Paper SD-08

Dealing with Missing Data for Credit Scoring

Steve Fleming, Clarity Services Inc.

ABSTRACT

How well can statistical adjustments account for missing data in the development of credit scores? We will demonstrate how credit scores are developed and why missing data is a problem especially in a waterfall decision environment. We will then compare two approaches to the missing data. One model will use all of the data by placing missing data into its own bucket. The second model will demonstrate the use of multiple imputation to account for the uncertainty created by missing data.

INTRODUCTION

Credit scores are built in an objective and consistent manner to help lenders identify the risk that applicants for credit will reach a negative outcome (Siddiqi, 2006). To build the scores, historical data from the time of application and the presence or absence of the negative outcome is used. A statistical model is fit to the historical data that predicts well in validation data not used in the development of the credit score.

Missing data can seriously bias credit scores. We will look at making adjustments for missing predictor variables in credit scoring. Past research has looked at using multiple imputation to adjust for missing outcomes (Fogarty, 2006).

DEVELOPING CREDIT SCORES

In typical credit scoring applications, the available predictor variables are transformed onto a weight of evidence (WOE) scale. This has the benefit of allowing non-linear effects, use of missing data, and the attenuation of model scores for outliers (Siddiqi, 2006). For each predictor variable, values are aggregated into 2 or more bins such that the level of risk as indicated by the outcome is similar across the bin (Liu). Bins are reviewed to make sure they make sense, represent enough observations to hold up in new samples, and have a linear relationship between the WOE values and the original predictor values unless there is a plausible explanation for the observed shape. This last requirement helps in the development of adverse action codes.

The WOE values for each potential predictor variables are aggregated into a statistic called the information value (IV) which is a single summary that judges predictive ability with larger values indicating higher predictive ability. A lower bound on the information value is often used as a first culling of the potential predictors. A common cutoff value for a minimally weak feature is 0.02 (Siddiqi, 2006). When the number of potential predictor variables is small, I have found some value in pushing the cutoff down to 0.01.

Even after this first cut, the available set of predictor variables for input into the credit scoring model remains large and many of them are highly correlated with one another. Feature selection is used to pare down inputs to a moderate number of relatively uncorrelated variables. This also helps avoid overfitting the model and yields a solution that usually generalizes well to the validation data. One way to do this in SAS is with PROC VARCLUS which divides the feature space into clusters of correlated variables. An example of using PROC VARCLUS is provided below.

PROC VARCLUS will continue splitting the feature space into additional clusters until a stopping criteria is met. Then one variable from each cluster is selected for input into the scoring model. One method selects the variable from each cluster with the highest IV. Another metric is the one minus R-squared ratio which measures how representative the variable is of the cluster. Features with low values of this ratio are closer to the cluster center.

PROC LOGISTIC is used to calculate the scoring model. A holdout sample is reserved to guard against model overfit. Predictor variables may be dropped from the model due to collinearity, small Wald Chi-Square values, or reversal of sign for the parameter estimate.

WATERFALL DECISIONING

Credit grantors may use data from a variety of products to make credit decisions. To reduce costs, lenders want to make a decision as early in the process as possible. For example, if an applicant looks like a particularly poor risk based on Product A, there is no need to incur the cost of pulling Product B.

While the waterfall decisioning strategy is a boon to lenders bottom line, it can leave analysts with missing data problems. Let's say there is a data element in Product B that is particularly useful in making credit decisions. However, most risky applicants may have been removed from the process prior to Product B being called. The

predictive value of data elements in Product B may be depressed due to missing data i.e. the unobserved results on Product B for those applicants most likely to have a bad outcome.

It is well known that ignoring missing data can lead to biased results (Fogarty). A more common approach is to group the missing data into a class for prediction. This study will investigate multiple imputation as an alternative approach.

FULL AND MASKED DATA SETS

A data set with 6,469 observations was used to develop the full model. An outcome variable with values of 0 for not defaulted and 1 for defaulted was used to divide the applicants by risk level. About 37% of applicants defaulted on their loan.

Complete data from two products from the time of application was appended to the outcome variable. There were 110 variables from product A detailing information such as inquiry counts, observed relationships in identify elements such as social security number and bank account, counts of changes in identity elements, and tradeline information.

Product B contains variables with information on the applicant's banking behavior. There were 42 variables from product B.

To simulate the effect of missing data, a variable known to indicate credit risk was used to set about 30% of the product B data to missing.

FULL ANALYSIS

All potential predictor variables were binned and those with information value of at least 0.02 were kept for the variable clustering stage. Seven variables clusters are identified (Table 1) with five product B variables all appearing in clusters 5 and 6.

```
proc varclus data=sesug.full_binned;
var w1_: ;
run;
```

A model was fit to the data using all of the variables with the highest information value in each cluster. Variables A108 and A109 represent counts of the same event over different time periods. Although they were in different clusters, when present in the model their coefficients had the highest standard errors and low values of the Wald Chi-Square. Variable A109 was removed from the model and the results looked more satisfactory. The code to fit the final model is shown below. The OUTPUT statement is used to produce a data set with the scores from the full model. The parameter estimates (Table 2) show that the two variables from product B that are in the model are among the top 3 predictive variables.

```
proc logistic data=sesug.full_binned ;
model outcome (event="1")
    =  A49 A55 A99 A108 B1 B42;
output out=predicted_full pred=score ;
run;
```

ANALYSIS OF MASKED DATA USING MISSING DATA AS A CATEGORY

In this version of the binning, missing data was included as a separate category. Whenever the risk exhibited by the missing data bin was similar to another bin, the bins were combined. All potential predictor variables with IV of at least 0.02 were kept for the variable clustering stage. Of interest, all product B variables are in the same cluster (Table 1). Even 30% missing data makes all of these variables look very similar for the purposes of predicting default.

Similar to the full analysis, variable A109 was removed from the model to remove collinearity with A108. The parameter estimates (Table 2) show that the two product B variable in the full model are replaced by B8 with a similar parameter estimate. None of the other parameter estimates change greatly. This model is would not be as robust as the full model.

ANALYSIS OF MASKED DATA USING MULTIPLE IMPUTATION

Multiple imputation is a statistical technique that fills in plausible values for missing data using the observed relationships in the non-missing data. This is done several times to represent the uncertainty caused by imputing unknown values. Analysis then proceeds separately on each imputed data set and the results are combined to create an overall model.

PROC MI offers many approaches for imputing missing data (Yuan). For this analysis, we take advantage of waterfall decisioning leading to a monotone missing data pattern. That is, there is a set of variables which are completely observed. These variables can be used to predict the missing values based on the observed relationships.

Binning of the variables was done taking care that the missing data category was not combined with any others. This led to results that were substantially similar to the previous analysis. For this study, the missing data category was kept in the IV calculation, but it would be interesting to see if removing it causes substantially different results.

The first step in the multiple imputation phase was to remove the WOE values related to the missing data. Multiple regression was used to impute the missing values so a model using the completely-observed variables was needed. To avoid overfitting, PROC VARCLUS was used to find a 5 cluster solution for the product A variables. The variables with the lowest one minus R-squared value from each cluster were used to predict the missing values.

In the PROC MI statement, the DATA= option provides the name of the data set with the WOE values to be used in the imputation. The SEED= option provides a seed to the random number generator so that results can be repeated if necessary, and the OUT= option provides a name for a data set to hold the imputed values. A variable called _IMPUTATION_ is added to this data set that holds the imputation number for use in a BY statement during the analysis. By default PROC MI produces 5 imputations which has been found to provide robust results in most situations. This can be adjusted by the NIMPUTE= option if needed.

```
proc mi data=work.masked_woe
        seed=524167184
        out=sesug.masked_imputed;
```

For each variable needing imputation, a MONOTONE statement is used. The REG option uses multiple regression to fill in missing values. Other options include DISCRIM which uses discriminant analysis to impute categorical variables, LOGISTIC for imputing binary variables, PROPENSITY which divides the observed data into ordered groups and assigns a value from the group with the highest propensity score, and REGPMM which assigns an observed value that is close to the predicted value from a regression equation.

```
monotone reg(B1 = A29 A49 A56 A69 A97 / details);
```

Finally, a VAR statement is used to provide the list of variables to be used in the imputation. The completely observed variables must be listed before the variables with missing values.

```
var A29 A49 A56 A69 A97 B: ;
```

The Missing Data Patterns output (Output 1) confirms the monotone missing data pattern. Two groups are identified, one with complete information representing about 69% of the observations, and one group with complete information on the product A variables and missing data for the product B variables accounting for the remaining 31% of observations.

Missing Data Patterns

Group	A69	A97	A29	A49	A56	B1	B5	B8	B14	B31	B42	Freq	Percent
1	X	X	X	X	X	X	X	X	X	X	X	4467	69.05
2	X	X	X	X	X	.	.	.	.	.	.	2002	30.95

Output 1. Missing Data Patterns Output from a PROC MI

For each imputation a random draw of the regression coefficients is made, and an imputed value is added to the data set for each observation with missing data (Output 2).

The imputed data was run through variable clustering (Table 1). It is interesting that the imputed data set separates out two clusters of product B variables like was found for the full data set.

Regression Models for Monotone Method

Imputed Variable	Effect	Imputation 3	Imputation 4
B5	Intercept	-0.017481	-0.000218
B5	A69	0.112882	0.083269
B5	A97	-0.043971	-0.031989
B5	A29	-0.069061	-0.033701
B5	A49	0.266813	0.239091
B5	A56	-0.054356	-0.022614

Output 2. Regression Models Output from a PROC MI

Cluster	Full Data		Masked Data with Missing as Category		Masked Data without Missing Data	
1	A109	0.044	Same		Same	
	A110	0.039				
	A80	0.034				
	A30	0.027				
	A31	0.026				
	A32	0.025				
	A81	0.022				
	A70	0.022				
	A67	0.021				
	A69	0.020				
2	A99	0.047	A99	0.047	A99	0.047
	A101	0.047	A101	0.047	A101	0.047
	A97	0.045	A97	0.045	A97	0.045
			A98	0.021		
3	A108	0.046	A108	0.046	A108	0.046
	A79	0.040	A79	0.040	A79	0.040
	A78	0.039	A78	0.039	A78	0.039
	A29	0.034	A29	0.034	A29	0.034
	A107	0.031	A107	0.031	A107	0.031
			A3	0.031		
4	A49	0.028	Same		A49	0.028
	A48	0.022			A48	0.022
	A1	0.020				
5	B42	0.062	n/a		B8	0.047
	B8	0.058			B42	0.045
	B31	0.030			B31	0.025
					B14	0.022

Cluster	Full Data		Masked Data with Missing as Category		Masked Data without Missing Data	
6	B1	0.031	n/a		B1	0.039
	A3	0.031			B5	0.021
	B9	0.024			A1	0.020
	A98	0.021				
7	A55	0.025	Same		Same	
	A56	0.023				
	A57	0.021				
8	n/a		B8	0.047	n/a	
			B42	0.045		
			B1	0.039		
			B31	0.025		
			B14	0.022		
			B5	0.020		
9	n/a		n/a		A3	0.031
					A97	0.21

Table 1. Variable Clustering Results with Information Values

With the imputed data set, separate logistic regression models were fit to each imputation using a BY statement. In the MODEL statement, the covariance matrix for the parameter estimate is requested using the COVB option. An ODS OUTPUT statement is used to save the parameter estimates and their covariance for use in obtaining the final model. Like the other analyses, a choice was made between variables A108 and A109 since they are similar. In this instance the p-value for A108 was higher so A109 was kept.

```
proc logistic data=sesug.masked_imputed ;
by _imputation_;
model outcome (event="1") =  A3 A49 A55 A99 A109 B1 B8 / covb;
ods output ParameterEstimates=work.parms_imputed
           CovB=work.covb_imputed;
run;
```

PROC MIANALYZE was used to combine the logistic regression models fit to each imputation set into one model (Table 2). It is a bit difficult to compare parameter estimats across models since different variables are present in each. The positive result from this model is the presence of two product B variables as found in the full model.

```
proc mianalyze
    parms=work.parms_imputed
    covb(effectvar=stacking)=work.covb_imputed
    ;
modeleffects Intercept A3 A49 A55 A99 A109 B1 B8 ;
ods output parameterestimates=sesug.param_imputed;
run;
```

Parameter	Full Model	Masked Model with Missing as Category	Masked Model with Imputed Missing Data
A3	n/a	n/a	0.9416
A49	0.5114	0.4527	0.3373
A55	0.5084	0.5669	0.4778
A99	1.0093	0.9608	0.9976
A108	0.5986	0.6030	n/a
A109	n/a	n/a	0.6447

Parameter	Full Model	Masked Model with Missing as Category	Masked Model with Imputed Missing Data
B1	0.9500	n/a	0.6340
B8	n/a	0.9763	0.6174
B42	0.9672	n/a	n/a
Intercept	-0.5294	-0.5296	-0.4457

Table 2. Model Parameter Estimates

MODEL COMPARISONS

To compare models, the two solutions for missing data were used to score the results on the full data set using the WOE values calculated on the full data. Pretty much across the board, the imputed model appears to do a better job at discriminating between good and bad accounts than the model using missing data as a category. It is also quite a bit smoother owing to the additional number of parameters which were selected for this model. It is interesting that both models built on the data with missing values peak in separation at a different spot than the model built on the full data set.

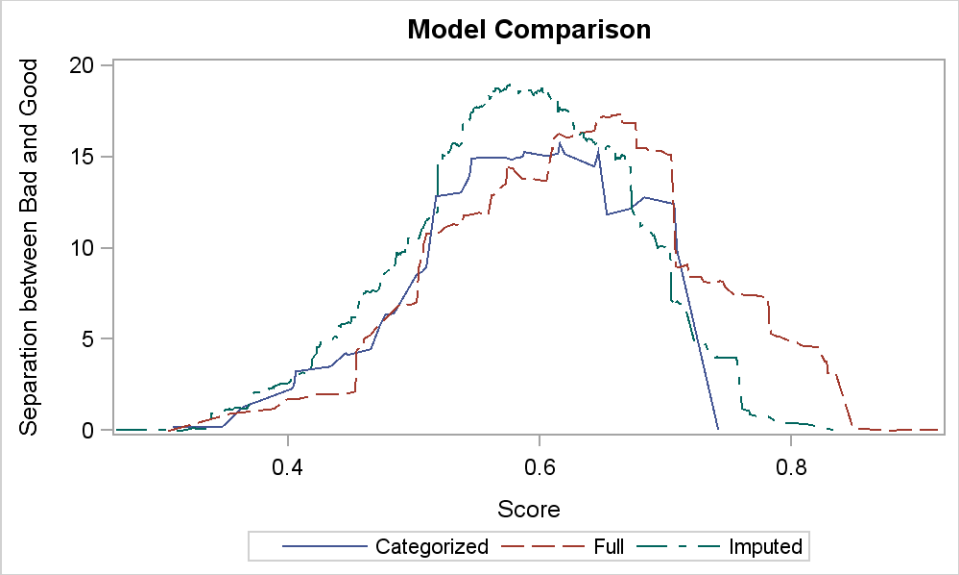

Figure 1. Comparison of Models for Separating Bad from Good Accounts

CONCLUSION

This work has proven the feasibility of basing credit scores on a model built on imputed data when there is missing data. In this instance, the imputed model appears to be superior to the model built by assigning missing data to its own WOE value. Whether this is the case generally remains to be determined. Building the imputed model did take more steps, and there is an open question as to what is an appropriate statistic to use to select the best variables from each cluster.

REFERENCES

Fogarty, David J. “Multiple Imputation as a Missing Data Approach to Reject Inference.” *Interstat*. May 28, 2013. Available at http://interstat.statjournals.net/YEAR/2006/articles/0609001.pdf.

Liu, WenSui. “A SAS Macro Implementing Monotonic WOE Transformation in Scorecard Development.” *Yet Another Blog in Statistical Computing*. June 10, 2012. Available at http://statcompute.wordpress.com/2012/06/10/a-sas-macro-implementing-monotonic-woe-transformation-in-scorecard-development/.

Siddiqi, Naeem. 2006. *Credit Risk Scorecards*. Hoboken, New Jersey: John Wiley & Sons

Yuan, Yang C. *Multiple Imputation for Missing Data: Concepts and New Development (Version 9.0)*. Rockville, Maryland: SAS Institute

ACKNOWLEDGMENTS

I would like to thank the Analytics Team at Clarity Services for their help improving this work.

CONTACT INFORMATION

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

Name: Steve Fleming
Enterprise: Clarity Services
Address: 9433 Bee Caves Rd
City, State ZIP: Austin, TX 78733
Phone: 512-582-7717
E-mail: sfleming@clarityservices.com