Paper 037-2013

Maximizing the Power of Hash Tables

David J. Corliss, Magnify Analytic Solutions, Detroit, MI

ABSTRACT

Hash tables provide a powerful methodology for leveraging bid data by formatting an n-dimensional array with a single, simple key. This advancement has empowered SAS® programmers to compile exponentially more missing data points than ever before, creating tables with hundreds of fields of all types in which the majority of data in this vast array is empty. However, the hash structure also supports analytics to calculate maximum likelihood estimates for missing values, leveraging extensive data resources available for each individual. An important application of this is in sentiment analysis, where social media text expresses likes or dislikes for particular products. Customer data, including sentiments for other products, are used to model sentiment where an individual's preference has not been made known.

Keywords: Hash, Sentiment, Imputation

INTRODUCTION

Consumer Sentiment Analysis is an important application that can be used to illustrate imputation in hash tables. In sentiment analysis, text from customer contacts, tweets, social media sites and other sources can be compiled for a large group of people, expressing their likes or dislikes for particular products. Most individuals will not have an expressed sentiment for most products. In this case, information on each consumer along with sentiments expressed for other products are used to estimate their sentiment for products where an individual's preference has not been made known.

Hash tables first became available in SAS 9.0 and have undergone much development in subsequent releases. Hash tables enable much faster lookup operations than other SAS structures, making them ideal for managing large datasets that can be keyed with indices. In the case of consumer data, extensive resources may be available for each one of a very large number of individuals. For example, credit bureaus may offer hundreds of variables for millions of individuals. Sales transaction data may incorporate long purchase histories for millions of customers. Sentiment data, in which a company's current and prospective customers express their likes and dislikes for products, services and brands, can be very challenging and extremely time consuming to manage using conventional SAS datasets and procedures. Hash tables provide a way to manage sentiment data quickly and efficiently.

Consumers in a database, whether persons or businesses, are often identified by a long series of alphanumeric fields such as name and address. While this kind of information is very helpful for making a mailing list, it is not necessary to run our entire mailing list through our models. In hash tables, a long, cumbersome and often variable length unique identifier is replaced by a short, often numeric key such as a customer number. This key is used as an index on the table. The indices found in hash tables allow separation of data *manipulation* from data *storage* by allowing fast look-up of pertinent data for individual records without calling an entire table into storage.

An Ordinary Customer List

Name	Street_Address	City	State	Zip_Code	prod_42	prod_44
Magnify Analytics	1 Kennedy Square	Detroit	MI	48226	4	3
Fedex Office	2609 Plymouth Road #7	Ann Arbor	MI	48105	4	2
Hyatt Regency Minneapolis	1300 Nicollet Mall	Minneapolis	MN	55403	1	5
Wrigley Field	1060 W. Addison St	Chicago	IL	60613	2	3

.
.
.

The same data in a Hash Table

Hash_ID	Zip_Code	prod_42	prod_44
00042540	48226	4	3
00063640	48105	4	3
00146328	55403	4	3
00243466	60613	4	3

.
.
.

In this table, name, address, city, state and zip code are used to identify an individual. In many data sources, even more fields are used as the name and street address are broken up into different parts. In a hash table, identification of individual records is accomplished by a compact key. In this example, used for consumer sentiment analysis, the USA postal code (zip code) has been retained because it, too, represents a hash key: connecting each consumer to geocoding, demographics and economic data indexed by postal code.

SENTIMENT DATA

Sentiment data is data on what individuals think or feel about some topic or thing. At the most basic level, sentiment data is a Boolean approve / disapprove; advanced sentiment data can provide a level of approval measured on a scale or attempt to categorize a person's emotional response. Businesses often seek to capture consumer sentiment to assess the approval of market interest in brand or set of products. This data can be captured from a number of sources. Direct sentiment data comes from consumer surveys, focus groups, and communication sent by the consumer to the company, including mail, email and call center contacts. With the growth of digital and social media, indirect consumer sentiment data has become very important. Indirect sentiment data is not communicated by the consumer directly to the company but rather made public through posts on social media sites, twitter, blogs and other sources. Web crawlers search for and collect these data, which then can be combined with data from direct sources and compiled into a customer sentiment database.

Advanced sentiment data - beyond "like" / dislike" - is often represented on a discrete scale with each level indicated by a number of stars. While a value of zero stars is sometimes used to denote instances where a specific consumer has no known sentiment for a particular product or brand, missing data are best represented by missing values. In such a case, a system with an odd number of stars yields an even number of values. For example, a system of up to 5 stars has six distinct values from 0 through 5. In rating systems with an even number of levels, called a "Forced Choice" model, consumers are never truly

neutral but must be rated at least slightly favoring or disfavoring a product. An odd number of choices – represented by an even number of stars – allows the possibility of no preference. For this reason, a Forced Choice model with an odd number of stars is preferred, with the absence of sentiment represented by a missing value rather than neutral.

In order to be actionable, sentiment data requires a unique key that identifies an individual consumer. While some combination of name and address fields is often used, hash tables reduce identification to a single compact matchkey such as a customer number. The first step in developing a consumer sentiment table is the creation of a unique consumer matchkey and assignment of direct and indirect sentiment data to the correct consumer. Once existing consumer sentiments are compiled, one can get a table like this one, capturing the consumer sentiment for a list of a company's seven products, ranked using a forced-choice 5-Star rating system:

Cust_Number	Product42	Product46	Product48	Product54	Product55
7884356	5	5		3	
12653887			5		
16793284		3			
19161413		4			
24722878	3				
26994350	3	4	5		
36144160			3		3
97599542		4	3		
175490127		3			
179481528		5	3	3	

.
.
.

Often, as in this example, many customers have expressed a sentiment for some products but not others. Some will have registered a sentiment for several products but perhaps very few will have offered a view on all of a company's products. Sentiment values may be frequently populated for some well-known products but only sparsely for others. As a result, much of the table is empty.

IMPUTATION OF MISSING DATA

Imputation is any process for assigning an estimated value to replace or represent missing values in a dataset. Several common techniques for the imputation of missing data exist, including mean imputation, regression modeling and multiple imputation. A critical factor in determining the best imputation technique is the reason for which the data are missing. In statistics, this is known as the Missingness Mechanism.

In the instance known as Missing Completely At Random (MCAR), the missingness of the data does not depend on either the independent variable (in this case, who the customer is) or the dependent variable (in this case, the sentiment value). The MCAR case will occur if data has been never been reported purely by chance or a reported sentiment has not been captured without regard to who the customer is or the value of the sentiment expressed. While this situation is easy to model, it is very rarely the case in practice.

In the case of Missing At Random (MAR), missingness depends on the value of the independent variable: that is, the customer. Some customers are simply less likely to express sentiments in a form that may be captured. This does not mean that the sentiments do not exist. While the likelihood of a person expressing sentiments where they can be found will have great impact on the likelihood of a sentiment value being missing, it is not likely to change the *value* of the sentiment. Since only missing values are imputed and which values are missing are known at the outset, MAR effects are expected to have little impact on sentiment analysis.

In the case of Not Missing At Random (NMAR), missingness depends on the value of the unobserved observations. In sentiment analysis, this means that the sentiment for a specific consumer for a particular product or brand in unknown simply because of the value of that sentiment.

While NMAR can be challenging to model, it is very often found in sentiment analysis. This leads to the problem of Selection Bias, where the individuals who respond to a survey, visit a website or otherwise make their sentiment known are not representative of the buying population in general. Such a bias is introduced as individuals are more likely to offer a comment on a product on which they have strong feelings. This is the NMAR case, where the likelihood of a value being missing depends on the value itself. This specific type of selection bias is known in statistics as Malmquist Bias, where the likelihood of an observation being captured - in this case, a sentiment - is proportional to the intensity. Malmquist bias represents a significant but under-studied problem in sentiment analysis. Fortunately, once Malmquist Bias is identified as the problem, statistical techniques exist to address it.

Conventional statistical methods will provide imputed values to use in place of missing ones. Regression techniques such as PROC GLM offer a regression value, while PROC NLMIXED can provide a maximum likelihood estimate. These values will be normalized to the sample population, which is subject to Malmquist Bias. Once an initial set of imputed values is found, they can be scaled to correct for Malmquist Bias, so long as the distribution from a representative sample is known.

Figure 1. Distribution of raw sentiment data

The raw data distribution give the percent of each sentiment level (star rating) of sentiment data collected from internet sources such as social media sites, blog posts and tweets. The bimodal distribution captures few sentiments at two or three stars, where the bulk of the population is expected to be found.

Figure 2. Distribution from a representative sample

A small incentivized sample can be designed to be representative of the population as a whole. Methods for creating a representative sample include carefully designed focus groups and incentivized surveys. The distribution from the raw data is divided by the sample distribution to produce a Malmquist Distribution. This is a response function, calculating the likelihood of a sentiment to be captured as a function of the value of the sentiment.

Figure 3. Response Function Distribution

The Response Distribution gives the weights needed to correct for selection bias. The results of an initial model from PROC GLM or NLMIXED can be weighted to produce a final distribution incorporating both observed and imputed data that matches the distribution from the representative sample.

It should be borne in mind that the sentiment values captured from internet searches etc. are valid and do not require any correction: the problem is not with their values but with their selection, as more intense sentiments are more likely to be expressed than less intense sentiments. As a result, only imputed values require correction.

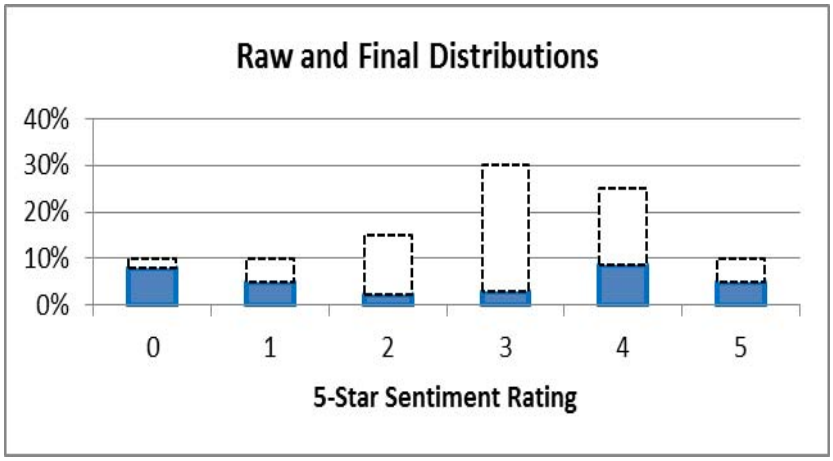

Figure 4. Response Function Distribution

FAST MODEL IMPLEMENTATION USING HASH OBJECTS

Sentiment analysis will often require a suite of imputation models – one for each product or brand of interest. Once the model scorecards have been completed, hash objects can be used to score a very large data set very quickly. This rapid scoring process may further allow the use of more fields from more sources than would otherwise be practical, resulting in superior model performance.

In order to use a hash process to implement a model scorecard, all fields in the scorecard must reside on hash tables. The list of records to be scored must also contain all keys used to access other hash tables. For example, in the B2B sentiment table described earlier, each record contains the customer ID – in this case, a business number – and a field containing the consumer's sentiment value for each of the company's products. In this example, the imputation scorecard for each product contains contributions from the type of business, designated by SIC code, and demographic and economic data indexed by zip code. This requires the table containing the list of record to include all the keys needed to access this data.

Cust_Number	SIC_Code	Zip_Code	Product42	Product46	Product48	Product54	Product55
7884356	2828	26693	5	5		3	
12653887	2221	34309			5		
16793284	8297	04170		3			
19161413	9443	26720		4			
24722878	7005	31597	3				
26994350	3573	28604	3	4	5		
.							
.							
.							

With all the required keys in place, hash lookups access the data needed to build the imputation model(s). In this B2B example, the imputation models use data from three separate tables: one with mean sentiment values for each product by type of business, keyed by SIC code, demographic and economic data keyed by zip code and data on individual consumers keyed by customer number. First, a hash lookup is used for the SIC code table:

```
data work.imp_model1;
   if 0 then set ftcc.hash_1pct_scaled work.sic_mean; /* Copy table structure from sources */

   if _n_ = 1 then do;

      /* Hash table declarations: the table name to be accessed, the key(s) used to lookup */
      /* the data and the fields to be read                                                 */

      declare hash sic (dataset:'work.sic_mean');
      sic.definekey ('sic_code');
      sic.definedata ('mean_42','mean_44','mean_46','mean_48','mean_51','mean_54',
                      'mean_55','mean_56','mean_57','mean_68','mean_69','mean_98');

      /* Complete the hash table declarations */

      sic.definedone ();
   end;

      /* Specify the table to which the new data is to be added */

          set ftcc.hash_1pct_scaled;

             /* Output records with the new fields added */

          if sic.find(key:sic_code) = 0 then output;
run;
```

This typical hash object lookup produces an output dataset much like a MERGE. However, no sorting is required. Instead, data in the second table is accessed using the hash key as an index, resulting in much faster performance. This process can be repeated for several successive tables. When the last required table is accessed, loading into memory all fields required for all the scorecards, the final model score can be calculated in the same hash table step:

```
data work.imp_model2;
   if 0 then set work.imp_model1 ftcc.duns_main_1pct;

   /* Hash object declarations */

   if _n_ = 1 then do;
      declare hash dm (dataset:'ftcc.duns_main_1pct');
      dm.definekey ('dunsno');
      dm.definedata ('empcount','salevol');
      dm.definedone ();
   end;

   set work.imp_model1;

   /* Model imputation of sentiment data using the scorecard for each product or brand */
   /* Records for which the sentiment value is known are not imputed                   */

   if count42sc not = . then final_42 = count42sc;
   else final_42 = (0.569 * mean_42) + (0.136 * zip_mean) + (0.00153 * sale_volume_class);

   if count44sc not = . then final_44 = count44sc;
   else final_44 = (0.582 * mean_44) + (0.143 * zip_mean) + (0.00193 * sale_volume_class);

   if count46sc not = . then final_46 = count46sc;
   else final_46 = (0.531 * mean_46) + (0.115 * zip_mean) + (0.00168 * sale_volume_class);
   .

   .
   if dm.find(key:dunsno) = 0 then output;
run;
```

CONCLUSION

Hash object programming, introduced in the SAS v9 to support faster sorts, joins and lookups, can rapidly access the data required by a model scorecards. Calculation of model scores can be performed within the hash table step.

The type of selection bias impacting consumer sentiment values is identified as Malmquist Bias, where the *probability of capturing* a sentiment *varies with the intensity* of the sentiment. This source of bias can be corrected in sentiment data through the use of a representative sample from a focus group, incentivized survey or other means. The response function, giving the weights needed to correct for selection bias, is given by dividing distribution of the sentiment in data collected from various sources (e.g., social media) by the distribution in the representative sample.

In imputing sentiment data using hash object programming, individuals may be efficiently identified using a numeric key to index a consumer data table. A lookup table indexed by the hash key provides more extensive identifying information as needed, e.g., name and address for a contact list. Multiple hash keys are used to leverage different classes of information, such as multiple records for one individual, individual, and corporate data applying to multiple individuals. The use of postal code as one component of a composite hash key will support the use of geographic, demographic and economic data in the development of imputation models.

REFERENCES

An Introduction to SAS® Hash Programming Techniques
Kirk Paul Lafler, 2011, “An Introduction to SAS® Hash Programming Techniques”, *Proceedings of the Southeast SAS Users Group Conference”.* Available at http://analytics.ncsu.edu/sesug/2011/BB08.Lafler.pdf

Dorfman, Paul, and Koen Vyverman. 2006. "DATA Step Hash Objects as Programming Tools." *Proceedings of the Thirty-First SAS Users Group International Meeting.* Cary, NC: SAS Institute Inc. Available at www2.sas.com/proceedings/sugi31/241-31.pdf.

ACKNOWLEDGMENTS

CONTACT INFORMATION

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

Magnify Analytic Solutions
1 Kennedy Square, Suite 500
Detroit, MI 48224
Phone: 313.202.6323
Email: dcorliss@marketingassociates.com