

# Householding and Its Relationship to Customer-to-Customer Marketing and Data Mining

Jacob Sacolick, Harte-Hanks Data Technologies

## ABSTRACT:

Householding, the process of grouping customer account records and prospect lists forming decision units, is a prominent part of customer to customer marketing by financial firms. It allows the marketer to consider the entire relationship of the household to the financial institution and allows measuring customers for management and marketing reporting purposes. Complexity exists because a) several match criteria are used - some fuzzy matched, b) data may contain free format name/addresses with mixed personal/business names and c) householding may change over time due to external factors – divorce/marriage/death/birth or due to data/processing changes. A Householding System typically could have 4-5 match criteria, 3 levels of output (household, individual, matched address), 30-40 distinct steps (data prep, parsing, postal matching, geocoding, sorting, matching, resolving multiple matches, extracting output, loading the warehouse/datamart), run in several modes (initial, incremental/static, re-householding/non-static, customers, prospects), and for a large financial institution involve processing tens of millions to hundred's of millions of data records in each step necessitating parallelization. The paper discusses these issues, householding's relationship to data mining and direct mail response modeling, and a specific unforeseen data discovery.

## 1.0 Householding Work

I was involved with building a) a householding prototype using SAS®, b) another prototype using a combination of SAS for data-transformation and Harte-Hanks Data Technology's Trillium<sup>1</sup> product for householding, and c) a production householding system on a large multi-node system using a language having parallel capability for data-transformation together with Harte-Hanks' Trillium product.

The Trillium based production system differed from the Trillium prototype only as follows:

1. The production system was an SP2 with a large number of nodes instead of an NT based PC.

2. A special language for parallelization, was used for non-Trillium data preparation steps while SAS was used in the prototype.
3. Trillium software was loaded to multiple nodes and run in parallel in the production system.
4. The production system had to renumber households and other sequentially numbered outputs after merging from separate nodes.
5. The production system had to load and maintain an Informix database with the householding and related information.

The use of prototypes for this problem was very beneficial in the following ways:

1. We were able to virtually "drop" in the Trillium business rule tables and "client" parsing tables from the Trillium prototype to the production system without any change.
2. The SAS code for transforming and preparing the data were easily put into specification form and used to develop the needed production code and overall flow of data through the system.
3. The prototype allowed continual testing, refining of client tables and rules to improve parsing and matching results.
4. Production development could take place in parallel with the iterative development and testing of business rules.

In the rest of the paper, I present a brief tutorial on householding, some of the interesting algorithmic and programming problems, and some of the programming techniques used in the SAS code.

## 2.0 Householding 101

Have you ever seen a movie where a person entering a store is treated rather rudely but turns out to be someone important, for example, the president of the company. Companies want to avoid these faux pas'. There were two. One, a customer – any customer- was treated rudely and two they didn't know who the customer was and didn't know how severe the penalties or how valuable the missed opportunity might be.

## 2.1 Householding Definitions

Householding is the process of organizing account and other data by customer. In householding we

---

<sup>1</sup> Trillium is a registered trademark and product of Harte-Hanks Data Technologies, Billerica, MA

generally deal with three entities 1) a household b) an individual and c) a customer. There can be many definitions of the above. In addition there is a definition of what we wish to achieve (the logical or abstract definition) and a definition based on how we actually obtain it (the physical or operational definition).

### **Households**

A household can mean a family. But a slightly broader definition is that it is a decision making unit. A household by either definition can have more than one physical address since they can have a primary residence and other locations such as summer homes.

### **Individuals**

Individuals are physical people.

### **Customers**

Customers are individuals and other entities that own accounts. Accounts are relationships with the company which maintain the financial interactions of the individual with the company. Complications occur when dealing with joint accounts. If John and Mary jointly own an account – the customer is the entity “John and Mary”. However, when getting individuals of an account, most companies do not treat “John and Mary” as a joint individual.

### **2.2 Operational Definitions**

Households, individuals, and customers do not self organize themselves or walk around with identification tags. Therefore there are two primary ways a company can find out who the individual is. One by asking and two by looking at the data supplied when opening up accounts or using accounts. In either case the resulting information is recorded on the account and that is the information used in individualization and householding.

With that in my mind operational definitions are given below:

A household is a unique identifier attached to the accounts that “belong” to that household.

An individual is a unique identifier attached to the accounts that “belong” to that individual.

A customer is an individual that has “open” accounts with the company.

A prospect is an individual that has no accounts with the company or who has only closed accounts.

### **2.3 Relationships among Accounts, Individuals, and Households.**

Since joint accounts can exist, accounts can belong to more than one individual. Therefore accounts and individuals have many to many relationships. In

other words an individual can have many accounts and an account can have many individuals.

Since the household is sometimes the primary unit for reporting marketing results, it is often desirable that accounts and individuals be in only one household. Therefore households have a many to one relationship with individuals and accounts. That means a household has many accounts and a household has many individuals, but an account belongs to only one household and an individual belongs to only one household.

### **2.4 Retail Industry**

In the retail industry a customer may not have any accounts – but walk in and execute a transaction. Under these circumstances transactions that don't belong to accounts can be treated in the same way as accounts in the householding process.

### **3 Household Components**

The following major system modules are included in householding:

1. Parsing Name and Address
2. Postal Matching
3. Geocoding
4. Applying one or more Household Match Rules in separate passes.
5. Combining Results of Multiple Household Match Rules
6. Applying one or more Individual match rules (usually one in a single pass usually)
7. Combining Results of Multiple Individual Match Rules, if necessary.
8. Loading results to a warehouse or marketing data mart.

As part of these steps or between them there may be various steps of reformatting data, transforming data, sorting, and other data processing steps.

In the above we start with the broadest grouping – household. Households are matched prior to matching individuals. The reverse can also be done and there may be a tradeoff in terms of efficiency and maintainability. However, in general one can get the same or similar household and individual account groupings doing it in either order.

### **3.1 Parsing Name and Address**

In many banks names and addresses are entered into systems in unformatted fashion. A name would typically consist of up to five - forty character name lines and the address similarly would consist of up to five - forty character lines.

Banks could enter names and address for new customers in a formatted field manner, but they would still have the job of parsing the names and addresses on old accounts. In fact though many are still entering new accounts unformatted.

Parsing is one of the more difficult tasks in householding and requires use of auxiliary tables that help in identifying components of a name and address. For example, Trillium has thousands of table entries containing common first names, last names, business names, city names, titles (e.g. MR, MRS, MD, Junior, Senior etc.), Address stuff (e.g. rd for road, dr for drive etc.) and formats/patterns. Account names in financial systems can be further complicated when they are trusts or other specialized accounts. In addition, not all banking products have commercial and retail customers separated from each other so that names in more than one format are interspersed, with some hard to distinguish.

Parsing of unformatted names and addresses is needed for a) improving the accuracy and appearance on names and addresses for mailing purposes, b) separating multiple individual names for joint accounts, and c) improving matching accuracy.

### **3.2 Postal Matching**

The Post Office supplies a database of valid street names with valid numeric house number ranges, city names, and zip codes. Using this data one can supply missing address components, correct city names, and identify addresses that may have problems.

### **3.3 Geocoding**

This function identifies the address in terms of census information such as block, tract, etc. and can also supply a latitude and longitude. The census location is useful in matching against databases that provide demographic information and related proprietary value added information based on the census. The latitude and longitude are important in any studies of distance of customers to the company or competitor facilities.

### **3.4 Applying one or more Household Match Rules in separate passes**

Marketing is the business unit usually in charge of householding and is responsible for articulating the business rules appropriate for householding and individualization. Grouping of account records for householding and individualization is usually done with the following information:

- Name Data
- Address Data

- Social Security Number and/or Tax-id
- Account Linkage Data

Parsing of name and address yields about 20 fields of data. In some cases there may be more than one version of a piece of information. For example, after Parsing with Trillium the first name field has a display version and a non-display (internal) version. The non-display version would have converted a name like "Bill" to a standardized version "William".

Account Linkage Data is data contained in production systems linking one or more accounts usually at the direction of the customer. Customers link accounts for convenience of access, getting consolidated statements, pricing plans, or other advantages and needs.

When matching one can do an exact match or a more robust match. Non-exact matches are useful because text information inherently has variation from account to account either due to what the customer has supplied or because of data entry errors. Numeric data has a higher amount of consistency and accuracy from account to account and approximate matching may not be needed.

In defining and assessing the matching rules one evaluates the incidence of over-householding e.g. putting accounts together that are inappropriate or under-householding e.g. leaving one or more accounts in a separate household where they belong together with another one.

All the householding rules could be applied in a single pass, but this would necessitate evaluating every account against every account. To reduce the comparisons to a manageable amount –we "guess" at which records are likely to come together by creating a key and then sorting on this key. Only records with the same key (e.g. in the same "match window" of the "candidate code" as it is called in Trillium) are compared to each other. With a single pass, the key would have to be a composite of data in all the match rules and would be much cruder than if separate passes are made for each match rule.

### **3.5 Combining Results of Multiple Household Match Rules**

Unless only a single match rule is used, results of the separate matching runs must be combined. This function called "Resolve" in Trillium is used to implement what is essentially a transitive property. The transitive property can be stated as follows:

If account A can be matched/householded with account B and account B with Account C then Account A is considered matched with Account C.

Thus if Account numbers 1-5 match on Social Security number and accounts 6-10 match on Name+Address and account 4 and 7 are linked together e.g. have the same account linkage number, then accounts 1-10 will all belong to the same household.

### **3.6 Applying one or more Individual match rules (usually one in a single pass usually)**

If an individual can belong to only one household than the key for forming a window would be the household key created in the previous step. Since this narrows down the records sufficiently only a single matching pass is needed and all the individualization rules can be put into this single pass.

### **3.7 Loading Data into a Warehouse or Marketing Data Mart**

This is the straightforward loading of the output of householding runs into a database. However, there are some complexities related to names and addresses when stability/constancy over time is desired and the name on account may change.

### **3.8 Other Aspects of Householding and Individualization**

The aspects described above are just the tip of the iceberg with respect to householding. Among issues that need to be addressed are:

- a) Is household constancy required – if not how is longitudinal reporting of data by household or individual handled. If constancy is maintained then for how long and when can changes in householding (e.g. what I call re-householding) be incorporated.
- b) What level of constancy is required? What if names, addresses, or social security numbers change or are corrected.
- c) How are closed accounts that are maintained in the warehouse/data mart handled? If a household breaks apart to two separate households because a joint account closed should that joint account now be assigned to only one of the new households or to both of them?
- d) When there are multiple names and addresses for a household or individual, how do we pick out the most appropriate one?
- e) What householding scheme should be used when considering both retail and commercial accounts jointly?
- f) How does one manage customers and prospects. Customers and prospects can jump between these categories by opening and closing accounts. How is consistent numbering

and history maintained? Are householding runs jointly done for both customers and prospects together?

It is beyond the scope of this paper to go in to all of these aspects of householding, but I will talk about re-householding later.

### **3.9 Multiple Householding Views**

In theory, there is no need to restrict householding to a single view or method within an organization. However if we are talking about marketing and management reporting, for example, there could be some serious negative consequences and confusion with having more than one householding scheme.

A frequent issue in banks is that the production systems that do day to day operations may have their own scheme to “dedupe” customers and assign them a customer number. As the goals and methodology used in these systems are usually much different than what is needed for marketing, the bank may end up with a highly undesirable dual methodology for dealing with customers.

### **3.10 Householding Levels**

We focused so far on two levels of householding - Household and Individual. I have also carried the scheme down another level e.g. address. The advantage of this is that with this data I could tell how many physical locations a household or individual was at, which individuals were at which physical locations, and which names and addresses were equivalent although textually not exact matches. This advanced information is highly relevant to Customer Relationship Marketing (CRM) and Customer-to-Customer marketing.

For commercial accounts the individual would represent a business or subsidiary and higher levels parent organizations in a hierarchy. As some companies are complicated, an comprehensive scheme for commercial customers may have several levels of hierarchy.

## **4 Illustration of SAS Code Used for Householding**

### **4.1 Parsing Names and Addresses**

For research and experimentation with householding I needed to parse data from a customer information database into separate fields. The data used had already been parsed once but the street line was a composite of several fields. Thus what I wanted to do is parse a street line like thus

123 S Main St.

into its components of a house number (123), direction (S), street name (Main) and street suffix (ST)

All the routines I use are coded as macros. For example the following routine is part of the routine for parsing the house number/street line of the address. It takes a text line (named as variable vvv) of length - lent and parses into separate words. Words are delimited by blank, ".", and "," (if ch not = ' ' and ch not = '.' and ch not = ',' and ch not = '-'). The "-" is treated differently – it is eliminated and anything it connects is concatenated. Up to maxword words are placed into the matrix labeled mmm where each word is a maximum of lword long.

```
%macro parseq1(vvv,mmm,ppp,maxword,lword,lent);
/*vvv is word lent is its length,mmm is where
to store parsed words, maxword is max # words)
*/
/*&vvv=upcase(&vvv);*/
fchar=1;

wx=0;
do z=1 to &lent;
ch=substr(&vvv,z,1);
if ch not = ' ' and ch not = '.' and ch not =
',' and ch not = '-'
then do;
if fchar=1 and wx<&maxword then do;
wx=wx+1;cx=0;fchar=0; end;
if cx<&lword and fchar<.01 then do;
cx=cx+1; &mmm{wx}=trim(&mmm{wx})||ch;
end;
end;
else if ch not = '-' then do; fchar=1; end;
end;
do z=1 to wx;
&mmm{z}=left(&mmm{z});
end;
&mend;
```

After breaking the text into words we assigned a type to the word based on whether it was alpha or numeric or found in various lists.

Here is some of the code that does this.

This macro checks if all characters except the last are numeric. If so it assigns a value 'n' to ppp otherwise it checks if all are numeric except for the last two characters. In that case the number is being used as a street name as in 11<sup>th</sup> Street and not a house number. Otherwise it is considered an alpha word - type "a".

```
%macro numb;
if &ppp{z}='a' then do;

nxx=0;
```

```
do zzz=1 to ll;
ch= substr(wdad,zzz,1);
if ch in ('1' '2' '3' '4' '5' '6' '7' '8' '9'
'0' '-')
then nxx=nxx+1;
end;
if nxx>=ll-1 and nxx>.01 then &ppp{z}='n';
else do;if nxx=ll-2 and nxx+1>.01 then do;
wdend= substr(wdad,nxx+1,2);
if wdend='TH' or wdend='RD' or wdend='ST'
or wdend = 'ND' and nxx>.01
then do; ad{z}= substr(wdad,1,nxx);
adt{z}='e'; end;
end;
end;
end;
&mend;
```

The alpha routine checks for the word "BOX"

If the word is not part of a box number then it is successively checked against various word lists. For instance the macro "chkapt" would check for words such as 'apt', 'apartment', etc. The "chkdir" routine would check for 'N', 'S', 'E', 'W', 'NW', 'NE' etc.

```
%macro alphaad;
if wdad=' ' then &ppp{z}=' ';
else if wdad='BOX' then &ppp{z}='x';
else do;
&ppp{z}='a';
%chkapt
if &ppp{z} not = 'y' then do;
%chkprf
if &ppp{z} not = 't' then do;
%chkdir
if &ppp{z} not = 'd' then do;
%chkrd
end;
end;
end;
end;
&mend;
```

Generally special words are loaded into arrays for use to check address words as in the routine "chkrd" below:

```
%macro chkrd;
mm=1;
do while (mm<=&maxrd and &ppp{z}='a' );
if lrd{mm}>=ll then do;
if wdad =substr(rd{mm},1,ll) then do;
&ppp{z}='b';
adv{z}=rdv{mm}; end;
mm=mm+1;
end;
&mend;
```

Here the array rd with maxrd entries is used to tell if the word in "wdad" is a street suffix such as "rd", "road", "drive" etc. The array rdv assigns each entry that is found a corresponding value. I want "rd" to be equated with "road" in matching so they would get corresponding values. The first values of 3 values of the array rdv would be 1,1,2 .

Array's like rd and rdv use for handling special words are defined/set up in the main routine as follows:

```
length rd1-rd&maxrd;
array rd{&maxrd};
retain rd1-rd&maxrd (%setrd);

array rdv{&maxrd};
retain rdv1-rdv&maxrd (%setrdv);
```

This illustrates another technique I devised. Because I had so many of these arrays, I needed a quick way of initializing values into these programs. This is done with the macro's %setrd and %setrdv which bring in a list of values. When initializing text variables you have put them in quotes. With use of these macro's I was able to edit in the quotes on mass in a separate files and then copy the values into the appropriate macro's.

In total, I count 11 such tables that I used to parse just the street line.

#### **4.2 SAS Code For Approximate Matching**

This routine does an approximate match of word "wxx1" of length "lxx1" with word "wxx2" of length "lxx2".

The routine goes through the following steps:

1. Initiates do loops allowing different trials.

One do loop allows sliding one word against the other by different amounts so that an extraneous letters at the beginning of one word do not result in no letters matched. Another loop controls letters out of place e.g. a certain number of positions before or after the letter being matched. In practice, the maximum for this was set at one letter before and after.

The loops are organized so that the most exact matching is done first.

2. Compares the letters in one word to another and counts the longest sequence of hits found.
3. Adjust the count downward based on what kind of liberties were taken in the matching and how long the match words were.

4. Compares the result to a criteria to see if a valid match has been obtained.

I found controlling the matching using a percentage of letters awkward and chose to use the adjusted number of letters. The results, after trial and error with the routine and its parameters, was surprisingly good for the data I used.

The overall matching of address was done using a combination of sorting, exact matches, and approximate matches.

In order to narrow down the matching window, it is good to pick a field or fields on which an exact match is required. For example, approximate matching on zip code and house/box number may yield too many errors. So the addresses to be matched were first sorted by house number and zip code. Because the zip would match exactly within this group we didn't worry about matching the city and state. This left the street name for approximate matching, the street suffix for exact matching except for specialized cases e.g. rd is the same as road, and apartment number.

#### **4.3 SAS Code for Exact Matching, and Transitive Property**

At this point in the matching we have done any approximate matching that is necessary in separate matches. Where records match we have created an output variable that has the same value for those accounts that match each other. The same is true for those variables which did not go through approximate matching e.g. those with the same values belong together.

The routine below kicks off the process.

I set the maximum number of iterations to 10 and then for each iteration, I bring more and more households together based on finding any match with the various match keys. In the routine below we use the following variables for matching: rltpnum, individn, addnum, ss, fuban.

"hhk" and/or "hhkn" are the household keys produced at each step. The process flip flops at each step so that first "hhk" is the starting household key assignments and "hhkn" is the new assignment and visa versa in the step afterward.

The algorithm works by sorting on each match variable. However, a household that doesn't change after a complete iteration is separated out so that the sort size gets progressively lower after each iteration.

When there are no more changes the process stops.

```
/* multiple iterations e.g. get the scan set */
```

```

/*the call symput stops process if there are no
changes*/
/*fuban should always be last */
%do jjj=1 %to 10;
%if &jjj=1 %then %do;
%hh(rltpnumn,hhk,hhkn,ss fuban addnum rltpnumn
indvidn,a,)
%end;
%else
%do;
%hh(rltpnumn,hhk,hhkn,ss fuban addnum rltpnumn
indvidn,a,hhz)
%end;
%hh(indvidn,hhkn,hhk,ss fuban addnum rltpnumn
indvidn,b,hhz)
%hh(addnum,hhk,hhkn,ss fuban addnum rltpnumn
indvidn,c,hhz)
%hh(ss,hhkn,hhk,ss fuban addnum rltpnumn
indvidn,d,hhz)
%hh(fuban,hhk,hhkn,ss fuban addnum rltpnumn
indvidn,e,hhz)
%if &jjj=1 %then
%do;%setx() %end;
%else %do; %setx(d.hhfin) %end;
run;
data _null_;
if _n_=1 then set d.chg;
set d.hhfin nobs=numobs;
if chg=numobs and _n_=1 then do;
put 'NUMBER OF RECORDS NO CHANGE' numobs chg;
call symput('jjj',11);
end;
run;
%end;
run;

```

I have also used this algorithm to determine how householding would change if one of the match keys were dropped e.g. analyzing what “glues” a household together.

#### **4.4 SAS Code for Re-householding**

When starting off a new householding methodology, all accounts must be householded e.g. initial householding. After that, as new accounts come in or changes in a data that is used in matching occurs, they are considered for householding e.g. incremental householding. A basic consideration in incremental householding is whether accounts can move from one household to another (re-householding) or are not allowed to move (household constancy). In re-householding, it is desirable that a whole new numbering scheme not be used. That is as far as possible households that do not change or change minimally should receive the same household number as before.

Here is a methodology for doing this.

1. Rerun the initial householding routine to get new household keys
2. Merge the old and new household keys into the same file based on account number.
3. Count the number of accounts in each new household key (NHC) and each old household key (OHC) and for each old household key - new household key pair (NHOHC).
4. Sort by NHC, new household key, NHOHC, old household key
5. For each new household assign the old household number of the first/largest NHOHC
6. When an assignment is contemplated check whether the old household number has already been assigned to another household. To do this we can maintain an array in SAS containing a list of already assigned household numbers.
7. If a new household cannot be assigned an old household number it gets an entirely new - never used before - number.
8. Write reports/log files identifying new households that are not perfect replica's of the old household number or have been assigned new household numbers.

I programmed a more primitive version of this for the householding prototype. I have also used this kind of algorithm for comparing any two householding schemes. I have used this in vendor selection to understand how the vendor was doing householding, in quality control of the warehouse production householding results, and in evaluating the impact of new rules.

#### **5. Improving Customer Information Quality**

In the householding process there can be redundant information. For example, we can identify individual by either name/address or social security number. Accounts we think belong to the same individual based on their name and address may have social security numbers differing by only one character. It would be reasonable to assume that there is a high likelihood that one of the social security numbers is in error. This information can be used for data cleansing and also potentially to initiate a marketing interaction to the customer.

I have written a SAS program to scan a database for such situations in existing accounts and in new accounts. The program distinguishes between types of potential errors. The position of the differing characters in the Social Security has different significance.

## **6. Conclusions**

Householding on the surface seems relatively simple. However, when considering all aspects of householding it is actually very complex. Companies are paying a great deal of attention to this issue because of Customer Relationship Management and one-to-one customer marketing. The methodology will have to work in a variety of modes and situations (constant householding, incremental householding, re-householding with open accounts and closed accounts and for both customers and prospects). It will also have to satisfy the need for a) reporting, b) targeting customers, and c) tracking customer response.

SAS has served as an excellent tool for prototyping and may also be a good tool for implementation. There are also tools like Trillium and Relationship Builder<sup>2</sup> both from Harte-Hanks Data Technologies that provide flexible parameter driven solutions to householding and Customer Relationship Management.

### **REFERENCE INFORMATION**

Jacob Sacolick

Harte-Hanks Data Technologies  
164 Lexington St.  
Billerica, MA 01821

Phone: 978 671 6187

email: [Jake\\_Sacolick@harte-hanks.com](mailto:Jake_Sacolick@harte-hanks.com)

### **TRADEMARKS**

SAS® and all SAS products are trademarks or registered trademarks of SAS Institute Inc.

Relationship Builder® is a registered trademark and product of Harte-Hanks Data Technologies, Billerica, MA

---

<sup>2</sup> Relationship Builder is a registered trademark and product of Harte-Hanks Data Technologies, Billerica, MA