MWSUG 2017 - Paper AA02

Logistic Model Selection with SAS® PROC's LOGISTIC, HPLOGISTIC, HPGENSELECT

Bruce Lund, Magnify Analytic Solutions, Detroit MI, Wilmington DE, Charlotte NC

ABSTRACT

In marketing or credit risk a model with binary target is often fitted by logistic regression. In this setting the sample size is large and the model includes many predictors. For these models this paper discusses the variable selection procedures that are offered by PROC LOGISTIC, PROC HPLOGISTIC, and PROC HPGENSELECT. These selection procedures include the best subsets approach of PROC LOGISTIC, selection by best SBC of PROC HPLOGISTIC, and selection by LASSO of PROC HPGENSELECT. The use of classification variables in connection with these selection procedures is discussed. Simulations are run to compare these methods.

INTRODUCTION

This paper focuses on fitting of binary logistic regression models for direct marketing, customer relationship management, credit scoring, or other applications where (i) samples for modeling are large, (ii) there are many predictors, and (iii) many of these predictors are classification variables.

The automated selection of predictor variables for fitting logistic regression models is discussed. Four SAS procedures are compared:

1. PROC LOGISTIC with SELECTION = SCORE
2. PROC HPLOGISTIC with SELECTION METHOD = FORWARD (SELECT=SBC CHOOSE=SBC)
3. PROC HPGENSELECT with SELECTION METHOD = LASSO (CHOOSE=SBC)
4. PROC GLMSELECT with SELECTION = LASSO (CHOOSE=SBC)

The use of PROC GLMSELECT (method #4) may seem inappropriate when discussing logistic regression. PROC GLMSELECT fits an ordinary regression model. But, as discussed by Robert Cohen (2009), a selection of good predictors for a logistic model may be identified by PROC GLMSELECT when fitting a binary target. Then these predictors can be refit in a logistic model by PROC LOGISTIC. Potentially, the many features of PROC GLMSELECT can be leveraged to produce a selection of good predictors.

All four methods listed above can utilize the notion that optimal models are those with the lowest Schwarz Bayes criterion (SBC).[1]

ASSUMPTIONS:

It is assumed there is an abundant population from which to sample and that large sub-samples have been selected for the training and validation data sets. It is also assumed that a large set of potential predictor variables X_1 - X_K have been identified for modeling against a binary target variable **Y.**

SCHWARZ BAYES CRITERION

In model building where there are large training data sets and many candidate predictors, it is easy and tempting to over fit a logistic regression model. It is natural to want to use all the information in the fitting of models that has been uncovered through data discovery. But the impact of using "all the data" in fitting a logistic model may be the creation of data management overhead with either minimal or no benefit for out-of-sample prediction by the model.

[1] But for Method #1 the ranking is done by using a proxy for SBC. In this paper this proxy is called ScoreP. See the discussion of the Schwarz Bayes Criterion in the paper, and see the Appendix for discussion of the connection of ScoreP and SBC.

A widely used penalized measure of fit for logistic regression models is the Schwarz Bayes criterion (SBC). The formula is given below:

$$SBC = - 2 * LL + \log(n) * K$$

where LL is log likelihood of the logistic model, K is degrees of freedom in the model (including the intercept) and n is the sample size.

The theory supporting the Schwarz Bayes criterion is complex, both conceptually and mathematically. For a logistic regression modeling practitioner, the primary practical consequence of the theory is that a model with smaller SBC value is preferred over a model with a larger SBC value.[2] The use of SBC favors the selection of a parsimonious model and can help the modeler to avoid over fitting.[3]

PRE-MODELING BINNING

In direct marketing, customer relationship management, or credit scoring, the predictors in a logistic model often have only a few levels. Some predictors are nominal and others are ordered, including ordinal (not numeric) and discrete (numeric with a few levels, such as a count). These predictors will be referred to as NOD (nominal, ordinal, discrete).

It is common practice to “bin” the NOD predictors before model fitting. Binning is the process of reducing the number of levels of a predictor to a smaller number of bins (i.e. consolidations of levels). The goals of binning are to achieve parsimony and to find a relationship between the bins and the event rate (or, alternatively, the odds) that satisfies business expectations and common sense.

Binning can also be applied to continuous predictors [4] or nominal predictors with a large number of levels.[5] For a continuous predictor PROC HPBIN can be utilized to prepare the predictor for binning by consolidating the many levels into a manageable number of intervals.[6]

BINNING ALGORITHMS

If a predictor X is nominal (no ordering), there are two approaches to binning. They are: (1) collapsing of levels where each level is initially in a separate bin, and (2) splitting where all levels are initially in one bin. An algorithm called NOD_BIN for collapsing is given in Lund (2017). [7] The splitting algorithm is usually based on a decision tree.[8] SAS Enterprise Miner implements splitting in the Interactive Grouping Node.

If the predictor is ordered, then an optimal collapsing process is given in Lund (2017). The best k-bin ordered solution, based on either information value or entropy, is guaranteed to be found.

FINAL BINNING IS FINAL

If a binned predictor is selected for a final logistic model, then all levels (bins) should appear in the model.[9]

In summary, “final binning is final”.

[2] See Hastie, Tibshirani, Friedman (2008), Elements of Statistical Learning, pages 233-235

[3] An introductory discussion of “Information Criteria” (SBC, AIC, and more) is given by Dziak, et al. (2012).

[4] A continuous (or interval) predictor is numeric with many levels. Examples include measures of time, distance, money, etc.

[5] Examples of a nominal predictor with a large number of levels include SIC codes, county of residence, job classification, etc.

[6] Binning of continuous predictors may be preferred to looking for a transformation (such as log(X) or X^2) because the sample sizes for training and validation data sets in marketing or credit risk are often very large and binning makes full use of the data.

[7] For a different approach to collapsing, see Manahan (2006, in Appendix 1) for a program to collapse levels based on PROC CLUSTER.

[8] See SAS documentation about PROC HPSPLIT for a decision tree procedure.

[9] Two approaches of how to use binned X in a model are: (1) As a classification variable (via a CLASS statement), or (2) As a weight of evidence coded variable. The pros and cons of (1) and (2) are not discussed in this paper.

PROC LOGISTIC WITH SELECTION = SCORE

The score chi-square for a logistic model is reported by PROC LOGISTIC in the report "Testing Global Null Hypothesis: BETA = 0". The score chi-square closely approximates the likelihood ratio chi-square. But in contrast to the likelihood ratio chi-square, the score chi-square can be computed without actually finding the maximum likelihood solution for the logistic model.[10] The computational efficiency provided by using score chi-square is the key idea behind the Best Subsets approach to finding good multiple candidate models.

Best Subsets explained below

Best Subsets is implemented by PROC LOGISTIC with SELECTION=SCORE. The syntax is shown here:

```
PROC LOGISTIC; MODEL Y = <X's> / SELECTION=SCORE START=s1 STOP=s2 BEST=b;
```

The SELECTION statements "START" and "STOP" restrict the models to be considered to those where the number of predictors is between s1 and s2. Then for each k in [s1, s2] the option "BEST" will produce the b "best" models having k predictors. These b "best" models are the ones with highest score chi-square.[11]

With SELECTION=SCORE the model coefficients are not found and log likelihood (LL) and related statistics are not computed. Notably, this includes SBC which is derived from log likelihood.

Penalized Score Chi-Square

A "best" model is best only within a set of models with the same number of predictors. For example, {X1} might have the largest score chi-square among 1 variable models but {X1, X2} will have a larger score chi-square simply because a predictor was added to the model.

To make the models comparable, a penalty term is needed to reflect the number of degrees of freedom in the model and the sample size. In the absence of SBC, a substitute is "ScoreP", a penalized score chi-square defined by:

$$\text{ScoreP} = -\text{Score Chi-Sq} + \log(n) * K$$ [12]

where K is the degrees of freedom in the model (counting the intercept) and n is the sample size.

If all possible models are ranked by ScoreP and also by SBC, there is no guarantee that the rankings will be the same but the rankings will be very similar.

A Problem

The SELECTION=SCORE option does not support the CLASS statement. A solution is to convert a classification variable into a set of dummy variables. But with even a modest number of classification variables the conversion to dummies could bring the total number of predictors to 100 or more. Run time for PROC LOGISTIC increases exponentially as the number of predictors for SELECTION=SCORE becomes 75 and greater. This makes large scale dummy variable conversion not practical when using SELECTON=SCORE.

One work-around is to run a preliminary PROC LOGISTIC with SELECTION=BACKWARD FAST to reduce the count of predictors to the point where SELECTION=SCORE can be run.[13] But BACKWARD

[10] See Appendix for code to compute score chi-squares in the setting of Best Subsets.

[11] Example of START, STOP, BEST: Assume there are 4 predictors, X1 - X4 and START=1, STOP=4, and BEST=3. Then a total of 10 models are selected:

- From the four 1 variable models: {X1}, {X2}, {X3}, {X4}, the 3 with highest score chi-square are found
- From the six 2 variable models: {X1 X2}, {X1 X3}, {X1 X4}, {X2 X3}, {X2 X4}, {X3 X4}, the 3 with highest score chi-square are found
- From the four 3 variable models: {X1 X2 X3}, {X1 X2 X4}, {X1 X3 X4}, {X2 X3 X4}, the 3 with highest score chi-square are found
- From the one 4 variable model: {X1 X2 X3 X4}, the 1 and only model is found

[12] See Appendix for connection between ScoreP and SBC

[13] SAS Institute (2012, chapter 3, page 78).

FAST is unlikely to select all the dummies associated with a classification variable. This distorts the results of final binning of classification variables.

Another work-around is to convert all classification variables to weight of evidence (WOE) coding. This work around is described in Lund (2016). But, as described in the paper, this approach also has issues.

In summary, the applicability of Best Subsets approach is limited to models with numeric predictors or to models with a modest number of classification predictors (which would be converted to dummies).

PROC HPLOGISTIC

PROC HPLOGISTIC is one of the SAS high performance procedures. The performance of these procedures is discussed by Nguyen, et al. (2016). However, only the model fitting functionality of HPLOGISTIC is of interest for this paper. HPLOGISTIC provides predictor variable selection using the following methods: FORWARD (including FAST), BACKWARD, STEPWISE.[14] These methods are also provided by PROC LOGISTIC. But HPLOGISTIC adds new methods of selecting predictor variables beyond the selection by best significance level, as used by PROC LOGISTIC.

FORWARD SELECTION

In FORWARD selection, PROC LOGISTIC adds the predictor at each step with the most significant score chi-square, assuming that this significance level satisfies the threshold which the user sets via SLENTRY. When a CLASS variable is selected for entry, L-1 degrees of freedom (where L gives the number of levels) is utilized in computing the significance level.

Importantly, PROC HPLOGISTIC adds new criteria for entering predictor variables. One criterion is SELECT=SBC which causes the predictor to be entered which gives the lowest (best) SBC for the new model among all predictors currently available for entry. The syntax is given below:

```
PROC HPLOGISTIC DATA = <dataset>;
MODEL Y (descending) = <predictors>;
SELECTION METHOD=FORWARD (SELECT=SBC CHOOSE=SBC STOP=NONE);
```

The CHOOSE=SBC finds the model, along the FORWARD selection path, which gives the smallest SBC. The coefficients and model statistics from HPLOGISTIC are for this chosen model. The STOP=NONE directs HPLOGISTIC to continue the FORWARD selection until all predictors have been entered.

The syntax for the alternative of fitting a model by significance testing is given below:

```
PROC HPLOGISTIC DATA = <dataset>;
MODEL Y (descending) = <predictors>;
SELECTION METHOD=FORWARD (SELECT=SL SLENTRY = <a>);
```

If no predictor has score chi-square with significance below SLENTRY, the FORWARD selection stops.

The need to specify SLENTRY requires the modeler to make an arbitrary guess as to the relationship of the data to the predictors. Criticisms of FORWARD (also BACKWARD and STEPWISE), based on significance testing, in the case of ordinary linear regression are given in Flom and Cassell (2007). Their criticisms in the case of linear regression have analogues to logistic regression. Hosmer, et al. (2013) are supportive of using FORWARD, BACKWARD, STEPWISE in logistic regression as a data exploration tool.

In contrast, SELECT=SBC follows the strategy of looking for the best SBC model. Of course, the FORWARD path may fail to include the best overall SBC model. But for a large number of predictors the alternative of computing SBC for all models is not practical.

See Lund (2016) for a discussion of collecting all models selected by FORWARD and BACKWARD as well as candidate (i.e. not selected) models in order to generate a large number of models with their associated SBC's. If there are K predictors, this approach generates approximately K^2 models. The modeler can then undertake further analyses of these (approximately) K^2 models.

[14] HPLOGISTIC does not provide SELECTION=SCORE

The degrees of freedom of a CLASS predictor is incorporated in the calculation of SBC by HPLOGISTIC.

PARTITION and CHOOSE=VALIDATE

Instead of using the CHOOSE=SBC an alternative CHOOSE criterion is based on having a validation sample. The PARTITION statement allows the user to designate a variable (called PART in the example below) that indicates which observations are to be used for training and which for validation. (Alternatively, the user may generate the training and validation data sets within the processing of PROC HPLOGISTIC. See the FRACTION option within PARTITION.)

```
PROC HPLOGISTIC DATA = <dataset>;
PARTITION ROLEVAR=PART(TRAIN="1" VALIDATE="0");
MODEL Y (descending) = <predictors>;
SELECTION METHOD=FORWARD (SELECT=SBC CHOOSE=VALIDATE STOP=NONE);
```

With CHOOSE=VALIDATE the chosen model is the model with smallest ASE (average squared error) that results from scoring the validation data set at each step. Although SELECT=SBC will favor parsimonious predictor selection, CHOOSE=SBC cannot fully protect against over optimizing the model that is fit on the training data. The CHOOSE=VALIDATE insures that the model can generalize. When large samples are available the use of PARTITION and CHOOSE=VALIDATE is a preferred approach.

The validation sample, as part of <dataset>, need not consist of a random split of observations from a master file. Instead the validation observations could come from an out-of-time data set. However, it is probably best to reserve an out-of-time data set for a final model performance test.

The SELECT=SBC is especially effective at reducing multicollinearity among the predictors available for a model. This is seen from an analysis of the impact on SBC when adding a collinear predictor. Since the predictor is collinear, the increase in -2*LL would be small. But meanwhile the penalty term would be increased by log(n). SBC may be *increased* for a highly collinear predictor.

The discussion above has focused on FORWARD but much the same discussion applies to BACKWARD and STEPWISE. The focus was on FORWARD because it has similarities to selection by LASSO, to be discussed next.

PROC HPGENSELECT

PROC HPLOGISTIC in version 14.2 does not support selection by LASSO (least absolute shrinkage and selection operator) as a SELECTION method but LASSO is available in HPGENSELECT.

Here is an outline of the LASSO process for a logistic regression by HPGENSELECT:

LASSO fits a logistic model, once a positive value of λ is given, by solving for the coefficients in the objective function shown here:

$$F(\beta_0,\beta,\lambda) = \min_{(\beta_0,\ \beta)} \{ -LL(\beta_0,\ \boldsymbol{\beta}) \ + \ \lambda * \textstyle\sum_{k=1}^{K} |\beta_k| \}$$

where LL denotes log likelihood, β_0 is the intercept, and $\boldsymbol{\beta} = \beta_1, \ldots \beta_K$ are the coefficients.

Each λ gives rise to a model. Different λ's may be associated with models with the same predictors but with different values for the coefficients.

Let "maxλ" be the smallest λ where $F(\beta_0,\beta, \lambda) = -LL(\beta_0, 0, \ldots, 0)$. As λ decreases from maxλ and approaches 0, predictors enter or exit the model 1-at-a-time or in small groups. At λ = 0 the LASSO model is the usual maximum likelihood estimation (MLE) model which would be found by HPLOGISTIC.

If the same predictors are in LASSO model and MLE model, then β's and LL are not the same since different objective functions are being solved.

A sequence of λ's must be specified where the objective function will be evaluated. LASSORHO is an HPGENSELECT parameter (between 0 and 1) that starts this generation of λ values. The default for LASSORHO is 0.8. The first λ in the sequence is:

$$\text{LASSORHO} * (\max\lambda)$$

The j^{th} λ in the sequence is:

$$\text{LASSORHO}^{\,j} * (\max\lambda)$$

The number of lambda steps is given by LASSOSTEPS with default = 20 … $\lambda_1, \lambda_2, \ldots, \lambda_{20}$

In order to simplify this outline of the LASSO algorithm, the preceding formulation does not include the case of classification variables.[15] The LASSO implementation in HPGENSELECT is actually "group lasso" which has a more general objective function than given above. In group lasso an entire classification variable enters or exits the model. Individual levels of the classification variable cannot enter or exit.

However, a classification variable can be converted to dummy variables in HPGENSELECT by the use of SPLIT in connection with the CLASS statement. SPLIT applies to any SELECTION METHOD. More discussion of SPLIT is given in a later section.

Logistic models from HPLOGISTIC and HPGENSELECT compared on data set TEST

Data set TEST has 10,000 observations, a binary target Y, and predictors for logistic regression.

```
DATA TEST;
do ID = 1 to 10000;
  cumLogit = ranuni(2);
  e = 1*log( cumLogit/(1-cumLogit ));
  X1 = rannor(9); /* in model as quadratic */
  X2 = rannor(9); /* in model as log */
  X3 = rannor(9); /* only weakly in model */
  X4 = rannor(9); /* not in model */
  X5 = rannor(9); /* in model as a factor in interaction X7 */
  X6 = rannor(9); /* in model as a factor in interaction X7 */
  X7 = X6*X5;     /* in model */
  B1 = (ranuni(1) < .4); /* in model */
  B2 = (ranuni(1) < .6); /* in model */
  B3 = (ranuni(1) < .5); /* in model */
  B4 = (ranuni(1) < .5); /* in model */
  C12 = B1 + B2; /* indirectly in model */
  C34 = B3 + B4; /* indirectly in model */
xbeta = X1**2 + log(X2+8) + .01*X3 + 2*X7 + 0.1*B1 + B2 + B3 + B4 + e;
  P_1 = exp(xbeta) / (1 + exp(xbeta));
  Y = (P_1 > 0.95);
  /* To create a PARTITION role variable, add this line */
  IF ID <= 5000 then PART = 1; ELSE PART = 0;
  output;
  end;
```

In the following model fitting, we will assume that the predictors which are available for modeling are **restricted** to numeric X1-X7 and to classification variables C12 and C34.

The Wald chi-squares and p-values for these predictors are shown in Table 1.

From this group the four strongest predictors for a one-variable model are X7, C34, C12, X2. A weaker 5th strongest is X5.

[15] SAS documentation:
https://support.sas.com/documentation/cdl/en/stathpug/68163/HTML/default/viewer.htm#stathpug_hpgenselect_details29.htm

TABLE 1. Strength of One-Variables Predictors for Logistic Model

Effect	DF	Wald Chi-Sq	Pr > ChiSq	Strength
X1	1	0.01	0.9053	
X2	1	6.16	0.0130	4th
X3	1	0.06	0.8108	
X4	1	0.44	0.5066	
X5	1	4.96	0.0259	5th
X6	1	0.02	0.8823	
X7	1	1365.52	<.0001	1st
C12	2	88.42	<.0001	3rd
C34	2	329.70	<.0001	2nd

The predictors X1-X6 are very weakly correlated, but X7 is the product of X5 and X6. There is only weak association between C12 and C34 and the numeric predictors X1-X7. The best model should include predictors X7, C12, C34, probably X2, and possibly X5.

Here is the syntax for running a LASSO logistic model on data set TEST. The statement DISTRIBUTION=BINARY is needed to specify a logistic model.

```
PROC HPGENSELECT Data=TEST
LASSORHO=.80 /* default */ LASSOSTEPS=20; /* default */
CLASS C12 C34;
MODEL Y (descending) = X1-X7 C12 C34 / DISTRIBUTION=BINARY;
    SELECTION METHOD=LASSO (CHOOSE=SBC STOP=NONE) DETAILS=ALL;
run;
```

The CHOOSE=SBC has the effect of selecting the λ which gives the model with smallest SBC. This value of λ is not necessarily a value where a variable is added or removed. The coefficients and other statistics that are reported by HPGENSELECT are for this chosen model.

Table 2 below shows that selection of predictors by LASSO depends on the choice of LASSORHO and LASSOSTEPS:

Table 2. LASSO Selection of Predictors in Model

Model #	LASSO-RHO	LASSO-STEPS	Variables in Model	SBC Step	LASSO SBC
1	0.80	20	X2 X3 X4 X5 X7 C12 C34	20	8803.98
2	0.80	40	X1 X2 X3 X4 X5 X7 C12 C34	24	8802.36
3	0.85	20	X2 X5 X7 C12 C34	20	8898.06
4	0.85	40	X2 X3 X4 X5 X7 C12 C34	28	8801.98
5	0.90	20	X7 C34	20	9377.81
6	0.90	40	X2 X4 X5 X7 C12 C34	39	8807.40
7	0.70	20	X1 X2 X3 X4 X5 X7 C12 C34	15	8802.38
8	0.70	40	X1 X2 X3 X4 X5 X7 C12 C34	15	8802.38

The best SBC model for the default values of LASSORHO=0.8 and LASSOSTEPS=20 occurred at step 20 (model #1). This suggests that more steps are needed. With 40 steps the best SBC model occurred at step 24 (model #2). This model includes three weak variables X1, X3, X4, and the marginally strong variable X5.

Changing to LASSORHO=0.85, the model (model #4) with lowest SBC from the list above is found. Models #2, #7 and #8 are (essentially) equal models and these models have only slightly higher SBC.

Based on the one-variable results of Table 1, the model #3 has appeal. But the SBC of 8898.06 is not even close to the lowest among the LASSO models. Also the model found by CHOOSE=SBC occurred at the LASSOSTEPS limit of 20. This suggests more steps are needed.

Notably, the predictors selected in models #1 to #8 vary significantly by the specification of LASSORHO and LASSOSTEPS. The modeler faces the uncertainty of how to specify the search using LASSORHO and LASSOSTEPS.

In contrast the use of FORWARD selection with HPLOGISTIC and SELECT=SBC, CHOOSE=SBC identifies as the best model { X7, C34, C12 } and as the second best model { X7, C34, C12, X2 }.

```
PROC HPLOGISTIC DATA= TEST;
CLASS C12 C34;
MODEL Y (descending) = X1-X7 C12 C34;
SELECTION METHOD=FORWARD
(SELECT=SBC CHOOSE=SBC STOP=NONE);
```

The Table 3 gives the entry of predictors, the model's SBC, and shows an asterisk for the chosen model.

Table 3. HPLOGISTIC FORWARD Entry of Predictions

Selection Summary			
Step	Effect Entered	Number Effects In	SBC
0	Intercept	1	11596
1	X7	2	9749.21
2	C34	3	8866.34
3	C12	4	8749.31*
4	X2	5	8752.35
5	X5	6	8756.47
6	X4	7	8764.07
7	X3	8	8772.69
8	X1	9	8781.65
9	X6	10	8790.76

Predictive Accuracy

Although the LASSO model for the default setting of LASSORHO = 0.8 and LASSOSTEPS = 20 includes two weak predictors, the reader can independently check that the predictive accuracy of the LASSO Model { X2 X3 X4 X5 X7 C12 C34 } and the MLE Model { X7 C12 C34 } is very similar.

LASSO and MLE Models are not Comparable Except by Measuring Predictive Accuracy

HPLOGISTIC models are fit by MLE (maximum likelihood estimation). MLE and LASSO involve different objective functions and this has the following consequences:

- If the same predictors are in a LASSO model and a MLE model, then coefficients and log-likelihoods are not the same.
- When there are classification variables in the model, then SBC is computed differently for LASSO. If classification variable C has L levels, then "L" is used in computing the number of parameters K in the SBC formula. In contrast, L-1 is used for MLE. For LASSO each of the L levels of C does have a coefficient. This is unlike MLE where one level becomes the reference level.

SPLIT Option in CLASS Statement for HPGENSELECT

For any selection method (LASSO or otherwise) the SPLIT statement converts the levels of a classification variable to dummies that can enter or exit a model independently. For example:

```
PROC HPGENSELECT Data=TEST
LASSORHO=.80 /* default */ LASSOSTEPS=20; /* default */
CLASS C12(SPLIT) C34(SPLIT);
MODEL Y (descending) = X1-X7 C12 C34 / DISTRIBUTION=BINARY;
   SELECTION METHOD=LASSO (CHOOSE=SBC STOP=NONE) DETAILS=ALL;
```

The use of SPLIT has the effect of violating “final binning is final”. Some, but not all, levels of the classification variable are likely to be selected. The modeler then has to decide whether to violate the pre-modeling binning results.

PARTITION and CHOOSE=VALIDATE

Similar to HPLOGISTIC, the PARTITION statement is offered in HPGENSELECT. In this case there is a new CHOOSE option: CHOOSE=VALIDATE.

```
PROC HPGENSELECT DATA=TEST LASSORHO=.80 LASSOSTEPS= 20;
PARTITION ROLEVAR = PART(TRAIN="1" VALIDATE="0");
CLASS C12 C34;
MODEL Y (descending)= X1-X7 C12 C34 / DISTRIBUTION = BINARY;
SELECTION METHOD=LASSO (CHOOSE=VALIDATE STOP=NONE) DETAILS=ALL;
```

The CHOOSE=VALIDATE finds the model with the best (lowest) SBC on the validation data set. ASE is also computed on both the test and validation data sets but is not available for CHOOSE.

PROC GLMSELECT

PROC GLMSELECT fits an ordinary regression model.[16] But, as discussed by Cohen (2009), a selection of good predictors for a logistic model may be identified by PROC GLMSELECT against a binary target. Then these predictors can be refit in a logistic model using PROC LOGISTIC. Potentially, the many features of PROC GLMSELECT can be leveraged to produce a selection of good predictors.

Of particular interest is the use of LASSO. PROC GLMSELECT provides SELECTION=LASSO. The algorithm for fitting LASSO is a variant of LAR (least angle regression).[17] Predictors are entered or removed one-at-a-time with no intermediate steps between entry or removal. This algorithm is very fast in comparison with LASSO of HPGENSELECT.

The *default* for SELECTION=LASSO is to SPLIT classification variables. The user could employ the rule that any classification variable where at least one level was selected by LASSO would be entered into a final logistic modeling step as a classification variable.

As an alternative, PROC GLMSELECT provides SELECTION=GROUPLASSO. This gives a parallel approach to LASSO from HPGENSELECT, except of course, the model being fit is an ordinary regression model with a binary target. The SPLIT statement is supported and converts the levels of a classification variable to dummies that can enter or exit a model independently. The GROUPLASSO in GLMSELECT runs much faster than LASSO in HPGENSELECT on the same data set.

A downside to the GROUPLASSO approach is that the analogues of LASSORHO and LASSOSTEPS need to be specified. The analogous statements are RHO and MAXSTEP. As seen with HPGENSELECT this leads to uncertainty in specifying a LASSO run and in determining the best model between LASSO runs.

The benefit of SELECTION=LASSO is that RHO is not used. MAXSTEP is not really an issue. The default for MAXSTEP is two times the number of predictors. This allows for entry and removal of all predictors. For TEST this number is 28.[18]

Here is the PROC GLMSELECT syntax for GROUPLASSO. The default value of RHO (comparable to LASSORHO) is 0.9. The default for MAXSTEP for GROUPLASSO is 100.

```
PROC GLMSELECT DATA=TEST;
CLASS C12 C34;
MODEL Y = X1-X7 C12 C34 /
SELECTION=GROUPLASSO (MAXSTEP=50 RHO=.9 CHOOSE=SBC STOP=NONE);
run;
```

[16] Target (response) variable is interval scaled and is linked by the identity function to a linear combination of predictors plus an error term with a normal distribution

[17] See https://support.sas.com/documentation/onlinedoc/stat/141/glmselect.pdf, page 3852.

[18] Numeric predictors: 7, intercept: 1, total class levels: 6 for 14 and multiplying by 2 gives 28

The chosen model was {X7 C34 C12 X2 X5 X4} and this model occurred at step 40. The predictor X4 is a weak predictor which was entered last. This selection was successful as a starting point for refining the predictors for a logistic model.

Alternatively, PROC GLMSELECT is run below with SELECTION=LASSO.

```
PROC GLMSELECT DATA = TEST;
CLASS C12 C34;
MODEL Y = X1-X7 C12 C34 /
SELECTION=LASSO (CHOOSE=SBC STOP=NONE);
run;
```

The chosen model was { X7, part of C34, part of C12, X2, X5, X4 }. The modeler would inspect the chosen model and see that classification variables C34 and C12 should be included when fitting a final logistic model.

PARTITION and CHOOSE=VALIDATE

With PARTITION and CHOOSE=VALIDATE the chosen model is the model with lowest ASE (Average Squared Error) on the scored validation data set.

```
PROC GLMSELECT DATA = TEST;
PARTITION ROLEVAR=PART(TRAIN="1" VALIDATE="0");
CLASS C12 C34;
MODEL Y = X1-X7 C12 C34 /
SELECTION = LASSO (CHOOSE=VALIDATE STOP=NONE);
run;
```

SPLIT Statement for the three Procedures (HPGENSELECT, GLMSELECT, HPLOGISTIC)

It might be hard to keep track of the SPLIT statement for the three procedures under discussion. Table 4 includes a summary.

Table 4. Summary of Procedures and the SPLIT Statement

HPGENSELECT (distribution=binary)	SELECTION METHOD=LASSO • Algorithm=group lasso • Parameters – LASSORHO – LASSOSTEPS Slow, comparatively	SPLIT statement for CLASS variables
GLMSELECT (ordinary regression with binary target)	SELECTION=LASSO • Algorithm=LAR • No Parameters Fast	Splitting of CLASS variables is default. (SPLIT statement is ignored.)
GLMSELECT (ordinary regression with binary target)	SELECTION=GROUPLASSO • Algorithm=group lasso • Parameters: – RHO – MAXSTEPS Fast	SPLIT statement for CLASS variables
HPLOGISTIC	SELECTION METHOD=<any> Algorithm=maximum likelihood estimation Fast	No SPLIT statement

COMPARING LOGISTIC PROCEDURES ON A SIMULATED DATA SET WITH MANY X'S

The data set WORK0 has 10,000 observations with binary target Y. It is designed to look like a data set that could arise in model fitting for marketing or credit. There are 20 classification variables and 20 binary variables.

Of these, 10 classification variables CIn1 - CIn10 are in the model. These 10 variables have moderate levels of inter-predictor association (dependence). The other 10 classification variables DOut1 - DOut10 are not in the model.

There are 10 binary variables BIn1 - BIn10 which are in the model and they have moderate levels of correlation. The other 10 binary variables BOut1 - BOut10 are very weakly in the model. These variables are not correlated.

Four tests of selection of predictor variables were run:

1. PROC GLMSELECT with LASSO and CHOOSE=SBC
2. PROC GLMSELECT with GROUPLASSO and CHOOSE=SBC (no splits)
3. PROC HPGENSELECT with (group) LASSO and CHOOSE=SBC (no splits)
4. PROC HPLOGISTIC with FORWARD and (SELECT=SBC CHOOSE=SBC)

```
DATA WORK0;
array CIn{10} CIn1 - CIn10;
array COut{10} COut1 - COut10;
array BIn{10} BIn1 - BIn10;
array BOut{10} BOut1 - BOut10;

do i = 1 to 10000;
  do k = 1 to 10;
    COut{k} = floor((k+2)*ranuni(1));
    end;
  xbeta = 0;
  RU = ranuni(1);
  do j = 1 to 10;
    BOut{j} = (ranuni(1) < j/20);
    xbeta = xbeta + 0.01*BOut{j};
    end;
  do j = 1 to 10;
    BIn{j} = (.55*RU + .45*ranuni(1) < (j+1)/20);
    xbeta = xbeta + 0.15*BIn{j};
    end;
  do j = 1 to 10;
    CIn{j} = floor((j+2)*(0.5*RU + 0.5*ranuni(1)));
    end;
  do j = 1 to 10;
    do jj = 1 to 10;
      if j-1 = CIn{jj}
        then xbeta = xbeta + (-1)**j*(1/j)*(CIn{jj}/(1 + CIn{jj}));
      end;
    end;
  e = rannor(1);
  xbeta = xbeta + e;
  P_1 = exp(xbeta) / (1 + exp(xbeta));
  Y = (P_1 > 0.50);
  output;
  end;
run;
```

```
PROC GLMSELECT DATA = WORK0;
CLASS CIn: COut: ;
MODEL Y = CIn: COut: BIn: BOut:
   / SELECTION=LASSO (CHOOSE=SBC STOP=NONE);
run;

PROC GLMSELECT DATA = WORK0;
CLASS CIn: COut: ;
MODEL Y = CIn: COut: BIn: BOut:
   / SELECTION=GROUPLASSO (CHOOSE=SBC RHO=.90 MAXSTEPS=40 STOP=NONE);
run;

PROC HPGENSELECT DATA = WORK0 LASSORHO=.80 LASSOSTEPS=30;
CLASS CIn: COut: ;
MODEL Y (DESCENDING) = CIn: COut: BIn: BOut:
   / DISTRIBUTION=BINARY;
SELECTION METHOD=LASSO (CHOOSE=SBC STOP=NONE);
run;

PROC HPLOGISTIC DATA = WORK0;
CLASS CIn: COut: ;
MODEL Y (DESCENDING) = CIn: COut: BIn: BOut: ;
SELECTION METHOD=FORWARD (SELECT=SBC CHOOSE=SBC STOP=NONE);
run;
```

For all models CHOOSE=SBC was specified.

GLMSELECT with LASSO selected 13 predictors, the most of any method. Levels from COut3 COut4 COut8 COut9 were included in the selection. These predictors are not in the actual model.

GLMSELECT with GROUPLASSO, with default RHO=0.9, selected 2 binary predictors, BOut5 BOut8, that are only weakly in the model. Additionally, all predictors actually in the model were selected.

HPGENSELECT with GROUPLASSO had very similar results to GLMSELECT with GROUPLASSO. Default LASSORHO=0.8 was used but steps were increased to 30.

HPLOGISTIC with FORWARD and SELECT=SBC gave the most parsimonious model. All classification variables actually in the model were selected, but some binary variables actually in the model were not selected. No variables not in the model were selected.

Table 5. Comparison of Predictors Selected by Four Methods

1. GLMSELECT LASSO	CIn1 CIn2 CIn3 CIn4 CIn5 CIn6 CIn7 CIn8 CIn9 CIn10 BIn2 BIn3 BIn4 BIn5 BIn6 BIn7 BIn8 BIn9 BIn10 COut3 COut4 COut8 COut9 (*)
2. GLMSELECT GROUPLASSO	CIn1 CIn2 CIn3 CIn4 CIn5 CIn6 CIn7 CIn8 CIn9 CIn10 BIn1 BIn2 BIn3 BIn4 BIn5 BIn6 BIn7 BIn8 BIn9 BIn10 BOut5 BOut8
3. HPGENSELECT (GROUP)LASSO	CIn1 CIn2 CIn3 CIn4 CIn5 CIn6 CIn7 CIn8 CIn9 CIn10 BIn3 BIn4 BIn5 BIn6 BIn7 BIn8 BIn9 BIn10 BOut5 BOut7 BOut8
4. HPLOGISTIC FORWARD SELECT=SBC	CIn1 CIn2 CIn3 CIn4 CIn5 CIn6 CIn7 CIn8 CIn9 CIn10 BIn3 BIn6 BIn7 BIn8 BIn9 BIn10

(*) Classification variable is deemed as entered if any of its levels were selected by LASSO.

Methods #2 and #3 performed well but this might be a lucky outcome from the use of default values of RHO. The run-time for Method #3 was very much longer than for the other 3 methods.

Method #1 may over-select classification variables based on the practice that one level is enough to select the entire variable.

Method #4 omitted BIn1, BIn2, BIn4, BIn5. The next 3 variables to enter, in order, would be BIn2, BIn1, BIn5, but Bin4 would not be entered.

Would inclusion of BIn1, BIn2, BIn4, BIn5 give a better model? A model comparison test of the final HPLOGISTIC model vs. the final HPLOGISTIC model with the addition of the predictors BIn1, BIn2, BIn4, BIn5 is a 4 d.f. chi-square test. The result of the comparison gave a very significant p-value of 0.008. But adding BIn1, BIn2, BIn4, BIn5 increases SBC by 23.1. If lower SBC is the guide, then adding these 4 predictors is not supported.

A SIMULATED DATA SET WITH MULTICOLLINEARITY

In a data set WORK1 the predictors BIn1 - BIn10 are binary variables and each has a correlation of more than 0.96 with all the other BIn<x>'s. The true logistic model includes CIn1 - CIn10 (classification variables) and BIn1 - BIn10. Additional binary variables BOut1 - BOut10 are only weakly in the true model.

Only BIn1 - BIn10 and BOut1 - BOut10 will be *allowed* in the MODEL statements.

Both training and validation data sets are included in the sample and each had 10,000 observations. The PARTITION statement is used in HPLOGISTIC and HPGENSELECT with the partitioning variable PART to identify the training and validation data sets.

The purpose of this section is to show that HPLOGISTIC with FORWARD and SELECT=SBC, CHOOSE=SBC selects only a minimum of the collinear predictors while HPGENSELECT with LASSO and CHOOSE=SBC selects more of the collinear predictors.

```
DATA WORK1;
array CIn{10} CIn1 - CIn10;
array BIn{10} BIn1 - BIn10;
array BOut{10} BOut1 - BOut10;
do ID = 1 to 20000;
  xbeta = 0;
  RU = ranuni(1);
  do j = 1 to 10;
    BOut{j} = (ranuni(1) < 0.3);
    xbeta = xbeta + 0.01*BOut{j};
    end;
  /* High Multicollinearity is added here */
  do j = 1 to 10;
    BIn{j} = (.96*RU + .04*ranuni(1) < 0.50);
    xbeta = xbeta + 0.15*BIn{j};
    end;
  do j = 1 to 10;
    CIn{j} = floor((j+2)*(0.5*RU + 0.5*ranuni(1)));
    end;
  do j = 1 to 10;
    do jj = 1 to 10;
    if j-1=CIn{jj} then xbeta=xbeta + (-1)**j*(1/j)*(CIn{jj}/(1+CIn{jj}));
      end;
    end;
  e = rannor(1);
  xbeta = xbeta + e;
  P_1 = exp(xbeta) / (1 + exp(xbeta));
  Y = (P_1 > 0.50);
  IF ID <= 10000 THEN PART = 1; ELSE PART = 0;
  output;
  end;

PROC CORR DATA = WORK1;
VAR BIn: ;
where PART=1;
run;
```

Here is the HPLOGISTIC model and the Summary of selected predictors.

```
PROC HPLOGISTIC DATA = WORK1;
PARTITION ROLEVAR=PART(TRAIN="1" VALIDATE="0");
MODEL Y (DESCENDING) = BIn: BOut: ;
SELECTION METHOD=FORWARD (SELECT=SBC CHOOSE=SBC STOP=NONE);
```

The selection summary for HPLOGISTIC is shown below. Only three predictors: BIn4, BIn6, and BIn7 were selected.

Table 6. Selection of Predictors by HPLOGISTIC

Selection Summary			
Step	**Effect Entered**	**Number Effects In**	**SBC**
0	Intercept	1	11541
1	BIn4	2	9209.18
2	BIn6	3	8901.74
3	BIn7	4	8901.66*
4	BIn9	5	8906.54
5	BOut9	6	8911.44
6	BOut2	7	8916.82
7	BIn10	8	8923.50
8	BOut3	9	8931.01
9	BIn8	10	8938.92
10	BOut1	11	8947.36
11	BIn3	12	8956.12
12	BOut8	13	8965.02
13	BOut4	14	8973.93
14	BOut10	15	8983.03
15	BIn5	16	8992.13
16	BOut6	17	9001.27
17	BIn1	18	9010.44
18	BOut7	19	9019.64
19	BIn2	20	9028.85
20	BOut5	21	9038.06

Next is HPGENSELECT with LASSO.

```
PROC HPGENSELECT DATA = WORK1 LASSORHO=.80 LASSOSTEPS=30;
PARTITION ROLEVAR=PART(TRAIN="1" VALIDATE="0");
MODEL Y (DESCENDING) = BIn: BOut: / DISTRIBUTION = BINARY;
SELECTION METHOD=LASSO (CHOOSE=SBC STOP=NONE) DETAILS=all;
run;
```

Four predictors BIn4, BIn6, BIn9, BIn10 were entered as a group in the first step. Then BIn7, BIn8, BIn5 were entered. Selection ends at Step 14.

Table 7. Selection of Predictors by HPGENSELECT

Selection Details				
Step	Description	Effects In Model	Lambda	SBC
0	Initial Model	1	1	11540.875
1	BIn4 entered	5	0.8	10723.145
	BIn6 entered	5	0.8	10723.145
	BIn9 entered	5	0.8	10723.145
	BIn10 entered	5	0.8	10723.145
2	BIn7 entered	6	0.64	10157.826
3	BIn8 entered	7	0.512	9776.755
4		7	0.4096	9508.506
5	BIn5 entered	8	0.3277	9332.64
6		8	0.2621	9204.764
7		8	0.2097	9116.576
8		8	0.1678	9055.997
9		8	0.1342	9014.609
10		8	0.1074	8986.508
11		8	0.0859	8967.556
12		8	0.0687	8954.859
13		8	0.055	8946.406
14		8	0.044	8940.812*
15	BOut9 entered	9	0.0352	8946.262
Steps 16 – 30 Omitted				

Despite the inclusion of seven correlated predictors in the HPGENSELECT model, the predictive accuracy of the two models is essentially the same. Here is the average squared error as measured on the validation data set.

Table 8. Average Squared Error on the Validation Data Set

Procedure	Predictors	ASE
HPLOGISTIC SELECTION METHOD=FORWARD (SELECT=SBC CHOOSE=SBC)	Step-by-step: BIn4 BIn6 BIn7	0.1485
HPGENSELECT SELECTION METHOD=LASSO (CHOOSE=SBC)	First: BIn4 BIn6 BIn9 BIn10 Then step-by-step: BIn7 BIn8 BIn5	0.1485

CONCLUSIONS

PROC HPLOGISTIC with SELECT=SBC and CHOOSE=SBC is fast, finds parsimonious models, and appears to minimize the inclusion of collinear predictors. I see no convincing reason to switch to logistic modeling with GROUPLASSO for models designed for direct marketing, customer relationship management, or credit scoring.

MWSUG 2017, St. Louis, MO [i]

REFERENCES

Cohen R. (2009). Applications of the GLMSELECT Procedure for Megamodel Selection, *Proceedings of the SAS Global Forum 2009 Conference*, Paper 259-2009.

Dziak, J, Coffman, D., Lanza, S., Li, R. (2012). Sensitivity and Specificity of Information Criteria, The Methodology Center, Pennsylvania State University, Technical Report Series #12-119. https://methodology.psu.edu/media/techreports/12-119.pdf. Last Tested 7/30/2017

Flom, P and Cassell, D (2007). Stopping stepwise: Why stepwise and similar selection methods are bad, and what you should use, Proceedings of the Northeast SAS User Group (NESUG),

Hosmer D., Lemeshow S., and Sturdivant R. (2013). *Applied Logistic Regression, 3rd Ed.,* John Wiley & Sons, New York.

Lund B. (2016). Finding and Evaluating Multiple Candidate Models for Logistic Regression, *Proceedings of the SAS Global Forum 2016 Conference*, Paper 7860-2016.

Lund B. (2017). SAS® Macros for Binning Predictors with a Binary Target, *Proceedings of the SAS Global Forum 2017 Conference*, Paper 969-2017.

Manahan, C. (2006). "Comparison of Data Preparation Methods for Use in Model Development with SAS Enterprise Miner", *Proceedings of the 31th Annual SAS Users Group International Conference*, Paper 079-31.

Nguyen, et al. (2016). Do SAS® High-Performance Statistical Procedures Really Perform “Highly”? A Comparison of HP and Legacy Procedures**,** *Proceedings of the SAS Global Forum 2016 Conference*, Paper 11670-2016.

SAS Institute (2012). *Predictive Modeling Using Logistic Regression: Course Notes*, Cary, NC, SAS Institute Inc.

ACKNOWLEDGMENTS

Dave Brotherton of Magnify Analytic Solutions of Detroit provided helpful insights and suggestions.

CONTACT INFORMATION

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

Bruce Lund
Magnify Analytic Solutions
777 Woodward Ave, Suite 500
Detroit, MI, 48226
blund.data@gmail.com, blund.data@gmail.com, or blund@magnifyas.com

All code in this paper is provided by Marketing Associates, LLC. "as is" without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Recipients acknowledge and agree that Marketing Associates shall not be liable for any damages whatsoever arising out of their use of this material. In addition, Marketing Associates will provide no support for the materials contained herein.

APPENDIX

The connection between SBC and ScoreP

Likelihood ratio chi-square is:

$\text{LR } \chi 2 = -2*\text{LL}_r - (-2*\text{LL}_f)$ where “r” is intercept-only model and “f” is model with all predictors.

The likelihood ratio chi-square appears in SBC as shown:

$$\text{SBC} = -2*\text{LL}_f + \log(n)*K = -\text{LR } \chi 2 + \log(n)*K - 2*\text{LL}_r$$

The PROC LOGISTIC SELECTION=SCORE option gives Score χ2 where LR χ2 ~ Score χ2

So now, SBC ~ - Score χ2 + log(n)*K -2*LL_r

Definition: ScoreP = - Score χ2 + log(n)*K … by dropping -2*LL_r (constant across models)

Score chi-square computed without finding the maximum likelihood solution

This code shows, in principle, how the Score Chi-Square can be computed without maximizing the likelihood function.

* This SAS program demonstrates that SCORE-CHI-SQUARE from

PROC LOGISTIC MODEL Y = <> / SELECTION=SCORE

can be obtained without ever maximizing the Likelihood function. In the PROC LOGISTIC that is run in the middle of this program the MAXITER is set to 0. There are no iterations performed in solving the log-likelihood equations. The purpose of this PROC LOGISTIC is to produce the Covariance matrix with the evaluation at specified initial values. This Covariance matric is needed in the calculation of the SCORE-CHI-SQUARE. At the end of this program there is a real PROC LOGISTIC that actually runs the SELECTION=SCORE. Compare these results with the results that are produced by the preceding DATA step.;

```
* STEP 0: Create example for PROC LOGISTIC;
DATA work;
   DO i = 1 to 1000;
     w1 = 0.10*rannor(3);
     w2 = 0.15*rannor(3);
     y = (ranuni(1) < .5 + 0.1*w1 + .005*w2);
     output;
     END;
run;
* STEP 1: Obtain Intercept for Logistic Model with no predictors;
* First find the mean of Y;
PROC means DATA = work noprint; var y;
output out = meanout(keep= meanY)
mean = meanY;
run;
* Now transform meanY to the Logistic Intercept (no predictors);
DATA Estimate; SET meanout;
intercept = log(meanY/(1 - meanY));
run;
* STEP 2: Compute Score statistics for w1 and w2;
* The Score statistics are conditional on the Intercept;
* The Score equations (below) are the first derivatives of the log-likelihood
function but with the Intercept "in the model";
DATA U; SET Estimate work END=eof;
RETAIN b0 Uw1 Uw2 0;
KEEP Uw1 Uw2 _LINK_;
IF _N_ = 1 THEN DO;
   b0 = intercept;
   put b0 = ;
   END;
IF _N_ > 1
THEN
DO;
* Score Statistics;
   Uw1 = Uw1 + y*w1 - w1*(exp(b0) / (1 + exp(b0)));
   Uw2 = Uw2 + y*w2 - w2*(exp(b0) / (1 + exp(b0)));
   END;
IF eof THEN DO;
length _LINK_ $8;
_LINK_ = "LOGIT"; /* Used for MERGE below */
   OUTPUT;
   END;
run;
/* For use in initializing coefficients in PROC LOGISTIC */
DATA Initial; SET Estimate;
     w1=0; w2=0;
```

```
run;
/* Suppress unneeded Logistic output */
ODS EXCLUDE ALL;
/* MAXITER = 0 gives Covariance Matrix evaluated at Initial Values of Parameters */
PROC logistic DATA = work descending OUTEST = est_w covout INEST= Initial;
MODEL y = w1 w2 / maxiter = 0;
run;
ODS EXCLUDE NONE;
* STEP 3: Compute Score Chi-Square for w1 and w2;
DATA score_chi_sq; MERGE U est_w(where = (_name_ in ("w1", "w2"))) END=EOF;
   by _LINK_;
length VariablesInModel $5000;
retain score_chi_sq_sum 0;
IF _name_ = "w1" THEN DO;
   score_chi_sq = w1*Uw1**2;
   VariablesInModel = "w1";
   output;
   score_chi_sq_sum = w1*Uw1**2 + w2*Uw1*Uw2 ;
   END;
IF _name_ = "w2" THEN DO;
   score_chi_sq = w2*Uw2**2;
   VariablesInModel = "w2";
   output;
   score_chi_sq_sum = score_chi_sq_sum + w1*Uw1*Uw2 + w2*Uw2**2;
   END;
IF EOF THEN DO;
   score_chi_sq = score_chi_sq_sum;
   VariablesInModel = "w1 w2";
   output;
   END;
run;
* COMPARE Calculations above to SELECTION=SCORE;
PROC logistic DATA = work desc;
MODEL y = w1 w2 / selection = score
start=1 stop=2 best=2;
title "From PROC LOGISTIC SELECTION=SCORE";
run;
PROC PRINT DATA = score_chi_sq;
var score_chi_sq VariablesInModel;
title "From Calculation of Score Statistic and Covariance Matrix at Initial Values";
run;
```

Table 9. From PROC LOGISTIC SELECTION=SCORE

Score Chi-Square	VariablesInModel
1.8409	w1
1.0660	w2
2.9093	w1 w2

Table 10. From calculation with Score Statistic and Covariance Matrix at Initial Values

Score Chi-Square	VariablesInModel
1.8409	w1
1.0660	w2
2.9093	w1 w2

[i] Final3, 8/30/17