Approximate Clustering In Association Rules

Lawrence J. Mazlack
Computer Science
University of Cincinnati
Cincinnati, Ohio 45221-0030
mazlack@uc.edu

Abstract

Data mining holds the promise of extracting unsuspected information from very large databases. A difficulty is that ways for discouvery are often drawn from methods whose amount of work increase geometrically with data quantity. Consequentially, the use of these methods is problematic in very large data bases. Categorically based association rules are a linearly complex data mining methodology. Unfotunately, rules formed from categorical data often gneerate many fine grained rules. The concern is how might fine grained rules be aggregated and the role that non-categorical data might have. It appears that soft computing techniques may be useful.

1. Introduction

Data mining is an advanced tool for the management of large masses of data. Data mining is the process of *secondary* analysis of large databases. It is aimed at discouvering previously unknown relationships of value.

Data mining is secondary analysis because the data were not collected to answer questions now posed. The data is examined in an attempt to discouver if there are patterns beyond those that were hypothesized before the data was collected. For example, perhaps we are at looking at long distance telephone call records. The records were originally collected for billing. Now, they can be examined to recognize calling patterns involving such things as: call length, time-of-day, customer calling plan, from where-to-where, etc.

There are several different strategies of performing data mining. One approach is to use classic machine learning methods, such as entropy based learning. However, most classic machine learning techniques are computationally complex and are ill-suited to completely consider a large data set. Also, most machine learning results are not well suited for use by most human clients. This is less than ideal as a goal of data mining is to help human analysts.

Consequently, data mining techniques suitable for the analysis of large databases have been developed. Some data mining techniques focus on the discouvery of simple rules. Of these, association rules are prototypical and the easiest to understand.

Association rules formed from categorical data often generate many fine grained rules. This work's concern is an exploration of how might fine grained rules be aggregated and the potential role of non-categorical data.

2. Association Rules

Association rules [Agrawal, 1993] meet a necessary[1] data mining subgoal of having efficient data structures and algorithms. Their algorithms also have the advantage of being linearly upwardly scaleable.

The process of discouvering association rules is often based on Apriori [Agrawal, 1994]. It is an unsupervised discouvery process. However, there is often a predefined hierarchical concept tree that is applied to cluster the data in preparation for mining; this is a form of supervised discouvery. For example, all brands of beer might be grouped into a single class called: "Beer." Essentially, hierarchy trees help cluster the data for analysis and reduce the number of attributes. If some clustering is not somehow accomplished, there will be a multitude of fine grained, relatively infrequent rules.

Predefined hierarchy trees may not serve the purposes of data analysis well. Perhaps, implicit in the data, there may be more useful hierarchy trees that go undiscouvered because of the pre-processing assumptions. As with much real world data, the clusters that hierarchy trees represent can possibly be best formed by using fuzzy sets; or alternatively, rough sets.

It is a speculation of this work that it might be better to first identify fine grained rules, then aggregate them. This would overcome one difficulty with association rules is that they may be overly fine grained. It may be possible to increase the granularity of the discouvered association rules either through (a) the use of discouvered concept hierarchies, (b) soft clustering the rules themselves, or increasing the granularity of the data before forming association rules.

Association Rules represent positive associations between attributes. Most commonly, the rules are developed from categorical data that is expressed as a 0/1 matrix. For example, consider *Figure 1.*

Parts of this work were performed while the author was a visiting at BISC, Computer Science Division, EECS Department, University of California, Berkeley

[1] Efficiency is necessary because realistic databases may well contain millions of records. Less efficient algorithms often resort to stratified sampling. However, may become less satisfactory as the volume of records increase.

0-7803-6274-8/00/$10.00 © 2000 IEEE

transaction	item	quantity
t_1	chips	1
t_1	Miller	1
t_2	chips	2
t_2	mustard	1
t_2	sausage	3
t_2	Coke	5
t_2	Tuberg	1
t_3	Miller	1
t_3	Tuberg	2

⇓

	Chips	Must -ard	Saus -age	Soft Drink	Beer
t_1	1	0	0	0	1
t_2	1	1	1	1	1
t_3	0	0	0	0	1

where t(A) is:

- *0/1 matrix; rows: customers, columns: products, and*
- *t(A) = 1 iff customer t bought product A*

Figure 1. Example transactions

This resulting tabular form, t(A), is the one often used in developing association rules. Notice that:

(a) Regardless of initial magnitude, t(A) values are represented either by categorical values of a "0" or a "1"; for example, for t_1, the magnitude of chips is "1" while for t_2, the transactional magnitude is "2". However, in the t(A) matrix, the actual magnitude of chips is lost and both $t_{1,chips}$ and $t_{2,chips}$ is set to "1".

(b) There can be implicit hierarchy trees; for example, in t_1, Miller (an American beer brand) is purchased and in t_2, Tuberg (a Danish beer is purchased. When these transactions are present in t(A), they are both represented as "1"s as $t_{i,beer}$. Further, in t_3, both Miller and Tuberg are purchased; and, they are represented by a single "1" as $t_{3,beer}$.

After the categorical data matrix is developed, the strength of association between term is determined.

Some restriction on result volume is useful lest too many rules to be examined by a human analyst may be developed. One way to do this is to only find association rules with a high level of support; i.e., occur with a frequency above a specified threshold. A potential problem with thresholding is that rules of great interest, but with only moderate support might not be captured. This is particularly true in large, relatively sparse matrices.

The format of an assoication rule is: When <*event*$_1$>, in <*confidence factor*> will also <*event*$_2$>; this occurs in <*support*> of cases. Where

- *Support:* relative occurrence
- *Confidence:* degree rule true across individual cases

For example:

When a customer buys beer & sausage,
in 80% of the cases, he will also buy mustard; this occurs in 15% of cases

Several efficient algorithms for mining categorical association rules have been published [Agrawal, 1994] [Mannila, 1994] [Toivonen, 1996].

4. Quantifying Association Rules:

In practice, the information in many databases is not limited to categorical attributes; but usually also contains quantitative data. It is a possibly undesirable oversimplification to translate quantitative data into categorical data. It may over-simplify results when data magnitudes (or, quantities) are lost. For example, should the purchase of two units of mustard be considered the same as the purchase of one unit? If not, how can magnitudes be best handled? There are at least two kinds of information that might be obscured: differences of behavior due to differing quantities; and, trends.

4.1 Increasing Data Granularity While Forming Quantified Association Rules

Several strategies have been pursued to deal with scalar, non-categorical data by incorporating quantification into the mined association rules. For the most part, they have attempted to increase the granularity of the data before forming the association rules.

Srikant and Agrawal [Srikant, 1996] mapped the quantitative association rules' problem and categorical rules into a Boolean association rules' problem, thus extending the work started in Agrawal[1994]. They discussed mining association rules over quantitative and categorical attributes. They dealt with quantitative attributes by fine-partitioning the values of the attribute and then combining adjacent partitions as necessary. For example, a rule might contain quantified age range: Age = [30,39] and a resulting Boolean rule might be:

(<Age: [30,39] > and <Married: Yes>) ⇒ <NumCars: 2>

They introduced a measure of partial completeness that quantifies the information lost due to partitioning. This measure is used to decide whether or not to partition a quantitative attribute and the number of partitions

Cai [1999] considers transactions with quantities as supporting "weighted" rules. Cai wanted to balance between weight and support, believing that a separation of the two might ignore some interesting knowledge. For example, a less frequently occurring or supported item set might still be interested if:

- An item is under promotion or if it is unusually profitable.
- If an item is not considered very important in terms of the weights of its item sets, but it is very a popular item in that many transactions contain it.

Aumann [1999] provides a different definition of quantitative association rules. It is based on statistics. They look for events that differ significantly from a

statistical norm and were then considered to be interesting. They consider that the best description of for a set of quantitative values is its behavior is its distribution. The approach is to look for a subset and its mean (or median, variance) and compare it to the mean of the whole. The idea is to identify a population subset that presents "interesting behaviour." Aumann finds a type of rule that is not found by other methods. For example, *when:*

Sex = female the female mean wage of $7.90/hour is interesting when overall mean wage = $9.02

Aumann handles both quantitatively based and caterogically based rules, such as

$age \in [20,40] \Rightarrow$ *Height: mean = 165 cm*

Liu [1999] presented a statistical approach for quantitative mining. It is supervised search. The user specifies the target attribute(s). Liu is concerned that typically a large number of associations are found; particularly when the attributes are highly correlated. To manage this, they first prune the number of associations; then find special subsets of the remaining rules. The chi-square test was used instead of a minimum confidence measure.

Another statistical approach has been presented by Yao [1999].

4.2.2 Soft Methods

There is a considerable history applying either fuzzy or rough set techniques to databases. To name a few approaches, there are: databases of fuzzy values, fuzzy query rules, and rough set partitioning. Both fuzzy and rough sets have been used in various data mining efforts. Our focus is on the relatively unexplored area of rule aggregation, the lightly explored clustering antecedents and/or consequents, and the role of quantification.

Chan [1997] presents an algorithm for increasing data granularity that eliminates the need for user-supplied thresholds for support and confidence, and to find negative as well as positive association rules. Using fuzzy set theory, linguistic terms are used to find the degree to which they characterize records in the database.

Another approach was suggested by Fukuda [1996a,b]. It focused on computing two optimized ranges. One that maximizes the support on the condition that the confidence ratio is at least a given threshold; and, another that maximizes the confidence ratio on the condition that the support is at least a given threshold. They mined association rules of the form

$(A \in [v_1, v_2]) \Rightarrow C$

They used techniques from computational geometry (convex hulls). They proposed algorithms that discovered optimized gain, support and confidence association rules for two classes of regions.

Zhang [1999] extended the equi-depth partition algorithm to mine fuzzy quantitative association rules. They built at times on the Apriori algorithm. They considered $Y \rightarrow X$, where an *item* $<a,v>$ represents either a crisp value, an interval (if numerical), or a fuzzy term and a is an attribute with value v. If v is fuzzy, the item is fuzzy. They use a minimum support for each attribute. The super-candidate technique [Agrawal, 1994] is used.

Kuok [1998] mines fuzzy association rules to avoid either ignoring or overemphasizing the elements near the boundaries in the mining process. After partitioning the attribute domain as suggested by Srikant [1996], Kuok suggests: *If X is A then Y is B* where X, Y are a set of attributes and A, B are fuzzy sets which describe X and Y respectively. A user-supplied threshold is used to test each side of the rule as to being "satisfied."

Other efforts at association rules with fuzzy antecedents and consequents include Au [1998, 1999], Fu [1998], Lee [1997]. Au and Chan [1997] are working on the same problem. Fu's work is closely related to Kuock [1998]. Du [1999] has a somewhat different approach to fuzzified ranges. Somewhat related is the work on incomplete data of Ng [1998] and Ali [1997].

The difficulty with all the methods discussed in section 4 is that their complexities are expensive.

5. Frequent Sets

An extension of simply forming association rules by counting is the technique of *frequent sets* [Mannila, 1994]. Because many matrices of interest are sparse, frequent sets help focus on the development of heuristically more interesting rules. (The heuristic is that patterns that occur more frequently are the more interesting.)

An example of a sparse matrix of interest is a collection of grocery store transactions. Grocery stores typically have well over 10,000 distinct items for sale. Customers typically purchase less than fifty items during a store visit. Combinations of items that are purchased together often are the most interesting. For example, perhaps when a customer purchases bananas (the most often purchased grocery store item), there is a good chance that she will also purchase milk and corn flakes.

The rules developed through the frequent sets algorithm may still be fine grained. Unfortunately, the antecedent/consequent data clustering methods discussed in section 4 appear to be infeasible for frequent sets. Instead, an approach focusing on rule aggregation or on initial rule reduction would be better.

6. Reducing Association Rules Generated

If all possible association rules are developed for high dimensional data (data with a large number of attributes), the number of rules may be too extensive to be useful.

The upper boundary for the number of association rules for a data set is (where n_a is the number of attributes):

$$\sum_{i=1}^{n_a-1}(C_n^i \sum_{j=1}^{n-i-1} C_{n_a-i}^j)/2$$

The number of associations would be much greater if attributes can take on multiple values and if a distinct association rule is developed for each value.

Four techniques can be used to reduce the count of generated association rules: limiting rule dimensionality, using a support level, reducing quantification, applying concept hierarchies. Using a support level is a part of many methods already. We need to seek elsewhere.

7.1 Reducing Rule Complexity

If there are n_a attributes, only consider association rules of maximum length m. This reduces the theoretical upper boundary of the number of association rules to:

$$\sum_{i=1}^{m}(C_n^i \sum_{j=1}^{j<m} C_{n_a-i}^j)/2$$

While producing rules that are more understandable, the count of rules is still too large.

If high support is used as rule interestingness, high support will almost always imply shorter rules. However, you cannot know their maximum length be unless there is a limit on length. Also, it is possible that a sliding scale of support might be used. That is, less support for longer rules. In any case, it might be necessary to experimentally determine what a useful support level might be for a particular data set and/or application. No particular number is written in stone.

Another approach to quantification that might be useful would be to granulate values, then apply quantified association rule techniques. For example, if academic grades are recorded from 100 to 0, they often are granulized into A, B, C, ... Possibly the *mountain method* [Yager, Filev, 1994] might be applied to form the granules. The *mountain method* may also me useful in clustering association rules together.

7.2 Concepty Hierarchies

It is unclear whether hierarchy trees should be part of the initial formation of the transaction matrix (categorical or non-categorical) or dealt with later. For example, should all beer brands be grouped together as a single class under all cases? Or, should they be presented separately?

Consider *Figure 1*. Should *Miller* and *Tuberg* be grouped together into a single class: beer? Should all beverages be grouped together; i.e., should *Miller*, *Tuberg*, and *Coke* be grouped together? Can items be grouped together depending on context; i.e., perhaps sometimes beers should be in one group and *Coke* in another; and, sometimes they should all be together (e.g., "picnic supplies"). Is there some way of computationally deciding what should be grouped together?

Extending the example further: Consider the case where we have more beer brands, say: *Budwiser*, *Miller*, *Sam Adams*, and *Tuberg*. Most regular beer drinkers would agree that there is a substantial taste difference between the four of them. (There may also be price differentiation.).

- Should *Budwiser*, *Miller*, *Sam Adams* and *Tuberg* be grouped into a single category; or
- Should each be grouped separately; or,
- Should *Budwiser* & *Miller* form one group while *Adams* & *Tuberg* form another group?

Currently, grouping is done by a human expert. This is not entirely satisfactory because (a) potentially useful groups may go unrecognized and (b) the expert may make an error.

Concept hierarchies group similar attributes together. Intuitively, they are desirable. However, the question of how to computationally recognize them is the question. For example, various types of wine might be all be placed into to concept of "wine." For example, if we have six wines[1]:

- Meier[2] alcohol-free[3] Burgundy[4]
- Carminet[5] alcohol-free Chardonnay[6]
- Gallo[2] Burgundy[5]
- Chalone[5] Pinot Noir[5]
- Almadin[2] Chardonnay[6]
- Carneros[5] Chardonnay[6]

If the transaction set is as shown in *Figure 2*: It might be possible to recognize that the premium, alcohol wines can be grouped together. They are all associated with both green grapes and brie (a premium cheese). Similarly the basic alcohol wines can be grouped together (they are all associated with basic cheddar cheese). However, if a pre-defined hierarchy was used to group all wines together, the association between type of wine and type of cheese would not be discouvered; and, the following would result. This would result in loosing the information that premium wines are related to the purchase of brie and green grapes. Why might this be important to a store? Well, if you stop carrying green grapes or brie, you may also lose your premium wine customers.

---

[1] As defined by their label.
[2] A "basic" wine
[3] Alcohol free wines are normally fermented, then their alcohol is extracted
[4] A red wine
[5] A "premium" wine
[6] A white wine

	green grapes	store brand cheddar	Jean D'Arc brie	cottage cheese	Meier alcohol free Burgundy	Carminet alcohol free Chardonnay	Gallo Burgundy	Chalone Pinot Noir	Almadin Chardonnay	Carneros Chardonnay
t_1	1	0	1	0	0	0	0	1	0	1
t_2	1	0	1	0	0	0	0	1	0	1
t_3	1	0	1	0	0	0	0	1	0	0
t_4	1	0	1	0	0	0	0	0	0	1
t_5	1	0	0	0	1	0	0	0	0	0
t_6	1	0	0	1	1	0	0	0	0	0
t_7	0	1	0	1	0	1	0	0	1	0
t_8	0	1	0	0	0	0	0	0	1	0
t_9	0	1	0	0	0	0	1	0	0	0

Figure 2. Wine and cheese transactions for items of difering perceived quality.

A human observer might not be able to easily recognize that the premium, alcohol wines can be grouped together. They are all associated with both green grapes and brie (a premium cheese). Similarly the basic alcohol wines can be grouped together (they are all associated with basic cheddar cheese). (Computationally, it might be possible to do this.) However, if a pre-defined hierarchy was used to group all wines together, the association between type of wine and type of cheese would not be discouvered; and, the *Figure 3* would result. This would result in loosing the information that premium wines are related to the purchase of brie and green grapes. Why might this be important to a store? Well, if you stop carrying green grapes or brie, you may also lose your premium wine customers.

	green grapes	store brand cheddar	Jean D'Arc brie	cottage cheese	wine
t_1	1	0	1	0	1
t_2	1	0	1	0	1
t_3	1	0	1	0	1
t_4	1	0	1	0	1
t_5	1	0	0	0	1
t_6	1	0	0	1	1
t_7	0	1	0	1	1
t_8	0	1	0	0	1
t_9	0	1	0	0	1

Figure 3. All wine grouped together

It gets even worse if all of the cheese varieties are grouped together as well, as in *Figure 4.* Now, the analysis would tell us that any cheese and any wine are associated. These results are clearly over-generalized.

	green grapes	cheese	wine
t_1	1	1	1
t_2	1	1	1
t_3	1	1	1
t_4	1	1	1
t_5	1	1	1
t_6	1	1	1
t_7	0	1	1
t_8	0	1	1
t_9	0	1	1

Figure 4. All cheeese and wine grouped together

A hierarchy such as: *items ∈ food ∈ wine* can be many levels deep. There are essentially two ways of establishing hierarchies: (a) have a human predefine them or (b) learn them. There are a number of different ways that they can be learned.

As part of a rough-set based approach, Han [1992] used crisp concept hierarchies provided by the user before the initiation of data mining. Alternatively, Mazlack [1997] developed and tested a methodology to learn useful concept hierarchies while partitioning databases. Potentially, the same methodology could be applied here. Questions that need to be answered are:

- How to best recognize intermediate level useful hierarchical categories. For example, in a grocery store, how do we learn and choose between:
  items ∈ food, non-food,
  food ∈ ... cheese, wine, ...
  wine ∈ ... no-alcohol, basic, premium ...
- The role of quantification in recognizing useful hierarchical categories

8. Summary

Data mining holds the promise of extracting new information from very large databases. One of the difficulties of data mining is that traditional ways of discouvering new information from data are largely drawn from classic machine learning methods. The work involved in these methods increases geometrically with the amount of data considered. Consequentially, the use of these methods is problematic in very large data bases.

Association rules are a linearly complex, user understandable way of doing data mining. The results are particularly useful when it is important that both the rules themselves and the methodology behind the rules' development are understandable.

References

References were not included because of space limitations. They may be had by contacting the author or from the web at:
http://www.ececs.uc.edu/~mazlack/NAFIPS2000.html