

Paper 068-2013

Product Affinity Segmentation Using the Doughnut Clustering Approach

Darius Baer, Ph.D., SAS Institute Inc., Cary, NC

Goutam Chakraborty, Ph.D., Oklahoma State University, Stillwater, OK

ABSTRACT

Product affinity means natural liking of customers for products. Product affinity segmentation divides customers into groups based on purchased products. While conceptually appealing to marketers and business analysts, it is difficult to implement in practice. Use of product-level data often yields inappropriate solutions such as one large segment and many tiny segments. Standard transformations such as logarithms or quantiles have not solved this problem. In this paper, we demonstrate how a combination of Softmax transformations with a “Doughnut” clustering approach (single central cluster) results in more evenly-sized product affinity segments for 30,000 customers of a B2B (business-to-business) company. The affinity segments show meaningful differences in product buying patterns across the customer base, and can be used for identifying cross-selling and up-selling opportunities. The segments are further profiled using customers’ background variables to provide deeper business insights.

INTRODUCTION

Segmentation is the first building block of any marketing strategy. The basic idea behind segmentation is to divide customers into groups such that within a group, customers are very similar based on certain selected characteristics (variables), and between the groups customers are very dissimilar based on the same set of characteristics (variables). Customers can be segmented using many variables such as geographic, demographic, attitudinal and behavioral (transactional) variables. Broadly speaking, there are two types of segmentation - foundation segmentation and targeting segmentation. Foundation segmentation carried out at the corporate level of a company is used to deliver consistent customer treatments and to create focus for a company’s long-term strategy. In such segmentations, all customers are usually included and each customer can fall into only one segment. Targeting segmentation, on the other hand, is often more useful for specific marketing programs and campaigns. Typically targeting segmentation is done using transactional variables and not all customers may be included in targeting segments. Product affinity segmentation is an example of such targeting segmentations.

UNDERSTANDING PRODUCT AFFINITY

The word affinity means “a natural liking or sympathy for someone or something.” Marketers use the term product affinity to loosely denote a natural liking between customers and the different products they buy. For any reasonable sized organization that has different types of products in its product line, the idea of a natural liking between its products and its customers is intuitively very appealing because it suggests the potential for segmenting customers based on their natural liking of different products. Product affinity segmentation uses customers’ purchases measured in revenue spent in each product class or the number of units purchased in each product class as segmentation variables. The goal is to find segments of customers such that within a segment, customers’ purchase patterns are similar with regards to the products they buy, but between the segments the purchase patterns are quite different. Once product affinity segments are identified, marketers can target a few of the segments selectively with customized special offers and promotions in product classes that are uniquely appealing to those segment members. In theory, such targeting should lead to more cross-selling and up-selling opportunities and enhanced business metrics such as market share and/or share of wallet.

There are three approaches for product affinity type segmentations. First, customers can be segmented using demographics or other aggregated transaction variables and then product-level

purchase data can be overlaid on top of those segments to give marketers additional insights into the segments with respect to their purchase patterns across product classes. Second, Market Basket Analysis using POS (point-of-sales) data can be used to understand what products tend to be bought together by customers during a shopping trip to a retail store. Third, product level data can be used directly in clustering algorithms to find product affinity segments. In this paper, we focus on the third approach as our research has suggested that this approach provides the most powerful and profitable information for marketing campaigns and communications with customers.

CHALLENGES TO CREATING SEGMENTS FOR PRODUCT AFFINITY

While finding product affinity segments sounds simple in concept, to execute it properly often poses enormous challenges as explained below. First, product-level data whether measured in number of units or revenue often has extremely high right-skewness and large kurtosis. This happens because in most cases, there are few customers who buy a lot in a product class while most others buy very little. Such patterns often create major problems in segmentation because the commonly used clustering algorithms perform poorly in the presence of such extreme skewness and high kurtosis. Of course, the recommended solution to such problems include transformations such as applying logarithms to bring the data shape back to a normal (or, uniform) distribution. The problem is that log transformation often fails to correct for such extreme distributions and often specialized transformations (such as Softmax) along with winsorizing (truncating) extreme values are needed to get any meaningful result. Even with the specialized transformations and winsorizing, sometimes segments emerge that are too disproportionate in sizes such as a single large cluster with many tiny clusters. Such solutions are deemed unsuitable by business executives for investing resources to develop targeted promotions.

Second, product level data for a customer often has a lot of zero values in many product classes. Large number of zero values pose another problem both methodologically (from clustering algorithm's ability to separate groups) and substantively (once we find segments to interpret those segments).

PRODUCT AFFINITY SEGMENTATION APPROACH

This paper is designed to be a 'soup-to-nuts' solution of product affinity segmentation. It covers everything from the data manipulation and transformation to begin the analysis to repeatedly running the clustering algorithm to produce good separation in the data space all the way to interpreting and profiling the affinity segments. SAS® procedures used in this paper include PROC UNIVARIATE, PROC STDIZE, and PROC FASTCLUS along with a fair amount of data step programming. We use data from a real B2B (business-to-business) catalogue company to give readers an idea of the nature of the problem. The data are summarized transaction counts for each customer for each product class over a period of three years. The company is located in the southeastern United States and has an active customer base of more than 100,000 customers. For the purpose of this paper, we used with a dataset of 30,000 customers selected at random having purchase records for 8 product classes.

SEGMENTATION BUSINESS VALUE

Our previous discussion has suggested that segmenting customers according to product affinity can be a powerful step in preparing for profitable marketing approaches and more satisfied customers. A segment that has an affinity for specific products may be marketed to for those products, upsold for those products, or even cross sold for related products. For marketing of specific products, it is useful to know which segments may be more receptive to those marketing campaigns. For example, if a store wants to get a customer to increase visits, then providing coupons or discounts for products that a customer has demonstrated an affinity for will be more successful than simply providing the same "marketing pitch" to everyone.

ISSUES WITH CLUSTERING CUSTOMERS FOR PRODUCT AFFINITY BASED ON PRODUCT PURCHASE FREQUENCY

Creating segments of customers according to product affinity is challenging. Although clustering techniques to group customers are readily available, the resulting clusters may not be readily useful to the marketing group. Two major issues with the clustering results must be resolved:

- Each segment must have a sufficient proportion of the overall population and the largest segment cannot overly dominate the other segments.
- The differences among the clusters with respect to product affinity must be large and meaningful enough to be of use to the marketing team as they prepare messages to and interactions with customers. Further they should provide a basis with which to better market the different products.

In addition to these issues, one may consider adding demographic or other customer variables (if available) to better understand the resulting segments. For example, one may know the age, gender, and possibly the geographic location of the customer. Using this information, one may know that certain products and product groups appeal more to certain ages, gender, or geographic locations, etc. This paper will not discuss adding other customer variables beyond the transactional frequencies.

Our analyses have used the k-means clustering approach to produce segments of customers with respect to product affinity. Using the SAS procedure called Fastclus, we have been able to produce segments that very clearly differentiate customers. However, in order to arrive at this successful result, we had to solve a host of problems that are typically found anytime one uses product-level data for segmentation.

UNMODIFIED DATA

Throughout this paper we used a 9-cluster solution to demonstrate our procedure. In practice, you will have to decide on the number of clusters based on business objectives, domain expertise and data considerations. A rule of thumb from years of consulting experience has suggested that a useful range of clusters is between 6 and 11. Using the purchase data directly in the Fastclus procedure, we requested a 9 cluster solution with this code:

```
proc fastclus data=SGF2013.Affinity_std1
              outseed=work.mean1 maxiter=20 summary MAXC = 9
              out=SGF2013.Affinity_FASTCLUS_PROD
              delete = 4 drift noprint;
var prod;;
id id;
run;
```

Only 5 clusters were produced from running the procedure even though we requested 9 clusters as follows:

CLUSTER	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	97	0.32	97	0.32
2	72	0.24	169	0.56
3	12	0.04	181	0.60
4	27628	92.09	27809	92.70
5	2191	7.30	30000	100.00

You can see that most of the customers fall into a single cluster, number 4, with over 92%. This is not very useful from a marketing perspective. Furthermore, the only differentiation between clusters (not shown) is not among the products, but simply with the overall quantity of purchases.

STANDARDIZED DATA

Clustering methodology suggests that one standardize the input variables prior to clustering. It is commonly accepted when using the k-means approach (which is the algorithm in the Fastclus Proc) to convert the values of the variables or customer attributes to Z-scores based on the mean and standard deviation of each independent variable. In our case, the variables are the product frequencies. This Z-score conversion (called standardization) enables each independent variable or dimension to have more similar contributions to the cluster formations. The following code illustrates the preparation and standardization of the data. First prepare the data so that the original variables are kept after standardizing. This is important because at the end of segmentation, you want to profile the segments using the original variables so that business executives understand your results.

```
data SGF2013.Affinity_mod; set SGF2013.Affinity_data;
  keep &all_name. &sm_name. id;
  array avar {*} &all_name.; array svar {*} &sm_name.;
  do i = 1 to dim(avar);
    svar(i) = avar(i);
  end;
run;
```

The macros `all_name` and `sm_name` refer to the list of the variables of product frequencies and the corresponding list of variables to be standardized (preceded by `SM_`), respectively. Then standardize the newly created variables.

```
proc stdize data=SGF2013.Affinity_mod out= SGF2013.Affinity_std;
  var sm_;;
run;
```

By doing this, the result set contains the values of the standardized variables as well as the original variables. Then we rename the `SM_` variables to `STD_` and obtain the Euclidean distance from the global mean for each customer. Then we sort the customers in descending distance from the global mean. This will enable the Fastclus procedure to choose initial centroids that are as far from the global mean as possible. Here is that additional code:

```
data SGF2013.Affinity_std1; set SGF2013.Affinity_std;
  array zz{*} &allname.; array ss{*} sm_;;
  drop i sm_;;
  ZZ_ALL = 0;
  do I = 1 to dim(zz);
    zz(I) = ss(I);
    ZZ_ALL = ZZ_ALL + (zz{I})**2;
  end;
  ZZ_ALL = sqrt(ZZ_ALL);
run;
proc sort data = SGF2013.Affinity_std1;
  by descending ZZ_ALL;
run;
proc fastclus data=SGF2013.Affinity_std1
```

```

outseed=work.mean1 maxiter=20 summary MAXC = 9
out=SGF2013.Affinity_FASTCLUS_STD_NL delete = 4 drift noprint;
var std;;id id;
run;

```

After running the code, we obtained the following cluster frequencies. Once again, we requested 9 clusters, but only 6 were created.

CLUSTER	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	89	0.30	89	0.30
2	12	0.04	101	0.34
3	26700	89.00	26801	89.34
4	988	3.29	27789	92.63
5	219	0.73	28008	93.36
6	1992	6.64	30000	100.00

As you can see, by clustering on the standardized values, we have slightly improved the frequencies of the clusters, but are still getting a maximum of 89% leaving only 11% for the other 5 clusters. This is still not good enough and the separation between clusters based on the product affinities is still not descriptive enough.

When examining the data for the average, minimum and maximum purchase frequencies for the 8 products among the 30,000 customers, we discover that the minimum purchase is zero for all products. In fact, a large portion of the customers have no purchases for each product. Here are the percent of customers having zero purchases for each product.

A	B	C	D	E	F	G	H
37.5%	69.2%	74.2%	74.1%	52.9%	41.0%	61.4%	83.2%

The average frequencies are as listed below.

A	B	C	D	E	F	G	H
4.07	2.44	0.86	0.74	2.59	3.53	2.00	0.61

The maximum frequencies are as follows:

A	B	C	D	E	F	G	H
768	958	131	79	349	369	603	118

From this, we realize that the distribution has a strong skewness to the right and a great percentage of zero purchases for each product that should be addressed in the analysis.

STANDARDIZED DATA WITH A CAP OF 3 STANDARD DEVIATIONS

One way of addressing this skewness issue is to cap or, winsorize the standardized values at some number of standard deviations. We chose to winsorize at 3 standard deviations. Customers who have purchases more than 3 standard deviations from the mean will be treated the same as those right at 3 standard deviations from the mean. From a customer relationship perspective, this is appropriate because these are all considered high purchase customers and we are more interested in product affinity than we are in understanding extreme buying patterns with any one product. The additional code to accomplish this is straight forward as follows:

```

data SGF2013.Affinity_std1;
  set SGF2013.Affinity_std;
  array zz{*} &allname.;
  array ss{*} sm_;;
  drop i sm_;;
  ZZ_ALL = 0;
  do I = 1 to dim(zz);
    if ss(I) > 3 then ss(I) = 3;
    if ss(I) < -3 then ss(I) = -3;
    zz(I) = ss(I);
    ZZ_ALL = ZZ_ALL + (zz{I})**2;
  end;
  ZZ_ALL = sqrt(ZZ_ALL);
run;

```

The Sort and Fastclus procedures remain the same. After running the code with the constrained standard deviations, we obtained the following cluster frequencies:

CLUSTER	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	1411	4.70	1411	4.70
2	1501	5.00	2912	9.71
3	1700	5.67	4612	15.37
4	405	1.35	5017	16.72
5	21827	72.76	26844	89.48
6	641	2.14	27485	91.62
7	563	1.88	28048	93.49
8	640	2.13	28688	95.63
9	1312	4.37	30000	100.00

By winsorizing the standardized values prior to clustering, we have again improved the frequencies of the clusters and are now getting all 9 requested clusters. However, the maximum is still almost 73% leaving only 27% for the other 8 clusters. Once again, this is not good enough and the separation between clusters based on the product affinities continues to be not descriptive enough.

We have one more approach to manage the skewness and even out the distribution.

APPLYING THE SOFTMAX TRANSFORMATION TO THE DATA PRIOR TO STANDARDIZATION

The Softmax function is a scaling transformation to reduce the effect of the two extremes – both zeroes as well as very large numbers. The transformation uses a form of generalized logistic function to scale the variable and results in output values between zero and one. There are 3 versions of the Softmax function: linear, unscaled Softmax, and scaled or squashed Softmax. In testing the function, we chose the unscaled Softmax version because heuristically it provides the best results in the greatest variety of cases. In this situation where a great number of the customers have zero purchases for one or more products, the Softmax transformation used results in the lowest value being somewhat above zero. Here is the code used to modify the purchase frequencies prior to standardization:

```

proc univariate data = a2013.Affinity_data noprint;
    output out=a2013.Affinity_uni
        mean = &avg_name.
        std  = &std_name.
        max  = &max_name.
        min  = &min_name.;
    var &all_name.;
run;
data SGF2013.Affinity_mod;
    if _n_ = 1 then set SGF2013.Affinity_uni;
    set SGF2013.Affinity_data;
    keep ID &all_name. &sm_name.;
    array avg_var {*} &avg_name.;
    array std_var {*} &std_name.;
    retain &avg_name. &std_name.;
    array avar {*} &all_name.;
    array svar {*} &sm_name.;
    do i = 1 to dim(avar);
        temp = (avar(i) - avg_var(i))/(1.283 * (std_var(i)/6.2831853));
        svar(i) = 1/(1 + exp(- temp));
    end;
run;

```

The Softmax transformation has been highlighted. It uses the average and standard deviation which was calculated from the Univariate Procedure. The Stdize Proc, the data step that winsorizes the standard deviations at 3, the Sort Proc and the Fastclus Proc remain the same. After running the code with the constrained standard deviations from the Softmax transformation, we obtained the following cluster frequencies:

CLUSTER	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	787	2.62	787	2.62
2	1131	3.77	1918	6.39
3	14953	49.84	16871	56.24
4	3494	11.65	20365	67.88
5	2775	9.25	23140	77.13
6	1900	6.33	25040	83.47
7	1673	5.58	26713	89.04
8	1034	3.45	27747	92.49
9	2253	7.51	30000	100.00

By using the Softmax transformation prior to standardization, we have once again improved the frequencies of the clusters and are getting a better distribution of the 9 requested clusters. However, the maximum is still almost 50% leaving only 50% for the other 8 clusters. This has left us with a challenge to reduce the frequency of the largest cluster. We know that the issue is the number of low frequency customers. How should we approach the issue of the majority of customers having low frequency purchases? As we understand the problem, we are able to come up with a solution which has been used in a wide variety of industries. The solution is called "The Doughnut Clustering Method".

USING THE DOUGHNUT CLUSTERING METHOD TO SEGMENT CUSTOMERS FOR PRODUCT AFFINITY

The Doughnut Clustering Method was presented at SGF 2012 in the paper titled, “*CSI: Customer Segmentation Intelligence for Increasing Profits*” (Paper 103-2012). In that paper, various segmentation approaches were covered and the Doughnut Method was covered in detail. Some portions of the code from that paper will be used in this paper as well. There can be a number of issues with the cluster frequencies and distributions when running the k-means approach without modification. You can end up with a higher density of observations nearer the global center of the population. In fact, that is precisely what we have seen with our sample population of 30,000 customers. Even when using all of our modifications and transformations, we are still left with 50% of our customers in a single cluster. If we were able to reduce the number in the center, that would leave us with a greater percentage for the outer clusters. This is exactly what the Doughnut Method does.

So, our first step, illustrated in the following code, will be to separate those customers closest to the global mean from the ones further away. We sort the customers in descending order by the distance from the global mean. We will then select and cluster on the ones furthest away or the “doughnut”. We will bring in the center or “doughnut hole” customers later.

```
proc sort data = a2013.Affinity_data_std1;
          by descending ZZ_ALL;
run;
data a2013.Affinity_avg
    a2013.Affinity_tran_Y;
    set a2013.Affinity_std1 nobs = num_obs;
    if _n_ < num_obs * .65 then output a2013.Affinity_tran_Y;
    else output a2013.Affinity_avg;
run;
```

This code puts 35% of the data closest to the global mean in the a2013.Affinity_avg dataset. The other 65% of the data will be clustered. The percent of the population to be held out in the “average” dataset was determined by running tests in which we held out from 25% to 40% for the “average” dataset. In this sample, holding out 35% provided the best frequency and cluster differentiation results.

Additionally, we want the requested 9 clusters to be as far apart as possible. Prior to clustering, we have sorted the customers by descending distance from the global mean. This was done to ensure that the initial centroids would be as far from the global mean as possible. An additional issue is that Proc Fastclus only allows us to specify either the number of clusters or the minimum distance criterion for selecting new centroids. So, we will use the latter and run the Fastclus Proc repeatedly using a macro do loop until we obtain the desired number of clusters while keeping the clusters as distant from one another as possible. Here is the initial code to do that:

```
%let nnum = 8;
%let tnum = %eval(&nnum+1);
%let clus_num = %eval(&nnum+1);
%macro clus_build1;
    %global curr_rad;
    /**** Initial Clustering ****/
    data _null_;
        set SGF2013.Affinity_tran_Y (obs=1);
```



```

    call symput ('first_rad',trim(left(ZZ_ALL)));
run;
title1 "Clustering for SGF2013 Affinity level";
%let curr_rad = &first_rad.; /* the largest value of ZZ_ALL = cur_rad */
%let ok = NO ; %let j=0; %let count=&tnum.;
%put 'curr_rad' = &curr_rad 'ok' = &ok;
%let multiplier = %sysevalf(&curr_rad / 50);
%do %until (&ok = YES);
    %if (&j. ge 150 and &count lt &nnum.) %then
        %do;
            %put WARNING: PROC FASTCLUS failed to converge.;
            %goto giveup;
        %end;
    %let j = %eval(&j +1);
    proc fastclus data=SGF2013.Affinity_tran_Y
        outseed=work.mean1 maxiter=20 summary radius=&curr_rad.
        delete = 4 drift noprint;
        var std;;
        id id;
    run;
    %put 'curr_rad' = &curr_rad 'j' = &j ;
    %if &j = 15 %then %let multiplier = %sysevalf(&curr_rad / (&j.*7));
    %if &j = 25 %then %let multiplier = %sysevalf(&curr_rad / (&j.*10));
    %if &j = 35 %then %let multiplier = %sysevalf(&curr_rad / (&j.*20));
    %if &j = 50 %then %let multiplier = %sysevalf(&curr_rad / (&j.*50));
    %if &j = 100 %then %let multiplier = %sysevalf(&curr_rad / (&j.*150));
    data work.cluster_temp1;
    set work.mean1 end=eof;
    if eof then do;
        count=_n_;
        count_1 = count + 1;
        call symput('clus_num',trim(left(count_1)));
        if count=&nnum. then call symput('ok','YES');
        else do;
            new_rad = &curr_rad. + (count-&nnum.)* &multiplier.;
            call symput('curr_rad',new_rad);
            call symput('count',count);
        end;
    output;
        put count= new_rad=;
    end;
run;
%end;
%giveup;;
%mend;
%clus_build1;

```

By running the do loop, we ensure that the requested number of clusters have been selected (in our case 8, saving 1 for the center cluster) and that they are as far apart as possible.

Now, we want to stabilize the centroids of each cluster by removing outliers, resetting the centroids, and then bringing the outliers back in. This will ensure the stability and usefulness of

the clusters in our marketing applications. In order to only remove between 4% and 12% of the data as outliers, we need to run the code through a do loop to satisfy that constraint. Previous work has provided us with the outlier range of 4% to 12% as being sufficient and effective. Here is that code:

```

%macro clus_build2;
  %let xstrict = &curr_rad.;
  %let max_dist = &curr_rad.;
  %let ok = NO ;
  %put 'max_dist' = &max_dist 'ok' = &ok;
  %do %until (&ok = YES);
    %put 'max_dist' = &max_dist 'ok' = &ok;
    proc fastclus data=SGF2013.Affinity_tran_Y seed=work.seed
      maxc=20 least=2 strict=&max_dist out=work.out
      outseed=SGF2013.mean2 converge=0 maxiter=200 noprint;
      var std;;
      id id;
    run;
    data work.cluster_temp;
      set work.out end=eof;
      if cluster<0 then outlier_count + 1;
      tot_count+1;
      if eof then do;
        fraction = outlier_count/tot_count;
        if fraction<.02 then max_dist = &max_dist - &XSTRICT/20;
        else if fraction<.04 then max_dist = &max_dist - &XSTRICT/100;
        else if fraction>.12 then max_dist = &max_dist + &XSTRICT/100;
        else if fraction>.14 then max_dist = &max_dist + &XSTRICT/20;
        if .04 <= fraction <= .12 THEN call symput('ok','YES');
        call symput('max_dist',max_dist);
        output;
      end;
    run;
  %end;
%mend;
%clus_build2;

/* Run PROC FASTCLUS with zero iterations to assign outliers to clusters. */
proc fastclus data=SGF2013.Affinity_tran_Y
  seed=SGF2013.mean2 maxc=20 maxiter=0 outseed=SGF2013.mean2x
  out=SGF2013.Affinity_&tnum._&clus_num.;
  var std;;
  id id;
run;

```

Finally, we need to include the observations from the “average” dataset excluded during the doughnut process. They are included as the “Average” or single central cluster. Then, each customer can move to the cluster that it is closest to.

```

/* Bring in average data - set cluster number */
/* Set cluster number 1 as average cluster. Increment other clusters by 1 */
data SGF2013.Affinity_&tnum._&clus_num.;
  set SGF2013.Affinity_&tnum._&clus_num.(in=a)
      SGF2013.Affinity_avg(in=b);
  if b then cluster = 1;
  else cluster + 1;
run;
proc freq data=SGF2013.Affinity_&tnum._&clus_num.;
  tables cluster;
run;
proc sort data=SGF2013.Affinity_&tnum._&clus_num.;
  by cluster;
run;
/** Add Average group to seed */
data SGF2013.mean3;
  set SGF2013.mean2 end=eof;
  ARRAY P_VARS {*} std;
  cluster+1;
  output;
  drop i;
  if eof then do;
    CLUSTER=1;
    do i = 1 to dim(P_VARS);
      P_VARS{i} = 0;
    end;
    output;
  end;
run;
proc sort data=SGF2013.mean3;
  by cluster;
run;
data SGF2013.Affinity_cl_&tnum._x;
  set SGF2013.Affinity_&tnum._&clus_num.;
run;
/**** Let each obs move to the cluster it is closest to ****/
proc fastclus data=SGF2013.Affinity_cl_&tnum._x
  seed=SGF2013.mean3 maxc=20 maxiter=0 outseed=SGF2013.mean4
  out=SGF2013.Affinity_&tnum._&clus_num.
  outstat=work.CLUS_&tnum._&clus_num._stat;
  var std;
  id id;
run;
proc freq data=SGF2013.Affinity_&tnum._&clus_num.;
  tables cluster;
run;

```

Here are the final frequencies prior to final movement:

CLUSTER	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	10501	35.00	10501	35.00
2	1703	5.68	12204	40.68
3	2226	7.42	14430	48.10
4	2250	7.50	16680	55.60
5	2141	7.14	18821	62.74
6	1811	6.04	20632	68.77
7	3348	11.16	23980	79.93
8	3007	10.02	26987	89.96
9	3013	10.04	30000	100.00

Here are the cluster frequencies after we allow each customer to move to the cluster that it is closest to.

CLUSTER	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	10272	34.24	10272	34.24
2	1703	5.68	11975	39.92
3	2221	7.40	14196	47.32
4	2225	7.42	16421	54.74
5	2123	7.08	18544	61.81
6	1797	5.99	20341	67.80
7	3102	10.34	23443	78.14
8	3220	10.73	26663	88.88
9	3337	11.12	30000	100.00

We have succeeded in bring the largest cluster down to 34.2%. Additionally, although we held out 35% for the average group, when we brought that group back in and allowed movement, very little movement among clusters occurred. Furthermore, as will be seen in the following tables and plots, we have differentiated the clusters according to product affinity.

PRODUCT AND CLUSTER DIFFERENTIATION BASED ON DOUGHNUT METHOD SEGMENTATION

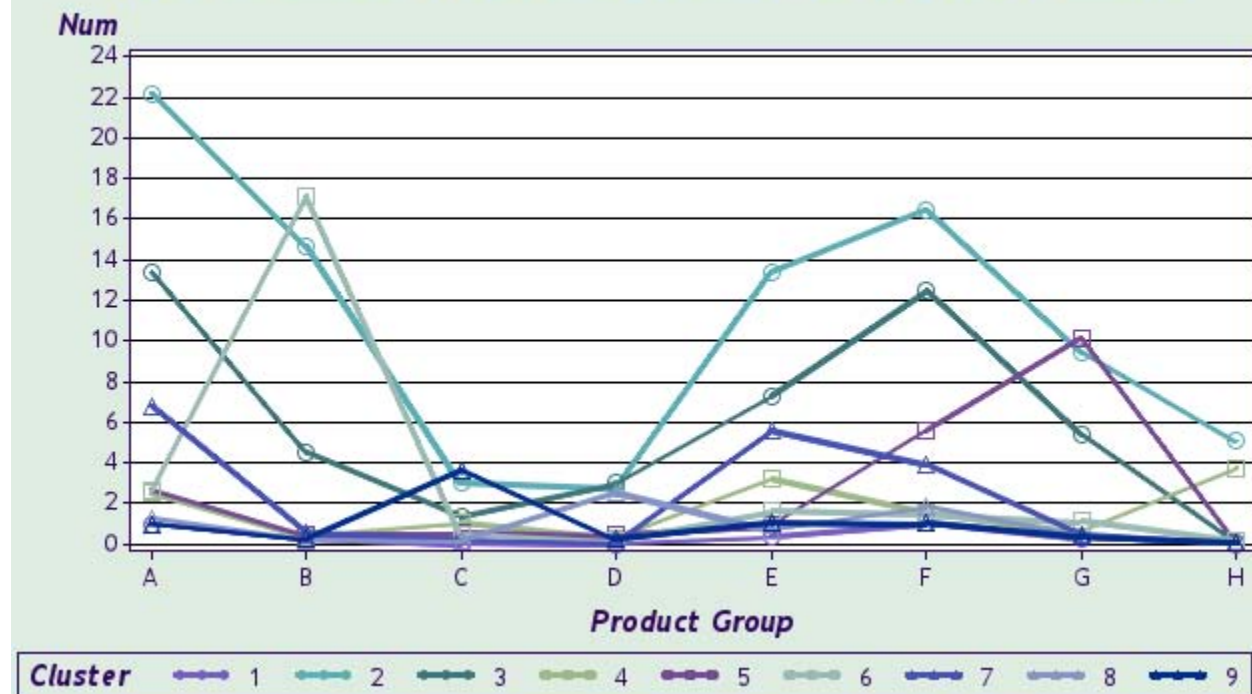
Here is a table illustrating the number of each product purchased for each cluster. As the clustering was done to keep customers with similar purchase patterns together, you can see distinct patterns:

Average Number of Sales Within Cluster by Product Group

Cluster	Freq	All	A	B	C	D	E	F	G	H
All	30,000	16.8	4.1	2.4	0.9	0.7	2.6	3.5	2.0	0.6
1	10,272	3.0	1.0	0.2	0.0	0.0	0.4	1.0	0.3	0.0
2	1,703	87.0	22.2	14.7	3.0	2.7	13.4	16.4	9.5	5.1
3	2,221	47.4	13.3	4.5	1.3	3.0	7.3	12.5	5.4	0.1
4	2,225	13.3	2.5	0.4	1.0	0.3	3.2	1.5	0.6	3.7
5	2,123	20.8	2.6	0.4	0.5	0.4	0.9	5.6	10.2	0.1
6	1,797	24.8	2.6	17.1	0.5	0.2	1.6	1.4	1.1	0.2
7	3,102	17.7	6.8	0.6	0.2	0.1	5.6	3.9	0.5	0.0
8	3,220	7.1	1.3	0.2	0.1	2.5	0.7	1.9	0.4	0.0
9	3,337	7.5	1.0	0.2	3.6	0.2	1.0	1.0	0.3	0.1

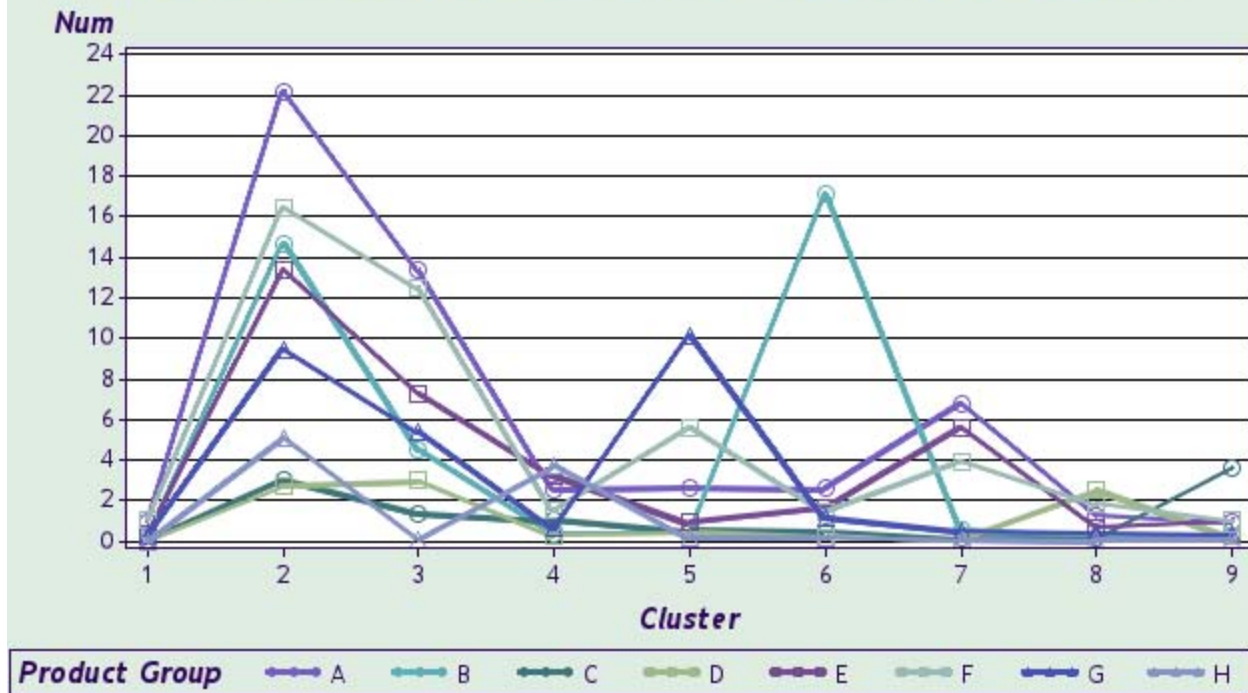
Examining this table, you might decide not to market to cluster 1 because they only average 3 items over all 8 products throughout the 3-year period. Cluster 9 only has 7.5 items purchased, but 3.6 of them are product C. You might want to market to them based on product C and do some up or cross selling. Clusters 2 and 3 are obviously very important due to their volume of purchases. Although product D is not a high volume item, Cluster 8 is just as important as clusters 2 and 3. For cluster 6, almost 70% of all items purchased are product B while almost 50% of all products purchased by cluster 5 are product G. Additional findings can be discerned by further inspection. Here are some plots that illustrate the preceding table:

Average Number of Sales Within Product Group by Cluster



Clusters 2 and 3 stand out across most products. However, cluster 6 stands out for product B and cluster 5 for product G as well as cluster 7 for product A.

Average Number of Sales Within Cluster by Product Group



With this plot, it is clear which clusters purchase the most products overall and within products. You can readily see which products are important to each cluster. Cluster 2 is a high volume cluster that purchases most products with frequency, particularly Product A. Cluster 6 purchases a lot of Product B while cluster 5 purchases Product G.

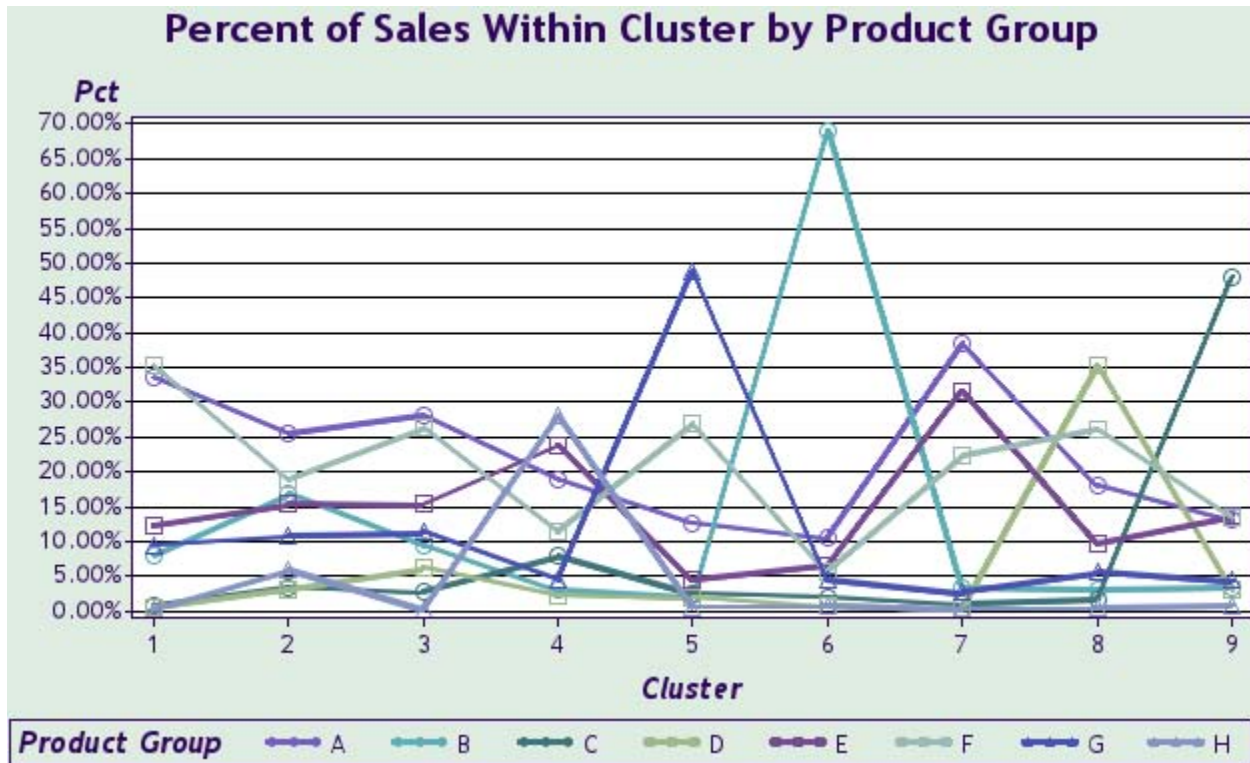
CLUSTER DIFFERENTIATION BASED ON PERCENT OF EACH PRODUCT

The following is the percent of sales within each cluster by product. Each column adds up to 100%.

Percent of Sales Within Cluster by Product Group										
Product	All	Clus1	Clus2	Clus3	Clus4	Clus5	Clus6	Clus7	Clus8	Clus9
A	24.2%	33.6%	25.5%	28.1%	18.9%	12.7%	10.6%	38.4%	18.1%	13.1%
B	14.5%	8.0%	16.9%	9.5%	3.0%	2.1%	69.1%	3.2%	3.0%	3.3%
C	5.1%	0.8%	3.4%	2.8%	7.8%	2.5%	2.0%	1.1%	1.8%	48.1%
D	4.4%	0.4%	3.1%	6.3%	2.3%	2.0%	0.7%	0.4%	35.2%	2.9%
E	15.4%	12.3%	15.4%	15.3%	23.9%	4.4%	6.5%	31.6%	9.8%	13.8%
F	20.9%	35.2%	18.9%	26.3%	11.3%	26.8%	5.7%	22.2%	26.2%	13.5%
G	11.9%	9.5%	10.9%	11.4%	4.6%	48.8%	4.5%	2.7%	5.6%	4.4%
H	3.6%	0.1%	5.9%	0.2%	28.1%	0.6%	0.8%	0.3%	0.5%	0.9%
All	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

If you are interested in the percent of each product purchased by a cluster, then you can use the preceding table to make your marketing decisions. Of course, you would have already considered the average number of items purchased of each product purchased by each cluster. For example, cluster 1 may be interesting in that almost 70% of the purchase items are from products A and F. On average, though, there are only 3 total products purchased by this cluster.

Nonetheless, once you have determined which clusters are of interest, the percentage table and the plot that follows can be of great use.



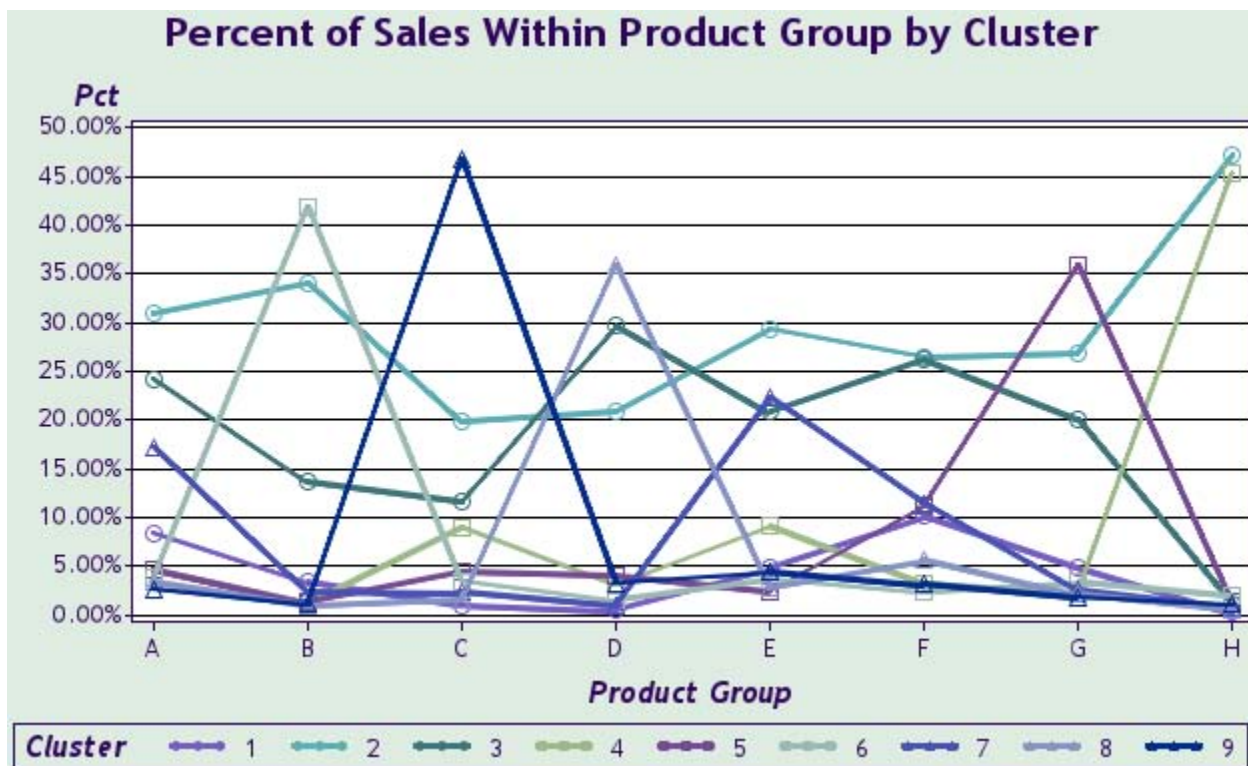
From the plot, it is apparent evident how important Product B is for cluster 6 and product G for cluster 5. Cluster 9 has a far greater affinity for product C than any other cluster.

PRODUCT DIFFERENTIATION BASED ON PERCENT OF EACH CLUSTER

One must be very careful when evaluating the products based on the percent purchased by each cluster. The reason for this is that some clusters are considerably larger than others. Therefore, although the total contribution of the cluster may be sizable, the individual contribution by each member of the cluster would be rather small. That is why one often looks at the average number of purchases rather than the percents. Nonetheless, the following is the percent of sales within each cluster by product. Each column adds up to 100%.

Percent of Sales Within Product Group by Cluster								
Cluster	Prd_A	Prd_B	Prd_C	Prd_D	Prd_E	Prd_F	Prd_G	Prd_H
1	8.4%	3.3%	1.0%	0.6%	4.8%	10.1%	4.8%	0.2%
2	31.0%	34.1%	19.8%	20.8%	29.3%	26.5%	26.9%	47.3%
3	24.3%	13.6%	11.6%	29.7%	20.8%	26.2%	20.1%	1.3%
4	4.6%	1.2%	9.0%	3.1%	9.1%	3.2%	2.3%	45.4%
5	4.6%	1.3%	4.4%	4.0%	2.5%	11.2%	36.0%	1.4%
6	3.9%	41.9%	3.4%	1.5%	3.7%	2.4%	3.4%	1.9%
7	17.2%	2.4%	2.4%	1.0%	22.3%	11.5%	2.5%	0.8%
8	3.4%	0.9%	1.6%	36.0%	2.9%	5.6%	2.1%	0.6%
9	2.7%	1.1%	46.9%	3.3%	4.5%	3.2%	1.8%	1.2%
10	100%	100%	100%	100%	100%	100%	100%	100%

One can see that cluster 1 contributes 10% to the sales of product F. However, cluster one has 35% of the customers. Therefore, the contribution of each customer from cluster 1 is still rather small. Here is the same information represented in a plot.



Be very careful in the interpretation and use of the preceding information. Low frequencies can make interesting percentage differences trivial in marketing implementation. On the other hand, if the cost of marketing communications is very low, then it might still be worthwhile to communicate with low frequency customers when the overall contribution of the segment is reasonably high.

CONCLUSION

Using the various data preparation techniques combined with the Doughnut Clustering Method has been shown to successfully create segments for valuable insights and marketing purposes. We have demonstrated how to process and analyze customer purchase data to create customer groups that are very similar based on the frequencies of purchases in different product groups. Using the results, the marketing analyst is more informed and can make intelligent decisions regarding campaigns and communications with the customer base. By treating customer differently according to their likes and dislikes, one is engaging in a customer relationship management resulting in higher customer satisfaction and higher corporate profits.

A customer who is treated according to their needs and wants is more likely to spend a larger portion of their wallet with that business. Businesses that target customers according to their needs and wants are likely to be more competitive and get a larger market share. This is why segmentation of customers according to product affinity is such a valuable marketing tool.

It would be beneficial if we also had profitability and demographic information to overlay on top of the purchase data by customer that would enable us to get a better perspective of the customer behavior and value.

This paper has demonstrated techniques and code that can be replicated and used by any marketing department to improve their customer relationship process.

REFERENCES

1. Baer, Darius S., 2012. "CSI: Customer Segmentation Intelligence for Increasing Profits", *Proceedings of the SAS Global Forum 2012 Conference, Cary, NC: SAS Institute Inc.* Available at <http://support.sas.com/resources/papers/proceedings12/103-2012.pdf>
2. Collica, R. S. 2007. CRM Segmentation and Clustering Using SAS Enterprise Miner. Cary, NC: SAS Institute Inc.
3. Dey, D., S. Garla, and G. Chakraborty. 2011. "Comparison of Probabilistic-D and k-Means Clustering in Segment Profiles for B2B Markets." Paper 193-2011, Conference Proceedings of the SAS Global Forum 2011, Cary, NC: SAS, Institute Inc., 1-9. available at: <http://support.sas.com/resources/papers/proceedings11/193-2011.pdf>
4. Garla, Satish, Gary Gaeth and G. Chakraborty, 2012. "Comparison of K-means, Normal Mixtures and Probabilistic-D Clustering for B2B Segmentation using Customers' Perceptions" Conference Proceedings of the SAS Global Forum 2011, Cary, NC: SAS Institute Inc., 1-8, available at: <http://support.sas.com/resources/papers/proceedings12/130-2012.pdf>
5. Pagolu, M. K. and G. Chakraborty. 2011. "Eliminating Response Style Segments in Survey Data via Double Standardization before Clustering." Paper 165-2011, Conference Proceedings of the SAS Global Forum 2011, Cary, NC: SAS Institute Inc., 1-15, available at: <http://support.sas.com/resources/papers/proceedings11/165-2011.pdf>

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Darius Baer
SAS Institute Inc
100 SAS Campus Drive
Cary, NC 27513
E-mail: Darius.Baer@sas.com

Goutam Chakraborty
Oklahoma State University
Stillwater OK 74078
E-mail: goutam.chakraborty@okstate.edu

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.