Umstieg von PROC GENMOD auf PROG HPGENSELECT: Scoren neuer Daten leicht gemacht

Dr. Olaf Kruse
VST Gesellschaft für Versicherungsstatistik mbH
Roscherstr. 10
30161 Hannover
olaf.kruse@vst-gmbh.de

Zusammenfassung

Für die Modellierung von Generalisierten Linearen Modellen wurde in SAS/STAT mit PROC HPGENSELECT eine performante Alternative zur bekannten PROC GENMOD eingeführt. Die unterschiedlichen Ausrichtungen werden mit Schwerpunkt auf die Optionen zum Scoren neuer Daten herausgearbeitet.

Schlüsselwörter: PROC GENMOD, PROC HPGENSELECT, SAS/STAT, Generalisierte Lineare Modelle, Scoring, High Performance Analytics Prozedur

1 Einleitung

Die Familie der Generalisierten Linearen Modelle (GLMs) wurde erstmalig von Nelder & Wedderburn [1972] zusammenhängend dargestellt. Wie der Namen andeutet, stellen GLMs eine Erweiterung des klassischen linearen Modells dar. Viele bekannte Modelle, wie das Logit-, Probit- oder loglineare Modell, gehören zu der Familie der GLMs. PROC GENMOD bietet seit SAS 6.09 einen umfassenden Zugang zu dieser Modellfamilie.

Mit der SAS/STAT Version 13.2 bzw. SAS 9.4M2 wurde mit PROC HPGENSELECT im Rahmen der Einführung High Performance Analytics Prozeduren eine hoch performante Alternative zu PROC GENMOD vorgestellt. HPA-Prozeduren sind optimiert für Multi-Threading und verteiltes Rechnen, was bereits bei „normalen“ Arbeitsplatz-Rechnern zu einer verbesserten Ausnutzung vorhandener Prozessorkerne führt (Abb.1).

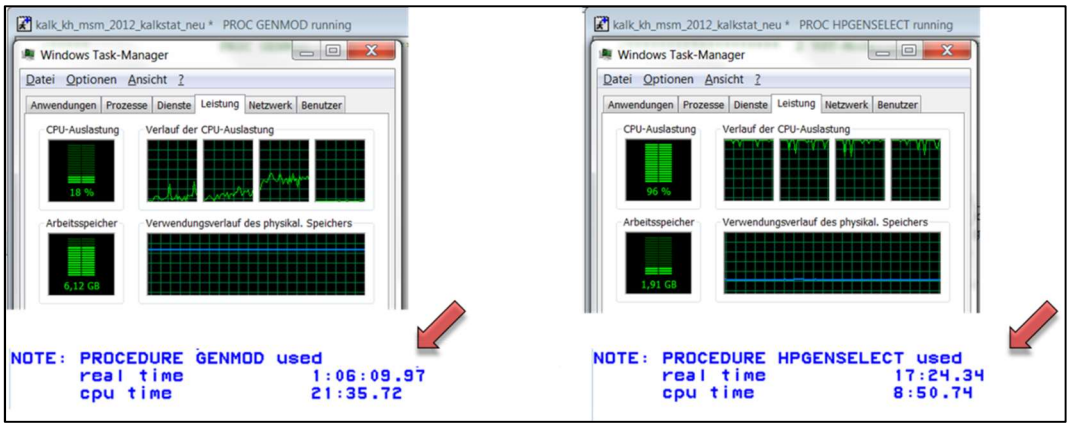

Abbildung 1: Performance-Vergleich PROC GENMOD und PROC HPGENSELECT

2 Prozedurvergleich

Im Kern bieten beide Prozeduren die gleiche Funktionalität und sind auch von der Syntax weitgehend deckungsgleich. Während PROC GENMOD als „klassische“ Statistik-Prozedur mehr Features im Bereich der Modelldiagnose (Contrast Estimates, Residuen-Analyse) aufweist, hat PROC HPGENSELECT als „moderne“ Big-Data-Prozedur seine Stärken neben der Performance z.B. bei Optionen für die automatische Variablenselektion (u.a. Lasso-Verfahren).

```
/*****************************************/
/**  Basis-Code GENMOD - HPGENSELECT  **/
/*************************   ************/

proc   <genmod> <hpgenselect>     data=mydata;
   class SFR REGIO KW ;
   model SCHADEN =  SFR REGIO KW /
         dist    = poisson
         link    = log
         offset = offvar;
   weight JE;
   output out=scoredata  xbeta=xb  predicted=pred;
   ods output  parameterestimates = parest;
   code file = 'scorecode.sas';
run;
```

Beispiel 1: Codevergleich GENMOD – HPGENSELECT

Der Wechsel zwischen beiden Prozeduren ist relativ problemlos, da der Basis-Code weitgehend identisch ist. Wie aus Beispiel 1 zu ersehen ist, „genügt“ bei einem Code mit vielen Standard-Optionen lediglich der Austausch des Prozedurnamens.

3 Scoring

Auf die in PROC GENMOD und PROC HPGENSELECT implementierten Features zum Scoren von Daten wird im Weiteren eingegangen. Ein zusätzliches Augenmerk liegt auf Besonderheiten bei der Berücksichtigung von a-priori-Informationen (Offset-Option) bei der Modellbildung bzw. beim Scoren.

Unter „Scoren“ wird hierbei die Berechnung des Schätzwertes bzw. Erwartungswertes für Beobachtungen, die nicht zur Modellbildung verwendet wurden, bzw. für ganz neue Beobachtungen verstanden.

Prinzipiell können die Modellparameter ausgelesen und in ein entsprechendes „Scoring-Modell“, z.B. in einem Data-Step, überführt werden. Zusätzlich müssen die Modellstruktur und alle Features (Verteilungs- und Link-Funktion des Modells, berücksichtigte Offsets etc.) manuell eingepflegt werden.

Als quasi universelle und implizite Scoring-Methode können die neu zu scorenden Datensätze auch in den Analysedatensatz mit einem Beobachtungsgewicht von „0“ eingebunden werden. Die Datensätze zur Modellberechnung behalten ihr ursprüngliches Gewicht bzw. werden mit einem Gewicht von „1“ berücksichtigt.

Über das Beobachtungsgewicht wird einerseits verhindert, dass diese nur zu scorenden Datensätze in die Modellbildung einfließen. Andererseits werden die durch das Modell vorhergesagten Werte auch für diese Datensätze mit berechnet. Z.B. über die Option

```
output out=<Datei Name> PREDICTED;
```

kann der gesamte Datensatz, d.h. auch mit den nur zu scorenden Daten, inklusive erklärenden Variablen und vorhergesagter Werte ausgegeben werden. Die zugrunde liegende Modellstruktur mit Offsets etc. wird hierbei automatisch berücksichtigt

4 Scoring und PROC GENMOD

PROC GENMOD kennt mehrere implizite Optionen zum Scoren neuer bzw. nicht zur Modellbildung verwendeter Daten, von denen auf das „estimate“- und „store“-Statement (Bsp. 2) detailliert eingegangen wird.

Über das „estimate“-Statement können einzelne Datensätze gescort werden, indem die Werte der erklärenden Variablen bzw. des entsprechenden Designvektors übergeben werden. Offsets etc. können nicht direkt mit übergeben werden, sondern müssen nachträglich manuell eingepflegt werden.

```
/*************************************/
/**      Scoring mit PROC GENMOD     **/
/*************************************/

proc genmod  data=mydata;
  class SFR REGIO KW ;
  model SCHADEN = SFR REGIO KW /
        dist = poisson  link = log;
  weight JE;

  estimate 'Neue Obs' int 1  SFR 0 1  REGIO 1  KW 0 1;
  store  work.mymodel;
run;
```

Beispiel 2: Scoring mit PROC GENMOD

Hierbei wird implizit die Funktionalität zur Konstruktion von Hypothesentests ausgenutzt. Mit einem Prozeduraufruf können mehrere verschiedene „estimate“-Statements angesetzt werden, deren Resultate z.B. über folgendes ods-Statement

```
ods output estimates =<Datei Name>;
```

in einer Output-Datei abgespeichert werden können. Diese Option ist dennoch nicht für Massendatenverarbeitung, sondern eher für die Berechnung einzelner Werte für weiterführende Analysen gedacht.

```
/*****************************************/
/** PROC PLM: Modellinformationen abrufen **/
/*****************************************/

proc plm restore = work.mymodel;
   show all;
run;
```

Beispiel 3: Modellinformationen abrufen mit PROC PLM

Interessant ist das „store“-Statement, über das alle relevanten Modellinformationen in eine Binär-Datei ausgegeben werden können. Über PROC PLM kann diese Binärdatei wieder ausgelesen und z.B. zum Scoren neuer Daten verwendet werden.

```
                The PLM Procedure

                Store Information

Item Store                  WORK.MYMODEL
Data Set Created From       WORK.MYDATA
Created By                  PROC Genmod
Date Created                23FEB20:00:02:09
Response Variable           schaden
Weight Variable             JE
Link Function               Log
Distribution                Poisson
Class Variables             SFR KW REGIO
Model Effects               Intercept SFR KW REGIO

                    [......]
```

Beispiel 4: Modellinformationen Output - PROC PLM

Mit der Option „show all“ (Bsp. 3) können die gespeicherten Job-Informationen und Modell-Ergebnisse, die dem Standard-Output der erstellenden Prozedur entsprechen, angezeigt werden (Bsp. 4).

```
/*************************************/
/**  Scoren einzelner Beobachtungen  **/
/*************************************/

proc plm restore = work.mymodel;
 estimate 'Datensatz 1' int 1 SFR 0 1  REGIO 1  KW 0 1;
 estimate 'Datensatz 2' int 1 SFR 1  REGIO 0 1 KW 1;

   ods output  estimates = estimates_out;
run;
```

Beispiel 5: Scoren einzelner Beobachtungen

Über das bekannte „estimate“-Statement können wiederum einzelne Beobachtungen gescort werden (Bsp. 5). Mit dem „score“-Statement (Bsp. 6) können ganze Datensätze gescort werden, wobei, im Gegensatz zum „Estimate“-Statement, verwendete Offsets implizit berücksichtigt werden.

Prinzipiell können so alle Post-Modell-Analysen, wie z.B. Residuen-Plots, für beliebige Datensätze in die PROC PLM „ausgelagert“ werden. Vorteilhaft ist neben der Flexibilität und Portabilität auch die hohe Revisionssicherheit.

```
/*****************************************/
/**  Scoren eines ganzen Datensatzes  **/
/*****************************************/

proc plm restore = work.mymodel;
   score data=mydata   out = mydata_scored;
run;
```

Beispiel 6: Scoren eines ganzen Datensatzes

Das „store“-Statement ist zwar in vielen Statistikprozeduren implementiert, wurde aber nicht bei PROC HPGENSELECT berücksichtigt. Gleiches trifft auf das „estimate“-Statement zu.

5 Scoring und PROC HPGENSELECT

PROC HPGENSELECT kennt hingegen andere implizite Optionen zum Scoren neuer Daten, von denen auf das „code“- und „partition“-Statement (Bsp. 7) detailliert eingegangen wird.

```
/*************************************/
/**    Scoring mit PROC HPGENSELECT  **/
/*************************************/

proc hpgenselect  data=mydata;
  class SFR REGIO KW ;
  model SCHADEN = SFR REGIO KW /
        dist = poisson  link = log;
  weight JE;
  output out=mydata_scored

  code file = 'mycode.sas';
  partition fraction (test=0.4 train=0.4 validate=0.2);
run;
```

Beispiel 7: Scoring mit PROC HPGENSELECT

Quasi als Ersatz für das fehlende „store“-Statement wurde in PROC HPGENSELECT mit dem „code“-Statement eine vergleichbare Option für das Auslesen von Modellergebnissen zum Scoren neuer Daten implementiert.

Über das „code“-Statement kann der komplette Scoring-Algorithmus inklusive aller Parameter, Offsets etc. in einer Datei abgelegt werden. Diese Code-Datei (Auszüge in Bsp. 9) ist ein „unselbstständiger“ Text-Sniplet, der z.B. einem Data-Step zum Scoren neuer Daten eingebunden werden kann:

```
/*****************************************/
/**     Scoren  eines  Datensatzes     **/
/*****************************************/

data gescorte_daten;
   set zu_scorende_daten;
   %include 'c:\pfad\<Dateiname>.sas';
run;
```

Beispiel 8: Scoring eines ganzen Datensatzes

Beispiel 9 zeigt zentrale Auszüge des Score-Codes. Im Prinzip werden in einem ersten Schritt notwendige Variablen, wie z.B. der Design-Vektor bzw. die notwendigen binären Variablen für die CLASS-Variablen, angelegt. In einem zweiten Schritt werden diesen Variablen die Parameter aus der Modellschätzung zugeordnet und in einem dritten Schritt die Score-Werte und ggf. Residuen bzw. weitere Informationen berechnet.

```
*****************************;
* Scoring-Code  HPGENSELECT *;
*****************************;

label P_SB = 'Predicted: SB' ;

* Design variables for SFR;
_st2=left(trim(put(SFR,$2.)));
   if _st2 = 'S1'  then do;
      _1_0 = 1;
   end;
   else if _st2 = 'S2' then do;
      _1_1 = 1;
   end;
   else if _st2 = 'S3' then do;
      _1_2 = 1;
   end;
   else if _st2 = 'S4' then do;
      _1_3 = 1;
   end;
            [......]
```

```
*** Compute Linear Predictors;
*** Effect: SFR;
   _LP0 = _LP0 + (0.2230   * _0_0;
   _LP0 = _LP0 + (0.1319)  * _0_1;
   _LP0 = _LP0 + (-0.1404) * _0_2;
   _LP0 = _LP0 + (-0.0168) * _0_3;

               [......]

*** Predicted values;
_LP0 = _LP0 + 5.17142;
_LP0 = exp(_LP0);
_SKIP_000:

               [......]

E_SB=2*(_Y*log(_Y/_LP0)+_LP0-_Y);
end;
```

Beispiel 9: Auszüge aus dem Scoring-Code

Über verschiedene Optionen der Code-Anweisung lassen sich die Struktur des Codes (z.B. Auswahl über if-then-else-Verschachtelungen oder ein „select“-Statement) und der Umfang des zu generierenden Scoring-Outputs steuern.

Vorteilhaft ist neben der Portabilität auch das einfache Handling. Anpassungen an den Scoring-Algorithmus, die sich z.B. auf Variablennamen oder einzelne Parameterwerte beziehen, können relativ einfach vorgenommen und dokumentiert werden. Dieses Statement ist auch in vielen anderen Statistik-Prozeduren implementiert.

Den „modernen“ datengetriebenen Inferenz-Überlegungen angelehnt, ist das „partition“Statement (Bsp. 10). Die Grundidee ist, den Analysedatensatz in Trainingsdaten zur Modellierung und Validate- bzw. Testdaten zur Modellüberprüfung aufzuteilen.

```
/*************************************/
/**   Zufallsgesteuerte Partition    **/
/*************************************/
proc hpgenselect  data=mydata;
partition fraction(train=0.4 test=0.4 validate=0.2 seed=42);
           [......]
run;

/*************************************/
/**      Kontrollierte Partition     **/
/*************************************/
Proc hpgenselect  data=mydata;
partition role = KW(train='K1' test='K2' validate='K3');
           [......]
run;
```

Beispiel 10: “Partition”-Statement

Diese Aufteilung kann entweder „zufällig“ erfolgen, wobei mit der „fraction“-Option die Stichprobengröße der drei Partitionen festgelegt wird. Über die „seed“-Option kann der Startwert des Zufallszahlengenerator gesteuert werden, um z.B. identische Aufteilungen reproduzieren zum können.

Bei der „kontrollierten“ Partition wird die Aufteilung mit der „role“-Option über eine vorzugebende Partitionsvariable gesteuert. Bei beiden Optionen wird im Output-Datensatz eine „Role“-Variable generiert, die die entsprechende Verwendung jeder Beobachtung indiziert.

The HPGENSELECT Procedure

Fit Statistics

	Training	Validation	Testing
-2 Log Likelihood	82939	91233	99527
AIC (smaller is better)	82959	91253	99547
AICC (smaller is better)	82964	91258	99551
BIC (smaller is better)	82980	91273	99567
Pearson Chi-Square	16885	18574	20262
Pearson Chi-Square/DF	351.78	386.96	422.13
Average Square Error	526.23	526.23	526.23

Beispiel 11: Output mit “Partition”-Statement

Die Modellschätzung erfolgt nur mit der Trainings-Partition. Die Berechnung der Fit-Statistiken und damit implizit auch der Score-Werte erfolgt hingegen für alle drei Partitionen (Bsp. 11). Bei vergleichbarer Aufteilung wird die Modellanpassung i.d.R. für die Trainings-Partition besser sein als für die beiden anderen Partitionen. Über das „Out-

put“- bzw. „id“-Statement kann der Inhalt der die Output-Datei mit den gescorten Datensätzen gesteuert werden.

6 Schlussbemerkung

Auch wenn PROC HPGENSELECT nicht in allen Bereichen mehr als PROC GENMOD zu bieten hat, ist als Vorteil vor allem die deutliche Verbesserung der Performance bzw. Rechenzeit zu nennen. Zudem bieten einige Features zum Scoren interessante Alternativen zu den entsprechenden Optionen von PROC GENMOD.

Literaturverzeichnis

[1] L.A. Baxter, S.M. Coutts & G.A.F. Ross, (1980): Applications of Linear Models in Motor Insurance, Transactions of the 21st International Congress of Actuaries, S. 11-29

[2] O. Kruse (1997): Modelle zur Analyse und Prognose des Schadenbedarfs in der Kfz-Haftpflichtversicherung, Karlsruhe: VVW

[3] O. Kruse (2004): Verborgene Schätze heben: Die Offset-Option in PROC GENMOD, in: R. Muche (Hrsg.) Proceedings der 8. KSFE, Aachen: Shaker

[4] SAS Institute (2015): SAS/STAT 14.1 User's Guide, Cary: SAS Institute