Print

Summary of Lesson 3: Preparing the Input Variables

Handling Missing Values

The data that you use for predictive modeling is likely to have missing values. It's important to replace missing values with reasonable estimates. Missingness (the probability that a value is missing) can be either dependent on or independent of the data. When the probability of missingness is independent of both the observed and unobserved measurements, the data values are said to be "missing completely at random" or MCAR. In predictive modeling applications, missingness is usually dependent on the data.

In PROC LOGISTIC, as in most other SAS modeling procedures, the default method of treating missing values is complete case analysis. Complete case analysis uses only the cases that have no missing values. However, complete case analysis is not a good choice for predictive modeling.

As an alternative to dropping observations that have missing values, you can replace missing values with reasonable values. This process is called imputation. There are many methods of imputing missing values.

To handle missing values effectively for predictive modeling, you can impute missing values and create missing value indicator variables — one indicator variable for each input that has any missing values. The steps of this method vary slightly depending on the type of input variable. For numeric inputs, you first create a missing value indicator variable for each numeric input that has any missing values. You treat all missing value indicators as new input variables in the analysis. Then you impute a value for each missing value. For categorical inputs, you don't need to create missing value indicators. Instead, you create a missing value level. Then you replace any missing values of the input with the new level. This approach to handling missing values accomplishes three important goals of predictive modeling: to retain all the original data for model development, to score all new cases, and to capture the relationship of missingness with the target.

View the "Imputing Missing Values" demonstration that shows how to use SAS, including PROC STDIZE, to impute missing values.

Imputing values by using the unconditional mean or median of the variable does not take into account the relationship with other inputs. One way to impute values that are conditional on other inputs is cluster imputation. In cluster imputation, you cluster the cases into relatively homogeneous subgroups, based on the values of several inputs.

Working with Categorical Inputs

In predictive modeling, categorical inputs with many levels — mainly, nominal inputs — can cause two main problems: high dimensionality and quasi-complete separation. Dummy variables work fine for categorical variables that have a small number of levels. However, expanding these inputs into dummy variables can greatly increase the dimension of the input space, which produces many redundant and irrelevant inputs in the model. Quasi-complete separation occurs when a level of the categorical input has a target event rate of either 0% or 100%. Quasi-complete separation can cause several problems. It complicates the interpretation of a logistic regression model. It can also affect the convergence of the estimation algorithm, and it might lead to incorrect decisions about variable selection. To solve the common problems associated with categorical inputs in a predictive model, you can choose among several methods. If a categorical input has too many levels to create dummy variables for each level, a smarter method is to use the categorical input to link to other data sets. This method uses subject-matter knowledge to create new numeric inputs (smarter variables) that represent relevant sources of variation. Collapsing categories by thresholding is another strategy for dealing with categorical inputs that have a large number of levels. This method requires a minimum number of cases in a level in order to create a dummy code input for that level.

Another method of dealing with categorical inputs that have many levels is to collapse the levels of the categorical variable based on an algorithm developed by Greenacre. This method hierarchically clusters the levels (that is, the rows of the two-way contingency table) based on the reduction in the chi-square test of association between the categorical input variable and the target.

View the "Collapsing the Levels of a Nominal Input" demonstration that shows how to use SAS, including PROC CLUSTER and PROC TREE, to collapse the levels of a nominal input according to Greenacre's method.

Another technique for working with categorical inputs is to replace the values with a single column that represents the event rate for each category. The categorical variable is converted to a continuous variable. The smoothed weight-of-evidence method (SWOE) is a general approach that avoids overfitting by taking into account the sampling variability. This method uses an adjusted log(odds). The smoothed weight-of-evidence method is similar to averaging the observed log(odds) in a particular group with the log(odds)in the overall sample.

View the "Computing the Smoothed Weight of Evidence" demonstration that shows how to use SAS to compute the smoothed weight of evidence.

Reducing Redundancy by Clustering Variables

When variables are redundant, they are highly correlated with each other. Redundancy is different from relevancy, which is the relationship between the inputs and the target. Redundancy does not involve the target variable. Including redundant inputs in your model can degrade your analysis. In high-dimensional data sets, the correlations among the inputs can make it difficult to identify relevant inputs. So it's a good strategy to first reduce redundancy and then tackle irrelevancy in a lower dimension space. When you have a relatively small number of variables, you can easily identify the correlations between them by looking at a correlation matrix. But when you have a large number of inputs, you must use other methods to detect correlations.

One recommended method of reducing redundancy is variable clustering. Variable clustering has two main steps. First, variable clustering identifies clusters of variables that are highly correlated among themselves and not highly correlated with variables in other clusters. Then, to reduce the number of inputs, this method selects a variable from each cluster.

When you're working with a large number of variables, it's not possible to identify clusters manually. Instead, you can use PROC VARCLUS to cluster the variables. PROC VARCLUS clusters variables by using an algorithm called iterative principal components analysis. Principal components are weighted linear combinations of the input variables where the weights are chosen to account for the largest amount of variation in the data. The principal components are produced by an eigenvalue-decomposition of the correlation matrix. The eigenvalues are the variances of the principal components. The eigenvalues are standardized so that their sum is equal to the number of principal components, which is equal to the number of variables.

PROC VARCLUS uses a divisive clustering algorithm to cluster inputs. At each stage, divisive clustering splits a given subset of variables into two groups. At each stage, divisive clustering splits a given subset of variables into two groups. All variables start in one cluster. Then, a principal components analysis is done on the variables in the cluster to determine whether the cluster should be split into two subsets of variables. The cluster needs to be split if the second eigenvalue exceed a cutoff or threshold. Selecting a cutoff is a subjective decision. The value 1 is a common choice for a cutoff because it represents the average size of the eigenvalues. However, to account for sampling variability, smaller values such as .7 have been recommended.

In a PROC VARCLUS step, you usually need only the VAR statement in addition to the PROC VARCLUS statement. The VAR statement specifies the numeric variables to be clustered.This lesson presents two options that you can specify in the PROC VARCLUS statement. The MAXEIGEN= option specifies a cuoff value other than *1*, which is the default. The SHORT option suppresses printing of the cluster structure, scoring coefficient, and intercluster correlation matrices.

```
PROC VARCLUS DATA=SAS-data-set <options>;
     VAR variables;
RUN;
```

After you've clustered your variables, you can reduce redundancy by selecting a representative variable from each cluster. An ideal representative has a high correlation with its own cluster and has a low correlation with the other clusters. To select a representative variable that meets these criteria, you can use the 1-R2 ratio. In addition to the 1-R2 ratio, there are other criteria that you can consider when you select inputs for your analysis. These criteria include subject-matter knowledge, a high correlation between the input and the target, variables that will cost the least amount of money to include, and variables that your peers and management think are important to control for.

View the "Reducing Redundancy by Clustering Variables" demonstration in SAS that shows how to use PROC VARCLUS to cluster variables.

Performing Variable Screening

Logistic regression was developed for inputs with effects that have a constant rate of change — that is, for inputs that have a linear relationship with the target. When this linear relationship is violated, the predictive accuracy of the model might decrease. One way to detect problematic nonlinear associations is to use a variable screening method that compares the ranks of the Spearman correlation statistic with the ranks of the Hoeffding's D statistic. When the Spearman rank is high, the association is monotonic, regardless of whether the Hoeffding value is high or low. Monotonic associations are not a problem for the model. However, when the Spearman rank is low and the Hoeffding rank is high, the association is non-monotonic. You should investigate this nonlinear pattern. When the Spearman rank is low and the Hoeffding rank is low, the association is weak. A weak association indicates a clearly irrelevant input that you can eliminate. Even though you have already used variable clustering to reduce redundancy, some further variable reduction might be needed prior to using the variable selection techniques in PROC LOGISTIC. Very liberal univariate screening might be helpful when the number of clusters created in PROC VARCLUS is still relatively large.

View the "Performing Variable Screening" demonstration that shows how to use SAS to perform variable screening.

In regression analysis, it is standard practice to look for nonlinear relationships between each input and the target by examining scatter plots. But when the target is binary, a scatter plot is not very enlightening. You have only a lot of ones and zeros. A plot of the empirical logits can be useful to detect nonlinear relationships. To create an empirical logit plot, you transform Y into the logit by using the log of *p* over 1-*p*. In the empirical logit plot, the Y axis is the logit axis. The relationship between X and the logit is clear; it's a nonlinear relationship that you might call approximately quadratic. If you have the time, it's worth the effort to create empirical logit plots for all of your continuous or ordinal inputs. For continuous inputs, you will need to bin the X values because there will be too few cases for each unique value of X. For binary inputs, it's a waste of time to create empirical logit plots. There is one problem with logit plots: when *p*=1 or 0, the logit is infinite. To solve the problem of infinite logits, you calculate a smooth estimate by adding a positive term, called a smoothing term, in both the numerator and the denominator of the empirical logit formula.

View the "Creating Empirical Logit Plots" demonstration that shows how to use SAS to create empirical logit plots.

There are various remedies for nonlinear relationships in your logistic regression model. One remedy is to create hand-crafted new input variables by transforming or discretizing the original inputs. Another possible way to handle nonlinear effects is to use a polynomial model; you can add quadratic and cubic terms. A multivariate function estimator can be a flexible alternative to logistic regression. Finally, some people think it's best to simply continue using standard logistic regression. If time permits, you can increase the predictive accuracy of the model by doing various modifications that can accommodate the nonlinear relationships.

View the "Accomodating a Nonlinear Relationship" demonstration that shows how to use SAS to accommodate a nonlinear relationship.

Selecting Variables Sequentially

In PROC LOGISTIC, you specify a subset-selection method in the MODEL statement by using the SELECTION= option. This lesson discusses three methods of subset selection: the best-subsets selection method, which is also called all-subsets selection (SELECTION=SCORE); stepwise selection (SELECTION=STEPWISE); and backward elimination (SELECTION=BACKWARD). By default, SELECTION=NONE.

```
MODEL response=<effects></options>;
```

```
SELECTION=BACKWARD | B
          | FORWARD | F
          | NONE | N
          | STEPWISE | S
          | SCORE
```

To change the default settings for the subset selection methods, you can also specify effect selection options. For example, the backward elimination method uses the significance level of the Wald chi-square to determine whether an effect stays in the model. The SLSTAY= option specifies the value of the significance level for an effect to stay in the model. The default value of the SLSTAY= option is .05. To specify a different significance level for an effect to stay in the model, you specify a value of the SLSTAY= option between 0 and 1, inclusive.

```
SLSTAY=value
```

The stepwise selection method also uses the significance level of the Wald chi-square to determine whether an effect enters the model. The SLENTRY= option specifies the value of the significance level for an effect to enter the model. The default value of the SLENTRY= option is .05. To specify a different significance level for an effect to enter the model, you specify a value of the SLENTRY= option between 0 and 1, inclusive.

```
SLENTRY=value
```

The best-subsets selection method does the most thorough search of the input variables. This method considers all possible models and rank-orders them based on the score chi-square. If you're working with a large number of inputs, the best-subsets selection method becomes prohibitively expensive with respect to computer resources.

Stepwise selection starts with an empty model. This method adds inputs incrementally until no more inputs meet the entry criterion for statistical significance. As variables are sequentially added to the model, previously selected variables are also evaluated, and are removed if they do not meet the criterion to stay in the model. Stepwise selection does not give good results when there is a high degree of redundancy.

Backward elimination starts with a full model. At each step, the test statistics are computed, and the variable with the largest *p*-value that exceeds the SLSTAY criterion is removed from the model. Backward elimination is less inclined to exclude important inputs or include spurious inputs than forward methods, such as stepwise selection. However, backward elimination does have a few problems related to starting with a full model.

To choose a method of subset selection, you might consider the amount of time that the method requires. In SAS, the relative efficiency of the subset-selection methods is different for logistic regression, as implemented in PROC LOGISTIC, than it is for linear regression. To compare the efficiency of best-subsets selection, stepwise selection, and fast backward selection (backward selection that uses the FAST option), a simulation was conducted with 50,000 cases and 200 intercorrelated inputs. The results show that the efficiency of each method changes as the number of inputs increases. For up to approximately 60 inputs, best-subsets selection requires the least time, backward elimination requires more time, and stepwise selection requires the most time. For any number of inputs, fast backward elimination is more efficient than stepwise selection. Fast backward elimination had the best overall performance, which is a linear increase in time as the number of inputs increased.

When you are selecting the most predictive inputs for a predictive model, it is important to look for interactions between inputs. Interaction occurs when the relationship between an input variable and the target differs by the level of another input variable. You can use various input selection methods to detect interactions. Forward selection is the method recommended in this course. In forward selection, you start with the main effects-only model. The algorithm then searches for any significant interactions between the effects. Starting with the main effects ensures that the model will be hierarchically well formulated. In a hierarchically well-formulated model, if an interaction is included, then the individual components of the interaction (which are main effects) must also be included.

In inferential statistics, it is common to use an arbitrarily selected significance level, such as .05. However, for predictive modeling, it is recommended that you use a significance level that is based on a model's fit statistics. The BIC is a measure of fit penalized for model complexity. Selecting the model with the smallest BIC favors a tight fit to the training data (that is, a large likelihood) and a small number of parameters.

View the 3.5 demonstrations that show how to use SAS to detect interactions, use backward elimination to subset the variables, display odds ratios for variables involved in interactions, create an interaction plot, use best-subsets selection, and use fit statistics to select a model.

Predictive Modeling Using Logistic Regression

Close