Partial Classification using Association Rules

Kamal Ali and **Stefanos Manganaris**
IBM Global Business Intelligence Solutions
650 Harry Rd, San Jose, CA 95120
{ali,stefanos}@almaden.ibm.com

Ramakrishnan Srikant
IBM Almaden Research Center
650 Harry Rd, San Jose, CA 95120
srikant@almaden.ibm.com

Abstract

Many real-life problems require a partial classification of the data. We use the term "partial classification" to describe the discovery of models that show characteristics of the data classes, but may not cover all classes and all examples of any given class. Complete classification may be infeasible or undesirable when there are a very large number of class attributes, most attributes values are missing, or the class distribution is highly skewed and the user is interested in understanding the low-frequency class. We show how association rules can be used for partial classification in such domains, and present two case studies: reducing telecommunications order failures and detecting redundant medical tests.

Introduction

Classification, or supervised learning, has been widely studied in the machine learning community, e.g. (Michie, Spiegelhalter, & Taylor 1994). The input data for classification, also called the training set, consists of multiple examples (records), each having multiple attributes or features. Additionally, each example is tagged with a special class label. The objective of classification is to analyze the input data and to develop an accurate description or model for each class using the features present in the data. The class descriptions can be used to classify future test data for which the class labels are unknown, or to develop a better understanding of each class in the data.

In *partial classification*, the main goal is not prediction of future values, but rather to discover characteristics of some of the data classes. The aim is to learn rules that are individually accurate. However, these rules may not cover all classes or all examples of a given class. Further the examples covered by different rules may overlap. Hence the model discovered by partial classification can provide valuable insights into the data, but cannot be directly used for prediction.

The problem of mining association rules was introduced in (Agrawal, Imielinski, & Swami 1993). The input consists of a set of transactions, where each transaction is a set of literals (called items). An example of an association rule is: "30% of transactions that contain beer also contain diapers; 2% of all transactions contain both of these items". Here 30% is called the *confidence* of the rule, and 2% the *support* of the rule. The problem is to find all association rules that satisfy user-specified minimum support and minimum confidence constraints. Applications include discovering affinities for market basket analysis and cross-marketing, catalog design, loss-leader analysis and fraud detection. See (Nearhos, Rothman, & Viveros 1996) for a case study of a successful application in health insurance.

In this paper, we show that association rules can often be used to solve partial classification problems. Associations can be used in domains where conventional classifiers would be ineffective due to one or more of the following conditions:

1. There are a very large number of attributes.
2. Most of the values of each attribute are missing.
3. The class distribution is very skewed, and the user is interested in understanding some low-frequency classes.
4. Each of the attributes must be modeled based on the other attributes.
5. The number of training examples is very large.

There has been some recent work on decision-tree classifiers that can scale to large training examples (Mehta, Agrawal, & Rissanen 1996) and set-valued features (Cohen 1996). However characteristics 2, 3 and 4 still make using decision-trees problematic. Characteristics 2 and 5 precluded the use of neural networks (Rumelhart & McClelland 1986).

Characteristic 5 made the use of existing inductive logic programming (ILP) techniques (Muggleton 1992) cumbersome. First order logic concept descriptions are very expressive, allowing variables, higher arity predicates, and negation of predicates. However, due to their flexibility and larger search space, current ILP algorithms are much slower than algorithms for finding association rules.

We illustrate this application of associations with two case studies.

The first project was in the telecommunications domain. The data consisted of around a million orders (examples), a few (2.5%) of which had failed. Each order had multiple sub-parts (attributes), but fewer than 1% of the possible sub-parts were present in any specific order. Thus most attribute values were "missing". The client was interested in rules characterizing the failure classes.

The second project was in the medical domain. The data consisted of medical tests (attributes) given to patients (examples). As before, there were hundreds of medical tests but only a few of them were given to any single patient. The goal was to see if any of the medical test results could be predicted by combinations of other test results. If such tests were found, they could potentially be used to avoid giving patients redundant tests, or complex/expensive tests could be replaced with simpler/cheaper tests. Thus *each* attribute value is a class that must be predicted by the other attributes.

Paper Organization We give a brief overview of association rules and frequent itemsets. Next, we describe the telecommunications and medical diagnosis case studies in detail. Finally, we summarize our results and conclude with a discussion of future work.

Association Rules Overview

Let $\mathcal{I} = \{l_1, l_2, \ldots, l_m\}$ be a set of literals, called items. Let $\mathcal{D}$ be a set of transactions, where each transaction T is a set of items such that $T \subseteq \mathcal{I}$. We say that a transaction T *contains* X, a set of some items in $\mathcal{I}$, if $X \subseteq T$. An *association rule* is an implication of the form $X \Rightarrow Y$, where $X \subset \mathcal{I}$, $Y \subset \mathcal{I}$, and $X \cap Y = \emptyset$. The rule $X \Rightarrow Y$ holds in the transaction set $\mathcal{D}$ with *confidence* c if $c\%$ of transactions in $\mathcal{D}$ that contain X also contain Y. The rule $X \Rightarrow Y$ has *support* s in the transaction set $\mathcal{D}$ if $s\%$ of transactions in $\mathcal{D}$ contain $X \cup Y$. Given a set of transactions $\mathcal{D}$, the problem of mining association rules is to generate all association rules that have support and confidence greater than the user-specified minimum support and minimum confidence respectively.

The problem of mining association rules is decomposed into two subproblems:

- Find all combinations of items that have transaction support above minimum support. Call those combinations *frequent itemsets.*
- Use the frequent itemsets to generate the desired rules. The general idea is that if, say, $ABCD$ and AB are frequent itemsets, then we can determine if the rule $AB \Rightarrow CD$ holds by computing the ratio $r = \text{support}(ABCD)/\text{support}(AB)$. The rule holds only if $r \geq$ minimum confidence. Note that the rule will have minimum support because $ABCD$ is frequent.

Figure 1 shows an example, assuming a minimum support of 50% (2 transactions) and a minimum confidence of 50%. The first step is responsible for most of the computation time, and has been the focus of considerable work on developing fast algorithms, e.g. (Agrawal *et al.* 1996) (Brin *et al.* 1997).

Dataset

Transaction ID	Items
1	Shoes, Shirt, Jacket
2	Shoes, Jacket
3	Shoes, Jeans
4	Shirt, Sweatshirt

Frequent Itemsets

Frequent Itemset	Support
{Shoes}	75%
{Shirt}	50%
{Jacket}	50%
{Shoes, Jacket}	50%

Rules

Frequent Itemset	Support	Confidence
Shoes $\Rightarrow$ Jacket	50%	66.6%
Jacket $\Rightarrow$ Shoes	50%	100%

Figure 1: Association Rules Example

Telecommunications Case Study

This case study focuses on one of the first components of a process re-engineering project. A typical service order for telecommunications related products undergoes a series of processing steps from entry to completion. Occasionally, a service order falls out of the path of normal processing, for unanticipated reasons, with a request for manual assistance (RMA). Automated processing for an order that has fallen out cannot continue until one has addressed the exception and put the order back into the system. The cost of fall-out for our client was significant, in terms of man-hours for handling orders on a case by case basis and in terms of reduced system throughput reflecting badly on customer satisfaction.

A service order was described by the set of product elements it involved, or Universal Service Order Codes (USOCs), in the telecommunications lingo. USOCs are low level components, which can be thought of as basic hardware or software resources; for example, there is a USOC for the "call-waiting" feature on a residential telephone line. In addition, each USOC in an order was associated with an *action*: a resource could be added or removed.

The goal of this project was to reduce fall-out, by modifying the process to eliminate the reasons causing it. Domain experts suspected that certain combinations of product elements in a service order were associated with requests for manual assistance. This case study presents our approach in identifying the root causes of fall-out.

In data mining terms, the problem can be cast as a partial classification problem: given historical data on fall-out, induce rules that predict it. We had more than a million examples of orders with associated RMA codes that indicated whether processing completed with or without a fall-out incident. Since an order may fall-out with different kinds of requests for manual assistance, we had a multi-class partial classification problem. There were two interesting challenges in our work: (i) instances were described in terms of a varying number of attributes, (ii) the prior probability of fall-out was much lower than that of normal processing. An order involved anywhere from one to 40 USOCs with an average of 3.5 USOCs, out of a set of several thousand possible USOCs. The prior probability of fall-out, regardless of type of RMA, was about 2.5%; there were about 25 different RMA codes in our study, with varying frequencies of occurrence.

Our approach involved two steps of analysis. In the first step, we identified frequent combinations of USOCs for each type of fall-out. We used the association discovery algorithm to analyze each RMA (i.e., each class) separately. For each type of RMA, the outcome was the set of all combinations of USOCs that appeared together in a service order with frequency exceeding a minimum threshold (*frequent itemsets*). This threshold was chosen based on recommendations from the domain experts. In the second step, we contrasted the patterns discovered for each RMA with those expected for service orders that either did not fall out or fell out with a different RMA. The purpose of this step was to eliminate patterns that were not predictive even though they were frequent. To this end, we looked for a statistically significant correlation between the presence of each pattern and the RMA class (for which the pattern was found), using a chi-square test of association. When a pattern was found to be significantly correlated, we estimated its relative risk,

$$r = \frac{P(\text{fall-out}|\text{presence of USOC combination})}{P(\text{fall-out}|\text{absence of USOC combination})},$$

and its 95% confidence interval.

Figure 2 shows an example of a service order, which fell-out with RMA code L000, involving the addition of USOCs RJ11C and 1FR and the removal of BSX. Note, the completion status of the order is implicit: orders in different classes were kept in separate files. After examining all orders in class L000, we found the following frequent item sets: [BSX:DEL], [RJ11C:ADD], and [BSX:DEL, RJ11C:ADD]. After testing the association of each itemset with RMA L000, taking into account all orders, only [BSX:DEL, RJ11C:ADD] was found significant:

[BSX:DEL, RJ11C:ADD] $\rightarrow$ L000, with $r = 25$,

indicating that order that include a removal of BSX and simultaneous addition of RJ11C are 25 times more likely to fail with RMA L000 than orders that do not.

Similar rules in our study pointed to an overhaul of the processing mechanisms for handling ISDN-related orders.

Order ID	USOC:ACTION
0001	RJ11C:ADD
0001	1FR:ADD
0001	BSX:DEL
...	...

Figure 2: File of Service Order in Class L000

An alternative approach would be to include the class of a service order in the "transaction" as another attribute, and find association rules over the whole data. We would then compute for each rule, $A \Rightarrow C$, the relative risk as

$$r = \frac{P(C|A)}{P(C|\neg A)} = \frac{\sup(C \wedge A)/\sup(A)}{(\sup(C) - \sup(C \wedge A))/(1 - \sup(A))}.$$

The subset of rules with an RMA as the consequent and high relative-risk values, would be roughly the same set of rules that was found with the first approach. To ensure that the the low-frequency classes are represented in the result, one is forced to use extremely low minimum-support thresholds. Thus, it becomes important to integrate in the discovery algorithm a constraint that we are only interested in rules that contain an RMA. Without such integrated constraints, the huge search space makes selection via post-filtering impractical. Recent work integrating item constraints into the association discovery algorithm (Srikant, Vu, & Agrawal 1997), enabled us to test this approach and verify that we got similar results.

Medical Diagnosis Case Study

In this proof-of-concept project, the Information Services department of a hospital cleaned the data and prepared two flat files for mining. The first file contained demographic measurements of the patients in a conventional format: one row per patient, one column per demographic variable. Approximately 20 demographic variables were used. The second file resembled transaction data: the first column consisted of the patient's visit identification number, the second column contained a medical test identifier and the third column contained the outcome of the test. Possible values for the outcome were low, normal and high. This format was used since the total number of tests per patient varied from patient to patient. We had data on a few thousand patients and there were about 600 medical tests. The doctors had several goals: (1) to model the outcome of a medical test in terms of other tests, (2) to see what kinds of tests were "prescribed" together, and (3) to see what kind of demographics were associated with medical tests that were given.

The data was processed to create a combined file in "transaction format" with two columns: the visit ID of the patient was in the first column and the

Visit ID	Tests/Demographics
001	gender:male
001	age:fifties
001	blood-pressure-test:normal
001	blood-pressure-test
001	blood-sugar-test:hypoglycemic
001	blood-sugar-test
…	…

Figure 3: Combined Medical Data

second column consisted of concatenated symbols of the form <medical-test>:<outcome> such as "blood-pressure:low". We also created additional records indicating that a certain test had been given without indicating the result of the test. This would enable discovery of associations between kinds of tests given without regard to their outcomes and thus achieve goal (3) above. Finally, the demographic information, such as "gender:male", was added to the transactions. We discretized each continuous variable into intervals in consultation with the domain experts. Altogether, then, the data for visit ID might look as shown in Figure 3. Including medical test data and demographic data together allowed for the learning of interplay between demographics and types of tests given and their outcomes.

The associations algorithm was applied to this data and we experimented with various levels of support and confidence. We discovered several rules that were new to the doctors, including an almost perfect model for one medical test that would allow the elimination of that test.

Conclusions

Complete classification is often infeasible or undesirable when there are a very large number of classes, most attributes values are missing, or the class distribution is highly skewed and the user is interested in understanding the low-frequency class. We showed how association rules can be used for partial classification in such domains, and illustrated this application with two case studies. The goal of the first study was reducing order failures in the telecommunications domain. Our results indicated an overhaul of the processing mechanism for handling ISDN-related orders. The goal of the second case study was to detect redundant tests during medical diagnosis. We discovered several rules that were new to the doctors, including an almost perfect model for one medical test that allowed the elimination of that test.

Future Work The successful use of associations for partial classification poses an intriguing question: can associations be used for complete classification? One approach to building a classifier on top of associations would be to add machinery for "conflict resolution" (Ali 1995). Let R_j be the set of rules for class j. The goal would be to combine evidence among the rules of R_j that fire for some test example x to come up with a posterior probability for class j. Then, whichever class had the highest posterior probability would "win"—x would be assigned to the winning class.

References

Agrawal, R.; Mannila, H.; Srikant, R.; Toivonen, H.; and Verkamo, A. I. 1996. Fast Discovery of Association Rules. In Fayyad, U. M.; Piatetsky-Shapiro, G.; Smyth, P.; and Uthurusamy, R., eds., *Advances in Knowledge Discovery and Data Mining.* AAAI/MIT Press. chapter 12, 307–328.

Agrawal, R.; Imielinski, T.; and Swami, A. 1993. Mining association rules between sets of items in large databases. In *Proc. of the ACM SIGMOD Conference on Management of Data*, 207–216.

Ali, K. 1995. A comparison of methods for learning and combining evidence from multiple models. Technical Report UCI TR #95-47, University of California, Irvine, Dept. of Information and Computer Sciences.

Brin, S.; Motwani, R.; Ullman, J. D.; and Tsur, S. 1997. Dynamic itemset counting and implication rules for market basket data. In *Proc. of the ACM SIGMOD Conference on Management of Data.*

Cohen, W. W. 1996. Learning trees and rules with set-valued features. In *Proc. of the 13th National Conference on Artificial Intelligence (AAAI).*

Mehta, M.; Agrawal, R.; and Rissanen, J. 1996. SLIQ: A fast scalable classifier for data mining. In *Proc. of the Fifth Int'l Conference on Extending Database Technology (EDBT).*

Michie, D.; Spiegelhalter, D. J.; and Taylor, C. C. 1994. *Machine Learning, Neural and Statistical Classification.* Ellis Horwood.

Muggleton, S., ed. 1992. *Inductive Logic Programming.* Academic Press.

Nearhos, J.; Rothman, M.; and Viveros, M. 1996. Applying data mining techniques to a health insurance information system. In *Proc. of the 22nd Int'l Conference on Very Large Databases.*

Rumelhart, D., and McClelland, J. 1986. *Parallel Distributed Processing: Exploration in the Microstructure of Cognition.* MIT Press.

Srikant, R.; Vu, Q.; and Agrawal, R. 1997. Mining Association Rules with Item Constraints. In *Proc. of the 3rd Int'l Conference on Knowledge Discovery in Databases and Data Mining.*