Modellauswahl in Generalisierten Linearen Modellen PROC GENMOD effizient nutzen

Dr. Olaf Kruse

VST Gesellschaft für Versicherungsstatistik mbH Hannover
30161 Hannover
Tel. 0511 – 339 599-21
E-Mail: Olaf.Kruse@vst-gmbh.de

Abstract

Ein Generalisiertes Lineares Modell wird durch den Linearen Prädiktor, die Link-Funktion und die Varianz-Funktion definiert. Die Modellauswahl, d.h. die inhaltliche Ausgestaltung dieser drei Elemente, kann auf theoretischen Überlegungen oder empirischen Erkenntnissen basieren. Auf Fragen der empirischen Modellauswahl mit PROC GENMOD wird im Folgenden eingegangen.

1. Einleitung

Die Familie der Generalisierten Linearen Modelle (GLMs) wurde erstmalig von Nelder & Wedderburn [1972] zusammenhängend dargestellt. Wie der Name andeutet, stellen GLMs eine Erweiterung des klassischen linearen Modells dar. Viele weitere bekannte Modelle, wie das Logit-, Probit- oder loglineare Modell, gehören zu der Familie der GLMs.

Mit vielen SAS-Prozeduren (u.a. REG, GLM, LOGISTIC, PROBIT) können einzelne Modelle aus der Familie der GLMs modelliert werden. PROC GENMOD eignet sich nicht nur zur Modellierung einzelner GLMs, sondern vor allem zur Modellauswahl bzw. zum Modellvergleich. Nach einer kurzen Einführung der theoretischen Grundlagen der GLMs wird auf einige spezielle Optionen von PROC GENMOD zur Modellauswahl eingegangen.

Was ist ein GLM

Das klassische lineare Modell als der bekannteste Vertreter der Familie der GLMs ist durch

$$y_i = \mathbf{x}_i' \boldsymbol{\beta} + e_i \quad \text{mit} \quad e_i \sim N(0, \sigma_i^2)$$

gegeben. Hierbei bezeichnen y_i die Responsevariable bzw. $\mathbf{x}_i$ den Vektor der nichtstochastischen erklärenden Variablen für die i-te der insgesamt T Beobachtungen und $\boldsymbol{\beta}$ den zu schätzenden Parametervektor mit insgesamt P Elementen. Weiterhin wird angenommen, dass die unbekannten Störterme e_i und damit auch die Responsevariablen y_i unabhängig normalverteilt sind.
Die Responsevariable y_i kann also in eine systematische Komponente, die durch ihren Erwartungswert μ_i gegeben ist, und einen Störterm e_i untergliedert werden. Die systematische Komponente μ_i wird durch eine (in diesem Fall sehr einfache, d.h. lineare) Funktion des linearen Prädiktors $\mathbf{x}_i' \boldsymbol{\beta}$ beschrieben:

$$y_i = \mu_i + e_i \quad \text{mit} \quad \mu_i = \mathbf{x}_i' \boldsymbol{\beta} \quad \text{und} \quad y_i \sim N(\mu_i, \sigma_i^2).$$

Mit dieser Darstellung sind die wichtigsten Eigenschaften eines GLMs, so wie sie sich im klassischen linearen Modell ausprägen, angesprochen. Allgemein wird ein GLM durch folgende drei Elemente charakterisiert:

- **Der lineare Prädiktor η als Linearkombination der erklärenden Variablen ist analog dem klassischen linearen Modell gegeben durch:**

  $$\eta_i = \mathbf{x}_i'\boldsymbol{\beta}$$

- Eine bijektive, zweimal stetig differenzierbare Link-Funktion *g* beschreibt die Verbindung zwischen dem Erwartungswert μ der Responsevariablen und dem linearen Prädiktor η:

  $$g(\mu_i) = \mathbf{x}_i'\boldsymbol{\beta} \quad bzw. \quad \mu_i = g^{-1}(\mathbf{x}_i'\boldsymbol{\beta}) \qquad (1)$$

- Die Dichte- bzw. Wahrscheinlichkeitsfunktion $f(y;\theta,\phi)$ der Responsevariablen *y* gehört zur Familie der einparametrigen natürlichen Exponentialverteilungen. Die für die Schätzung von $\boldsymbol{\beta}$ wesentliche Eigenschaft der Verteilung ist neben dem Erwartungswert μ die Varianzfunktion $V(\mu)$, die den Einfluss des Erwartungswerts auf die Varianz der Responsevariablen beschreibt:

  $$E(y_i) = \mu_i \quad \text{und} \quad Var(y_i) = \frac{\phi}{w_i} V(\mu_i) \qquad (2)$$

  Der Dispersionsparameter ϕ ist konstant über alle Beobachtungen. Er ist z.B. bei der Poisson-Verteilung mit ϕ = 1 a-priori bekannt oder muss z.B. im klassischen linearen Modell geschätzt werden. Über die Beobachtungsgewichte *w* kann der Einfluss der Beobachtungen auf das Modell individuell bestimmt werden.

Die Familie der Exponentialverteilungen umfasst sowohl stetige Verteilungen, wie die Normal- und Gammaverteilung, als auch diskrete Verteilungen, wie die Binomial- und Poissonverteilung. Die Wahl einer geeigneten Verteilung ergibt sich oft implizit durch theoretische Modellüberlegungen oder die Struktur des Datenmaterials.

Für jede dieser Verteilungen existiert eine sog. kanonische Linkfunktion, die vorteilhafte statistische Eigenschaften aufweist und die Modellschätzung vereinfacht. Die Linkfunktion sollte auch den unbeschränkten Wertebereich des linearen Prädiktors auf den ggf. beschränkten Wertebereich (wie z.B. bei der Binomialverteilung) der Responsevariablen abbilden. Zudem sollte die Modellanpassung bei der Auswahl der Link-Funktion berücksichtigt werden.

Verteilung	$V(\mu)$	KANONISCHE LINKFUNKTION
Normal	1	$\eta = \mu$
Poisson	μ	$\eta = \log(\mu)$
Gamma	μ^2	$\eta = \mu^{-1}$
Invers-Normal	μ^3	$\eta = \mu^{-2}$
Binomial	$\mu(1-\mu)$	$\eta = \log\left[\mu(1-\mu)^{-1}\right]$

Tabelle 1: Varianzfunktion und kanonische Linkfunktion wichtiger GLM

In Tabelle 1 sind die wichtigsten Verteilungen aus der Familie der Exponentialverteilungen mit ihren Varianzfunktionen und kanonischen Linkfunktionen zusammengefasst.

Devianz und verallgemeinerte X²-Statistik

Jedes GLM definiert eine Log-Likelihood-Funktion $l(\boldsymbol{\mu}; \mathbf{y}, \phi)$, die in Abhängigkeit von $\boldsymbol{\beta}$ maximiert wird. Die nummerische Lösung der Maximum-Likelihood-Schätzung kann mit einer iterativen Kleinste-Quadrate-Schätzung, wie z.B. einem Newton-Rhapson-Algorithmus, erfolgen. Aus der Schätzung leiten sich mit der Devianz $D(\mu; y)$ und der verallgemeinerten X²-Statistik zwei für die Modellbeurteilung zentrale Statistiken ab. Die Devianz

$$D(\mu; y) = 2\phi\,(l(y; y) - l(\mu; y)) \qquad (3)$$

ist eine Funktion der Differenz zwischen der modellunabhängigen maximalen Log-Likelihood und der Log-Likelihood des betrachteten Modells. Der ML-Schätzer für den Parametervektor $\boldsymbol{\beta}$ ist gleichzeitig der Devianz-minimierende Schätzer. Jede Varianzfunktion $V(\mu)$ definiert genau eine Devianz-Funktion. Die verallgemeinerte X²-Statistik

$$\chi^2(\mu; y) = \sum_{i=1}^{T} w_i \frac{(y_i - \mu_i)^2}{V(\mu_i)}$$

ist eine quadratische Approximation der Devianz. Aus der verallgemeinerten X²-Statistik wird zudem ersichtlich, dass in die Parameterschätzung lediglich die durch (2) gegebene Erwartungswert- und Varianzbeziehung der Exponentialverteilungen eingehen. Weiterhin wird deutlich, dass die Varianzfunktion $V(\mu)$ das Skalenniveau beider Statistiken bestimmt.

Verteilung	$D(\mu; y)$
Normal	$\Sigma w_i (y_i - \mu_i)^2$
Poisson	$2\Sigma w_i (y_i \log(y_i / \mu_i) - (y_i - \mu_i))$
Gamma	$2\Sigma w_i ((y_i - \mu_i)/\mu_i - \log(y_i / \mu_i))$
Invers-Normal	$\Sigma w_i ((y_i - \mu_i)^2 / (y_i \mu_i^2))$

Tabelle 2: Devianz-Funktionen

Die skalierte Devianz bzw. skalierte verallgemeinerte X²-Statistik ergibt sich durch Division der jeweiligen unskalierten Statistiken durch den Dispersionsparameter:

$$D^*(\mu; y) = \frac{D(\mu; y)}{\phi} \quad \text{bzw.} \quad \chi^{2*}(\mu; y) = \frac{\chi^2(\mu; y)}{\phi} \qquad (4\text{ a,b})$$

Beide skalierte Statistiken folgen unter gewissen Regularitätsannahmen asymptotisch einer X²-Verteilung mit I-P Freiheitsgraden. Aus diesen Zusammenhängen leiten sich zwei konsistente Schätzer für den Dispersionsparameter ϕ ab:

$$\phi = \frac{D(\mu; y)}{T - P} \quad \text{bzw.} \quad \phi = \frac{\chi^2(\mu; y)}{T - P} \qquad (5\text{ a,b})$$

Im klassischen linearen Modell entspricht die Devianz $D(\mu; y)$ und die verallgemeinerte X²-Statistik der Summe der Abweichungsquadrate (vgl. Tab. 2) und der Dispersionsparameter ϕ der unbekannten Varianz σ^2 der Responsevariablen. Aus (3) leitet sich auch die Verallgemeinerung des Likelihood-Ratio-Tests zum Vergleich zweier geschachtelter Modelle ab:

$$\frac{D(\mu_b; y) - D(\mu_a; y)}{\phi} \sim \chi^2_{a-b}$$

In diesem Fall ist das Modell μ_b mit b zu schätzenden Parametern eine Vereinfachung des Modells μ_a mit a zu schätzenden Parametern. Diese Teststatistik folgt asymptotisch einer X²-Verteilung mit a-b Freiheitsgraden. Ist der Dispersionsparameter ϕ z.B. durch (5a) zu schätzen, folgt die Teststatistik asymptotisch einer F-Verteilung

$$\frac{(D(\mu_b; y) - D(\mu_a; y))/(a-b)}{D(\mu_a; y)/(T-a)} \sim F^{a-b}_{T-a} \qquad (6)$$

und wird als „Verallgemeinerter F-Test“ bezeichnet. Beide Teststatistiken eignen sich sowohl zur Bewertung des linearen Prädiktors, z.B. bei einer schrittweisen Variablenselektion, als auch zum Test von Parametern in der Link- und Varianzfunktion.

Power-Varianz und Power-Link-Funktionen

Für die Auswahl der optimalen Link- bzw. Varianzfunktion ist es hilfreich, dass alle relevanten Link- und Varianzfunktionen (siehe Tab. 1) zu der Familie der Power-Link-Funktionen $\eta(\mu;\xi)$ bzw. Power-Varianz-Funktionen $V(\mu;\psi)$ zählen:

$$\eta(\mu;\xi) = \begin{cases} \mu^{\xi} & \text{für } \xi \neq 0 \\ \log(\mu) & \text{für } \xi = 0 \end{cases} \quad \text{bzw.} \quad V(\mu;\psi) = \mu^{\psi} \quad \text{für} \quad \psi \geq 0 \qquad (7)$$

Wie aus Tabelle 3 zu ersehen ist, umfasst die Familie der Power-Link- bzw. Power-Varianz-Funktionen alle wichtigen GLMs in ihren kanonischen Formen.

Verteilung	$V(\mu,\psi)$	$\eta(\mu,\xi)$
Normal	$\psi = 0$	$\xi = 1$
Poisson	$\psi = 1$	$\xi = 0$
Gamma	$\psi = 2$	$\xi = -1$
Invers-Normal	$\psi = 3$	$\xi = -2$

Tabelle 3 GLMs mit Power-Link- bzw. Power-Varianz-Funktionen

Andere Kombinationen von ψ und ξ, die zu keiner Verteilung aus der Exponentialfamilie gehören, werden als Quasi-Likelihood-Modelle bezeichnet. Da in die Modellschätzung lediglich die Erwartungswert- und Varianzbeziehungen aus (2) eingehen, gelten für die Parameterschätzer alle wesentlichen Eigenschaften der GLM-Schätzer zumindest asymptotisch. Diese sog. Quasi-Likelihood-Schätzer bzw. -Modelle ermöglichen es, z.B ein multiplikatives Modell mit konstanten Variationskoeffizienten ($\xi = 0$ und $\psi = 2$) zu modellieren, ohne sich über Verteilungsannahmen Gedanken zu machen.

Für die technische Umsetzung ist es hilfreich, dass für Power-Varianz-Funktionen die Devianz (3) direkt in Abhängigkeit von ψ ausgedrückt werden kann, wobei die Link-Funktion über die Erwartungswertbeziehung aus (1) implizit in μ_i enthalten ist:

$$D_Q(\mu; y, \xi, \psi) = 2\sum_{i=1}^{T} w_i \left[-\frac{y_i}{1-\psi}\left(\mu_i^{1-\psi} - y_i^{1-\psi}\right) + \frac{1}{2-\psi}\left(\mu_i^{2-\psi} - y_i^{2-\psi}\right)\right] \quad \text{für } \psi \neq 1, 2 \qquad (8)$$

Für die beiden Ausnahmen ψ = 1,2 sei auf die Devianz-Formeln in Tab. 2 verwiesen. Die Sensitivität von Modellen gegenüber inkrementellen Änderungen in der Varianz- und Link-Funktion kann analytisch und graphisch über die Veränderungen der skalierten Quasi-Devianz $D_Q^*(\mu; y, \xi, \psi)$ mit

$$D_Q^*(\mu; y, \xi, \psi) = \frac{D_Q(\mu; y, \xi, \psi)}{\phi} + \sum_{i=1}^{T} \log\left[2\pi \frac{\phi}{w_i} V(y_i) \right] \qquad (9)$$

beurteilt werden. Zu dem ersten Term, der skalierten Devianz aus (4a), wird ein von der Modellschätzung unabhängiger Korrekturfaktor addiert, der erst den Vergleich über verschiedene Varianzfunktionen ermöglicht.

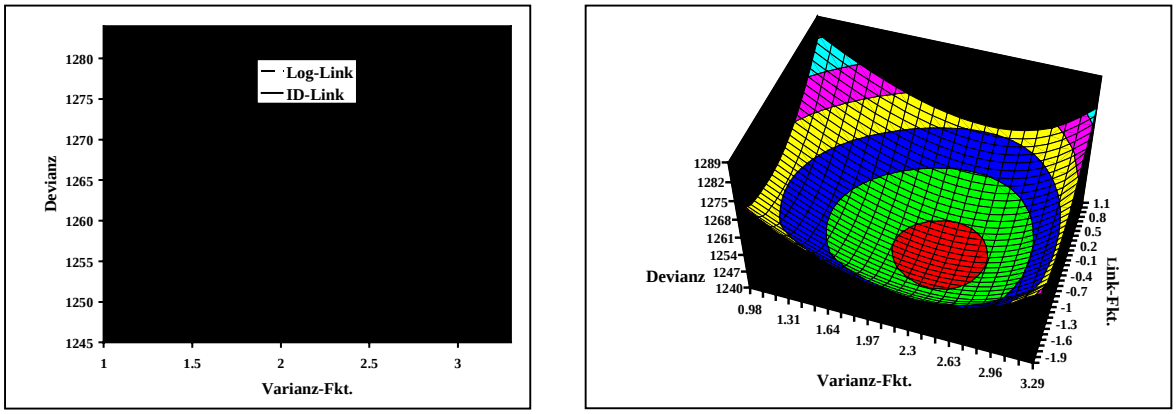

Abb. 1a) Devianz-Profile für zwei Link-Fkt.
Abb. 1b) Dreidimensionales Devianz-Profil

Die Veränderung der skalierten Quasi-Devianz $D_Q^*(\mu; y, \xi, \psi)$ in Abhängigkeit von ξ bzw. ψ lässt sich anschaulich in Devianz-Profil-Plots abbilden. Abb. 1a zeigt ein zwei-dimensionales Devianz-Profil für zwei gegebene ξ bei variierenden ψ. Abb. 1b zeigt ein dreidimensionales Devianz-Profil bei der simultanen Variation von ξ und ψ.

Modellierung von GLMs mit PROC GENMOD

Von der Struktur gleicht PROC GENMOD anderen SAS–Prozeduren für lineare Modelle, wie z.B. Proc GLM oder Proc REG. Zwei interessante Optionen von PROC GENMOD sind die Modellierung von benutzerdefinierten Link- und Varianzfunktionen und die Performance-Optimierung durch Vorgabe von Startwerten für die Parameterberechnung.

Ausgehend von dem Gamma-Modell mit logarithmischer Linkfunktion wird in mehreren Schritten gezeigt, wie Power-Link- bzw. Power-Varianz-Funktionen in PROC GENMOD implementiert werden können, um die Datenpunkte für Devianz-Profil-Plots zu berechnen.
Bei der Modellierung mit PROC GENMOD (Beispiel 1) müssen neben der z.B. aus PROC GLM bekannten MODEL-Anweisung mit Angabe der Modellvariablen lediglich die Verteilung der Responsevariablen und die Linkfunktion (DIST- und LINK-Option in der MODEL-Anweisung) definiert werden. Mit der SCWGT-Anweisung kann die Gewichtungsvariable *w* berücksichtigt werden. Über die MAKE- Anweisung werden alle wichtigen Ergebnisse, wie z.B. die geschätzten Parameter, zur weiteren Bearbeitung in SAS-Dateien abgelegt.

```
/*******************************************/
/**      Bsp. 1 Gamma-Modell „Einfach“      **/
/*******************************************/

  proc genmod data=MYDATA;
    class X1 X2 X3;
    model Y = X1 X2 X3 / dist=gamma link=log;
    scwgt W;

   make 'modelfit' out = MODDAT;
   make 'parameterestimates' out = PAREST;
   make 'obstats' out = OBSDAT;
  run;

/*******************************************/
```

Beispiel 1: Gamma-Modell „Einfach“

Durch die DIST-Option wird die für die Schätzung wichtige Devianz- bzw. Varianz-Funktion festgelegt. Im Beispiel 2 werden beide Funktionen für das Gamma-Modell aus Beispiel 1 gemäß den Funktionen aus Tab. 1 und 2 direkt implementiert. Hierzu stehen in PROC GENMOD u.a. die Systemvariablen _RESP_ und _MEAN_ für die Responsevariable y bzw. ihren geschätzten Erwartungswert μ zur Verfügung.

```
/*******************************************/
/**    Bsp. 2 Gamma-Modell „Kompliziert“    **/
/*******************************************/

  proc genmod data=MYDATA;
    class X1 X2 X3;

    variance var = _MEAN_**2;
    deviance dev = 2*((_RESP_-_MEAN_)/_MEAN_-log(_RESP_/_MEAN_));

    model Y = X1 X2 X3 / link=power(0);
    scwgt W;
  run;

/*******************************************/
```

Beispiel 2: Gamma-Modell „Kompliziert“

In Beispiel 3 werden die in (6) vorgestellten Power-Link- bzw. Power-Varianz-Funktionen implementiert. Power-Link-Funktionen werden über die LINK-Option POWER(Wert) abgebildet. Die entsprechende Devianz-Funktion aus (8) wird analog dem Beispiel 2 implementiert. Zur besseren Strukturierung können alle relevanten Variablen und Formeln vorab definiert werden, um sie dann komprimiert in dem VARIANCE- und DEVIANCE-Statement zusammenfließen zu lassen. So können z.B. IF-THEN-ELSE-Beziehungen für komplexere Devianz-Strukturen implementiert werden. In Beispiel 3 wird wiederum die Gamma-Verteilung mit logarithmischer Linkfunktion, d.h. 2 als Exponent der Power-Varianz-Funktion und 0 als Exponent der Power-Link-Funktion, modelliert.

```
/*********************************************/
/**  Bsp. 3 Gamma-Modell „sehr Kompliziert“  **/
/*********************************************/

  proc genmod data=MYDATA;
    class X1 X2 X3;
      P = 2;
      M = _MEAN_;
      Y = _RESP_;
      If _Y = 0 then _D = 0;
      else if p=1 then D=2*(_Y*log(Y/A)-(Y-A));
      else if P=2 then D=2*( -log(Y/A) + (Y - A)/A);
      else  D = 2 * ( (-Y)/(1-p)*(a**(1-p)-y**(1-p))
                  + 1/(2-p)*(a**(2-p)-y**(2-p)));
    variance var = A**2;
    deviance dev = D;
    model Y = X1 X2 X3 / link=power(0) ;
    scwgt W;
  run;

/*********************************************/
```

Beispiel 3: Gamma-Modell „sehr kompliziert“

Zur Datengenerierung für Devianz-Profil-Plots muss eine Vielzahl von Modellen iterativ berechnet werden. Über ein Macro kann diese Aufgabe automatisiert werden, wobei die Indizes für die Power-Link- und Power-Varianz-Funktionen als Laufindizes der Iterationen dienen. Die Berechnung der skalierten Quasi-Devianz-Werte $D_Q^*(\mu; y, \xi, \psi)$ gemäß den Formeln (5b), (8) und (9) erfolgt durch Auslesen der Modellinformationen über die MAKE-Anweisung und Weiterverarbeitung in einem DATA-Step. Die graphische Darstellung der Devianz-Profile (vgl. Abb. 1a,b) kann z.B. über SAS-GRAPH erfolgen.

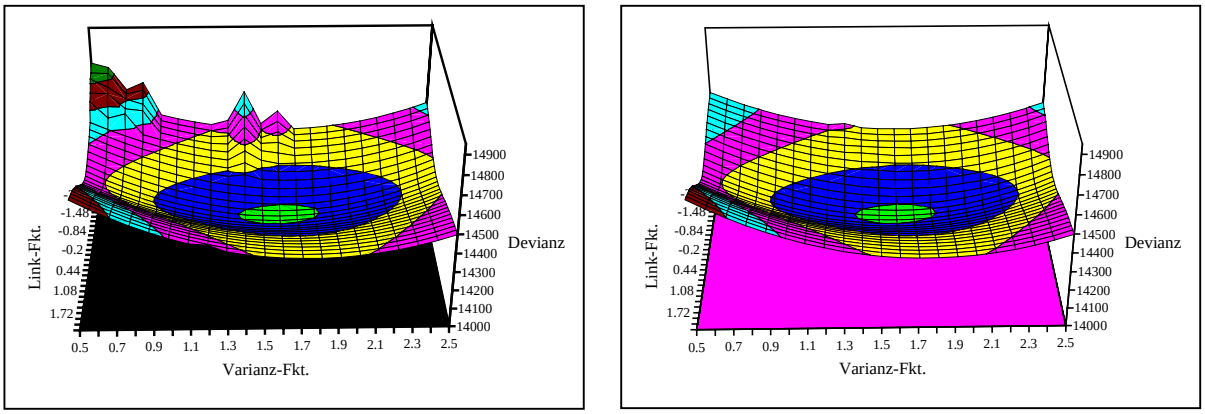

Abb. 2a) Devianz-Profil ohne Startwerte
Abb. 2b) Devianz-Profil mit Startwerte

Je weiter ξ und ψ von ihren Devianz-minimierenden Optima entfernt liegen, desto größer ist die Wahrscheinlichkeit, dass die Modelle nicht bzw. nur an einem lokalen Optimum konvergieren. Die hier ermittelten Quasi-Devianz-Werte $D_Q^*(\mu; y, \xi, \psi)$ werden als extreme (nicht verwertbare) Ausreißer in den Plots (vgl. Abb. 2a) sichtbar. Diese Probleme werden häufig von folgenden Warn- oder Fehlermeldungen im SAS-Log-File begleitet:

```
WARNING: The relative Hessian convergence criterion of 0.001782 is greater than
         the limit of 0.0001. The convergence is questionable.
WARNING: Procedure is continuing but the validity of the model fit is questionable

WARNING: The specified model did not converge.
ERROR:   Error in computing inverse link function.
```

Beispiel 4: Fehlermeldungen in PROC GENMOD

Leider stellt PROC GENMOD keine Systemvariable o.ä. zur Verfügung, um dieses Problem anzuzeigen. Diese Probleme in der Modellschätzung können in der MODEL-Anweisung mit der Option STRTVALS zur Berücksichtigung von Startwerten begrenzt werden. Ziel ist es, Startwerte zu finden, die sich bereits in Nähe der endgültigen Parameterschätzer befinden. Es liegt auf der Hand, dass sich die Modellparameter bei einer inkrementellen Änderung des Index ψ der Power-Varianz-Funktion i.d.R. nur marginal ändern. Die Parameter einer Iteration sind also sinnvolle Startwerte der jeweils nächsten Iteration. Beispiel 5 zeigt ein einfaches iteratives Makro zur effizienten Nutzung dieser Option STRTVALS.

```
/*******************************************************/
/**  Bsp. 5 Verwendung von Startwerten in einem Makro  **/
/*******************************************************/

%macro QUASI;

%let startwerte=; /*Erste Iteration hat keine Startwerte*/
%do index = -1 %to 1 %by 1;  /*Laufindex für Power-Link */

  proc genmod data=MYDATA;
      [Weiterer Code wie Bsp. 3]
   model Y = X1 X2 X3 / link = pow(&index)  &startwerte;
   make 'parameterestimates' out = PAREST; /*S.-W. nächste It.*/
  run;

/***  Auslesen der Modellparameter in Macro-Variablen ***/;
   data  _null_;
   set   PAREST;
      call symput( 'B'||left(_N_), put( estimate, 16.8 ) );
      call symput( 'parms', put( _N_, 4.) );
   run;

/****     Text-String der Startwerte generieren     *****/
     %let parms = %eval(&parms-1);   /*Letzter Parm 'weg*/
     %let startwerte = intercept=&B1  initial=  ;
        %do i=2 %to &parms;
           %let startwerte=&startwerte &&B&i;
        %end;
    run;
 %end;          /* Ende der Schleife */;
%mend QUASI;      /* Ende des Macros */

/*******************************************************/
```

Beispiel 5: Verwendung von Startwerten

In dem Makro QUASI_LH im Anhang wird Beispiel 5 um die Schleife für die Power-Varianz-Funktion und die DATA-Steps zur Berechnung und Ausgabe der Quasi-Devianz-Werte $D_Q^*(\mu; y, \xi, \psi)$ erweitert. Es empfiehlt sich, die Schleife für die Power-Varianz-Funktion als innere Schleife zu programmieren. Die Laufindizes sollten so gewählt werden, dass die Itera-

tionen sich von einer sicher konvergierenden Ausgangskombination der Indizes ξ bzw. ψ in die Randbereiche vorarbeiten.

Ein positiver Nebeneffekt der Verwendung von Startwerten ist die teilweise erhebliche Reduktion der Rechenzeit. In Abhängigkeit der Datenkonstellation und der Güte der Startwerte ist mit einer Verringerung der CPU-Zeit zwischen 20 und 50 Prozent zu rechnen. Die Verwendung von Startwerten kann z.B. auch bei der schrittweisen Variablenauswahl, wie sie u.a. in PROC REG fest implementiert ist, sinnvoll eingesetzt werden.

Schlussbemerkung

Im Bereich der linearen Modelle stellt PROC GENMOD ein sehr vielseitiges und leistungsfähiges Werkzeug dar. Die Stärke von PROC GENMOD liegt weniger in der Tiefe der Funktionalität für einzelne Teilfragen. Diese werden durch die auf das jeweilige lineare Modell zugeschnittenen Prozeduren, wie z.B. PROC REG oder PROC PROBIT besser abgedeckt. Die Stärken liegen vor allem im explorativen Bereich bei Modellbewertung und -auswahl.

Literaturverzeichnis

O. Kruse (1997): Modelle zur Analyse und Prognose des Schadenbedarfs in der Kfz-Haftpflichtversicherung, Karlsruhe: Verlag Versicherungswirtschaft

J.A. Nelder & D. Pregibon (1987): An Extended Quasi-Likelihood-Funktion, Biometrika 74, S. 221–232

J.A. Nelder & R.M.W. Wedderburn (1972): Generalized Linear Models, Journal of the Royal Statistical Society A, 1972, S.370-384

A.E. Renshaw (1994): Modelling the Claims Process in the Presence of Covariates, ASTIN-Bulletin 24, S. 265-285

SAS Institute Inc. (1993): SAS Technical Report P-243: The GENMOD Procedure, Cary: SAS Institute Inc.

Anhang: SAS-Macro QUASI_LH

```
%macro QUASI_LH(InData=,OutFile=, Y_Var=, X_Vars=, Ex_Var=,Cls_Vars=,
              StartV=, StopV= ,IncrV=, StartL=, StopL= ,IncrL= );
/*********************************************************************/
/*          AUTHOR : Olaf Kruse, Summer 1995                         */
/*                   Revised Spring 2001                             */
/*********************************************************************/
/*  USAGE:                                                           */
/*   %QUASI_LH(InData   = <Input Data>,                              */
/*             OutData  = <Output File>                              */
/*             Y_Var    = <Response-Variable>,                       */
/*             X_Vars   = <Prediktoren>,                             */
/*             Cls_Vars = <Class-Variablen, Faktoren >,              */
/*             Ex_Var   = <Beobachtungs-Gewichte >,                  */
/*   StartV = <Startwert*100 der Power-Varianz-Funktion>             */
/*   StopV  = <Endwert*100 der Power-Varianz-Funktion>               */
/*   IncrV  = <Iterationsschrittweite*100 für Power-Varianz>         */
/*   StartL = <Startwert*100 der Power-Link-Funktion>                */
/*   StopL  = <Endwert*100 der Power-Link-Funktion>                  */
/*   IncrL  = <Iterationsschrittweite*100 für Power-Link>            */
/*********************************************************************/
/*****************    Allgemeine Vorbereitungen    *****************/
ods listing close;
%if %upcase(&Cls_Vars) ne %then %let clsstmt = %str(CLASS &Cls_Vars;);
%else %let clsstmt= ;
%let StrtVals= ;
%let IncrV = %sysfunc(Abs(&IncrV));
%let IncrL = %sysfunc(Abs(&IncrL));
%if &StartL > &StopL %then %let IncrL = %eval(&IncrL*(-1));
%if &StartV > &StopV %then %let IncrV = %eval(&IncrV*(-1));

DATA &OutFile;
set _null_;
run;
/*********   Äußere Schleife: Power-Link-Funktion          *********/
/*********   Innere Schleife: Power-Varianz-Funktion       *********/
/*********                Hinweis:                 -      *********/
/*********   Schleifen-Reihenfolge kann durch Tauschen der *********/
/*********   nächsten beiden Code-Zeilen geändert werden!  *********/

%do vau = &StartL %to &StopL %by &IncrL;
  %do tau = &StartV %to &StopV %by &IncrV;

  %let beta = %sysevalf(&vau/100);/* Laufindex Power-Link-Fkt.    */
  %let alpha =%sysevalf(&tau/100);/* Laufindex Power-Varianz-Fkt. */
  %put **Log-File:  Link: &beta  Var: &alpha  **;

  proc genmod data=&InData;
    &clsstmt
    make 'modelfit' out = MODDAT;
    make 'parameterestimates' out = PAREST;
    make 'obstats' out = OBSDAT;
    _P = &alpha;
    _A = _MEAN_;
    _Y = _RESP_;
    If _Y = 0 then _D = 0;
      else if _P=1 then _D=2*(_Y*log(_Y/_A)-(_Y-_A));
      else if _P=2 then _D=2*((_Y-_A)/_A-log(_Y/_A));
      else _D = 2 * ( (-_Y)/(1-_P)*(_A**(1-_P)-_Y**(1-_P))
                  + 1/(2-_P)*(_A**(2-_P)-_Y**(2-_P)));
    variance _VAR = _A**_P;
    deviance _DEV = _D;
    scwgt  &Ex_Var;
    model &Y_Var = &X_Vars / link = pow(&beta) obstats &StrtVals ;
    run;
```

```
/*********        String der Startwerte generieren         *********/
      data  _null_ ;
      set   PAREST;
         call symput( 'B'||left(_N_), put( estimate, 16.8 ) );
         call symput( 'parms', put( _N_, 4.) );
         run;
      %let parms = %eval(&parms-1); /*Letzter Parm 'weg*/
      %let Strtvals= intercept=&B1  initial=  ;
         %do i=2 %to &parms;
            %let Strtvals=&Strtvals &&B&i;
         %end;
       run;
/******  Berechnen der Werte der Extended Quasi-Devianz  **********/
/*******   Auslesen der Anzahl Freiheitsgrade für Teta   **********/
      data _NULL_;
      set MODDAT;
         if _N_ = 1 then call symput( 'DF', put( df, 10. ) );
         run;      /***  Anzahl Freiheitsgrade für Teta  ***/
/******  Berechnen der Werte der Extended Quasi-Devianz  **********/
      data ExQL;
      merge &InData(keep= &Y_Var &Ex_Var
                     rename=(&Y_Var=y &Ex_Var=wgt ))
            obsdat(keep=pred rename=(pred=a));
         p=&alpha;
         if wgt= 0 or y=0 then do;
            nobs=0; /* Anzahl gültiger Beobachtungen */
            Chi2=0; /* Für Teta */
            Dev=0;  /* Normale Devianz */
         end;
         else do;
            if p=1 then Dev =2*wgt*(y*log(y/a)-(y-a));
            else if p=2 then Dev =2*wgt*((y-a)/a-log(y/a));
            else Dev =2*wgt*((-y)/(1-p)*(a**(1-p)-y**(1-p))+
                              1/(2-p)*(a**(2-p)-y**(2-p)));
            Chi2=wgt*(y-a)**2/a**p;
            nobs=1;
            DY=log(6.2832/wgt*y**p)/* Für Korrekturterm */;
         end;
         run;

   proc means sum noprint data=ExQL;
      var Dev DY Chi2 nobs;
      output out=statistics sum=;
      run;

   data statistics(keep=PLink PVar Teta D1 D2 QDev );
      retain PLink PVar Teta D1 D2 QDev ;
   set statistics;
      PLink=&beta;          /* Power-Link-Index */
      PVar =&alpha;         /* Power-Var-Index */
      Teta=Chi2/&DF;        /* Dispersions-Par. */
      D1=Dev/Teta;          /* skalierte Dev.   */
      D2=DY+nobs*log(teta); /* Korrekturterm    */
      QDev=D1+D2;           /* Quasi-Devianz    */
      run;

   data &OutFile; /* Ergebnisse der Iterationen zusammenführen */
   set &OutFile statistics;
   run;

   %end;  * innere Schleife *;
%end;  * äußere Schleife *;

ods listing;

%mend QUASI_LH;
```