Paper SAS1742-2015

Introducing the HPGENSELECT Procedure: Model Selection for Generalized Linear Models and More

Gordon Johnston and Robert N. Rodriguez, SAS Institute Inc.

Abstract

Generalized linear models are highly useful statistical tools in a broad array of business applications and scientific fields. How can you select a good model when numerous models that have different regression effects are possible? The HPGENSELECT procedure, which was introduced in SAS/STAT® 12.3, provides forward, backward, and stepwise model selection for generalized linear models. In SAS/STAT 14.1, the HPGENSELECT procedure also provides the LASSO method for model selection. You can specify common distributions in the family of generalized linear models, such as the Poisson, binomial, and multinomial distributions. You can also specify the Tweedie distribution, which is important in ratemaking by the insurance industry and in scientific applications.

You can run the HPGENSELECT procedure in single-machine mode on the server where SAS/STAT is installed. With a separate license for SAS® High-Performance Statistics, you can also run the procedure in distributed mode on a cluster of machines that distribute the data and the computations.

This paper shows you how to use the HPGENSELECT procedure both for model selection and for fitting a single model. The paper also explains the differences between the HPGENSELECT procedure and the GENMOD procedure.

Introduction

Generalized linear models are highly versatile statistical models that have a huge range of applications. For example, these models are used in the insurance industry to set rates, in the airline industry to reduce the frequency of flight delays, and in health care to find relationships between cancer incidence and possible causes.

What makes these models so versatile? Generalized linear models accommodate response variables that follow many different distributions, including the normal, binomial, Poisson, gamma, and Tweedie distribution. Like other linear models, generalized linear models use a linear predictor. They also involve a link function, which transforms the mean of the response variable to the scale of the linear predictor.

The HPGENSELECT procedure was introduced in SAS/STAT 12.3 in July 2013. Like the GENMOD procedure, the HPGENSELECT procedure uses maximum likelihood to fit generalized linear models. In addition, PROC HPGENSELECT provides variable selection (including forward, backward, stepwise, and LASSO selection methods) for building models, and it supports standard distributions and link functions. It also provides specialized models for zero-inflated count data, ordinal data, and unordered multinomial data.

PROC HPGENSELECT is a high-performance analytical procedure, which means that you can run it in two ways:

- You can run the procedure in single-machine mode on the server where SAS/STAT is installed, just as you can with other SAS/STAT procedures. No additional license is required.
- You can run the procedure in distributed mode on a cluster of machines that distribute the data and the computations. Because each node in the cluster does a slice of the work, PROC HPGENSELECT exploits the computing power of the cluster to fit large models to massive amounts of data. To run in distributed mode, you need to license SAS High-Performance Statistics.

Comparing the HPGENSELECT and GENMOD Procedures

Like the GENMOD procedure, the HPGENSELECT procedure uses maximum likelihood to fit generalized linear models. Whereas the GENMOD procedure offers a rich set of methods for statistical inference such as Bayesian analysis and postfit analysis, the HPGENSELECT procedure is designed for predictive modeling and other large-data tasks. In addition, PROC HPGENSELECT enables you to do variable selection for generalized linear models, which is new in SAS/STAT. You can run PROC HPGENSELECT in single-machine mode and exploit all the cores on your computer. And as the size of your problems grows, you can take full advantage of all the cores and large memory in distributed computing environments.

Building Generalized Linear Models with the HPGENSELECT Procedure

In order to fit a generalized linear model, you specify a response distribution that is appropriate for your data, a set of independent variables (covariates), and a link function that transforms the linear predictor to the scale of the response. Covariates can be either continuous variables or classification variables, or they can be effects that involve two or more variables.

Table 1 shows the response distributions that PROC HPGENSELECT provides.

Table 1 Response Probability Distributions from PROC HPGENSELECT

Distribution	Default Link Function	Appropriate Response Data Type
Binary	Logit	Binary
Binomial	Logit	Binomial events/trials
Gamma	Inverse	Continuous, positive
Inverse Gaussian	Inverse square	Continuous, positive
Multinomial with generalized logit link function		Nominal categorical
Multinomial	Logit	Ordered categorical
Negative binomial	Log	Count
Gaussian	Identity	Continuous
Poisson	Log	Count
Tweedie	Log	Continuous or mixed discrete and continuous
Zero-inflated negative binomial	Log/logit	Count with zero-inflation probability
Zero-inflated Poisson	Log/logit	Count with zero-inflation probability

Examples

The following examples illustrate key features of the HPGENSELECT procedure.

Fitting a Poisson Model to Auto Insurance Data

This example uses an automobile insurance data set called OntarioAuto, which has about 500,000 observations. The data set contains a response variable, **NumberOfClaims**, which represents the number of claims that an individual policyholder submits in a certain time period. The log transform of its mean depends on the continuous regressors **PolicyAge**, **DriverAge**, and **LicenseAge** and on four classification regressors, **MultiVehicle**, **Gender**, **RatingGroup**, and **TransactType**. The logarithm of an exposure variable, **logExposure**, is used as an offset variable to normalize the number of claims to the same time period. The following statements use the HPGENSELECT procedure, running in single-machine mode, to fit a Poisson regression model that has all the variables:

```
libname Data 'C:\Data';
proc HPGenselect data=Data.OntarioAuto;
   class Gender RatingGroup MultiVehicle TransactType;
   model NumberOfClaims=MultiVehicle Gender RatingGroup
                        TransactType PolicyAge DriverAge
                        LicenseAge / dist=Poisson
                                     link=Log CL
                                     offset=logExposure;
   performance details;
   code file = 'AutoScore.txt';
run;
```

The LIBNAME statement specifies data that in this example happen to be saved locally on the computer on which SAS® is running. The CLASS statement identifies the classification variables in the model, and the MODEL statement specifies the response variable, the regression variables, and options such as the distribution, the link function, and the offset variable. The CL option requests that confidence limits for all model parameters be displayed.

The PERFORMANCE statement requests that procedure execution times be displayed. The CODE statement produces a text file named AutoScore.txt that can be used for scoring. This file contains fitted model information that can be included in a DATA step for scoring, as shown on page 4.

The procedure output in Figure 1 provides the settings that are used in this analysis. The "Performance Information" table shows that PROC HPGENSELECT executed in single-machine mode on four concurrent threads, which is the number of CPUs on the machine. The "Model Information" table shows model information, such as the distribution and link function that were used. The "Number of Observations" table shows the number of observations that were read and the number that were used in the analysis. More observations were read than were used in the analysis because some observations had missing values for either the response or regression variables.

Figure 1 Model Settings

Performance Information	
Execution Mode	Single-Machine
Number of Threads	4

Model Information	
Data Source	DATA.ONTARIOAUTO
Response Variable	NumberOfClaims
Offset Variable	logexposure
Class Parameterization	GLM
Distribution	Poisson
Link Function	Log
Optimization Technique	Newton-Raphson with Ridging

Number of Observations Read	567962
Number of Observations Used	386729

Figure 2 shows the levels of the classification variables that were listed in the CLASS statement, important fit statistics such as Akaike's information criterion (AIC), and the resulting parameter estimates, confidence limits, and standard errors.

Figure 2 Model Fit Results

Class Level Information		
Class	**Levels**	**Values**
Gender	2	F M
RatingGroup	12	02 05 08 11 14 17 20 23 26 29 30 31
MultiVehicle	2	Multi Single
TransactType	3	MOD NEW REN

Fit Statistics	
-2 Log Likelihood	29822
AIC (smaller is better)	29860
AICC (smaller is better)	29860
BIC (smaller is better)	30066
Pearson Chi-Square	517848
Pearson Chi-Square/DF	1.3391

Figure 2 *continued*

Parameter Estimates							
Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	-3.672780	0.196874	-4.05865	-3.28691	348.0274	<.0001
MultiVehicle Multi	1	-0.289906	0.041559	-0.37136	-0.20845	48.6613	<.0001
MultiVehicle Single	0	0	.	.	.	.	.
Gender F	1	0.051268	0.040167	-0.02746	0.12999	1.6291	0.2018
Gender M	0	0	.	.	.	.	.
RatingGroup 02	1	-0.770014	0.295287	-1.34877	-0.19126	6.8000	0.0091
RatingGroup 05	1	-0.157194	0.197180	-0.54366	0.22927	0.6355	0.4253
RatingGroup 08	1	-0.045116	0.193078	-0.42354	0.33331	0.0546	0.8152
RatingGroup 11	1	0.077805	0.189152	-0.29293	0.44854	0.1692	0.6808
RatingGroup 14	1	0.120489	0.185472	-0.24303	0.48401	0.4220	0.5159
RatingGroup 17	1	0.116955	0.184574	-0.24480	0.47871	0.4015	0.5263
RatingGroup 20	1	0.258596	0.184863	-0.10373	0.62092	1.9568	0.1619
RatingGroup 23	1	0.228393	0.184471	-0.13316	0.58995	1.5329	0.2157
RatingGroup 26	1	0.294483	0.186638	-0.07132	0.66029	2.4896	0.1146
RatingGroup 29	1	0.213461	0.191124	-0.16113	0.58806	1.2474	0.2640
RatingGroup 30	1	-0.014585	0.289647	-0.58228	0.55311	0.0025	0.9598
RatingGroup 31	0	0	.	.	.	.	.
TransactType MOD	1	0.262652	0.043490	0.17741	0.34789	36.4745	<.0001
TransactType NEW	1	0.139413	0.082705	-0.02269	0.30151	2.8415	0.0919
TransactType REN	0	0	.	.	.	.	.
PolicyAge	1	-0.005330	0.003386	-0.01197	0.00131	2.4770	0.1155
DriverAge	1	-0.000102	0.001619	-0.00327	0.00307	0.0040	0.9496
LicenseAge	1	-0.013132	0.002377	-0.01779	-0.00847	30.5271	<.0001

The timing table in Figure 3 shows that the procedure took a little more than two seconds to run.

Figure 3 Timing

Procedure Task Timing		
Task	Seconds	Percent
Reading and Levelizing Data	0.41	18.46%
Full model fit	1.81	81.54%

The following DATA step statements score the first 100 observations of the original data set by using the fitted model information in AutoScore.txt. The variable **P_NumberOfClaims** represents predicted values in the scored data set ScoreData.

```
data ScoreData;
   keep P_NumberOfClaims NumberOfClaims MultiVehicle Gender
        RatingGroup  TransactType PolicyAge DriverAge
        LicenseAge Exposure;
   set Data.OntarioAuto(obs=100);
   %inc 'AutoScore.txt';
run;
```

Figure 4 shows the first 10 observations in the scored data set. You can score any data set by using this method; the only requirement is that all the regression variables and the offset variable that are in the original model be present.

Figure 4 Scoring Data

Obs	NumberOfClaims	LicenseAge	PolicyAge	DriverAge	Exposure	TransactType	MultiVehicle
1	0	28	9.0	44	0.10685	MOD	Multi
2	0	23	10.0	63	0.10137	REN	Multi
3	0	24	7.0	45	0.00000	MOD	Multi
4	0	41	8.0	58	0.04110	MOD	Multi
5	0	29	20.5	47	0.00000	MOD	Multi
6	0	21	11.0	74	0.32329	MOD	Multi
7	0	19	7.0	76	0.00000	MOD	Multi
8	0	6	6.0	22	0.18904	MOD	Multi
9	0	49	6.0	69	0.01918	MOD	Multi
10	0	19	5.0	67	0.90959	MOD	Multi

Obs	RatingGroup	Gender	P_NumberOfClaims
1	17	M	0.001951
2	26	M	0.001802
3	17	F	.
4	17	M	0.000635
5	31	M	.
6	17	F	0.006718
7	02	M	.
8	17	M	0.004692
9	14	F	0.000284
10	29	M	0.020977

Fitting a Tweedie Model to Auto Insurance Data

Now, suppose you want to fit a model for the cost of claims instead of the number of claims. The OntarioAuto data set contains the variable **DollarClaims**, which represents the cost of an individual policyholder's claims over a period of time. Many observations have a value of 0 for **DollarClaims** because there were no claims for those observations. However, for observations that have nonzero cost, a continuous distribution is appropriate. The Tweedie distribution is sometimes used for this type of data because it can model continuous data that have a discrete component at 0.

The following statements use the HPGENSELECT procedure, running in single-machine mode, to fit a Tweedie regression model for **DollarClaims** by using the same regressors as in the previous example:

```
libname Data 'C:\Data';
proc HPGenselect data=Data.OntarioAuto;
   class Gender RatingGroup MultiVehicle TransactType;
   model DollarClaims=MultiVehicle Gender RatingGroup
                      TransactType PolicyAge DriverAge
                      LicenseAge / dist=Tweedie
                                   link=Log CL
                                   offset=logExposure;
   performance details;
run;
```

The “Model Information” table in Figure 5 shows the Tweedie model settings.

Figure 5 Model Information

Model Information	
Data Source	DATA.ONTARIOAUTO
Response Variable	DollarClaims
Offset Variable	logexposure
Class Parameterization	GLM
Distribution	Tweedie
Link Function	Log
Optimization Technique	Quasi-Newton

Figure 6 shows the resulting Tweedie model fit statistics and parameter estimates.

Figure 6 Fit Statistics

Fit Statistics	
-2 Log Likelihood	77912
AIC (smaller is better)	77954
AICC (smaller is better)	77954
BIC (smaller is better)	78183
Pearson Chi-Square	1.4562E9
Pearson Chi-Square/DF	3765.71

Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	5.231829	0.272081	4.69856	5.76510	369.7533	<.0001
MultiVehicle Multi	1	-0.351287	0.063940	-0.47661	-0.22597	30.1840	<.0001
MultiVehicle Single	0	0	.	.	.	.	.
Gender F	1	0.069925	0.060776	-0.04919	0.18904	1.3237	0.2499
Gender M	0	0	.	.	.	.	.
RatingGroup 02	1	-2.128568	0.405827	-2.92397	-1.33316	27.5102	<.0001
RatingGroup 05	1	-1.161764	0.273965	-1.69873	-0.62480	17.9823	<.0001
RatingGroup 08	1	-1.079393	0.269130	-1.60688	-0.55191	16.0855	<.0001
RatingGroup 11	1	-0.654231	0.260553	-1.16491	-0.14356	6.3048	0.0120
RatingGroup 14	1	-0.261659	0.251862	-0.75530	0.23198	1.0793	0.2989
RatingGroup 17	1	-0.122954	0.249669	-0.61230	0.36639	0.2425	0.6224
RatingGroup 20	1	-0.039529	0.251577	-0.53261	0.45355	0.0247	0.8751
RatingGroup 23	1	0.215756	0.249169	-0.27261	0.70412	0.7498	0.3865
RatingGroup 26	1	0.175484	0.253458	-0.32129	0.67225	0.4794	0.4887
RatingGroup 29	1	0.422019	0.256827	-0.08135	0.92539	2.7001	0.1003
RatingGroup 30	1	-0.512928	0.417520	-1.33125	0.30540	1.5092	0.2193
RatingGroup 31	0	0	.	.	.	.	.
TransactType MOD	1	0.336215	0.064126	0.21053	0.46190	27.4891	<.0001
TransactType NEW	1	0.061979	0.134865	-0.20235	0.32631	0.2112	0.6458
TransactType REN	0	0	.	.	.	.	.
PolicyAge	1	-0.009993	0.004904	-0.01960	-0.00038175	4.1527	0.0416
DriverAge	1	0.001526	0.002467	-0.00331	0.00636	0.3823	0.5364
LicenseAge	1	-0.009299	0.003383	-0.01593	-0.00267	7.5556	0.0060
Dispersion	1	1579.296618	25.892051	1529.35580	1630.86824	.	.
Power	1	1.562776	0.005967	1.55113	1.57451	.	.

The parameters **Intercept** through **LicenseAge** are regression parameters, and **Dispersion** and **Power** are Tweedie dispersion and power parameters, respectively.

The timing table in Figure 7 shows that PROC HPGENSELECT took slightly more than 1.5 minutes to run. This is considerably more time than the Poisson model took, because the Tweedie likelihood takes more resources to compute than the Poisson likelihood.

Figure 7 Timing

Procedure Task Timing

Task	Seconds	Percent
Reading and Levelizing Data	0.36	0.38%
Full model fit	94.57	99.62%

Model Selection with a Zero-Inflated Model

The examples in this section use a simulated data set named GLMData, which has 10 million observations. These data contain a response variable named **yZIP**, which is constructed to have a zero-inflated Poisson (ZIP) distribution. The response variable **yZIP** depends on a number of regression variables that are listed in Table 2. In addition to including these regression variables, the data set contains a number of noise variables that are unrelated to **yZIP**.

Table 2 Regressors for ZIP Model

Regressor Name	Type	Number of Levels	Role
xIn1–xIn20	Continuous		Regressor for Poisson mean
xSubtle	Continuous		Regressor for Poisson mean
xTiny	Continuous		Regressor for Poisson mean
xOut1–xOut80	Continuous		Noise
cIn1–cIn5	Classification	2–5	Regressor for Poisson mean and zero-inflation probability
cOut1–cOut5	Classification	2–5	Noise

The Poisson mean part of the ZIP model depends on the variables **xIn1–xIn20** and **cIn1–cIn5** through a logarithmic link. It also depends on the variables **xTiny** and **xSubtle**, but the dependence is considerably weaker. The zero-inflation probability depends on the classification variables **cIn1–cIn5** through a logit link function. The variables **xOut1–xOut80** and **cOut1–cOut5** are noise variables that are included in the model selection process but do not influence the response. A model selection procedure should screen out these variables as being unimportant to the model.

The following statements fit a zero-inflated model that uses **yZIP** as the response and all the variables in Table 2 as regressors. The HPGENSELECT procedure runs in single-machine mode in this example and uses only the first 50,000 observations from the data set GLMData to perform stepwise model selection. As in the preceding example, the data are saved locally on the computer on which SAS is running.

```
libname Data 'C:\Data';
proc hpgenselect data=Data.GLMData(obs=50000);
   class c:;
   model yZIP = x: c: / dist=ZIP;
   zeromodel c:;
   selection method=stepwise(choose=sbc);
   performance details;
run;
```

The MODEL statement specifies the model for the Poisson mean part of the model, and the ZEROMODEL statement specifies the model for the zero-inflation probability. The symbols **x:** and **c:** are shorthand for all variables that begin with **x** and **c**, respectively. The SELECTION statement requests that the stepwise selection method be used and that the final model be chosen on the basis of the best Schwarz Bayesian criterion (SBC).

The "Performance Information" table in Figure 8 shows that PROC HPGENSELECT ran in single-machine mode on four concurrent threads. The "Model Information" table shows model settings for the zero-inflated model. The "Number of Observations" table shows that 50,000 observations were used in the analysis.

Figure 8 Performance Information

Performance Information	
Execution Mode	Single-Machine
Number of Threads	4

Model Information	
Data Source	DATA.GLMDATA
Response Variable	yZIP
Class Parameterization	GLM
Distribution	Zero-Inflated Poisson
Link Function	Log
Zero Model Link Function	Logit
Optimization Technique	Newton-Raphson with Ridging

Number of Observations Read	50000
Number of Observations Used	50000

The “Selection Summary” table in Figure 9 shows that the model in step 30 was selected on the basis of the minimum SBC. None of the noise variables were selected. However, **xSubtle** and **xTiny** were not included in the model. Would performing model selection with more data provide a better model by including these variables?

Figure 9 Selection Summary

Selection Summary

Step	Effect Entered	Number Effects In	SBC	p Value
0	Intercept	1		
	Intercept_Zero	2	146817.850	.
1	cIn5_Zero	3	131452.827	<.0001
2	cIn4_Zero	4	119946.851	<.0001
3	cIn3_Zero	5	114440.541	<.0001
4	xIn20	6	110924.857	<.0001
5	xIn19	7	107695.553	<.0001
6	xIn18	8	104609.925	<.0001
7	xIn17	9	101844.442	<.0001
8	xIn16	10	99349.952	<.0001
9	xIn15	11	96947.036	<.0001
10	cIn2_Zero	12	94827.139	<.0001
11	xIn14	13	92923.729	<.0001
12	xIn13	14	91186.990	<.0001
13	xIn12	15	89697.891	<.0001
14	xIn11	16	88390.109	<.0001
15	xIn10	17	87236.680	<.0001
16	cIn4	18	86337.206	<.0001
17	cIn5	19	84618.512	<.0001
18	cIn3	20	83550.746	<.0001
19	xIn9	21	82695.565	<.0001
20	xIn8	22	81912.081	<.0001
21	xIn7	23	81311.094	<.0001
22	xIn6	24	80875.454	<.0001
23	cIn2	25	80561.009	<.0001
24	cIn1_Zero	26	80335.362	<.0001
25	xIn5	27	80119.301	<.0001
26	xIn4	28	79921.678	<.0001
27	xIn3	29	79810.099	<.0001
28	xIn2	30	79786.064	<.0001
29	cIn1	31	79774.564	<.0001
30	xIn1	32	79770.630*	0.0001
31	xOut53	33	79771.580	0.0017
32	xOut14	34	79775.166	0.0072
33	xOut51	35	79780.299	0.0171
34	xOut56	36	79785.855	0.0218
35	xOut33	37	79792.191	0.0342
36	cOut2	38	79807.117	0.0346
37	xOut5	39	79813.927	0.0452
38	cOut5	40	79847.492	0.0459

*** Optimal Value of Criterion**

Selected Effects:	Intercept xIn1 xIn2 xIn3 xIn4 xIn5 xIn6 xIn7 xIn8 xIn9 xIn10 xIn11 xIn12 xIn13 xIn14 xIn15 xIn16 xIn17 xIn18 xIn19 xIn20 cIn1 cIn2 cIn3 cIn4 cIn5 Intercept_Zero cIn1_Zero cIn2_Zero cIn3_Zero cIn4_Zero cIn5_Zero

The timing table in Figure 10 shows that the procedure took about 70 seconds to run in single-machine mode.

Figure 10 Timing

Procedure Task Timing		
Task	**Seconds**	**Percent**
Reading and Levelizing Data	0.41	0.60%
Candidate evaluation	27.88	40.92%
Candidate model fit	38.38	56.32%
Final model fit	1.47	2.16%

Performing the analysis by using the full data set of 10 million observations would take several hours in single-machine mode. You can use distributed mode to do the same analysis in far less time. The entire data set of 10 million observations was loaded on a Hadoop server. The following statements read the data from the Hadoop server and perform the computations in distributed mode on a different server that contains 10 server nodes:

```
option set=SAS_HADOOP_JAR_PATH='C:\hadoop\cloudera';
option set=GRIDHOST='bigmath.unx.sas.com';
option set=GRIDINSTALLLOC='/opt/TKGrid';
option set=GRIDMODE='asym';

libname gridlib HADOOP
   server="hpa.sas.com"
   user=XXXXXX
   HDFS_TEMPDIR="temp"
   HDFS_PERMDIR="perm"
   HDFS_METADIR="meta"
   config="demo.xml"
   DBCREATE_TABLE_EXTERNAL=NO;

proc hpgenselect data=gridlib.GLMData;
   class c:;
   model yZIP = x: c: / dist=ZIP;
   zeromodel c:;
   selection method=stepwise(choose=sbc);
   performance details nodes=10;
run;
```

The “Performance Information” table in Figure 11 shows that the analysis was performed in distributed mode by using 10 computing nodes, each with 32 threads. The table also shows that PROC HPGENSELECT ran in asymmetric mode, where the computations are performed in a distributed computing environment that is separate from the database where the data are stored. The “Model Information” table shows the same model as in the previous analysis. The “Number of Observations” table shows that all 10 million observations were used.

Figure 11 Performance Information

Performance Information	
Host Node	bigmath.unx.sas.com
Execution Mode	Distributed
Grid Mode	Asymmetric
Number of Compute Nodes	10
Number of Threads per Node	32

Model Information	
Data Source	GRIDLIB.GLMDATA
Response Variable	yZIP
Class Parameterization	GLM
Distribution	Zero-Inflated Poisson
Link Function	Log
Zero Model Link Function	Logit
Optimization Technique	Newton-Raphson with Ridging

Number of Observations Read	10000000
Number of Observations Used	10000000

The “Selection Summary” table in Figure 12 shows that this analysis included all the variables that are included in the model in the previous analysis. In addition, the variable **xSubtle** is included, reflecting the larger amount of data.

Figure 12 Selection Summary

Selection Summary

Step	Effect Entered	Number Effects In	SBC	p Value
0	Intercept	1		
	Intercept_Zero	2	29524449.3	.
1	cIn5_Zero	3	26540103.3	<.0001
2	cIn4_Zero	4	24211661.5	<.0001
3	cIn3_Zero	5	23058742.7	<.0001
4	xIn20	6	22360427.5	<.0001
5	xIn19	7	21704302.2	<.0001
6	xIn18	8	21092219.0	<.0001
7	xIn17	9	20525192.6	<.0001
8	xIn16	10	20015823.1	<.0001
9	xIn15	11	19555603.2	<.0001
10	cIn2_Zero	12	19106403.6	<.0001
11	xIn14	13	18693647.9	<.0001
12	xIn13	14	18333245.6	<.0001
13	xIn12	15	18021684.7	<.0001
14	xIn11	16	17759850.3	<.0001
15	xIn10	17	17541040.8	<.0001
16	cIn5	18	17347544.1	<.0001
17	cIn4	19	16995861.2	<.0001
18	cIn3	20	16766910.6	<.0001
19	xIn9	21	16580877.7	<.0001
20	xIn8	22	16433007.4	<.0001
21	xIn7	23	16318265.8	<.0001
22	xIn6	24	16234037.8	<.0001
23	cIn2	25	16164935.0	<.0001
24	xIn5	26	16106958.3	<.0001
25	cIn1_Zero	27	16053016.7	<.0001
26	xIn4	28	16015246.4	<.0001
27	xIn3	29	15994179.6	<.0001
28	xIn2	30	15984996.4	<.0001
29	cIn1	31	15978262.2	<.0001
30	xIn1	32	15975845.3	<.0001
31	xSubtle	33	15975836.8*	<.0001
32	xTiny	34	15975843.1	0.0017
33	xOut52	35	15975852.6	0.0104
34	xOut22	36	15975862.8	0.0149
35	xOut28	37	15975873.2	0.0164
36	xOut18	38	15975884.7	0.0328
37	xOut73	39	15975896.3	0.0336
38	cOut1	40	15975908.3	0.0420
39	xOut2	41	15975920.4	0.0450

*** Optimal Value of Criterion**

Selected Effects:	Intercept xIn1 xIn2 xIn3 xIn4 xIn5 xIn6 xIn7 xIn8 xIn9 xIn10 xIn11 xIn12 xIn13 xIn14 xIn15 xIn16 xIn17 xIn18 xIn19 xIn20 xSubtle cIn1 cIn2 cIn3 cIn4 cIn5 Intercept_Zero cIn1_Zero cIn2_Zero cIn3_Zero cIn4_Zero cIn5_Zero

Parameter estimates for the selected model are shown in Figure 13 and Figure 14.

Figure 13 Parameter Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept	1	-0.319361	0.003691	7485.3170	<.0001
xln1	1	-0.050986	0.001034	2433.1003	<.0001
xln2	1	0.099071	0.001033	9199.5521	<.0001
xln3	1	-0.150147	0.001034	21087.3993	<.0001
xln4	1	0.201001	0.001035	37748.7598	<.0001
xln5	1	-0.249349	0.001035	58090.6836	<.0001
xln6	1	0.300777	0.001035	84458.6925	<.0001
xln7	1	-0.351769	0.001036	115273.105	<.0001
xln8	1	0.400321	0.001037	149022.440	<.0001
xln9	1	-0.449548	0.001038	187488.110	<.0001
.					
.					
.					
cln5 2	1	0.747241	0.002696	76825.5556	<.0001
cln5 3	1	0.497155	0.002724	33308.6157	<.0001
cln5 4	1	0.246343	0.002859	7424.3835	<.0001
cln5 5	0	0	.	.	.

Figure 14 Zero-Inflation Parameter Estimates

Zero-Inflation Parameter Estimates

Parameter	DF	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept_Zero	1	6.527324	0.012450	274868.944	<.0001
cln1_Zero 1	1	-1.003686	0.004723	45164.8001	<.0001
cln1_Zero 2	0	0	.	.	.
cln2_Zero 1	1	4.011342	0.007866	260070.923	<.0001
cln2_Zero 2	1	1.998211	0.006136	106040.112	<.0001
cln2_Zero 3	0	0	.	.	.
cln3_Zero 1	1	-9.024949	0.014108	409226.351	<.0001
cln3_Zero 2	1	-6.014037	0.010579	323163.031	<.0001
cln3_Zero 3	1	-3.017093	0.007793	149886.519	<.0001
cln3_Zero 4	0	0	.	.	.
.					
.					
.					
cln5_Zero 3	1	-10.037194	0.016290	379661.851	<.0001
cln5_Zero 4	1	-5.022051	0.011111	204286.226	<.0001
cln5_Zero 5	0	0	.	.	.

The timing table in Figure 15 shows that PROC HPGENSELECT took slightly more than five minutes to run, most of the time spent in model evaluation and fitting.

Figure 15 Timing

Procedure Task Timing

Task	Seconds	Percent
Distributing Data	2.03	0.65%
Reading and Levelizing Data	112.12	35.87%
Candidate evaluation	71.30	22.81%
Candidate model fit	123.81	39.60%
Final model fit	3.35	1.07%

Model Selection by the LASSO Method

This example shows how you can use the HPGENSELECT procedure in single-machine mode to perform model selection by using the LASSO method. For more information about the LASSO method see, for example, Hastie, Tibshirani, and Friedman (2009).

The following statements use the first 50,000 observations from the data set GLMData to perform model selection by the LASSO method. Here, the Poisson response variable **yPoisson** depends on the regression variables in Table 2. The SELECTION statement specifies that the LASSO method be used and that the final model be selected on the basis of the minimum SBC criterion.

```
libname Data 'C:\Data';
proc hpgenselect data=Data.GLMData(obs=50000);
   class c:;
   model yPoisson = x: c: / dist=Poisson;
   selection method=lasso(choose=sbc) details=all;
run;
```

PROC HPGENSELECT uses a group LASSO method so that all parameters that are associated with levels of CLASS effects are included or excluded together.

Model selection is performed by varying a regularization parameter, which controls the amount of shrinkage in the regression coefficients. Those coefficients that are shrunk to zero are deemed to be out of the model, and coefficients that are not zero are in the model. The “Selection Details” table in Figure 16 shows the sequence of steps in the model selection process. Each successive step corresponds to a smaller regularization parameter (named **Lambda** in Figure 16), which corresponds to less regression coefficient shrinkage. Unlike the stepwise method, the LASSO method includes zero or more effects in each step.

Figure 16 LASSO Method Selection Summary

Selection Details						
Step	Description	Effects In Model	Lambda	AIC	AICC	BIC
0	Initial Model	1	1	215879.209	215879.209	215888.029
1	xIn17 entered	5	0.8	209124.963	209124.964	209169.062
	xIn18 entered	5	0.8	209124.963	209124.964	209169.062
	xIn19 entered	5	0.8	209124.963	209124.964	209169.062
	xIn20 entered	5	0.8	209124.963	209124.964	209169.062
2	xIn13 entered	10	0.64	196159.045	196159.053	196273.703
	xIn14 entered	10	0.64	196159.045	196159.053	196273.703
	xIn15 entered	10	0.64	196159.045	196159.053	196273.703
	xIn16 entered	10	0.64	196159.045	196159.053	196273.703
	cIn5 entered	10	0.64	196159.045	196159.053	196273.703
3	xIn11 entered	13	0.512	184409.836	184409.851	184577.412
	xIn12 entered	13	0.512	184409.836	184409.851	184577.412
	cIn4 entered	13	0.512	184409.836	184409.851	184577.412
4	xIn8 entered	17	0.4096	172994.506	172994.532	173215.001
	xIn9 entered	17	0.4096	172994.506	172994.532	173215.001
	xIn10 entered	17	0.4096	172994.506	172994.532	173215.001
	cIn3 entered	17	0.4096	172994.506	172994.532	173215.001
5	xIn7 entered	18	0.3277	163951.437	163951.465	164180.751
6	xIn6 entered	20	0.2621	157281.319	157281.354	157537.092
	cIn2 entered	20	0.2621	157281.319	157281.354	157537.092
7	xIn4 entered	22	0.2097	152456.156	152456.196	152729.569
	xIn5 entered	22	0.2097	152456.156	152456.196	152729.569
8		22	0.1678	148941.187	148941.227	149214.600
9	xIn3 entered	24	0.1342	146514.918	146514.963	146805.970
	cIn1 entered	24	0.1342	146514.918	146514.963	146805.970
10		24	0.1074	144859.219	144859.264	145150.272
11	xIn2 entered	25	0.0859	143758.891	143758.939	144058.764
12		25	0.0687	143012.513	143012.561	143312.386
13		25	0.055	142522.998	142523.045	142822.870
14	xIn1 entered	26	0.044	142204.818	142204.869	142513.510
15		26	0.0352	141992.171	141992.221	142300.863
16	xOut53 entered	29	0.0281	141855.391	141855.450	142190.542
	xOut56 entered	29	0.0281	141855.391	141855.450	142190.542
	xOut60 entered	29	0.0281	141855.391	141855.450	142190.542
17	xOut14 entered	33	0.0225	141768.812	141768.892	142156.883
	xOut72 entered	33	0.0225	141768.812	141768.892	142156.883
	xOut77 entered	33	0.0225	141768.812	141768.892	142156.883
	cOut3 entered	33	0.0225	141768.812	141768.892	142156.883
18	xOut5 entered	36	0.018	141704.761	141704.851	142119.291*
	xOut20 entered	36	0.018	141704.761	141704.851	142119.291*
	xOut44 entered	36	0.018	141704.761	141704.851	142119.291*
19	xOut6 entered	45	0.0144	141680.110	141680.252	142200.477
	xOut10 entered	45	0.0144	141680.110	141680.252	142200.477
	xOut16 entered	45	0.0144	141680.110	141680.252	142200.477
	xOut33 entered	45	0.0144	141680.110	141680.252	142200.477
	xOut51 entered	45	0.0144	141680.110	141680.252	142200.477
	xOut52 entered	45	0.0144	141680.110	141680.252	142200.477

Figure 16 *continued*

Selection Details						
Step	Description	Effects In Model	Lambda	AIC	AICC	BIC
	xOut78 entered	45	0.0144	141680.110	141680.252	142200.477
	cOut1 entered	45	0.0144	141680.110	141680.252	142200.477
	cOut4 entered	45	0.0144	141680.110	141680.252	142200.477
20	xOut3 entered	59	0.0115	141674.422	141674.656	142344.725
	xOut7 entered	59	0.0115	141674.422	141674.656	142344.725
	xOut8 entered	59	0.0115	141674.422	141674.656	142344.725
	xOut11 entered	59	0.0115	141674.422	141674.656	142344.725
	xOut35 entered	59	0.0115	141674.422	141674.656	142344.725
	xOut42 entered	59	0.0115	141674.422	141674.656	142344.725
	xOut54 entered	59	0.0115	141674.422	141674.656	142344.725
	xOut57 entered	59	0.0115	141674.422	141674.656	142344.725
	xOut64 entered	59	0.0115	141674.422	141674.656	142344.725
	xOut66 entered	59	0.0115	141674.422	141674.656	142344.725
	xOut70 entered	59	0.0115	141674.422	141674.656	142344.725
	xOut73 entered	59	0.0115	141674.422	141674.656	142344.725
	xOut74 entered	59	0.0115	141674.422	141674.656	142344.725
	cOut5 entered	59	0.0115	141674.422	141674.656	142344.725

*** Optimal Value of Criterion**

The model in step 18 was selected on the basis of the minimum SBC, and the effects that are selected are shown in Figure 17. The variables **xIn1–xIn20** and **cIn1–cIn5** were included in the model, and a few of the noise variables were also selected.

Figure 17 Effects Selected by the LASSO Method

Selected Effects:	Intercept xIn1 xIn2 xIn3 xIn4 xIn5 xIn6 xIn7 xIn8 xIn9 xIn10 xIn11 xIn12 xIn13 xIn14 xIn15 xIn16 xIn17 xIn18 xIn19 xIn20 xOut5 xOut14 xOut20 xOut44 xOut53 xOut56 xOut60 xOut72 xOut77 cIn1 cIn2 cIn3 cIn4 cIn5 cOut3

Parameter estimates for the selected model are shown in Figure 18.

Figure 18 Parameter Estimates

Parameter Estimates		
Parameter	**DF**	**Estimate**
Intercept	1	-0.306403
xIn1	1	-0.028968
xIn2	1	0.071474
xIn3	1	-0.132928
xIn4	1	0.194972
xIn5	1	-0.207019
xIn6	1	0.289436
xIn7	1	-0.338620
xIn8	1	0.393349
xIn9	1	-0.434013
.		
.		
.		
cIn3 3	1	0.124542
cIn3 4	0	0
cIn4 1	1	-0.767686
cIn4 2	1	-0.573186
cIn4 3	1	-0.389338
cIn4 4	1	-0.194644
cIn4 5	0	0
cIn5 1	1	0.945248
cIn5 2	1	0.707911
cIn5 3	1	0.445848
cIn5 4	1	0.207569
cIn5 5	0	0
cOut3 1	1	0.002422
cOut3 2	1	-0.001612
cOut3 3	1	-0.002164
cOut3 4	0	0

Distributed Mode

For more information about distributed mode, see Cohen and Rodriguez (2013) and *SAS/STAT 13.1 User's Guide: High-Performance Procedures*, at `http://support.sas.com/documentation/onlinedoc/stat/`. For more information about the LIBNAME statement, see *SAS/ACCESS 9.4 for Relational Databases: Reference, Third Edition*, at `http://support.sas.com/documentation/onlinedoc/access/`.

Summary of Benefits

The HPGENSELECT procedure, added in SAS/STAT 12.3, provides the following:

- model selection and model fitting for standard generalized linear model distributions and link functions
- a growing number of model selection techniques, now including the LASSO method
- zero-inflated models, ordinal and nominal multinomial models, and the Tweedie model
- predictive modeling for large data problems in a distributed computing environment
- the ability to use all available CPUs in single-machine mode on the server where SAS/STAT is installed

References

Cohen, R., and Rodriguez, R. N. (2013). "High-Performance Statistical Modeling." In *Proceedings of the SAS Global Forum 2013 Conference*. Cary, NC: SAS Institute Inc. http://support.sas.com/resources/papers/proceedings13/401-2013.pdf.

Hastie, T. J., Tibshirani, R. J., and Friedman, J. H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. New York: Springer-Verlag.

Contact Information

Your comments and questions are valued and encouraged. Contact the authors:

Gordon Johnston
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
Gordon.Johnston@sas.com

Robert N. Rodriguez
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
Bob.Rodriguez@sas.com