The Marriage of Market Basket Analysis to Predictive Modeling

Sanford Gayle, SAS Institute Inc, Cary, NC

INTRODUCTION

A primary objective in data mining is to develop and improve upon the accuracy of predictive models, and an essential challenge toward this end lies in the discovery of new features, inputs or predictors. This paper illustrates how rules generated from market basket analysis (MBA) might be used to enhance predictive models.

To create a host of candidate inputs for predictive modeling, a transactional table, characterized by its having multiple rows per transaction-id, can be transposed into a "modeling" table, characterized by its having a single row per id. However, the curse of dimensionality often rears its ugly head, and a rules-based (RB) dimensional reduction scheme is proposed to skirt the problem.

KEYWORDS OR PHRASES

Market basket analysis (MBA); predictive modeling; curse of dimensionality; rules-based (RB) dimensional reduction schemes; RB input discovery.

MBA REVIEW

Market basket analysis concerns the analysis of various subsets of items taken from a population of items. The subsets of concern, or rules, are those identified as having a minimum value of confidence, support, or lift. An example rule is denoted as A➔B, where A is referred to as the rule's antecedent condition and B the consequent. The rule is interpreted as "If A occurs in the market basket, then B also occurs in the market basket."

The interpretation of MBA rules is not to be confused with the use of the same notation in logic to denote implication. To be sure, in MBA the rule A➔B can be "true", and both A and not B can also be true or occur in a market basket, a contradiction in logic. Instead, rules in MBA might be thought of as having degrees of implication associated with them, where the confidence, support, and lift statistics each represent various measures of degree.

The support and confidence for rule A➔B are defined as:

- If A and B occur together in at least X% of the market baskets, then the support for this rule is X
- Of all market baskets containing A, if at least X% also contains B, then the confidence for this rule is X.

For marketing campaigns, the confidence statistic ensures that the rule is true often enough to make a RB campaign effective. Or it ensures that the rule is true often enough to justify a rearrangement of products on a store floor based upon purchase patterns inferred from the rule. One must be careful when using confidence, for when the consequent is popular the confidence will be large irrespective of whether the two items are indeed associated in the same way that, for example, eggs & milk are.

The support statistic is often used to justify the financing of a marketing campaign or product rearrangement on a store floor. Support provides a measure of how often the rule occurs in market baskets and enables the marketer to assess whether the rule is worthy of his/her attention.

MARRYING MBA TO PREDICTIVE MODELING

The ad targeted at a surfer profile in a banner ad campaign is typically chosen from a series of possible banner ads based upon the probability that the surfer will click on the banner ad. To implement such a campaign successfully the marketer must provide answers to the following two questions. First, how can the surfer's profile be identified from the surfer's click stream or web log files? And, second, which banner ad should be displayed or has the highest probability of being clicked on for each surfer profile?

These two questions can be answered by marrying MBA to predictive modeling, with MBA being used to identify the surfer's profile and predictive modeling being used to decide which of the alternate banner ads most appeals to the profile. That is, interesting combinations of web pages might be identified with MBA and flagged with dummy variables to better profile the web surfer, and such dummy variables and profiles might then prove useful in fostering the predictive accuracy of existing predictive.

CONVERTING RULES INTO INPUTS

Generally, MBA tools don't automatically create features or dummy variables to identify individuals (i.e., the values of the transactional-id variable) having a particular rule in their market basket, for the number of possible rules is typically too large. For example, suppose there are 1000 possible items (i.e., web-pages) that a surfer can visit or put into their market basket. Then there are 1000*999/2 = 499,500 distinct possible combinations of items taken two at a time, many of which MBA will identify as rules.

The analyst often consults with, for example, the marketer to decide which rules are of interest prior to creating candidate inputs.

Various MBA criteria exist for creating inputs. Instead

of creating a dummy variable corresponding to a particular rule of interest, for example, the analyst might create a dummy variable based upon a set of rules meeting some minimum value of confidence, support, or lift.

For the purpose of using confidence as a criterion for creating RB inputs, the question as to how transactional categorical data can be collapsed into a single row per person or transaction-id value arises, and a method for transforming transactional tables into "modeling" tables is of relevance.

COLLAPSING OR ROLLING-UP TRANSACTIONAL DATA

Representing a challenging aspect of using predictive modeling tools when the original source data is transactional (i.e., rows are not independent of each other), predictive modeling tools require that each row in the source or "modeling" table be independent of the other rows,

In contrast, a transactional table is characterized by it's having dependencies or multiple rows per person or transaction-id value, and this structure is the norm for MBA tools.

The task of collapsing the transactional data into one independent row per id can be classified according to whether the column to be collapsed is continuous or categorical. The solution is typically very easy for a continuous column since for any given numeric column either the sum, count, or mean of the transactional rows will typically suffice for collapsing the data by id.

Suppose a person puts the six items listed below in their market basket. This data is a subset taken from the SAS internal web-server log files. In the transactional table the "purchase" is represented by six rows, each containing the id column, the Item column, the Count column, the Amount column, and the Banner column:

id	Item	Count	Amount	Banner-ad
10019	/hr/cafe/	1	0	0
10019	/hr/cafe/deli/	1	0	0
10019	/hr/cafe/forms/	1	0	0
10019	/hr/cafe/recipes/	1	$5.65	1
10019	/hr/cafe/recipes/Soup	1	0	0
10019	/hr/vendor.html	1	0	0

Here the person "10019" placed these six items in their market basket; that is, this person visited the pages represented by the corresponding directories (each directory references an index.html file). The Count, Amount, and Banner-Clicked columns for this person can be collapsed into the following row using the sum statistic:

Id	Total Amount	Page Count	Banner-ad
10019	$5.65	6	1

When the rows are collapsed in this way, each row corresponds to a distinct id, and the resulting table can be characterized as a modeling table.

Note, however, that by summarizing in this fashion information is lost. Here the specific item information or list of items placed in the market basket is lost, illustrating that the real challenge in collapsing transactional data lies in dealing with the categorical rather than the continuous data.

How, then, might the information that the person placed the six specific items in their market basket be retained?

TRANSPOSITION OR THE TRANSPOSE METHOD

Transposition is one approach to preserve the specific item information. Using the transpose procedure in SAS, the resulting row of the "modeling" table corresponding to the previous transactional data has the id column and six new columns, each corresponding to a distinct item and containing the value one. Here the count is used, but the sum could likewise have been chosen, in which case all columns would take a value of 0 except the "/hr/cafe/recipes/" column, which would take the value $5.65. Perhaps the analyst wants both the total amount and total count of each item placed in the market basket; in this case the modeling table would have 13 new columns.

The candidate inputs can be joined with any other "modeling" information available on the person "10019", including demographic information, the total count or total amount (e.g., 6 or $5.65) of items "purchased", and modeling tools can legitimately be used, for a single row now exists for each id.

THE PROBLEM WITH TRANSPOSITION

Transposition works well in collapsing the categorical data, but it is inappropriate when a large number of distinct values of a categorical column exist. If 1,000 distinct items (i.e., pages visited) were placed in market baskets during a given month, then the transposed transactional table would have 1,000 new columns instead of the six new ones illustrated above, with the columns corresponding to items that person "10019" didn't put in their market basket having a value of missing or zero.

Suppose now the analyst is interested in the values of a second categorical column, say time. In addition to the item column, now the time of the page visit is recorded and bucketed into 24 hourly bins. Collapsing the transactional table for both item and hour using transposition results in 1000+24 additional new columns.

In general, assuming that the interaction between the various levels of the categorical columns are of no interest, the number of new columns created as a result of transposing categorical columns will equal the sum of the number of distinct values of all categorical columns that are collapsed into the transposed or modeling table.

The potential dimensionality problem gets entirely out of hand when the interaction between, for example, time and item is of interest, for in this case the total number of new variables becomes the product of the numbers of distinct values of the categorical variables. Using the above example, 24,000 new columns need be created to model this interaction with regression. Even the greatest modeling algorithms in existence would have difficulty sifting through this many interaction terms to discover a significant relation between a target and the alleged interaction among a particular rule (e.g., diapers ➔ beer) and the time of day.

The potentially large number of columns created via transposition often creates problems for neural network, regression, and cluster tools. This problem falls under the rubric of the curse of dimensionality; i.e., as the number of levels in any categorical variable increases linearly, the number of new variables required by, for example, regression to address the high-order interaction among the variable and other categorical variables increases exponentially.

Beyond the shear number of new variables created by transposing transactional categorical data, sparse data often presents additional challenges for neural network, regression, and cluster tools. And even when missing values are converted to zero or imputed, the preponderance of identical values often diminishes the utility or significance of these inputs when used in modeling and clustering tools.

To better appreciate the curse of dimensionality in this context, consider how many distinct pages exist on a given web-server. For the purposes of this paper, the universe of distinct pages is limited to internal SAS web server pages, which is easily in excess of 10,000. Now consider moving outside the domain of a web-server and into the context of an ISP, where the distinct pages of all companies having web-servers becomes the universe and the number of levels becomes essentially infinite.

HOW MIGHT THE CURSE OF DIMENSIONALITY BE ADDRESSED?

From a general modeling perspective, reducing the number of levels of a categorical input often serves to add stability or robustness to a model using the input.

In many situations a natural, more general level of classification exists to reduce the number of levels in the categorical variable. For example, SKU numbers might be mapped into alcohol, consumer, and food products. Here the generalized detail information is preserved, in contrast to the worst-case scenario, where none of the detail information is retained and only the sum, count, or mean of the items placed in the market basket is used.

Often times a natural hierarchy doesn't exist or is simply not appropriate because it represents too much of an abstraction for the business purpose at hand. Suppose a catalog retailer is hoping to determine which items should go into each of 8 distinct catalogs. A product hierarchy might be considered; however, there often is no reason to expect item purchase patterns to parallel product hierarchies. That is, ideally all items that are commonly purchased in combination fall within the same catalog, but this is likely a different arrangement than that dictated by product hierarchies.

Similarly, suppose the web retailer is hoping to determine what items should be displayed when the banner-ad is clicked on by a particular surfer-profile. Ideally all items that are commonly purchased in combination would fall within the page that the banner ad links to, but this is likely a different arrangement than that dictated by product hierarchies.

RB INPUTS FOR MODELING TOOLS

MBA represents a means for establishing a reclassification scheme for reducing the number of levels of a categorical variable, and it is based upon purchase patterns.

There are at least three ways to use MBA to generate features for predictive models.

First, MBA can be used to identify interesting rules. In this scenario the dimensionality problem is of little or no relevance, for some exogenously existing business objective dictates interest in only, for example, one specific rule out of, for example, the thousands of other possible item combinations that exist. Here an "interesting" combination of items is epitomized with the notion that beer and diapers are allegedly purchased in combination after 10:00 P.M.

Second, the value of support or lift might be used as a criterion for creating RB inputs.

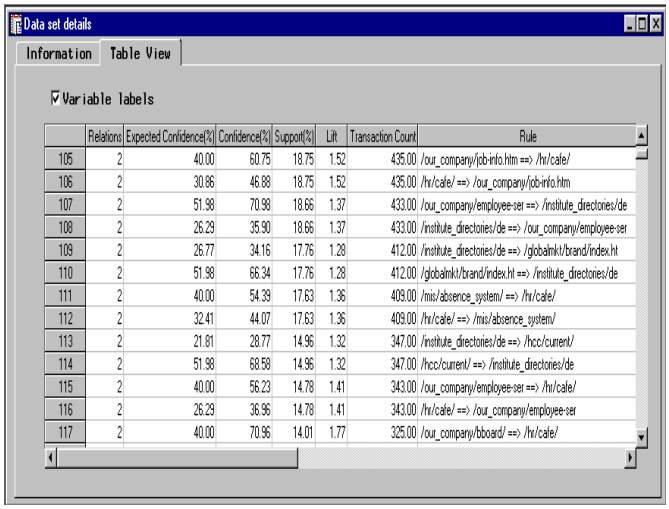
Third, the confidence statistics can be used as a criterion for a reclassification scheme to reduce the dimensionality of the categorical variable in the transactional table. The reclassified data can subsequently be transposed to collapse the

transactional table into a modeling table.

In the first case, the analyst creates a dummy variable that flags those individuals having the interesting rule or combination of items in their market basket, and the dummy variable is then added to an existing modeling table and used in modeling tools as either an input or a target. Further, it is entirely conceivable that more than one rule might be interesting, and a multitude of dummy variables, each flagging the folks having the corresponding rule of interest in their market basket, might be created and utilized by both modeling and clustering tools.

In the context of the person 10019, perhaps the marketer wants to flag those folks who commonly visit a certain combination of pages in an effort to target them with a banner ad trumpeting a special café offer; that is, the marketer seeks to add value to the product line with a new niche product and add revenue to the bottom line by increasing the life-time value of the customers in this market segment.

To illustrate, consider again the web-log file, which contains id numbers for SAS employees and the corresponding internal pages they visited. The corresponding MBA output is shown below.



	Relations	Expected Confidence(%)	Confidence(%)	Support(%)	Lift	Transaction Count	Rule
105	2	40.00	60.75	18.75	1.52	435.00	/our_company/job-info.htm ==> /hr/cafe/
106	2	30.86	46.88	18.75	1.52	435.00	/hr/cafe/ ==> /our_company/job-info.htm
107	2	51.98	70.98	18.66	1.37	433.00	/our_company/employee-ser ==> /institute_directories/de
108	2	26.29	35.90	18.66	1.37	433.00	/institute_directories/de ==> /our_company/employee-ser
109	2	26.77	34.16	17.76	1.28	412.00	/institute_directories/de ==> /globalmkt/brand/index.ht
110	2	51.98	66.34	17.76	1.28	412.00	/globalmkt/brand/index.ht ==> /institute_directories/de
111	2	40.00	54.39	17.63	1.36	409.00	/mis/absence_system/ ==> /hr/cafe/
112	2	32.41	44.07	17.63	1.36	409.00	/hr/cafe/ ==> /mis/absence_system/
113	2	21.81	28.77	14.96	1.32	347.00	/institute_directories/de ==> /hcc/current/
114	2	51.98	68.58	14.96	1.32	347.00	/hcc/current/ ==> /institute_directories/de
115	2	40.00	56.23	14.78	1.41	343.00	/our_company/employee-ser ==> /hr/cafe/
116	2	26.29	36.96	14.78	1.41	343.00	/hr/cafe/ ==> /our_company/employee-ser
117	2	40.00	70.96	14.01	1.77	325.00	/our_company/bboard/ ==> /hr/cafe/

From the first row of the output shown above, it follows that if a surfer visits the site represented by "/our_company/job-info.htm", then approximately 61% of the time they also visit the site represented by "/hr/café".

Is this an interesting rule? This particular rule might be interesting to the marketer endeavoring to boost the number of specials sold in the company cafeteria. Indeed, one obvious question is which employees might best be targeted with a campaign trumpeting a new lunch special? The lift value of 1.52 for the rule suggests that if a surfer visits the job-info page, they are 1.52 times more likely to visit the café page. What's more, the support statistic indicates that these pages occur in combination in 18.75% of the market baskets, suggesting that there may be sufficient critical mass to justify the costs associated with designing and executing a banner-ad campaign for this market niche.

SQL code is used to create a dummy variable corresponding to any given rule or rules of interest. The SQL code references a "Rules" table produced by SAS' MBA tool. The Rules table is used in conjunction with the original transactional source table to identify those id values having the rule of interest in their market basket. The SQL code is shown below.

```
create table source as
  select distinct id, path as item
  from source;
create table all_info as
  select source.id, rule, source.item as item1,   source2.item as
item3, conf, lift, support
  from source, source as source2, rules
  where source.id = source2.id
    and source.item = rules.item1
    and source2.item = rules.item3
    and set_size eq 2;
 create table identify as
  select id,1 as dummy
   from all_info
   where rule = '/our_company/job-info.htm ➔ /hr/café/';
```

This code creates, among others, a table named "All-info" containing the rules, the associated values of support, confidence, and lift, as well as the id values or people having each of the rules in their market basket.

The All-info table is then subset using the rule of interest to create the table "Identify", which contains all surfers having the rule of interest in their market basket, as well as the RB dummy variable, which is set to 1.The where-clause used in the query that creates the "Identify" table can be modified to select those id values for any given rule of interest or for any set of rules corresponding to a minimum value of support, confidence, or lift.

Now suppose the SAS has information gleaned from other sources of employee information, such as department, years employed, gender, etc. Then the "Identity" table can be joined with these other sources of information using the id column and subsequently used with modeling tools to develop a predictive model for targeting CAFE-INFO prospects with a campaign offer. In the augmented modeling table, the dummy variable will take a value of 1 if the surfer's market basket contains the rule JOB-INFO ➔ CAFE-INFO and a value of 0 otherwise.

In this scenario, one would use the new, RB feature as an input for developing the predictive model, using a separate dummy variable flagging all existing CAFE-INFO banner-ad clickers (i.e., those surfers who have clicked on, or better yet, accepted the offer trumpeted with the banner ad in the past) as the target.

Note that one cannot observe rules in a customer's market basket; instead, the analyst can only observe whether the rule's antecedent and consequent items exist in the customer's market basket. Thus, the rule A➔B, for the purpose of creating a dummy variable, is equivalent to A & B.

OTHER RB FEATURE CREATION CRITERIA

In contrast to specifying a particular rule of interest, MBA provides the analyst with at least three other criteria for creating dummy variables; namely, confidence, support, and lift.

USING CONFIDENCE AS A CRITERION FOR A RECLASSIFICATION SCHEME

For the purposes of converting a transactional table that has a categorical column with "too" many levels into a modeling table, the general approach is to reduce the number of levels of the categorical variable using a reclassification scheme and use transposition to generate the candidate inputs.

The confidence statistic can be implemented as a reclassification scheme by modifying the where clause used in the previous SQL code to select all rules having a specified minimum value of confidence.

One obvious criterion for reducing the dimensionality of a categorical variable having "too" many levels is to collapse all levels that correspond to a rule having a value of confidence of 100%, for such combinations occur together in every situation and can therefore be considered as a single item.

Consider, for example, the legendary rule, diapers ➔beer (or D➔B for short). And suppose that every time diapers are purchased beer is purchased. Then the rule D➔B would have a 100% confidence value, and it seems that the two levels "diapers" and "beer" could justifiably be collapsed into the new single level labeled "diapers➔beer." By replacing the two previous levels, "diapers" and "beer", with the one new level "diapers➔beer", it seems that the analyst reduces the number of levels of the categorical variable by one.

There is actually an additional condition to make the mapping suggested by the rule D➔B having a confidence value of 100% indeed reduce the dimensionality. Namely, both the primary rule D➔B and its opposite, B➔D, must each have a confidence value of 100% for the dimensionality to be reduced by one.

To illustrate, suppose the rule D➔B has a value of confidence of 100%, but B➔D has a value of confidence of 80%. The confidence value of 80% for the rule B➔D implies that there are market baskets containing beer without diapers. If these two values of confidence, 100% and 80%, exist and a dummy variable is created flagging all those folks having the D➔B rule in their market basket, then the number of levels of the categorical variable has remained the same. For we have mapped all occurrences of the value "diapers" into the value "diapers➔beer", thereby eliminating the value "diapers". However, because some market baskets contain beer but not diapers, the value "beer" must be retained to preserve accuracy.

When the confidence is 100% for both the primary rule and its opposite, then the RB mapping or reclassification scheme serves to reduce the number of levels of the original categorical variable by one, two, and three if the rule corresponds to, respectively, a two item, a three item, and a four item combination.

This knowledge is relevant because the only way to identify all individuals having, for example, the rule D➔B in their market basket is to see if they have both diapers and beer in their market basket, for one only observes that both diapers and beer are in an individual's market basket; one cannot look into a market basket and see "rules" or the logic that went into putting the items into the market basket. Thus, for the purpose of identifying all those people having a given rule in their market basket, D➔B is equivalent to D&B (i.e., diapers & beer).

But would the analyst ever want to collapse two or more levels when the corresponding rule has a value of confidence that is almost 100%? Might the analyst consider collapsing two or more levels into a single new RB level if the rule has a confidence value of, say, 90% or higher? After all, in the face of being overwhelmed with "too" many levels of a categorical variable, why would the analyst be opposed to introducing a little inaccuracy if the tradeoff is a sufficient reduction in the dimensionality of the categorical variable?

Suppose that the analyst decides to round up to 100% all values of confidence greater than or equal to, say, 90%. We might refer to this scheme as the round-up-confidence (RUC) reclassification scheme. Then if both the primary rule and it's opposite have a rounded up confidence value of 100%, the RUC reclassification scheme dictates that all antecedent (A) and consequent (B) items occurring in the rule and transactional table be mapped to the RB level,

A&B.

In the RUC reclassification scheme, some market baskets contain only the antecedent item or only the consequent item (i.e., without the other), but these items are still (inaccurately) mapped into the value "A&B". How badly is the information contained in the reclassified categorical variable prostituted by this scheme? For the purposes of this paper, this question remains open. It certainly seems an improvement upon what was previously referred to as the worst-case scenario, where no item detail information was retained in lieu of the sum, count, or mean for all levels for each id value.

USING SUPPORT OR LIFT AS A CRITERION FOR CREATING A RB DUMMY VARIABLE

As an alternative to choosing some particular rule of interest as the criterion for creating a RB dummy variable, the analyst might choose people based upon whether they have any rule in their market basket with a value of support greater than or equal to, say, 50, which is chosen solely for the purpose of illustration.

The path that the interpretation of such a dummy variable might take would be something similar to the following. If the support for the rule is 50, then the rule occurs in 50% of the market baskets. If in addition the confidence is 100% for both this rule and it's opposite, then a dummy variable created using this rule would be expected to have half the values equal one and half the values would equal zero.

Suppose, for example, fraudulent drug claims are being modeled, and that two of the columns in the claims table are id and drug. If a person claims two drugs, then the person has two rows in the claims table. One might then create potential new features using MBA. It is conceivable, for example, that a rule is produced that corresponds to what is often referred to as the AIDS cocktail or combination of drugs, for this cocktail has become a common treatment. Indeed, to the extent that doctors on the whole treat various diagnoses with the same drug combinations, we would expect the rules output from MBA to identify common drug treatments.

If fraudulent behavior is atypical, one might create a fraud predictor (dummy) variable based upon all rules having a level of support greater than or equal to, say, 5, for such a variable would serve to identify those individuals in the claims table having one or more of these rules in their market basket. This feature would then indicate typical behavior or drug combinations. And if the fraud predictor proved to be significant, the analyst would certainly want to add the newly discovered input to the previously used or "traditional" modeling table and develop a new and improved fraud detection model.

CONCLUSION

One of the primary objectives in data mining is to develop predictive models and improve the accuracy of existing predictive models. And one of the essential challenges toward this end lies in the discovery of new features or inputs to foster predictive accuracy.

A whole array of candidate inputs can be created from a transactional table via transposition. In practice, however, the curse of dimensionality often rears its ugly head. Here the confidence statistic can be used to develop a reclassification scheme to reduce the dimensionality of a categorical column prior to using transposition to create the candidate inputs.

In addition, we have seen how RB candidate inputs can be created from either a particular "interesting" rule or from a specified minimum level of support.

Ideally software would be sufficiently automated to find those RB inputs that are significantly related to a given target. Perhaps, however, trial and error is the only such means and expecting software to automate the process of rule based feature discovery is asking too much.

ACKNOWLEDGMENTS

I am grateful to Jon White for the log-file data, and for early feedback on the idea provided by Mark Stranieri, Will Potts, Rajen Doshi, and David Duling.

BIBLIOGRAPHY

[1] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami, Mining Association Rules between Sets of Items in Large Databases. ACM SIGMOD International Conference on Knowledge Discovery and Data Mining, SIGMOD 1993 Proceedings pp. 207-216

[2] Sergey Brin, Rajeev Motwani, Jeffery D Ullman, and Shalom Tsur. Dynamic Itemset Counting and Implication Rules for Market Basket Data. ACM SIGMOD International Conference on Management of Data, SIGMOD 1997 Proceedings, pp. 255-264.

[3] Roberto J. Bayardo Jr. and Rakesh Agrawal, Mining the Most Interesting Rules. ACM SIGMOD International Conference on Knowledge Discovery and Data Mining, SIGMOD 1999 Proceedings, pp. 145-154.

[4] Tom Brijs, Gilbert Swinnen, Koen Vanhoof, and Geert Wets, Using Association Rules for Product Assortment Decisions: A Case Study. ACM SIGMOD International Conference on Knowledge Discovery and Data Mining, SIGMOD 1999 Proceedings, pp. 254-260.