Paper 257-25

A Methodology for Customer Segmentation Using Existing Product Category Schemes and The SAS® System

Christopher S. Andrews, DiaLogos, Inc., Boston, MA

Abstract:

Nearly every retailer (or e-tailer) has a methodology for categorizing the products that they sell. However, most of them do not have a clearly defined system (or any system for that matter) for categorizing their customers by the types of products they buy. This is unfortunate since, in the rapidly evolving world of customer relationship management, this level of customer understanding is critical. This paper will develop a methodology for turning product categorization on its ear (so to speak) and into a customer segmentation scheme that is not only straightforward in its production, but also simple in its interpretation.

In addition to the development of customer segments, this paper will also present methods for evaluating the performance of segments from a financial perspective. Enhanced targeting is a natural consequence of this, as well as similar evaluation techniques that will suggest themselves.

The paper is designed to be 'soup-to-nuts' covering everything from the data manipulation necessary to begin the analysis, all the way through to the financial calculations used to evaluate the segmentation performance. As such, the PROC SUMMARY and PROC SQL from BASE SAS will be prominently utilized, as will PROC FASTCLUS from SAS STAT.

The Issue:

In most retail marketing shops the focus is product centric. That is, customers get offers based on what they bought most recently, or sometimes what they bought only once. The problem with this type of marketing is that it is highly susceptible to unusual or seasonal purchases (such as gifts). In the brick and mortar world this can be annoying to customers who receive mail offers for products they have no interest in buying. In the e-commerce world, this can be devastating. Because of the relative ease with which customers can opt-out of all future email solicitations, an e-commerce company that does not target effectively will find it's lifeblood list of customers to whom it may solicit rapidly dwindling. One solution to this is to develop a profile of the types of products that customers tend to buy.

Pragmatic Segment Development:

One way to develop profiles of customers is to examine, individually, all the products that each customer has purchased over his/her lifetime of interaction with the company. While this approach is certainly thorough, it has the drawback of not being very easy to use since the entire order history of each customer under consideration must be scrutinized each time a new campaign is run. Furthermore, relatively new customers are not treated well under this scenario since they have little history on which to base offer generation. Another approach, which this paper advocates, assumes that while all customers are certainly different, there exist only a few broad categories in terms of the type of items customers tend to purchase.

By leveraging all available customer order history, general segments can be developed which can then be used to predict the profile of new customers. This scheme is also easy to use since only a few customer segments and their corresponding profiles need be considered.

Hypothetical Example:

The easiest way to understand the process proposed herein is through an example. Let us suppose that the online camera store 'Shutterbugs' wishes to develop a profile of the types of customers that shop at their site. They currently have a system for classifying their products through a 3-byte field stored in their database; The first byte indicates the high-level type of product, the second it's brand, and the third indicates the specific kind of product. That is, every product can be completely defined by these 3 hierarchical qualifiers. The possible values for these bytes are shown in Table 1. Note that this example is intentionally over-simplified for ease of presentation. Actual schemes may be significantly more complicated.

Table 1: Shutterbugs' product category scheme

Byte 1	Byte 2	Byte 3
Accessories (A)	Canyon (C)	Camera (C)
Hardware (H)	Fungi (F)	Digital Camera (D)
Image Media (I)	Independent (I)	Standard Film (F)
	Kodiak (K)	Lens (L)
	Polaris (P)	Tripod (P)
		Scanner (S)
		Digital Storage (T)

So, for example, a Polaris digital camera would be classified as a HPD.

This may be a great way to classify products, but it doesn’t tell much about the customers buying them. Nor does it lend itself to any statistical analysis since there are only three variables, none of which is quantitative. However, a little creativity reveals a simple way to rectify both of these issues.

We can now transpose the category scheme so that every possible *value* in each byte becomes it’s own variable. To continue this customer centric idea, instead of having products as rows, we now have customers. Each of the newly generated variables is now the count of the number of items the customer purchased that fell into that category value. In summary, we have transformed a table that consisted of rows of items with columns of categories populated by characters, into one of rows of customers with columns of category values populated by integers. An example will help. Table 2 shows the data as it currently exists in the database, and Table 3 shows the transformed data where the columns ‘c#_?’ stand for the count of items purchased that fell into the value ‘?’ of category (byte position) ‘#’.

Table 2: An example of Shutterbugs' order history

customer_id	product_id	product_category
3	1	AFL
3	10	AFP
4	3	APS
1	4	HCD
4	4	HCD
3	5	HFC
1	6	HFD
2	6	HFD
3	6	HFD
4	7	HIT
1	2	HKC
1	8	HPC
3	9	IFF
5	9	IFF
5	11	IKF
5	12	IPF

Table 3: The transformed order history data from Table 2

customer_id	c1_A	c1_H	c1_I
1	0	4	0
2	0	1	0
3	2	2	1
4	1	2	0
5	0	0	3

. . .

c2_C	c2_F	c2_I	c2_K	c2_P
1	1	0	1	1
0	1	0	0	0
0	5	0	0	0
1	0	1	0	1
0	1	0	1	1

. . .

c3_C	c3_D	c3_F	c3_L	c3_P	c3_S	c3_T
2	2	0	0	0	0	0
0	1	0	0	0	0	0
1	1	1	1	1	0	0
0	1	0	0	0	1	1
0	0	3	0	0	0	0

While transforming the data from Table 2 into Table 3 is probably best left to a data engineer or DBA, the following is a brute force SAS code for conducting the transformation:

```
data in1;
  infile 'c:\ord_hist.csv' dlm = ',';
  input cust_id prod_id prod_cat $;
  cat1 = substr(prod_cat,1,1);
  cat2 = substr(prod_cat,2,1);
  cat3 = substr(prod_cat,3,1);
  count = 1;
run;
proc sort data = in1 out = cat1;
  by cust_id cat1;
run;
proc summary data = cat1 sum;
  var count;
  by cust_id cat1;
  output out = in1cat1 sum = cnt;
run;
```

```
data in1cat1;
  set in1cat1;
  if      cat1 = 'A' then c1_A = cnt;
  else if cat1 = 'H' then c1_H = cnt;
  else if cat1 = 'I' then c1_I = cnt;
run;
proc summary data = in1cat1 sum;
  var c1_A c1_H c1_I;
  by cust_id;
output out = in1cat1 sum = c1_A c1_H c1_I;
run;
```

This results in the following output…

```
OBS   CUST_ID   _TYPE_   _FREQ_   C1_A   C1_H   C1_I

 1       1        0        1        .      4      .
 2       2        0        1        .      1      .
 3       3        0        3        2      2      1
 4       4        0        2        1      2      .
 5       5        0        1        .      .      3
```

…which is the contents of the SAS dataset in1cat1, and is exactly the category 1 section of Table 3. Note that there are missing values in the data set which will need to be set to zero prior to further analysis.

This code generates only the results for category 1, but the others follow similarly. Once all three data sets have been generated, combine them using a 'set' statement to obtain the Table 3 in its entirety.

Scaling The Data:

The final step before we can sick the FASTCLUS algorithm on our data is to scale it. This is important since we are trying to find patterns in the *types* of products rather than the *quantity* of products that customers tend to buy. To see where this can be a problem let us examine the purchases that customers #1 & #2 have made with respect to category 1. Here, customer #1 has the vector (0,4,0) and customer #2 has the vector (0,1,0). Now, if we were to compare these customers based solely on this vector (in Euclidean space, as FASTCLUS does) they would look quite different. However, the fact is that all the products they have ever purchased fall under 'H' in category 1 – which makes them quite similar in terms of the products they tend to buy.

There are probably many ways to scale these vectors to make them reflect this similarity. The method that this paper endorses is to simply change the counts into proportions by dividing each entry in the vector by the total items purchased by the individual. If this scaling is done for the example at hand, customer #2 retains the same vector (0,1,0) but now customer #1's vector becomes (0,4,0)*(1/4) = (0,1,0) which, as desired, makes both customers look identical in terms of their category 1 purchases.

Clustering the Data – Step I, Choosing an Appropriate Technique and Cohort:

Once all the above processing has been completed, we are now ready to run the cluster analysis. Since there will usually be a very large amount of data, PROC FASTCLUS is the SAS procedure recommended. However, should only small amounts of data be available, the flexibility provided by PROC CLUSTER may be a better alternative.

Next, a cohort of customers must be chosen. This should be done so that the behavior of 'typical' customers may be observed. One way to approach this is to pick a time frame, select all those customers who made their first purchases within that time frame, and then follow these customers to the present day. This allows us to observe customer behavior throughout their life cycle. If instead we were to simply take a sample of all customers, both customers who are one-time purchasers and regular customers would be thrown together in the mix. This puts the clustering algorithm at a disadvantage since not all observations have had the same opportunity to exhibit the purchase behavior that is being targeted.

Since PROC FASTCLUS requires the user to specify the number of clusters to be generated with each run, it is recommended that analysts run the procedure multiple times with different numbers of target clusters. By doing this, and keeping track of where each observation was placed, the analyst can see how each cluster is expanding. Choosing the 'correct' number of clusters is more an art than a science, and is highly dependent on the way that the clusters will be used. However, conventional wisdom states that at a minimum 3 solutions (with 2, 4 and 8 cluster targets) should be run. The results of these runs can then be used to determine if further runs are necessary, and should give the analyst some indication of how many clusters should be in the final solution.

The general code for running the clustering procedure follows:

```
proc fastclus data=in1all maxc=4 maxiter=20 out = in1clus4 mean = in1mean4;
               var pc1_A pc1_H pc1_I
                   pc2_C pc2_F pc2_I pc2_K pc2_P
                   pc3_C pc3_D pc3_F pc3_L pc3_P pc3_S pc3_T;
```

There are a few things to note here: the maxc option specifies the number of clusters to be generated – in this case 4. The maxiter option specifies the maximum number of iterations through the data allowed. The default depends on some other options, but can be as low as 1. This is potentially dangerous since the procedure will happily produce error-free output, even when convergence has not been reached. Therefore, it is usually a good idea to set this to a high number; also, be sure that convergence occurred by checking the log file. The out option writes out the SAS data set in1clus4 that contains all the original data with the variables cluster (the cluster number to which the observation was assigned) and distance (the Euclidean distance that the observation is from the cluster center). The mean option creates the SAS data set in1mean4 that contains the clusters (as observations) and various statistics (as variables). Of particular importance are the mean values of the input variables – taken together, these represent the cluster centers.

Clustering the Data – Step II, Results Validation and Other Caveats:
Unfortunately, cluster analysis in general, and the variation performed by FASTCLUS in particular, are not very robust procedures. FASTCLUS tends to be highly influenced by the initial seeds. In a typical situation, the initial seeds are just the first observations that the procedure encounters. One way to validate results is to separate the input file into a few groups and run each of them through the algorithm independently. If all the ending solutions look similar, we can be fairly confident that the clusters are indeed legitimate representations of what is happening in the data.

Decisions must also be made as to what is the correct number of clusters. Deciding this is more an art than a science. As previously mentioned, several passes through the data should be made specifying a different number of clusters each time. From here, each of the cluster centers should be scrutinized. It may also be useful to join the resulting output data sets using PROC SQL to observe the way individuals are being separated out as the number of clusters increases. One key to knowing when to stop is when extremely small clusters start to be generated. This may indicate that the algorithm is being asked to find too many clusters and is becoming too granular as a result. Much also depends on the objective of the cluster analysis. This will often be a good guide as to what is the correct number of clusters.

One final comment about validation: Cluster analysis, as a multivariate technique, has the distinct disadvantage of not actually having a right answer. In contrast, other techniques, such as regression, have a target variable upon which the model can be trained. Unfortunately, for cluster development there is no methodology to tell whether or not you have produced the correct solution; indeed, there is no correct solution. Here, the proof is in the pudding. The truest test of whether or not the clusters work is in using them to distinguish other behaviors.

Defining, Evaluating, and Using the Clusters:
Let's return to our example. After running the Shutterbugs data through the above-described process, a 4-cluster solution was determined. Based on the cluster centers, the following intuitive naming conventions were given to these clusters:

1.	Fungi –Heads	(customer #3)
2.	Dual Band Independents	(customer #1)
3.	Digital Warriors	(customer #2 and customer #4)
4.	Analog Film Folks	(customer #5)

To give you an idea what these clusters mean, each of the 5 customers in our simplified example has been assigned to a cluster in the parentheses above. Notice, for example, that customer #3 has made widely different category purchases (accessories, hardware, digital cameras, regular cameras, etc.). But the one common theme in these purchases is that all of these products were made by the Fungi Corporation. The fact that our clustering pulled this out as a cluster means that it was a fairly common purchasing pattern. The other cluster definitions follow through similar reasoning. One side note here involves the classification of customer #2. The issue here is that this customer has made only one purchase from Shutterbugs. Classifying this customer becomes a challenge due to lack of information. However, the algorithm has chosen to place this customer in with the Digital Warriors. This classification may change over time as the customer makes more purchases, but at the moment, this is the algorithm's best guess.

Next we address evaluation and use. The first step is to compute the financial differences between the clusters. To do this, keep the same cohort of customers and compute the revenue generated by each individual. Then separate out these individuals by their cluster assignment. This results in the following table:

Table 4: Basic Shutterbugs cluster metrics

cluster	count	population percent	revenue	revenue/customer	percent revenue
Fungi - Heads	1,027	16.58%	$ 522,749.55	$ 509.01	18.44%
Dual Band Independents	2,701	43.59%	$1,309,989.90	$ 485.00	46.21%
Digital Warriors	892	14.40%	$ 707,405.35	$ 793.06	24.95%
Analog Film Folks	1,576	25.44%	$ 294,688.85	$ 186.99	10.40%
total	6,196	100.00%	$2,834,833.65	$ 457.53	100.00%

There are several things that can be inferred about the clusters based on this chart. One of the most obvious things is that Digital Warriors, while representing only about 15% of Shutterbugs' customers, make up nearly 25% of their total revenue. Conversely, Analog Film Folks make up over 25% of the population, but account for barely more than 10% of overall revenue. One immediate take away from this is that Shutterbugs may want to focus more on digital photography and less on its traditional lines. Also, in an effort to not alienate its loyal customers, it may be worthwhile to try to migrate the analog folks into digital folks by sending them targeted promotions.

Improved campaign targeting is a natural consequence of these cluster definitions. For example, if Shutterbugs' previously sent email newsletters to all its customers, it can now tailor these messages to the cluster's specific purchase behavior. But this is only the beginning of the process of understanding Shutterbug's customers. While it is only natural to think sending analog film promotions to Digital Warriors will be a miserable failure, this needs to be tested. Furthermore, these segments will evolve over time. Careful scrutiny of the clusters and how they change over time needs to be done. For example, if Shutterbugs is able to migrate its analog users over into the more profitable digital area, its customer base will have dramatically changed. At this point it may be necessary to start the whole process over from scratch.

Conclusions:

Customer segmentation is a powerful and increasingly essential tool for customer relationship management. With little effort, retailers (and e-tailers) can turn product categorization schemes into a full-blown customer facing segmentation. From here, through targeted campaigns and test and learn processes, the segmentations can be refined and updated so they continually reflect the customer base of any company.

Contact Information:

Christopher Andrews
Senior Consultant, Metrics & Intelligence
DiaLogos, Inc.
12 Farnsworth Street
Boston, MA 02210

(p) 617-357-4722 x170
(f) 617-357-4727
(e) candrews@dialogos.com
(w) www.dialogos.com