ coefficient can be written as:

$$\phi_{xy} = \frac{P(x,y) - P(x)P(y)}{\sqrt{P(x)(1-P(x))P(y)(1-P(y))}}$$

where $P(.)$ is equivalent to support. It can be shown that for certain interesting range of support values (i.e. when P(x) << 1, P(y) << 1 and P(x,y)/$(P(x)P(y))$ >> 1):

$$\phi_{xy} \approx \sqrt{I(x,y) \times P(x,y)} \equiv IS\ (x,y)$$

We will denote the right-hand side of this expression as the *IS* measure [5]. This measure is desirable because it takes into account both the statistical dependence (I(x,y)) and statistical significance (P(x,y)) aspects of a pattern. We do not use ϕ as the dependence measure because it treats both presence and absence of items in the same manner (see Table 1). In many data mining application domains, the presence of items are more important that their absence.

	Y=1	Y=0
X=1	0.1	0.15
X=0	0.15	0.6

(a)

	Y=1	Y=0
X=1	0.6	0.15
X=0	0.15	0.1

(b)

Table 1: The correlation (ϕ) between X and Y for both cases (a) and (b) are the same; even though the support for X and Y in (b) is higher than (a). However, the *IS* measure for case (b) is larger than case (a).

In this paper, we will use *IS* as the dependence measure for Condition 2(b). Nevertheless, our general framework can accommodate other appropriate interest measures. In fact, it can be shown that interest measures such as Piatetsky-Shapiro's rule-interest, J-measure and Gini index, are equally good at capturing statistical correlation in many natural datasets [5].

ALGORITHM

An algorithm for mining indirect association is summarized in Table 2. There are two phases in this algorithm. During the first phase, all frequent itemsets are initially derived using standard frequent itemset generation algorithms such as Apriori [2] or FP-tree [4] (step 1). During the second phase, the frequent itemsets of size k, L_k, are used to generate candidate indirect associations for pass k+1, C_{k+1}. Each candidate in C_{k+1} is represented as a triplet, < *a, b, M* >. The candidate pairs are generated in the following way. During the join step (step 4), a pair of frequent itemsets of

size-k, f_1 and f_2, is combined together as long as they both have exactly k-1 items in common. The intersection between f_1 and f_2 produces the mediator set M while their set differences form the indirect items, a and b. By joining itemsets that are frequent, the mediator support condition (condition 2(a)) is guaranteed to be satisfied. Furthermore, since we are only interested in indirect association between pairs of items, it is sufficient to join frequent itemsets of the same size. The remaining steps (steps 6 to 8) are needed to verify the itempair support and mediator dependence conditions (conditions 1 and 2(b) respectively).

1. Extract frequent itemsets, $L_1, L_2, \ldots, L_n$ using frequent itemset generation algorithm.
2. $P = \emptyset$ (set of indirect associations)
3. for $k = 2$ to n do
4. $\quad C_{k+1} = \text{join}(L_k, L_k)$
5. $\quad$ for each $< a, b, M > \in C_{k+1}$ do
6. $\quad\quad$ if ($\text{sup}(a, b) < t_s$ and $\text{d}(a, M) \geq t_d$ and $\text{d}(b, M) \geq t_d$)
7. $\quad\quad\quad P = P \cup \{< a, b, M >\}$
8. $\quad\quad$ end
9. $\quad$ end
10. end

Table 2: Algorithm for mining indirect associations.

Example: Suppose {a,b,d} and {a,b,e} are two frequent itemsets of size 3 (Figure 3). After joining the two itemsets, we would obtain the candidate indirect association pattern <*d*, *e*, {*a*, *b*}>. After checking for the itempair support and mediator dependence conditions, we can determine whether d and e are indeed indirectly associated via {a,b}.

Transaction Id	Items
1	{a,b,c,d}
2	{a,b,e}
3	{b,c}
4	{a,b,e}
5	{a,b,d,e}

Frequent 3-itemset
(support threshold = 40%)

Pattern	Support
{a,b,d}	2
{a,b,e}	3

Candidate Indirect Association:

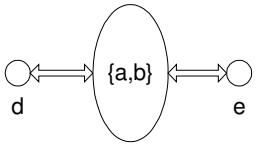

Figure 2: Example of candidate generation.

SYSTEM PROTOTYPE

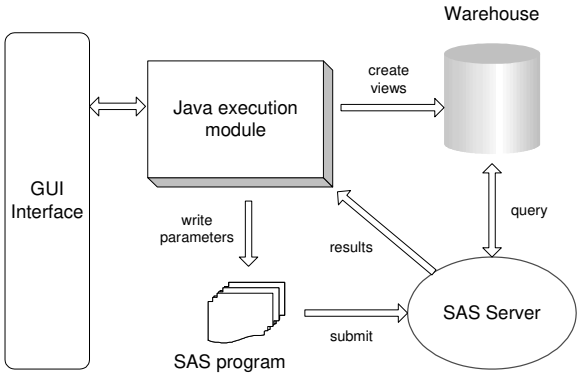

Figure 3: Prototype system for mining association patterns

In order to demonstrate the utility and feasibility of indirect associations, we implemented a prototype system that enables analysts to detect indirect associations in their data. This prototype was built using SAS version 6.2 and SAS Enterprise Miner V3.0 as the back-end analytical module.

The system provides a front-end that walks users through the process of indirect association discovery. A graphical user interface has been developed to handle user inputs (e.g. login name, password, threshold parameters, etc) and to display the results. The Java execution module is a single-threaded program that controls the overall execution of the system. This module is supported by a back-end SAS server that performs all the analytical computations. First, the execution module extracts the historical data from the warehouse and converts the data into SAS datasets. The SAS server is invoked whenever the execution module submits a SAS program in the batch mode. A SAS procedure called DMDB is used to create the SAS catalog and dataset files. Next, the execution module will call the Assoc procedure (from SAS Enterprise Miner) to generate frequent itemsets from the SAS datasets. The resulting frequent itemsets are stored in a SAS table called ITEMSET. The ITEMSET table contains various attributes such as set_size, support and list of items in the frequent itemset. A temporary dataset called tempdat1 is then created which is essentially the same as ITEMSET except for an additional column called set_key. The procedure for generating indirect association is implemented in the following way:

1. find all candidate indirect associations. This is done in the following way:
   a. join two tempdat1 tables (say, A and B) on set_size where A.set_key < B.set_key. The result of the join operation is stored in a SAS view called vw1.
   b. for each tuple in vw1, check if all items from the corresponding row in A has k-1 items in common with the corresponding row in B (where k is the value of set_size for the two joint tuples).
2. The output of step 1 is a SAS table called dbview. This table contains all candidate indirectly associated pairs and their corresponding mediators. For each tuple in dbview, we then compute their itempair

support and dependence values. This will filter out indirect itempairs that have large itempair supports or low dependence values. The results will be stored in a temporary result file and control will be returned to the execution module.

Finally, the execution module will read the result file and display the indirect associations using the graphical user interface.

RESULTS

The prototype system was tested on a masked dataset collected from Hewlett-Packard online purchase transactions for the past two years. With $t_s = 0.005\%$, $t_f = 0.025\%$ and $t_d = 0.1$, we found more than 200 indirect associations. The Java execution module would automatically summarize the discovered patterns by grouping together similar patterns according to their product categories before presenting them to the analyst (Figure 4). This reduces the number of patterns down to 37. The categorized patterns are shown on the left window of Figure 4. By clicking on any one of the categorized patterns, analysts can drill down to the finest granularity patterns (shown on the right window).

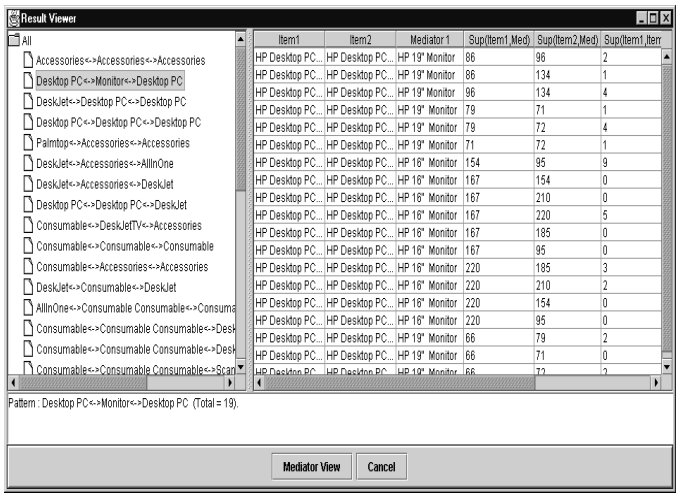

Figure 4: Display of indirect association results.

Using indirect associations, we have identified several potentially actionable patterns. An example of the up-sale pattern is the indirect association between PC Desktop model X5 and PC Desktop model X4 via the mediator 17" Multimedia Monitor where the X5 model is more expensive than the X4 model[3]. We have also discovered several potential cross-sale patterns. An example of such pattern is <*a*=PC Desktop R7, *b*=Deskjet Printer PP9, *M*=15" Monitor MM2>. Another cross-sale example is the pattern <*a*=greeting card paper, *b*=deskjet, *M*=inkjet cartridge> which indicates that people tend not to buy greeting card paper right away when they purchase a desk jet but that people who buy desk jets might later buy greeting card papers. So perhaps we could bundle the greeting card paper with the desk jet to encourage people to try the product early.

Indirect associations can sometimes reveal other interesting information about the customers. Consider the pattern <a=Jornada, b=Jornada stylus pen, *M*=leather Jornada case>. This pattern is obtained because most people tend not to realize that they are going to need an extra stylus pen until they lose their first. There are two groups of customers who own the Jornadas: (1) those who buy their Jornadas at the online store within the last 2 years, and (2) those who buy their Jornadas elsewhere or more than 2 years ago. Since the support of the indirect pair (Jornada and stylus pen) is almost non-existent, this suggests that most of the Jornada owners who need the extra pens have either bought their Jornadas elsewhere or more than 2 years ago.

CONCLUSION

In this study, we describe how indirect associations can be used to discover interesting patterns in their data. An itempair is said to be indirectly associated if both items are highly dependent on a common set of items even though their joint support is low. We propose an algorithm for deriving these patterns and show how to implement this algorithm in SAS.

For future research, there are still several unresolved issues we need to address. First of all, it will be useful to extend our work to indirectly associated itemsets rather than itempairs. In this case, computational scalability becomes a major issue because one is no longer restricted to joining equal sized frequent itemsets. We are also investigating how to combine direct and indirect associations into higher order structures for more effective visualization. Finally, threshold selection is another issue that needs further investigation. Our previous study [6] showed that a meaningful dependence threshold can be chosen to ensure that the majority of mediators found are statistically correlated.

ACKNOWLEDGMENTS

We wish to thank Vineet Singh, Meichun Hsu, Umeshwar Dayal and other members of the data mining group at Hewlett Packard Laboratories for their suggestions and assistance. We would also like to thank Hewlett Packard for kindly sharing their data with us.

This work was done by Pang-Ning Tan during his summer internship at Hewlett Packard Laboratories. Pang-Ning Tan and Vipin Kumar are currently supported by NSF ACI-9982274 and by Army High Performance Computing Research Center contract number DAAH04-95-C-0008.

REFERENCES

1. R. Agrawal, T.Imielinski, A.Swami, Database mining: a performance perspective. *IEEE Transactions on Knowledge and Data Engineering*, 5:914-925 (1993).
2. R.Agrawal, R.Srikant, Fast algorithms for mining association rules. In *Proc. of the 20th VLDB Conference*, 487-499, Santiago, Chile (1994).
3. S.Brin, R.Motwani, C.Silverstein, Beyond market baskets: Generalizing association rules to correlations. In *Proc. ACM SIGMOD Int'l Conf. on Management of Data*, 265-276, Tucson, AZ (1997).
4. J. Han, J. Pei, and Y. Yin, Mining Frequent Patterns without Candidate Generation, In *Proc. 2000 ACM-SIGMOD Int. Conf. on Management of Data (SIGMOD'00)*, Dallas (2000).
5. P.N.Tan, V.Kumar, Interestingness Measure for Association Patterns: A Perspective, In *KDD'2000 Workshop on Postprocessing in Machine Learning*

and Data Mining, Boston, MA (2000)

6. P.N.Tan, V.Kumar, J.Srivastava, Indirect Association: Mining Higher Order Dependencies in Data. In *Proc. of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 632-637, Lyon, France (2000)

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Pang-Ning Tan
Department of Computer Science and Engineering
University of Minnesota
200 Union St SE
Minneapolis, MN 55454
ptan@cs.umn.edu

Vipin Kumar
Department of Computer Science and Engineering
University of Minnesota
200 Union St SE
Minneapolis, MN 55454
kumar@cs.umn.edu

Harumi Kuno
Software Technology Laboratory
Hewlett Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94304
hkuno@hpl.hp.com

---

[1] This example is for illustrative purposes only and does not necessarily reflect the actual performance of these stocks. Microsoft is a registered trademark of Microsoft Corporation, while Intel is the registered trademark of Intel Corporation.

[2] Red Hat is a registered trademark for Red Hat, Inc.

[3] Note that the actual name and model of the products are modified due to confidentiality reasons.