SAS South Central User Group

Contextualized Market Basket Analysis – How to learn more from your Point of Sale Data in Base SAS and SAS Enterprise Miner

Andrew Kramer, Louisiana State University

ABSTRACT

Recent advances in unsupervised learning have led both academics and private-sector data science teams to scan consumer market basket data, looking to create advanced predictive models such as recommender systems. However, these new statistical techniques fail to address the fundamental questions that a Market Basket Analysis (MBA) presents for any retailer: do these associations create profitable, long-term relationships with my valued customers?

This paper will address how Macro Variables, PROC SQL and Database steps can be used understand the profit implications of any retailer's Market Basket Analysis.

INTRODUCTION

With the advent and growth of POS systems to gather, organize, and store point of sale data, researchers began developing methods to discover patterns and useful insight from this new data source. Perhaps most influential was the Apriori Algorithm, originally proposed in the early 1990s, with the goal of saving computational power by not looking at all possible subsets of Stock Keeping Units(SKUs), but rather looking at the relationships between SKUs that occur commonly in the dataset. Retail stores can hold 20,000+ SKUs, but only a small percentage of those SKUs will result in meaningful sales. By identifying these meaningful SKUs, the algorithm can more efficiently sift through the data looking for the most important patterns.

This paper will cover the following categories of interest to both analysts and managers:

1. Why retailers need Contextualized MBAs
2. Preparing a data set for analysis
3. Analysis of Macro output
4. Technical explanation of Macro code

THE NEED FOR MARKET BASKET CONTEXT

The biggest issue that retailers face with MBAs is that common algorithms ignore several key parameters, such as Customer ID, Date, Quantity Purchased, and Price that many retailers live and die by. While it's not necessarily wrong to exclude these parameters the analysis, the interpretation can be very difficult without this additional context to back up the results. A manager can see there is a relationship, but it is hard to see why.

The retail industry has become well known for becoming an ever-consolidating landscape with razor thin profit margins, sometimes a low as 1%. In many supermarkets, sale items or dry grocery goods are often sold at a loss with the hope customers come in and purchase other specialty goods, often at higher markups. These relationships are likely to change in a time series fashion as a function of price, store positioning, sales and seasonality. Traditional MBAs cannot tell the whole story in retail setting, but rather we must consider how the associated items relate to each other, and to their market baskets as a whole to truly understand MBA significance.

PREPARING A DATA SET FOR ANALYSIS

The data set used in this analysis is from an anonymous Taiwanese retailer, consisting of a detailed sales history, from November 2000 to February 2001. While the data is somewhat outdated, the fundamentals of a relational database and POS technologies as utilized by retailers maintains largely unchanged. This paper will focus on the retailer's sales history from November 2000 to show how to set up and interpret the Macro. A small portion of the dataset is shown below for reference. Each row represents one SKU purchased by a customer.

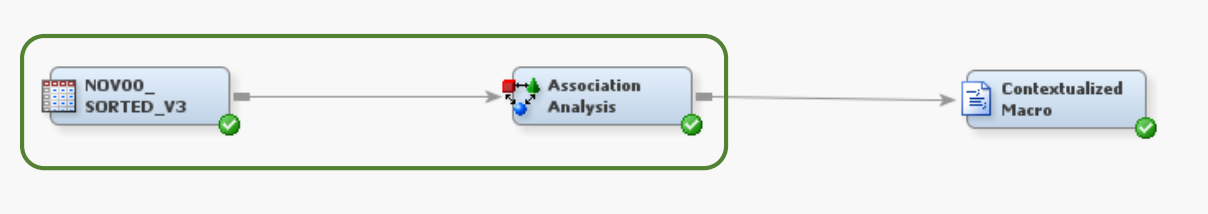
Sample Taiwanese Data Set

Obs	Transaction #	Date	Customer_ID	Age	Area	subclass	ProdId	AMT	Cost	SalePrice
1	1	01NOV2000	00038317	J	E	130315	4714981010038	2	56	48
2	1	01NOV2000	00038317	J	E	120105	4719090105002	1	28	28
3	2	01NOV2000	00045902	H	E	100304	4710147100018	1	24	28
4	2	01NOV2000	00045902	H	E	130204	4710088434692	1	114	119
5	2	01NOV2000	00045902	H	E	100511	4710594912028	6	210	313

Ensure a new dataset contains the following information before running the analysis in Enterprise Miner:

1. A "Target" variable in Enterprise Miner, most likely a SKU (Prodid in this example), or a subclass
2. An "ID" variable (Transaction # here) ordered sequentially with the earliest transaction having the smallest ID value. Each line in the database should represent one SKU purchased in one transaction
3. Variables representing quantity purchased, sale price, and unit cost to company. This information will be needed for the variable to populate correctly.

After importing the dataset into Enterprise Miner, we must run an association analysis to get preliminary results. We will then use a custom node to build the contextualized macro presented later in this papre that appends to the results of the Association Analysis Node

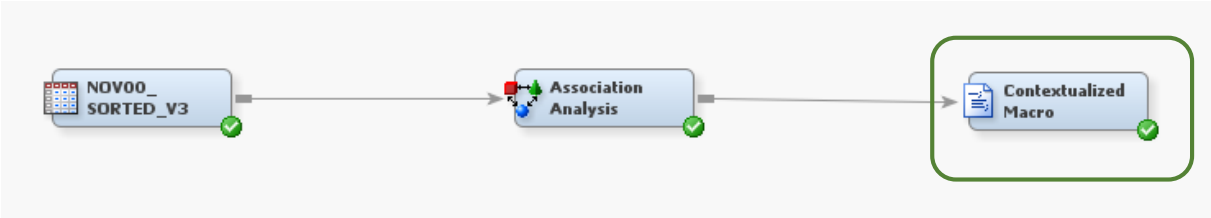


With the data set created, the information can now be imported into SAS Enterprise Miner and the Association Analysis node can now be run. The results from this node will feed into the macro presented in this paper and the macro will append the output. Expected Confidence, Confidence, Support, and Lift help to define the statistical relationship between the two items in both rules. In this paper, we will assume the algorithm has identified these statistical relationships, and look to understand the profit implications of the relationship. A sample result for the node is as follows:

EXP_CONF	CONF	SUPPORT	LIFT	COUNT	RULE	_LHAND	_RHAND
1.35	76.00	0.66	56.18	209.00	4710085120697 ==> 4710085120680	4710085120697	4710085120680
0.86	48.49	0.66	56.18	209.00	4710085120680 ==> 4710085120697	4710085120680	4710085120697
10.76	47.03	0.72	4.37	230.00	4711663700010 ==> 4714981010038	4711663700010	4714981010038
2.82	12.04	0.89	4.26	284.00	4711271000014 ==> 4710421090059	4711271000014	4710421090059
7.40	31.56	0.89	4.26	284.00	4710421090059 ==> 4711271000014	4710421090059	4711271000014

THE CONTEXTUALIZED MACRO

Having run the Association Analysis Node, we must connect a custom node to the results and insert the code for the Contextualized Macro.



The macro presented in this paper is designed to do the following steps:

1. **Item level profitability analysis:** The program will scan all the rules, and for each rule will look for all the transactions that contain both the SKUs in question. The macro will then determine if the combination of just those SKUs was profitable or not. For the same transactions, the macro will display the profit on the first SKU, the profit on just the second SKU, and two profit summaries representing the profit on just the two SKUs together, and the total market basket profit for all purchases in these transactions.

2. **Market basket profitability analysis:** The second part of the macro will print out more detailed information looking at the market baskets as a whole for all the transactions where both SKUs for the rule in question were purchased. This will allow comparison on how spending, profit, market basket size and price paid compare for profitable and non-profitable transactions. This is important because a company may lose money on the item-level analysis in #1, but this may be made up for large profits at the market-basket level from their best, most loyal customers.

MACRO OUTPUT AND ANALYSIS

When the code in the appendix is run in the custom node, the macro will print out a revised Enterprise Miner Association Analysis output displaying extra statistics for the two SKUs that make up each rule. The macro looks at each individual transaction where both items listed in the RULE category were purchased, and looks to see if the two items sold together were profitable or not. The MB profit looks at the profit or loss of the whole market basket for the transactions that purchased both items in the rule. As we can see, the last rule is of concern since the combination of both SKUs was almost always sold at a loss. A manager needs additional context to further investigate the relationship between the two SKUs.

RULE	profitable	Not Profitable	Percent Profitable	VarOneProfit	VarTwoProfit	Association Profit	MBProfit
4710085120697 ==> 4710085120680	116	93	0.5550	$3,526	$1,035	$4,561	$66,341
4710085120680 ==> 4710085120697	116	93	0.5550	$1,035	$3,526	$4,561	$66,341
4711663700010 ==> 4714981010038	4	226	0.0174	$134	$-11,510	$-11,376	$12,661
4711271000014 ==> 4710421090059	2	282	0.0070	$-5,084	$-8,213	$-13,297	$30,708
4710421090059 ==> 4711271000014	2	282	0.0070	$-8,213	$-5,084	$-13,297	$30,708

To provide extra context, the macro will print out additional information for each rule on a Market Basket level. It compares the market baskets as a whole, breaking them down based on profitable market baskets and non-profitable market baskets. It also provides the profit/loss from each group. Below is the additional information that the macro printed for rule number 5:

Profitable Transaction	N Obs	Variable	N	Mean	Std Dev	Minimum	Maximum
0	123	TotalSales	123	-38.89	21.09	-130.00	-1.00
		VariableInQuestionQty	123	5.60	1.24	1.00	12.00
		VariableInQuestionQty2	123	1.96	0.30	1.00	4.00
		BasketSize	123	10.83	4.17	4.00	28.00
		UniqueItemsPur	123	4.49	2.74	2.00	14.00
		VariableInQuestionProfit	123	-31.73	9.77	-36.00	0.00
		VariableInQuestionProfit2	123	-19.67	6.51	-80.00	-5.00
1	161	TotalSales	161	220.44	300.48	1.00	1946.00
		VariableInQuestionQty	161	5.12	1.79	1.00	12.00
		VariableInQuestionQty2	161	1.78	0.46	1.00	4.00
		BasketSize	161	23.64	11.24	5.00	60.00
		UniqueItemsPur	161	13.91	8.14	2.00	40.00
		VariableInQuestionProfit	161	-26.77	19.31	-36.00	144.00
		VariableInQuestionProfit2	161	-16.55	6.75	-40.00	2.00

ProfitableTransaction	TotalProfit
0	-4783
1	35491

As we can see, for the unprofitable transactions, we lose a lot of money, especially on the two SKUs in rule 5. However, for the profitable transactions, although we still lose money on the association, these customers buy over 13 more items and have an average of $220 in profit for the whole market basket. The association may seem unprofitable, but one can argue that it rewards loyal customers who end up spending more at the market basket level, making up for the money lost by those who cherry pick certain items

CODE ANALYSIS

The full code for the macro can be found in the appendix of this paper. The section below contains simplified code representing key parts of the macro that users can implement in Base SAS or SAS Enterprise Miner.

Extracting the SKUs for each rule

In order to loop through all the rules defined in the report, we need a macro to store the number of rules we have in the data step. This can be accomplished with the following code:

```
PROC SQL;
       SELECT count(_LHAND)
       INTO :rules
       FROM &EM_IMPORT_RULES;
QUIT;
```

The Macro &EM_IMPORT_RULES evokes the dataset that results from running an Association Analysis Node in Enterprise Miner. The custom macro in this paper will append to this macro

The macro then creates a way to extract the SKUs for each rule in the Association Analysis output. The code shown here is designed to work for only two SKUs per rule, but can be modified to accommodate for more SKUs:

```
PROC SQL;
       CREATE Table varlist AS
       SELECT _LHAND, _RHAND
       FROM &EM_IMPORT_RULES;
QUIT;
```

PROC SQL saves the left hand rule (_LHAND) and the right-hand rule (_RHAND) in the table varlist.

```
DATA _NULL_;
      SET VARLIST;
      id=_N_;
      CALL SYMPUT('ids'||left(put(id,20.)), _LHAND);
      CALL SYMPUT('idst'||left(put(id,20.)), _RHAND);
RUN;
```

Using a _NULL_ data step, two macro variables are created: ids (For left-hand SKU) and idst (For right hand SKU). Each macro takes the form of ids1-ids5 or idst1-idst5, since there are 5 rules in our dataset. Calling on &ids1 will extract the left-hand SKU for the first rule.

Crawling the Database

The next step in the macro is a program to crawl the transactional dataset and collect different summary statistics that we need to add context to our Market Basket Analysis results. This code is extracted from a macro used in a DO loop, allowing us to execute the code one time for each rule produced from the Enterprise Miner program. A simplified version of the code is shown below with four unique steps:

```
DATA Nov00_Analyzed;
      SET &EM_IMPORT_TRANSACTION;
      BY ID;
      IF first.ID THEN DO;
            TotalProfit=0;
            BasketSize=0;
            VariableInQuestionProfit=0;
1           VariableInQuestionProfit2=0;
            VariableInQuestionQty=0;
            VariableInQuestionQty2=0;
      END;

      IF prodid="&&ids&i" THEN DO;
            VariableInQuestionProfit+(AMT*(SalePrice-Asset));
2           VariableInQuestionQty+1;
            END;

      IF prodid = "&&idst&i" THEN DO;
            VariableInQuestionProfit2+(AMT*(SalePrice-Asset));
            VariableInQuestion2Qty+1;
            END;

      TotalProfit+(Amt*(SalePrice-Asset));
           Basketsize+AMT;

3     IF TotalSales GT 0 THEN ProfitableTransaction=1;
            ELSE ProfitableTransaction=0;

      IF SUM(VariableInQuestionProfit, VariableInQuestionProfit2) GT 0
           THEN ProfitableAssoc=1;
           ELSE ProfitableAssoc=0;

4     IF last.ID and VariableInQuestionQty GT 0 AND VariableInQuestionQty2
            GT 0;
```

Below is a summary of each of the steps:

1. Set counter variables to zero at the start of each new transaction ID
2. Evoke the macro variables “ids” and “idst” for the correct loop count as defined by “i”. If the pointer reads a row showing a purchase of one of the two SKUs evoked by the macro, the code will generate summary statistics for the SKU
3. Defines variables representing the profitability and basket size of each transaction
4. Extracts the calculated fields into a new data set containing one row for each transaction where both items in the rule were purchased

Once the database is crawled, PROC SQL is used to extract the information pertinent to the macro and save it in a table called "ProfitPerc". The code below extracts the count of profitable associations (number of transactions where both items were purchased at a profit) and the total profit for each market basket selected from the PROC SQL statement. A similar code with more items selected is used in the macro in the Appendix.

```
PROC SQL;
       CREATE TABLE ProfitPerc AS
       SELECT sum(profitableAssoc) as profitable, sum(TotalProfit) AS
              MBProfit
       FROM nov00_analyzed;
QUIT;
```

The macro is designed to do multiple loops, so it needs a way to append the results from the above PROC SQL statement to an empty dataset and build upon it with subsequent loops. The code in the appendix creates an empty dataset and uses the text below to append the table ProfitPerc to "dataset" after each iteration of the DO loop.

```
DATA dataset;
       SET dataset ProfitPerc;
RUN;
```

Do Loops in a Macro

In the appendix, the code listed above is modified into four different macros. The DO loop shown below instructs the SAS program to loop once for reach rule outputted in the Association Analysis tab.

```
%MACRO AllSubs;
       %DO i=1 %TO &rules; /*Macro storing the number of rules*/
              %DsCrawl
              %ProfitTrans /*Four Macros Evoking the code described above*/
              %DatasetUse
              %REPORT
       %END;
%MEND AllSubs;
```

CONCLUSIONS

The purpose of this macro is show the profit implications of an Association Analysis for retailers. With the additional information provided by the presented macro, a manager can quickly see which relationships are significant as well as the implications for his or her business. Managers should use this macro as part of a larger decision making process in which the company looks to reward loyal customers and build a profitable business in the long term. For those without Enterprise Miner, analysts can takeaway concrete programming skills that can be incorporated into their analyses and processes in Base SAS or other languages. Even as the internet becomes more prevalent for retail sales, brick and mortar stores are here to stay and companies must be able to leverage Point of Sale data to achieve supply chain efficiencies, satisfy their best customers, and build long-term profitability.

CONTACT INFORMATION:

Please contact the author for comments or questions:

Name: Andrew Kramer
Email: Andrew.kramer526@gmail.com

REFERENCES

Faron, M & Chakraborty, G. (2012). Easily Add Significance Testing to your Market Basket Analysis in SAS Enterprise Miner. From the SAS Global Forum 2012.

First, S & Ronk, K. SAS Macro Variables and Simple Macro Programs. SUGI 30 Hands-on Workshop

Lewandowski, D. (2008). A step-by-step Introduction to PROC REPORT. SAS Global Forum 2008.

McGowan, Kevin & Spruell, Brian. Proc SQL Tips and Techniques – How to get the most out of your queries.

APPENDIX – MACRO CODE

```
************************************************************
* Program Purpose = Add contextual data to the results of an
Association Analysis Node in Enterprise Miner*

* ADD THIS CODE TO THE "SAS CODE" NODE UNDER UTILITY IN
ENTERPRISE MINER AND CONNECT IT TO THE PROCEEDING ASSOCIATION
ANALYSIS NODE*
************************************************************;
************************************************************;

************************************************************
* Fill in Macros according to their definition in your data*
************************************************************;
%LET IDvar=ID; /*Insert the name for your ID Variable in EM*/
%LET Target=prodid; /*Insert variable name of target variable*/
%LET Quantity=AMT; /*Insert variable name of unit quantity purchased*/
%LET Price=SalePrice; /*Insert variable for unit sale price*/
%LET Cost=Asset; /*Insert variable name for unit cost to company*/


************************************************************
* Code to extract the number of rules and a macro variable
to extract the left and right side rules from the Enterprise
Miner Association Analysis Node Output*
************************************************************;
PROC SQL;
      SELECT count(_LHAND)
      INTO :rules
      FROM &EM_IMPORT_RULES;
QUIT;

PROC SQL;
      CREATE TABLE varlist AS
      SELECT _LHAND, _RHAND
      FROM &EM_IMPORT_RULES;
QUIT;

DATA _NULL_;
      SET VARLIST;
      id=_N_;
      CALL SYMPUT('ids'||left(put(id,20.)), _LHAND);
      CALL SYMPUT('idst'||left(put(id,20.)), _RHAND);
RUN;

************************************************************
* This code creates an empty dataset to append Macro Results *
************************************************************;
DATA dataset;
      SET varlist;
            IF &target eq " ";
            IF &target eq " " THEN DELETE;
      A=" ";
      KEEP A;
RUN;
```

```
*********************************************************
* This macro crawls the dataset and generates specific summary
statistics for all transactions, and creates special statistics
for the transactions that contain both SKUs in the Enterprise Miner Rules.
*********************************************************;
%MACRO DsCrawl;

      Data Nov00_Analyzed(DROP=subclass &target &Quantity &cost &price);
            SET &EM_IMPORT_TRANSACTION;
            BY &IDvar;
            IF FIRST.&IDvar THEN DO;
                  TotalProfit=0;
                  ProfitableTransaction=0;
                  VariableInQuestionQty=0;
                  VariableInQuestionQty2=0;
                  BasketSize=0;
                  UniqueItemsPur=0;
                  VariableInQuestionProfit=0;
                  VariableInQuestionProfit2=0;
            END;


            IF &target="&&ids&i" THEN DO;
                  VariableInQuestionQty+&Quantity;
                  VariableInQuestionProfit+(&Quantity*(&price-&cost));
            END;


            IF &target = "&&idst&i" THEN DO;
                  VariableInQuestionQty2+&Quantity;
                  VariableInQuestionProfit2+(&Quantity*(&price-&cost));
            END;


            TotalProfit+(&Quantity*(&price-&cost));
            Basketsize+&Quantity;
            UniqueItemsPur+1;

            IF TotalProfit gt 0 THEN ProfitableTransaction=1;
                  ELSE ProfitableTransaction=0;

            IF SUM(VariableInQuestionProfit, VariableInQuestionProfit2)
                  gt 0 THEN ProfitableAssoc=1;
                  ELSE ProfitableAssoc=0;

            IF LAST.&IDvar and VariableInQuestionQty gt 0 AND
                  VariableInQuestionQty2 gt 0;

            RUN;

%MEND DsCrawl;
```

```
*************************************************************
* Macro to calculate summary statistics from the output of
the %DSCRAWL Macro*
*************************************************************;
%MACRO ProfitTrans;
      PROC SQL;
            CREATE TABLE ProfitPerc AS
            SELECT SUM(profitableAssoc) AS profitable,

                  (COUNT(profitableAssoc)-SUM(profitableAssoc))
                   AS NotProfitable,

                  (SUM(profitableAssoc)/COUNT(profitableAssoc)) AS
                   PercentProfitable,

                   SUM(VariableInQuestionProfit) AS VarOneProfit,

                   SUM(variableinquestionprofit2) AS VarTwoProfit,


(SUM(variableinquestionprofit)+sum(variableinquestionprofit2))
                  AS AssociationProfit,

                  SUM(TotalProfit) AS MBProfit

            FROM nov00_analyzed;
      QUIT;
%MEND ProfitTrans;
*************************************************************
* MACRO TO append the results from the ProfitTrans macro
to the empty dataset created above. Will create a dataset
where the number of rows equals the number of rules form Enterprise Miner*
*************************************************************;
%Macro DatasetUse;
      DATA dataset;
           SET dataset ProfitPerc;
      RUN;
%MEND DatasetUse;
*************************************************************
* MACRO to create summary data for each iteration of the
DO loop in the %AllSubs macro below*
*************************************************************;
%MACRO REPORT;
      PROC MEANS DATA=Nov00_Analyzed MAXDEC=2;
            VAR TotalProfit VariableInQuestionQty VariableInQuestionQty2
                BasketSize UniqueItemsPur VariableInQuestionProfit
                VariableInQuestionProfit2;
            TITLE "Report for rule &i";
            CLASS ProfitableTransaction;
      RUN;

      PROC REPORT DATA=Nov00_Analyzed;
            COLUMN ProfitableTransaction Totalprofit;
            DEFINE ProfitableTransaction/Group;
            DEFINE TotalProfit/analysis SUM;
      RUN;
%MEND;
```

```
************************************************************
* MACRO instructing the macros defined above to iterate i
number of times where i = the number of rules as defined in
the Association Analysis Output*
************************************************************;
%MACRO AllSubs;
      %DO i=1 %TO &rules; /*&number;/*&counts;*/
            %DsCrawl
            %ProfitTrans
            %DatasetUse
            %REPORT
      %END;
%MEND AllSubs;
%AllSubs

************************************************************
*  Appended output produced by the Association Analysis node
with the defined summary statistics in this program*
************************************************************;
DATA dataset2;
      MERGE &EM_IMPORT_RULES dataset;
      DROP A ITEM4 ITEM5 transpose SET_SIZE _LHAND
           _RHAND ITEM1 ITEM2 ITEM3 index;
      FORMAT PercentProfitable 10.4 MBProfit varoneprofit
             vartwoprofit AssociationProfit dollar15.;
RUN;

PROC PRINT DATA=dataset2;
RUN;
```