Preparing for Basel II

Common Problems, Practical Solutions

Part 2: Modeling Strategies

by Jeffrey S. Morrison

Continuing the series dealing with requirements for Basel II and building PD and LGD models, this article offers some tips in model-building strategy so the analyst can get the most out of the data available. The focus is on selecting the right predictors, dummy variables, transformations, and interactions, evaluating influential outliers, and developing segmentation schemes.

Volumes of literature have been written on probability of default, loss given default, and the statistical techniques used to estimate them. There's a *deficit* of literature, however, on practical advice so the analyst can produce better models more quickly. For each topic there are a variety of approaches—many of which can work equally well. Regardless of whether the analyst is building a PD or LGD model, some practical strategies can provide a good jump-start in aligning your modeling process to the New Basel Capital Accord. In this article, we will refer to a hypothetical dataset to model the probability of default using such predictive attributes as LTV, payment history, bureau scores, and income. And although these techniques can be implemented in just about any statistical language, examples will be given using the SAS software language as well as S-PLUS.

Selecting the Right Predictors

Although good modeling data may be initially hard to find, once your bank has put together the informational infrastructure for the New Capital Accord, you might have more data than you know what to do with. In building a PD model using a regression technique, typically only 10 to 15 predictor variables are finally chosen. However, you may have 200 or more potentially predictive attributes coming from your loan accounting systems, not to mention outside vendors such as economic service providers. So how do you narrow down all that information? As indicated in the May 2003 issue of *The RMA Journal*, sound judgment, combined with knowledge of simple correlations in your data, is a good starting point. Luckily, there are some additional tools to make the model-building process a little less stressful.

 Jeff Morrison is vice president, Credit Metrics—PRISM Team, at SunTrust Banks, Inc., Atlanta, Georgia.

The first is the automatic variable selection routines found in most statistical packages. Generally, they are called *forward stepwise*, *backward stepwise*, and *best-fit* procedures. Forward selection builds the model from the ground up, entering a new variable and eliminating any previous variables not found statistically significant according to a specified criteria rule. The backward stepwise procedure works just the opposite. It starts with all available predictors and eliminates them one by one based upon their level of significance. Best-fit procedures may take much more computer time because they may try a variety of different variables to maximize some measure of fit rather than the significance level of each variable. *What's the advice?* Experiment with them all, but don't be surprised if the backward stepwise procedure works a little better in the long run. However, a word of caution—only include variables in your model that make business sense, *regardless* of what comes out of any automatic procedure.

One major drawback of these procedures is that they can lead to variables in your final model that are too correlated with one another. For example, if local area employment and income are highly correlated with one another, the stepwise procedures may not be able to correctly isolate their individual influence. As a result, they could both end up in your model. This is a condition called *multicollinearity* and may cause problems in the model-building process. One way of avoiding this problem is to perform a correlation analysis before you run any regression procedure. The advice—only include predictor variables in your regression that have a correlation with each other less than .75 in absolute value. If you do run into variables that are too highly correlated with one another, choose the one that is the most highly correlated with your dependent variable and drop the other.

A second tool is called *variable clustering*, a set of procedures that can help the analyst jump some of the hurdles associated with using an automatic selection routine. This procedure seeks to find groups or clusters of variables that look alike. The optimal predictive attributes would be those that are highly correlated with variables within the cluster, but correlated very little with variables in other clusters. SAS, for example, has an easy-to-use procedure called PROC VARCLUS that produces a table such as the one shown in Figure 1.

Clustering routines do not offer any insight into the relationship of the individual variables and our dependent variable (PD or LGD). What they do offer, however, is a much easier way of looking at your predictive attributes. It may be that Cluster 1 represents delinquent payment behavior data. Cluster 2 could reflect geographic or economic data. Therefore, in a regression procedure you should be sure to test variables from each cluster. So how do you know which to pick? One way is to choose those variables with the lowest ratio values as indicated in Column D—a measure easily computed by most popular statistical software packages such as SAS. In our example, we might select VAR7 from Cluster 1 and VAR5 from Cluster 2 to try in a stepwise regression.

Dummy Variables

Much of the information you might have on an account is in the form of a categorical variable. These are variables such as product type, collateral code, or the state in which the customer or collateral resides. For example, let's say you have a product type code stored as an integer ranging from 1 to 4. If the variable was entered into a regression model without any changes, then its coefficient's meaning might not make much sense, let alone be predictive. However, if we recoded it as follows, then the

Figure 1

Variable Clustering

A Variable	B R-Squared Own Cluster	C R-Squared Next Closest	D 1-R-Squared Ratio
Cluster 1			
VAR1	.334	.544	1.460
VAR2	.544	.622	1.206
VAR7 ➤	**.322**	**.242**	**0.894**
Cluster 2			
VAR5 ➤	**.988**	**.433**	**0.021**
VAR8	.544	.231	0.592
VAR3	.764	.434	0.416
VAR4	.322	.511	1.386

regression could pick up the differences each product category makes in the PD or LGD, all other things remaining equal:

```
IF PRODUCT_CODE=1 THEN DUMMY_1=1;
        ELSE PRODUCT CODE=0;
IF PRODUCT_CODE=2 THEN DUMMY_2=1;
        ELSE PRODUCT CODE=0;
IF PRODUCT_CODE=3 THEN DUMMY_3=1;
        ELSE PRODUCT CODE=0;
IF PRODUCT_CODE=4 THEN DUMMY_4=1;
        ELSE PRODUCT CODE=0;
```

What you now have are four product code *dummy* variables that will allow the model to estimate the default impact among different product types. By convention, one of the variables has to be left out in the regression model or an error will result. Therefore, you would include dummy_1, dummy_2, and dummy_3 in your regression model rather than a single variable presenting all values of the product code.

Variable Binning

Building on the dummy variable concept, a procedure called *binning* sometimes helps the analyst increase the model's predictive power even further. Although there are a variety of different binning approaches available, all leverage off the idea that breaking down a variable into intervals can provide additional information. When dealing with a variable that has continuous values, such as a bureau score, a two-step process can be implemented. First we do something called *discretizing*, where the original value is changed into ranges in order to smooth the data—creating a small number of discrete values. An example of this process is shown in Figure 2, where the bureau score was recoded into having only five values.

Now that you have regrouped the data into five range values, simply create dummy variables from them as shown in Figure 3.

Note that the last grouping variable was not done because we would automatically leave this out of the model by convention. The reason this procedure sometimes works to increase the accuracy of the model is because it allows the model to be more flexible in estimating the relationship between the bureau score, for example, and the default when the relationship may not be strictly linear. This is a great technique to try on other variables such as LTV, income, or months on books.

Transformations

Many times, simple transformations of the data end up making the model more predictive. If you are working with a PD model, one thing you could do as part of the preliminary analysis is to run a logistic regression using only a single predictor

Figure 2

Step 1—Discretizing your Data

Grouping Range	SAS Code
455-569	IF B_SCORE >455 AND B_SCORE<=569 THEN B_SCORE=569;
570-651	IF B_SCORE >569 AND B_SCORE<=651.5 THEN B_SCORE=651;
652-695	IF B_SCORE >651 AND B_SCORE<=695 THEN B_SCORE=695;
696-764	IF B_SCORE >695 AND B_SCORE<=764 THEN B_SCORE=764;
765-850	IF B_SCORE >764 AND B_SCORE<=850 THEN B_SCORE=849;

Figure 3

Step 2—Dummy Variables / Binning

B_SCORE_1 =0;
IF B_SCORE=569 THEN B_SCORE_1 =1;
B_SCORE_2 =0;
IF B_SCORE=651 THEN B_SCORE_2 =1;
B_SCORE_3 =0;
IF B_SCORE=695 THEN B_SCORE_3 =1;
B_SCORE_4 =0;
IF B_SCORE=764 THEN B_SCORE_4 =1;

Figure 4

Transformation

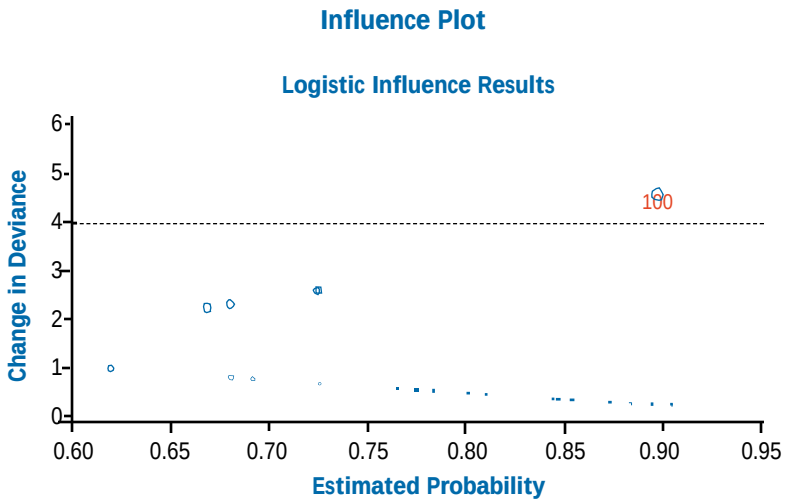
Transformation Analysis—Logistic

Obs	_Status_	_Lnlike_	Name	T_Type	Recomm
1	0 Converged	-423.744	INCOME	XSQ	<-------*
2	0 Converged	-423.744	INCOME	QUAD	
3	0 Converged	-434.979	INCOME	NONE	
4	0 Converged	-443.669	INCOME	SQRT	
5	0 Converged	-454.364	INCOME	LOG	
6	0 Converged	-478.693	INCOME	INV	

variable, calculating the log likelihood measure as shown in Figure 4. This measure is automatically produced by most statistical software packages.

If you do the same for various transformations such as the square, the square root, and the log of the variable, pick the transformation where the log likelihood measure is closest to 0. Include this "winning" transformation in your full regression model along with other predictors. However, there is no guarantee that such a transformation will improve the overall fit of the model once the other predictor variables have been added. Remember, modeling is as much of an art as it is a science!

Figure 5



Interactions

Additional variables can be created reflecting the interaction of two variables together that may sometimes end up increasing your model's accuracy. Usually the creation of these variables comes from prior knowledge as to what should influence default probabilities or LGD. To create such a variable, simply multiply together the variables you wish to interact and include the new variable in the regression.

Influential Outliers

Observations called *influence points* can impact the parameter estimates more than others. During the model development process, it is always good to produce an influence plot so you can decide what corrective action (if any) is necessary. Figure 5 is a bubble chart showing that observation #100 exerts significant influence on the size of your regression estimates. This type of analysis usually uses a shortcut approach to see how the predicted probabilities (and estimated parameters) would change if each observation were removed. The rule of thumb is to examine any observations that have a change in *deviance* greater than 4.0, as represented by the dotted horizontal line in Figure 5. Some software packages call these measures by different names, but most produce some option for looking at influential observations. In reality, there may always be some observations that show up on a graph of this type, but the worst ones should be reviewed and a decision made as to whether they should be deleted. With respect to a model's implementation code, it is always a good idea to limit extreme predictor values to certain maximums or minimums. For example, you might want to restrict an LTV value to be between 0 and 100 in the coding algorithm before coefficients or weights are used to predict the probability of default.

Segmentation

Sometimes a better overall PD or LGD model can be built if a segmentation analysis is performed. Segmentation in this context refers to the development of more than one PD or LGD model for better overall accuracy. The effectiveness of devoting the time needed to build additional models depends on the predictive information available and whether or not the segmented populations are really that different from one another. Sometimes the segmentation groups are obvious, such as modeling consumer loans apart from loans to small businesses. Sometimes, the segmentation is not so obvious. For example, you could build a PD model for loans with high, medium, or low outstanding balances, or separate models for accounts with different levels of income or LTV.

So how do you know how to define the high, medium, or low segmentation breaks? One common approach is something called CART. CART stands for Classification and Regression Trees—a method

quite different from regression analysis. It uses an approach referred to as *recursive partitioning* to determine breaks in your data. It does this by first finding the best binary split in the data, followed by the next best binary split, and so forth.

CART is an exploratory analysis tool that attempts to describe the structure of your data in a tree-like fashion. S-Plus, one of the more popular CART vendors, uses deviance to determine the tree structure. As in regression, CART relates a single dependent variable (in our case, default or LGD) to a set of predictors. An example of this tree-like structure is shown in Figure 6. CART first splits on accounts where LTV is greater than and less than 54.5%. Therefore, you could design a PD model for accounts with LTVs below this cutoff and one for accounts greater than this cutoff. If you had even more time on your hands, you might want to consider using *score* and *income* to further define your segmentation schemes.

Sometimes it is possible to use CART analysis directly in a logistic regression model. This is a much quicker process than building and implementing separate regression models for each segment. So how can we integrate the CART methodology into a single PD model? The answer is to construct a variable that reflects the tree structure for the branches in the tree. Figure 6 shows that the tree structure produces five nodes. A node represents the end of the splitting logic and the segmentation branch of the tree. Therefore, in order to create a CART segmentation variable for our regression model, we would code something like this:

```
IF LTV<.545 AND SCORE<651.5 THEN NODE=1;
IF LTV<.545 AND SCORE>=651.5 THEN NODE=2;
Etc.
```

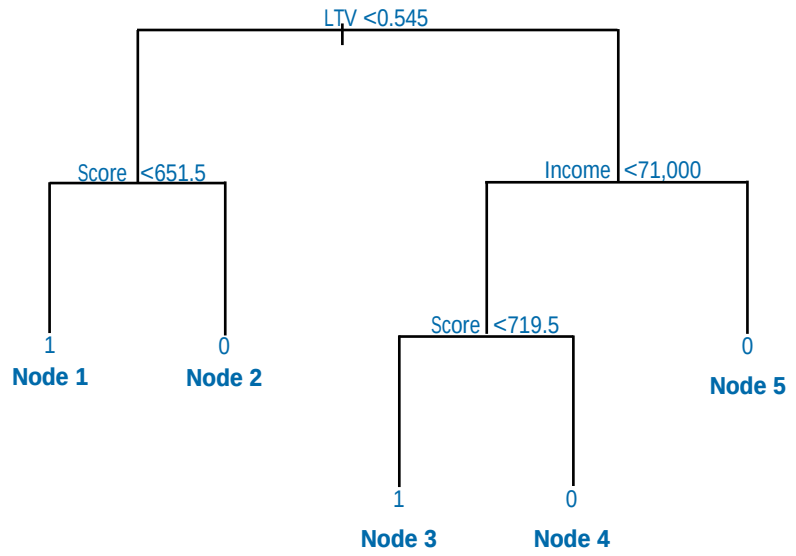
The same type of coding would be done for nodes 3-5. To introduce this information content into logistic regression, we would create dummy variables for each node and use them as predictors in our PD model.

Summary

Getting the most out of the data is one of the biggest challenges facing the model builder. *Anyone* can build a model. However, the challenge is building the most predictive model possible. This article has offered some tips on "building a better mousetrap" and how to go about it in a way that is based on sound statistical approaches that should satisfy most regulators. On the other hand, it must be remembered that building PD and LGD models is more an art than a science. Today's model builders come from a variety of disciplines, have different academic training and experience, and are accustomed to using certain statistical tools. Regardless of these differences, there is simply no substitute for knowing your data. Understanding how it was developed, what values are reasonable, and how the business works is essential in the model development process. ❐

Figure 6

Typical CART. Segmentation Analysis in S-PLUS Software



Contact Morrison by e-mail at Jeff.Morrison@suntrust.com.

References

1 Jeffrey S. Morrison, "Preparing for Modeling Requirements in Basel II, Part 1: Model Development," *The RMA Journal*. May 2003.

2 David Hosmer and Stanley Lemeshow. *Applied Logistic Regression*,©1989 John Wiley & Sons, Inc.

3 Bryan D. Nelson,"Variable Reduction for Modeling Using PROC VARCLUS," Paper 261-26. SAS User-group publication.