Paper 079-30

On GSForward—The Experimental Variable Selection Node in SAS® Enterprise Miner™ 5.1.

Leonardo Auslender, SAS Institute Inc., Bedminster, NJ

ABSTRACT

The GSForward node is an implementation of the methods described in Foster and Stine (2004), who present a novel approach in variable selection with a large number of variables.

INTRODUCTION

Foster and Stine provide a method to select a very good predictive model for a binary dependent target out of more than 67,000 predictors. They arrive at this large number of predictors by creating all product interactions of the original centered 200 or so variables, to which missing value indicators are also added. Variable centering (mean removal) ameliorates but does not eliminate colinearity arising from variable products. The missing values themselves are mean imputed. In proceeding in this overall fashion, they identify an efficient (sub)set of variables and disregard any potential alternative models while claiming to have averted the risk of model overfitting with their novel approach.

FOSTER AND STINE'S METHOD

Foster and Stine provide a formula for the acceptance or rejection of a predictor that eliminates the problem of multiple comparisons that renders invalid the inference process in the stepwise family. The traditional methods do not incorporate such a correction.

They call their inferential procedure "Adaptive Thresholding," as compared to the "Hard Risk Inflation Criterion" discussed in Foster and George (1994). In the latter, given p predictors, the RIC selects predictors the z-scores of which are larger than the threshold $\sqrt{2 \log p}$. Foster and George show that by using this threshold, the resulting MSE of the model is within a factor of $2 \log p$ of that obtained by estimating the true model. For $100 \leq p \leq 100{,}000$, the RIC threshold of $\sqrt{2 \log p}$ corresponds to a Bonferroni criterion with $\alpha = 0.20$.

Adaptive Thresholding reduces the threshold as more significant features are found; that is, it adjusts the threshold to accommodate problems in which more predictors appear useful. Assume p orthogonal predictors and rank their p-values in ascending order, $p_1 < p_2 < p_3 \ldots p_p$, with the associated $X_1, X_2, X_3, \ldots, X_p$. Enter X_1 if and only if $p_1 < \alpha / p$, otherwise stop and retain the null model. If X_1 is entered, then test $p_2 < 2\alpha / p$, … and more generally $p_q < q\alpha / p$, and so on, where q is the number of predictors already in the equation plus the predictor being tested, and $1 - \alpha$ is the confidence level. Stop when the inequality is not fulfilled. In the nonorthogonal case, it is necessary to orthogonalize the predictors at every step.

Foster and Stine aimed their procedure at the case of a binary target. The search procedure is computed by a modification of the traditional sweep operator. They further modified the search and predictor acceptance because they noted that in the case of a binary target with sparse data (especially due to interactions), influential variables would tend to dominate and produce highly inflated t-values with inflated significance. They bound the significance by way of Bennett's inequality (Bennett 1962) that provides a conservative estimate, which replaces the partial correlations of the typical forward search to determine entered predictors.

GSForward is based on a combination of forward selection and orthogonalization. Once a variable is chosen as a predictor candidate, its effect is removed from the dependent variable, when continuous, as well as from the remaining independent variables by way of Gram-Schmidt orthogonalization.

It is very important to remove multicollinear effects because the resulting models are more parsimonious without loss of explanatory power. Moreover, inclusion of unnecessary variables, however defined, increases the variance of the estimated coefficients. In many simulations that were performed GSForward always produced the most parsimonious representations.

SPECIFIC CHARACTERISTICS OF GSFORWARD

A contribution of GSForward is that it provides ways to speed up the processing time considerably. Since a forward search is greedy in the sense that it looks at one variable at a time, it is not necessary to work with the entire data set at once; instead, the user can work with batches and a number of samples within batches to speed up the search. In this sense, GSForward provides the following set of alternatives:

a) Work with the entire data set in one pass per model.

b) Work with *Batches* by specifying their desired number or the initial sample size. A batch is a step with a specific sample size associated with it. If the user opts to input the number of batches, the sample size for the first batch is calculated, and doubled at every new batch. The system is constrained to obtain the total number of observations in its last batch. The output of a typical batch is a subset of selected variables that, given the forward nature of the approach, will appear in the final model. However, a batch may not find any subsets.

On the other hand, if the user inputs the initial sample size, the number of batches is calculated and the initial sample size smoothed to correspond to the constraint just mentioned.

The *Sample Size* option has priority over the *Number of Batches*. Sampling is random and without replacement.

c) Within each batch and its corresponding sample size, the user can select the *Number of Samples* to work with. These samples are randomly generated with replacement. The variables chosen by *all* samples within a batch are retained, and not reanalyzed in later batches.

If the user chooses the number of batches to be 1, the entire data set is analyzed and the number of samples is also 1. The early stopping criteria detailed below are bypassed in this instance.

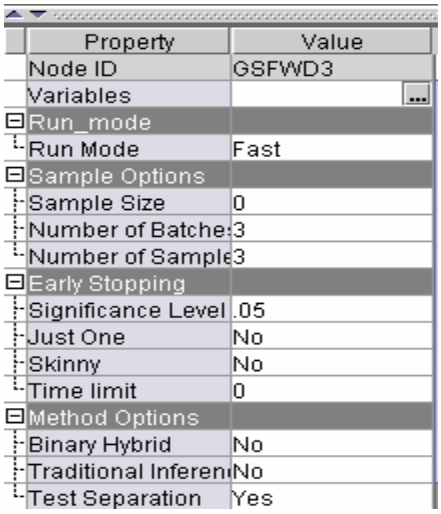
d) *Early Stopping* criteria: If some variables have been found in a batch, the user has the option of stopping the search earlier. Setting *Just One* to *Y* stops the search once a batch has found a subset of variables. Setting *Skinny* to *Y* stops the search if once a batch has found a subset the next two batches find no additional variables.

Implementation

GSForward has been implemented in SAS code using macros, the DATA step, IML, and existing procedures. The code is tuned to run inside the Enterpise Miner environment as an Extension node. Such nodes represent end-user additions to the EM tool set programmed in the SAS Language. See the Enterprise Miner documentation for more information on writing code to be used as an Extension node. The GSForward code only executes the variable search and selection process. The final models for these examples were computed using the Regression node in Enterprise Miner that computes both linear and logistic regressions using PROC DMREG.

Enterprise Miner Extension nodes may have their run time options presented to the user in a property sheet that is defined by an XML file. The property sheet for the GSForward node is shown below, and the options are discussed.

Options Available in the Advanced Setting



Property	Value
Node ID	GSFWD3
Variables	...
Run_mode	
Run Mode	Fast
Sample Options	
Sample Size	0
Number of Batche	3
Number of Sample	3
Early Stopping	
Significance Level	.05
Just One	No
Skinny	No
Time limit	0
Method Options	
Binary Hybrid	No
Traditional Inferen	No
Test Separation	Yes

Run Mode: Since there is no optimal setting for number of batches or initial sample size in combination with the early stopping criteria, GSForward enables the user to set them automatically. *Run Mode* has three settings:

Fast
Fastest
Advanced

Fast sets the initial sample size at 20 times the number of variables for a continuous dependent variable, and at 40 times the number of variables for a binary target. The number of batches is calculated thereafter. It sets on the *Skinny* option.

Fastest performs in the same way as *Fast* but sets on the *Just One* option instead. *Advanced* requires that the user input values for the number of batches, etc. as explained above.

By using *Early Stopping* criteria, the running time can be curtailed considerably, since in most situations without interactions the required number of observations to find significant effects is not large regardless of the number of variables. Still, simulations with large data sets with 50,000 observations and 5,000 variables of which 500 were "true" variables on a UNIX Sun Fire 3800, 8x900MHz USIII, 64GB RAM, StorEdge 3510 SAN configuration running Solaris 9 HTH took up to seven hours of computing time for a continuous dependent variable. GSForward identified 493 of the "true" variables and did not add any "false" ones. A stepwise run on the same run could not be completed.

Binary targets take much longer. In a recent run with genomics data that contained 94 observations and 24,000 variables, the running time was about eight hours on the same equipment. Since the data set contained only 94 observations, it was run without batching but with the introduction of a frequency variable that simulated a larger data set.

In the case of using all product interactions, they tend to dominate the selected model due to their sheer number. Since a computer search does not distinguish between a product interaction and the main components that created it, the final model might contain the former but not any member of the latter. Thus, Foster and Stine reject the principle of marginality that states that a model that contains product interactions must contain the main components (McCullaugh and Nelder 1989).

The *Significance Level* is set at a default (1 -) 5%. Lowering this value selects fewer variables and increasing this value selects a larger number.

Time Limit enables the user to input a maximum run time measured in minutes. GSForward will stop once a search is completed that has reached at least the time limit requested and report on the results up to that point. The default value of 0 enables the system to run without checking the time limit option.

Binary Hybrid is a combination of the searches for a binary and a continuous target. For a binary target, the first predictor is obtained in the same manner described above according to Foster and Stine (2004). The residuals are obtained and transformed into the 0–1 range, the obtained predictor is orthogonalized out of the remaining ones, and GSForward proceeds now as if the target were a continuous variable. The *Binary Hybrid* approach is faster than the standard binary approach, but it does not recalibrate and the results will differ.

The *Traditional Inference* option enables the user to obtain a standard Forward Selection in the case of a continuous target. If the user wants to produce a standard Forward Selection, it is more advisable to use the version already available in the regression node, which is faster.

Test Separation enables the user to prevent the found subset of variables to end up in a *quasi-separation* or *complete-separation* event. The system stops and reports on the subset generated just before the introduction of an additional variable that generates the event. While this is not a true solution, a better solution to this problem may be found in the future. This option is only available for the binary target, and not for the *Binary Hybrid*.

Given the forward nature of the method, interactions and dummies that increase the sparseness and dimension of the data require larger initial sample sizes. Resampling decreases the unstable nature of a stepwise search and the unanimous rule imposes a tyrannical acceptance threshold at every batch.

Comparison to Other Methods

As briefly mentioned above, in all simulations with artificial as well as with real data, GSForward outperformed present popular methods such as stepwise, forward, backward, and Bayesian (Shtatland et al. 2000) in terms of parsimony and accuracy of variable selection.

In a simulation study, a data set was created with 2,000 observations and 100 variables, of which 8 were ”true”’ variables that created a continuous target. The 100 variables had different degrees of correlation and interaction, and no time dependency. GSForward, Forward, Stepwise, and Backward regression were run, along with Decision Tree (with the default settings of SAS Enterprise Miner). Additional methods were also compared, but are omitted here for brevity’s sake. No method identified all 8 variables. The following tables illustrate the findings:

Method	# True Vars	# False Vars
GSForward	7	0
FORWARD	7	51
STEPWISE	7	12
BACKWARD	7	11
TREE	4	17

While all methods save trees identified 7 out of 8 “true” variables, all the methods except GSForward also included “false” variables. Inclusion of unnecessary variables increases the standard error of the coefficients of the true variables.

Method	Adj R^2	R^2	AIC	BIC	# Indep	RMSE
GSForward	0.95	0.95	20602.03	20604.1	7	172.18
FORWARD	0.95	0.95	20605.98	20613.9	58	170.2
STEPWISE	0.95	0.95	20575.76	20578.8	19	170.54
BACKWARD	0.95	0.95	20576	20578.9	18	170.59

In the Fit diagnostics table, GSForward has higher AIC and BIC statistics, while achieving the best model. AIC and BIC are standards for variable selection at present. In this instance, they do not penalize enough for inclusion of spurious variables.

In all simulations with known “true“variables, GSForward surpassed all other methods in accuracy of variable selection. In simulations where noise variables were added, GSForward did not select them, while other methods did occasionally. In all cases, the system did not require analyzing the full data set to obtain the best possible selection.

Comparative Case Study

A department store is interested in understanding the relationship of toy revenues to other items in the different stores. A database of transactions and revenues is created and rolled up at the individual store level. As is customary in today’s business databases, county-level demographic data is appended in the expectation of enriching the analysis.

The original data set contains 3,068 observations and 13 predictors. A quick exploratory run compared the variables selected by GSForward and Stepwise runs as a starting point of the modeling effort. For the sake of brevity, all variable descriptions are omitted.

Stepwise selected 19 predictors with R squares (adjusted and otherwise) just above 0.85, AIC = 37309.72.

Parameter		DF	Estimate	Standard Error	t Value	Pr > |t|
Intercept		1	-628.7	82.8957	-7.58	<.0001
COUGH_DLRS		1	0.0169	0.00131	12.96	<.0001
COUGH_UNITS		1	-0.0853	0.00546	-15.62	<.0001
FRAG_DLRS		1	0.00477	0.00169	2.82	0.0049
FRAG_UNITS		1	0.2651	0.00846	31.34	<.0001
ICDGRP_COUGH	1	1	74.1812	22.7199	3.27	0.0011
ICDGRP_COUGH	2	1	-170.9	47.2632	-3.62	0.0003
ICDGRP_COUGH	3	1	43.7691	23.4904	1.86	0.0625
ICDGRP_COUGH	4	1	26.4203	30.8899	0.86	0.3924
ICDGRP_SKIN	1	1	48.4504	35.6116	1.36	0.1738
ICDGRP_SKIN	2	1	15.6376	28.5567	0.55	0.5840
ICDGRP_SKIN	3	1	-68.9535	20.8878	-3.30	0.0010
MEDIAN_INCOME		1	0.00396	0.000729	5.43	<.0001
PCT_AMERICAN_INDIAN		1	73.1073	7.0306	10.40	<.0001
PCT_ASIAN		1	12.0191	1.5248	7.88	<.0001
PCT_HISPANIC		1	2.2598	0.5639	4.01	<.0001
PCT_WHITE		1	-1.4849	0.6442	-2.31	0.0212
POPULATION		1	-0.00318	0.000765	-4.16	<.0001
SKIN_DLRS		1	0.00350	0.000304	11.52	<.0001
SS_SQR_FT		1	0.0354	0.00591	5.98	<.0001

GSForward selected 6 predictors, with R squares (adjusted and otherwise) above 0.84, AIC = 37550.78.

PARAMETER ESTIMATES	Parameter Estimate	Standard Error	t Value	Pr > |t|
Variable				
Intercept	-399.9394022	48.057673377	-8.322	0.000
FRAG_UNITS	0.298157175	0.0037978191	78.507	0.000
PCT_ASIAN	12.204872074	1.3121300412	9.302	0.000
PCT_AGE_0_4	27.84184392	6.2538236094	4.452	0.000
SKIN_DLRS	0.0032245461	0.0002891999	11.150	0.000
COUGH_UNITS	-0.097061661	0.0050888923	-19.073	0.000
COUGH_DLRS	0.0198114719	0.0012133941	16.327	0.000

All of the GSForward chosen variables except PCT_age_0_4 were chosen by Stepwise, which means that the models are non-nested. While Stepwise has slightly higher R square and lower AIC measures, it is accomplishing that with 19 predictors instead of 6. In addition, 4 of the 19 coefficients are not significant at the 95% level.

In order to compare the non-nested models, we applied the J-test's 1981 version (Davidson and Mackinnon 1981; there are more recent versions of this same test) by way of the J_test1 macro in the SAS code. The corresponding output favors the results of GSForward:

```
**** J_TEST REJECTED MODEL REG2. THE ACCEPTED MODEL
**** CONTAINS THE FOLLOWING VARIABLES:
****
**** COUGH_DLRS COUGH_UNITS FRAG_UNITS PCT_AGE_0_4 PCT_ASIAN SKIN_DLRS
```

CONCLUSION

GSForward provides an additional tool to the already present arsenal of variable selection techniques in Enterprise Miner. The models selected by GSForward tend to be simpler than those found by other linear searchers. This result does not imply overall superiority because the search cannot be solely guided by automatic procedures. The cautious analyst should weigh the different results of the different tools and try to provide cogent reasoning for choosing one model over another, in addition to the results provided by SAS Enterprise Miner.

REFERENCES

Bennett, G. (1962), "Probability Inequalities for the Sum of Independent Random Variables," *Journal of the American Statistical Association,* 33–45.

Davidson, R. and Mackinnon, J. (1981), "Several Tests for Model Specification in the Presence of Alternative Hypotheses," *Econometrica,* 49, 781–793.

Foster, D. and George, E. (1994), "The Risk Inflation Criterion for Multiple Regression," *Annals of Statistics,* 1947–1975.

Foster, D. and Stine, R. (2004), "Variable Selection in Data Mining: Building a Predictive Model for Bankruptcy," *Journal of the American Statistical Association,* 303–313.

McCullagh, P. and Nelder, J. (1989), *Generalized Linear Models,* 2nd ed., London: Chapman & Hall.

Shtatland, E., Moore, S., Dashevsky, I., Miroshnik, I., Cain, E. and Barton, M. (2000), "How to Be Bayesian in SAS: Model Selection Uncertainty in PROC LOGISTIC and PROC GENMOD," *NESUG 2000 Proceedings,* Northeast SAS Users Group, Inc., 724–732.

ACKNOWLEDGMENTS

The author gratefully acknowledges the support of editor Virginia Clark.

CONTACT INFORMATION

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

Leonardo E. Auslender
1430 Route 206N, Suite 234
Bedminster, NJ 07921
908 470 0080 x 8217 (O)
908 470 0081 (F)
leonardo.auslender@sas.com