

Technische Universität München

ZENTRUM MATHEMATIK

**Verallgemeinerte Poisson und Nullenüberschuß –
Regressionsmodelle mit regressiertem
Erwartungswert, Dispersions- und
Nullenüberschuß-Parameter
und
eine Anwendung zur Patentmodellierung**

Diplomarbeit

von

Vinzenz Erhardt

Themenstellerin: Prof. Dr. Claudia Czado

Betreuer: Dr. Aleksey Min

Abgabetermin: 08. Mai 2006

Hiermit erkläre ich, daß ich die Diplomarbeit selbstständig angefertigt und nur die angegebenen Quellen verwendet habe.

Garching, den 08. Mai 2006

Danksagung

Mein Dank gilt Frau Prof. Claudia Czado für die kontinuierliche Betreuung und die zahlreichen Anregungen während der Erstellung der Diplomarbeit. Außerdem danke ich Dr. Aleksey Min für die vielen Anregungen, Hinweise auf Skripten und die Beantwortung meiner Fragen sogar außerhalb der Bürozeiten. Zuletzt danke ich Stefan Wagner vom Forschungsinstitut INNO-tec für die betriebswirtschaftliche Überprüfung meiner Modellinterpretation.

Inhaltsverzeichnis

Einleitung	2
1 ZIGP-Modell mit Regression auf μ und ω	4
1.1 GP - und ZIGP – Verteilungen	4
1.1.1 Wahrscheinlichkeitsfunktionen mit Eigenschaften	4
1.1.2 Erzeugung von Zufallsvariablen	10
1.2 Konstruktion eines $ZIGP(\mu_i, \varphi, \omega_i)$ – Regressionsmodells	12
1.2.1 ML-Schätzer der Regressionsparameter $\boldsymbol{\delta} = (\boldsymbol{\beta}^t, \boldsymbol{\alpha}^t, \boldsymbol{\gamma}^t)^t$	14
1.2.2 Loglikelihoodfunktion der $ZIGP(\mu_i, \varphi, \omega_i)$ –Regression	15
1.2.3 Score-Vektor der $ZIGP(\mu_i, \varphi, \omega_i)$ –Regression	16
1.2.4 Hessematrix der $ZIGP(\mu_i, \varphi, \omega_i)$ –Regression	17
1.2.5 Erwartete Fisher-Information der $ZIGP(\mu_i, \varphi, \omega_i)$ –Regression	20
1.3 Anmerkungen zur Software für die Modellanpassung und -analyse	24
1.4 Simulationsstudie	25
2 ZIGP - Modell mit Regression auf μ, φ und ω	35
2.1 Definition und Eigenschaften	35
2.1.1 Loglikelihoodfunktion der $ZIGP(\mu_i, \varphi_i, \omega_i)$ –Regression	36
2.1.2 Score-Vektor der $ZIGP(\mu_i, \varphi_i, \omega_i)$ –Regression	37
2.1.3 Hessematrix der $ZIGP(\mu_i, \varphi_i, \omega_i)$ –Regression	38
2.1.4 Erwartete Fisher-Information der $ZIGP(\mu_i, \varphi_i, \omega_i)$ –Regression	41
2.2 Anmerkungen zur Software für die Modellanpassung und -analyse	42
2.3 Konsistenz und Konvergenz	43
2.3.1 Notationen und Voraussetzungen	43
2.3.2 Konsistenz und Konvergenz	44
2.3.3 Beweis der asymptotischen Existenz und der schwachen Konsistenz der Lösungen der Scoregleichungen	47
2.3.4 Beweis der asymptotischen Normalität der Scorevektoren \mathbf{s}_n	51
2.3.5 Beweis der asymptotischen Normalität der ML-Schätzer $\hat{\boldsymbol{\delta}}_n$	56
2.4 Simulationsstudie	61
3 Zero Modified Generalized Poisson - Modelle	69
3.1 Einführung	69
3.2 Definition und Eigenschaften	72
3.2.1 $ZMGP(\mu_i, \varphi, \omega_i)$ – Regressionsmodell	72

3.2.2	Loglikelihoodfunktion der $ZMGP(\mu_i, \varphi, \omega_i)$ -Regression	73
3.2.3	Score-Vektor der $ZMGP(\mu_i, \varphi, \omega_i)$ -Regression	73
3.2.4	Hessematrix der $ZMGP(\mu_i, \varphi, \omega_i)$ -Regression	74
3.2.5	Erwartete Fisher-Information der $ZMGP(\mu_i, \varphi, \omega_i)$ -Regression . . .	77
3.3	Simulationsstudie	78
4	Anwendung zur Patent-Modellierung	86
4.1	Datenbeschreibung	86
4.2	Explorative Datenanalyse	91
4.2.1	Kovariablen für die Modellierung des Erwartungswertes	91
4.2.2	Kovariablen für die Modellierung der Überdispersion	107
4.2.3	Kovariablen für die Modellierung des Nullenüberschuß	113
4.3	Vergleichskriterien für die Anpassungsgüte	117
4.3.1	Vuong - Test	119
4.3.2	Akaike Informationskriterium	121
4.4	Vergleich mit AIC-Werten und Vuong-Tests	122
4.5	Interpretation des Modells	125
4.5.1	Einfluß der Kovariablen auf die Outsourcing-Quote	125
4.5.2	Einfluß der Kovariablen auf die Überdispersion	133
4.5.3	Einfluß der Kovariablen auf die Nullen-Wahrscheinlichkeit	137
5	Zusammenfassung	139
A	Software $ZIGP(\mu_i, \varphi, \omega_i)$-Regressionsmodell	148
B	Software $ZIGP(\mu_i, \varphi_i, \omega_i)$-Regressionsmodell	177
C	Software $ZMGP(\mu_i, \varphi, \omega_i)$-Regressionsmodell	211
D	Summaries für den Patent-Datensatz	238

Einleitung

Zählraten stammen aus der Menge der natürlichen Zahlen mit der Null. Für die Modellierung dieser Zählraten wird üblicherweise die Poissonverteilung verwendet; in der Poissonregression wird angenommen, daß die Responsevariablen unabhängig Poisson-verteilt sind. Anwendungen sind etwa die Modellierung der Anzahl von Versicherungsschäden oder Überlebenszeitmodelle. Bei der Poissonverteilung stimmen Erwartungswert und Varianz überein. In der Praxis beobachtet man jedoch häufig, daß die Daten dies nicht erfüllen. Als Überdispersion bezeichnet man daher in der Poissonregression das Phänomen, daß die empirische Varianz der Daten ihren Mittelwert übersteigt. Der umgekehrte Fall wird als Unterdispersion bezeichnet, ist jedoch selten. Die Gründe für Überdispersion sind vielfältig: erklärende Variablen in Regressionsmodellen bleiben unberücksichtigt, die Linkfunktion wurde falsch gewählt. Eine weitere Ursache, die in dieser Arbeit untersucht wird, ist ein Überschuß an Nullen (engl. "Zero Inflation").

Consul und Jain [6] führten eine Verallgemeinerte Poissonverteilung (GP-Verteilung) ein, die die Abbildung der Dispersion mittels eines weiteren Parameters ermöglicht und die gewöhnliche Poissonverteilung als Spezialfall enthält. Zur Abbildung eines Nullenüberschuß gibt es von Mullahy [28] ein Zero Inflated Poisson Regressionsmodell (ZIP), Lambert [24] hat es noch erweitert. Famoye und Singh [13] haben in einem Mixture-Ansatz ein Zero Inflated Generalized Poisson - Modell (ZIGP) untersucht. In Kapitel 1 wird ein ZIGP-Regressionsmodell vorgestellt, das Regression auf den Erwartungswert und den Zero Inflation - Parameter zuläßt. In Kapitel 2 wird dieses dann noch um die Regression auf den Überdispersionsparameter erweitert. Für dieses umfassende Modell werden die Eigenschaften Konsistenz und Konvergenz untersucht. Der seltene Fall der Unterdispersion kann mit diesem Modell aufgrund der Wahl der Verknüpfungsfunktion für φ_i jedoch nicht abgebildet werden. In Kapitel 3 wird mit der Entwicklung von "Zero Modified Generalisierten Poissonmodellen" (ZMGP) der Versuch unternommen, einfache Tests für die Fragestellung zu ermöglichen, ob ein Nullenüberschuß vorhanden ist. Eine "R"-Software berechnet für all diese Modelle die Maximum-Likelihood-Schätzer der Koeffizienten und die Fisherinformation. Die Software findet sich in Anhang A bis C. Zusätzlich werden in mehreren Simulationsszenarien werden zunächst jeweils Zufallsvariablen erzeugt und danach die verwendeten Koeffizienten geschätzt sowie deren Güte beurteilt. Zur Erzeugung von ZIGP - verteilten Zufallsvariablen wird die Invertierungsmethode vorgestellt, für die lediglich gleichverteilte Zufallszahlen als Input benötigt werden. Mit den angefügten "R"-Programmen ist eine übersichtliche und sehr einfach zu bedienende Software gegeben, deren Ausgabe sich stark an den glm - Summaries orientiert, die in "R" fest implementiert sind. Darüber hinaus wird in strenger Analogie zu "R" - Hilfsdateien jede Funktion beschrieben, somit ist eine Einarbeitung sogar in Hilfsprogramme leicht möglich. Ferner

wird in Kapitel 4 anhand eines Patent-Datensatzes gezeigt, daß durch Regression auf den Dispersions- und den Nullenüberschußparameter eine bessere Anpassungsgüte erreicht wird. Es wird vorgeführt, wie man Variablen für die Modellierung von Erwartungswert, Überdispersion und Nullenüberschuß auswählen kann. Zum Vergleich mit anderen Modellen werden das Akaike Informationskriterium und der Vuong Test herangezogen.

Da diese Diplomarbeit auf der Diplomarbeit von Stekeler [35] aufbaut, wird an einigen Stellen lediglich auf diese Arbeit verwiesen. Der Schwerpunkt bei Stekeler [35] lag eher in einer umfassenden theoretischen Darstellung verschiedener Anpassungsmodelle, die vorliegende Arbeit konzentriert sich eher auf die Entwicklung einer komplexen Software und einer eingehenden Untersuchung des Patent-Datensatzes. Im Vordergrund steht außerdem ein umfassender Vergleich von neun Regressionsmodellen, die auf der Poisson-Verteilung basieren.

Kapitel 1

ZIGP-Modell mit Regression auf μ und ω

1.1 Verallgemeinerte Poisson – und Zero Inflated – Verteilungen

1.1.1 Wahrscheinlichkeitsfunktionen mit Eigenschaften

Nach Consul und Jain [6] kann man die gewöhnliche Poissonverteilung um einen Dispersionsparameter erweitern. Die folgende Darstellung entsteht aus der Verteilung von Consul und Jain [6] durch Umparametrisierung und wird 'Mean Parametrization' genannt. Sie hat den Vorteil, daß analog zur Poissonverteilung einer der Parameter - nämlich μ - bereits dem Erwartungswert entspricht. Man substituiert lediglich $\theta = \frac{\mu}{\varphi}$ und $\lambda = 1 - \frac{1}{\varphi}$. Die so entstehende Verteilung $GP(\mu, \varphi)$ mit Erwartungswert μ und Dispersionsparameter φ enthält die Poissonverteilung als Spezialfall für $\varphi = 1$.

Definition 1.1 (Verallgemeinerte Poissonverteilung (GP)).

Sei X eine Zufallsvariable mit diskreter Verteilung, die auf \mathbb{N} definiert ist. $p^*(x | \mu, \varphi)$ bezeichne die Wahrscheinlichkeit $P(X = x)$, daß X den nicht-negativen ganzzahligen Wert x annimmt. X ist Verallgemeinert Poisson-verteilt mit Erwartungswerten μ und Dispersionsparameter φ , falls

$$p^*(x | \mu, \varphi) = \begin{cases} \frac{\mu(\mu+(\varphi-1)x)^{x-1}}{x!} \varphi^{-x} e^{-\frac{1}{\varphi}(\mu+(\varphi-1)x)} & x = 0, 1, 2, \dots \\ 0 & \text{für } x > m, \text{ wenn } \varphi < 1 \end{cases} \quad (1.1)$$

und Null sonst (d.h. für $x \notin \mathbb{N}$). Für die Parameter gelten dabei folgende Einschränkungen:

- $\mu > 0$,
- $\varphi > \max(\frac{1}{2}, 1 - \frac{\mu}{m})$,

wobei m die größte natürliche Zahl ist, für die gilt, daß $\mu + m(\varphi - 1) > 0$, wenn $\varphi < 1$.

Zur Erklärung: Falls $\varphi < 1$ und $x > m$ wird der Term $(\mu + (\varphi - 1)x)^{x-1}$ negativ. Da in diesem Fall die Wahrscheinlichkeitsfunktion in der Darstellung für $x \leq m$ auch $p^*(x | \mu, \varphi) < 0$, wird sie für $x > m$ auf 0 gesetzt. Damit gibt m also an, wieviele echt positive Ausprägungen ("Klassen") $x = 0, \dots, m$ existieren.

Die zweite Einschränkung läßt nun zu, φ so groß zu bestimmen, daß m Klassen entstehen. Bei gegebenem μ ist m so zu wählen, daß $\mu + (\varphi - 1)x > 0$.

$$\begin{aligned} \mu + (\varphi - 1)x &> 0 && \Leftrightarrow \\ \varphi &> 1 - \frac{\mu}{m} \end{aligned}$$

Die generelle Einschränkung $\varphi > \frac{1}{2}$ ist analog zu Consul's Beschränkung $-1 < \lambda \leq 1$ gewählt und ergibt sich durch Umformung mit $\lambda = 1 - \frac{1}{\varphi}$.

Wir schreiben von nun an $X \sim GP(\mu, \varphi)$. Erwartungswert und Varianz der GP-Verteilung in der *Mean Parametrization* wurden von Consul und Famoye [4] angegeben. Sie geben Aufschluß über die Bedeutung von φ .

Satz 1.2 (Erwartungswert und Varianz der GP-Verteilung).

Sei $X \sim GP(\mu, \varphi)$ Verallgemeinert Poisson-verteilt und habe die Wahrscheinlichkeitsfunktion $P_x(\mu, \varphi)$. Dann sind

$$E(X) = \mu \quad \text{und} \quad (1.2)$$

$$Var(X) = \varphi^2 \mu \quad (1.3)$$

Beweis.

Der Beweis für den Erwartungswert wird ähnlich zu Consul [5, S. 14] ausgeführt. Allerdings wird dort die ursprünglich in Consul und Jain [6] eingeführte Parametrisierung verwendet. Es wird mit Variablensubstitution und der geometrischen Reihe gearbeitet. Bezeichne $E(X | \mu, \varphi)$ den Erwartungswert der verallgemeinert poissonverteilten Zufallsvariablen X mit Parametern μ und φ

$$E(X | \mu, \varphi) = \sum_{x=1}^{\infty} x P_x(\mu, \varphi).$$

Da das Ereignis $X = 0$ keinen Beitrag zum Erwartungswert liefert, kann die Summation bei $x = 1$ begonnen werden.

$$\begin{aligned} E(X) &= \sum_{x=1}^{\infty} x P_x(\mu, \varphi) \\ &= \sum_{x=1}^{\infty} x \frac{\mu(\mu + (\varphi - 1)x)^{x-1}}{x!} \varphi^{-x} e^{-\frac{1}{\varphi}(\mu + (\varphi - 1)x)} \\ &\quad \text{(Substituiere } s := x - 1) \\ &= \sum_{s=0}^{\infty} \frac{\mu + \varphi - 1 + (\varphi - 1)s}{\varphi} \cdot \frac{\mu(\mu + \varphi - 1 + (\varphi - 1)s)^{s-1}}{s!} \varphi^{-s} e^{-\frac{1}{\varphi}(\mu + \varphi - 1 + (\varphi - 1)s)} \end{aligned}$$

$$\begin{aligned}
& \text{(Substituiere } \tilde{\mu} := \mu + \varphi - 1) \\
&= \frac{\mu}{\varphi} \sum_{s=0}^{\infty} (\tilde{\mu} + (\varphi - 1)s) \frac{(\tilde{\mu} + (\varphi - 1)s)^{s-1}}{s!} \varphi^{-s} e^{-\frac{1}{\varphi}(\tilde{\mu} + (\varphi - 1)s)} \\
&= \frac{\mu}{\varphi} \left(\sum_{s=0}^{\infty} \frac{\tilde{\mu}(\tilde{\mu} + (\varphi - 1)s)^{s-1}}{s!} \varphi^{-s} e^{-\frac{1}{\varphi}(\tilde{\mu} + (\varphi - 1)s)} \right. \\
&\quad \left. + \frac{\varphi - 1}{\tilde{\mu}} \sum_{s=0}^{\infty} s \frac{\tilde{\mu}(\tilde{\mu} + (\varphi - 1)s)^{s-1}}{s!} \varphi^{-s} e^{-\frac{1}{\varphi}(\tilde{\mu} + (\varphi - 1)s)} \right) \\
&= \frac{\mu}{\varphi} \left(\sum_{s=0}^{\infty} P_s(\tilde{\mu}, \varphi) + \frac{\varphi - 1}{\tilde{\mu}} \sum_{s=0}^{\infty} s P_s(\tilde{\mu}, \varphi) \right) \\
&= \frac{\mu}{\varphi} \left(1 + \frac{\varphi - 1}{\tilde{\mu}} E(s|\tilde{\mu}, \varphi) \right)
\end{aligned}$$

Der erste Summand in der Klammer wird zu 1, da $P_s(\tilde{\mu}, \varphi)$ die Wahrscheinlichkeitsfunktion einer Wahrscheinlichkeitsverteilung ist. Im zweiten Ausdruck findet man den Erwartungswert einer $GP(\tilde{\mu}, \varphi)$ verteilten Zufallsvariablen s . Nun kann man $\tilde{\mu}$ wieder durch $\mu + \varphi - 1$ ersetzen und die oben beschriebenen Transformationen erneut für $E(s|(\mu + \varphi - 1), \varphi)$ einsetzen.

$$\begin{aligned}
E(X) &= \frac{\mu}{\varphi} \left(1 + \frac{\varphi - 1}{\mu + \varphi - 1} E(s|(\mu + \varphi - 1), \varphi) \right) \\
&= \frac{\mu}{\varphi} \left(1 + \frac{\varphi - 1}{\mu + \varphi - 1} \left(\frac{\mu + \varphi - 1}{\varphi} \left(1 + \frac{\varphi - 1}{\mu + 2\varphi - 2} E(s|(\mu + 2\varphi - 2), \varphi) \right) \right) \right) \\
&= \frac{\mu}{\varphi} \left(1 + \frac{\varphi - 1}{\varphi} \left(1 + \frac{\varphi - 1}{\mu + 2\varphi - 2} E(s|(\mu + 2\varphi - 2), \varphi) \right) \right) \\
&= \frac{\mu}{\varphi} \left(1 + \frac{\varphi - 1}{\varphi} + \frac{(\varphi - 1)^2}{(\mu + 2\varphi - 2)\varphi} E(s|(\mu + 2\varphi - 2), \varphi) \right)
\end{aligned}$$

Wird diese Vorgehensweise iterativ fortgesetzt, so erhält man eine geometrische Reihe (s. z.B. Königsberger [22, S. 60]): $\sum_{i=0}^{\infty} a^i$. Für $|a| < 1$ konvergiert diese gegen $\sum_{i=0}^{\infty} a^i = \frac{1}{1-a}$. Im vorliegenden Fall ist dies wegen $|\frac{\varphi-1}{\varphi}| = |1 - \frac{1}{\varphi}| < 1$ erfüllt und man erhält:

$$\begin{aligned}
E(X) &= \frac{\mu}{\varphi} \left(1 + \frac{\varphi - 1}{\varphi} + \left(\frac{\varphi - 1}{\varphi} \right)^2 + \dots \right) = \frac{\mu}{\varphi} \left(\sum_{i=0}^{\infty} \left(\frac{\varphi - 1}{\varphi} \right)^i \right) \\
&= \frac{\mu}{\varphi} \cdot \frac{1}{1 - \frac{\varphi-1}{\varphi}} = \mu.
\end{aligned}$$

Der Beweis für die Varianz kann jedoch nicht in Analogie zum Erwartungswert berechnet werden, da keine geometrische Reihe mehr entsteht. Ein umfangreicher Beweis kann bei Stekeler [35, S. 7-12, 70f] gefunden werden. Stekeler verwendet dafür zunächst die Parametrisierung nach Consul und Jain [6] und leitet eine Rekursionsgleichung für nicht zentrierte

Momente her. Dazu bestimmt er die Momentenerzeugende Funktion der GP-Verteilung. Zuletzt transformiert er die Varianz auf die *Mean Parametrisierung* $GP(\mu_i, \varphi_i)$. \square

Nun hat man die gewünschte Form: Erwartungswert und Varianz müssen nicht mehr identisch sein. Übersteigt die Varianz den Erwartungswert, spricht man von *Überdispersion*. Dieser Fall tritt für Dispersionsparameter $\varphi > 1$ ein. Entsprechend bezeichnet man den Fall identischer Varianz und Erwartungswertes als *Äquidispersion*. Ist die Varianz geringer als der Erwartungswert, spricht man von *Unterdispersion*. Man erhält insgesamt

- Äquidispersion für $\varphi = 1$
- Überdispersion für $\varphi > 1$
- Unterdispersion für $\varphi < 1$.

Consul [5, S. 300] verwendet den Begriff *Hyperdispersion* anstelle von *Überdispersion*.

Einen *Nullenüberschuß* kann man mit Hilfe von Zero-Inflated-Verteilungen erzeugen. Dazu wird der Zählvariablen X^* eine binäre Variable $\Omega \in \{0, 1\}$ vorgeschaltet. Diese wird als Auswahlvariable bezeichnet, die Wahrscheinlichkeit $\omega = P(\Omega = 0)$ sei der Auswahlparameter. Falls $0 < \omega < 1$, enthält das Modell zusätzliche Nullen. Betrachtet man ω als deterministischen Parameter, so wird mit $\omega < 0$ ein Nullendefizit erzeugt. Dazu jedoch mehr in Kapitel 3.

Die resultierende Zählvariable X kann als Produkt dieser (unabhängigen) Größen betrachtet werden. Die "zero inflated" Variable X entsteht demnach aus zwei unabhängigen Prozessen:

- Auswahlprozeß durch die Auswahlvariable $\Omega \sim \text{Bernoulli}(1 - \omega)$
- Prozeß zur Erzeugung der Zählvariablen $X^* = x^*$

Eine Null in X kann daher auf zwei Arten entstehen: einerseits durch den Auswahlprozeß ($\Omega = 0$) mit Wahrscheinlichkeit ω , andererseits durch den Ursprungsprozeß ($\Omega = 1$, $X^* = 0$) mit Wahrscheinlichkeit $(1 - \omega) \cdot P(X^* = 0)$. Dies erklärt die folgende Wahrscheinlichkeitsfunktion, die von Johnson et al. [20, S. 312-318] untersucht wurde.

Satz 1.3 (Wahrscheinlichkeitsfunktion der ZI-Verteilung).

Ist $\omega \in [0, 1]$ die Wahrscheinlichkeit für $\Omega = 0$ und P^* die Wahrscheinlichkeitsfunktion der Verteilung von X^* , so gilt für die Wahrscheinlichkeitsfunktion der Verteilung von X :

$$\begin{aligned} P(X = x) &= \begin{cases} \omega + (1 - \omega)P_0^* & x = 0 \\ (1 - \omega)P_x^* & x = 1, 2, \dots \end{cases} \\ &= \omega \cdot \mathbb{1}_{\{x=0\}} + (1 - \omega)P_x^* \quad \forall x \in \mathbb{N} \end{aligned}$$

Dabei bezeichnet $\mathbb{1}_{\{A(x)\}}(x)$ die Indikatorfunktion, die bei Eintritt des Ereignisses A den Wert 1 hat und 0 sonst. Man schreibt kurz $\mathbb{1}_{\{A\}}$.

Beweis.

Der Beweis ist z.B. in Johnson et al. [20, S. 312-318] zu finden. Mit $P(\Omega = 0) = \omega$ und der Unabhängigkeit der Auswahl- und der Zählvariablen gilt

$$\begin{aligned}
P(X = x) &= P(\Omega = 0) (P(X^* = x)[1 - \mathbb{1}_{\{x=0\}}(0)] + \mathbb{1}_{\{x=0\}} \mathbb{1}_{\{x=0\}}(0)) \\
&\quad + P(\Omega \neq 0) (P(X^* = x)[1 - \mathbb{1}_{\{x=0\}}(1)] + \mathbb{1}_{\{x=0\}} \mathbb{1}_{\{x=0\}}(1)) \\
&= P(\Omega = 0) (P(X^* = x)[1 - 1] + \mathbb{1}_{\{x=0\}} 1) \\
&\quad + P(\Omega \neq 0) (P(X^* = x)[1 - 0] + \mathbb{1}_{\{x=0\}} 0) \\
&= P(\Omega = 0) \mathbb{1}_{\{x=0\}} + P(\Omega \neq 0) P(X^* = x) \\
&= \omega \mathbb{1}_{\{x=0\}} + (1 - \omega) P(X^* = x).
\end{aligned}$$

□

Dieser Ansatz wurde für die Poissonverteilung schon von Lambert [24, S. 3] genutzt. Erwartungswert und Varianz können in Abhängigkeit des Erwartungswertes und der Varianz der zugrundeliegenden Verteilung angegeben werden.

Satz 1.4 (Erwartungswert und Varianz von ZI-Verteilungen).

Die ZI-Zufallsvariable X folge einer Verteilung mit Wahrscheinlichkeitsfunktion wie in Satz 1.3. ω sei die Eintrittswahrscheinlichkeit des Ereignisses $\Omega = 0$. Dann gilt:

$$\begin{aligned}
E(X) &= (1 - \omega) E(X^*) \\
\text{Var}(X) &= (1 - \omega) \text{Var}(X^*) + \omega(1 - \omega) [E(X^*)]^2
\end{aligned}$$

Beweis.

Da X^* und Ω unabhängig sind, gilt für den Erwartungswert von X :

$$\begin{aligned}
E(X) &= E_{\Omega}[E(\Omega X^* | \Omega)] = E_{\Omega}[\Omega E(X^* | \Omega)] \\
&= E_{\Omega}[\Omega E(X^*)] = E_{\Omega}(\Omega) E_{\Omega}[E(X^*)] \\
&= (1 - \omega) E(X^*).
\end{aligned}$$

Bildet man nun die Varianz von Ω , so erhält man

$$\begin{aligned}
E(\Omega) &= 0P(\Omega = 0) + 1P(\Omega = 1) \\
&= 1 - \omega
\end{aligned} \tag{1.4}$$

$$\begin{aligned}
E(\Omega^2) &= 0^2 P(\Omega = 0) + 1^2 P(\Omega = 1) \\
&= 1 - \omega
\end{aligned} \tag{1.5}$$

$$\begin{aligned}
\Rightarrow \text{Var}(\Omega) &= E(\Omega^2) - [E(\Omega)]^2 = (1 - \omega) - (1 - \omega)^2 \\
&= 1 - \omega - 1 + 2\omega - \omega^2 \\
&= \omega(1 - \omega).
\end{aligned} \tag{1.6}$$

Mit Ausdruck (1.6) läßt sich die Varianz von X für unabhängige X^* und Ω bestimmen. Dazu benutzt man, daß für Zufallsvariable X und Y die Identität

$$\text{Var}(X) = \text{Var}(E(X|Y)) + E(\text{Var}(X|Y)) \tag{1.7}$$

gilt (s. z.B. Evans und Rosenthal [9, S. 172]). Dann ist

$$\begin{aligned}
 \text{Var}(X) = \text{Var}(\Omega X^*) &\stackrel{(1.7)}{=} E_{\Omega}[\text{Var}(\Omega X^*|\Omega)] + \text{Var}_{\Omega}[E(\Omega X^*|\Omega)] \\
 &= E_{\Omega}[\Omega^2 \text{Var}(X^*|\Omega)] + \text{Var}_{\Omega}[E(X^*|\Omega)] \\
 &= E_{\Omega}(\Omega^2) \text{Var}(X^*) + [E(X^*)]^2 \text{Var}_{\Omega}(\Omega) \\
 &\stackrel{(1.5), (1.6)}{=} (1 - \omega) \text{Var}(X^*) + \omega(1 - \omega)[E(X^*)]^2.
 \end{aligned}$$

□

Nullen können von Überdispersion oder einem Nullenüberschuß herrühren. Daher kommen Ridout et al. [31, S. 219] zu dem Schluß, daß Zero Inflated - oder Überdispersions - Modelle allein oft keine zufriedenstellende Erklärung der Daten, insbesondere der Nullen, erlauben. Daher soll nun ein Mixture-Ansatz aus Verallgemeinerter Poissonverteilung zusammen mit einer Variablen zur Modellierung des Nullenüberschuß benutzt werden. Die so entstehende Verteilung wird als 'Zero Inflated Generalized Poisson' - Verteilung ('ZIGP') bezeichnet.

Definition 1.5 (Zero Inflated Verallgemeinerte Poissonverteilung (ZIGP)).

Sei X eine Zufallsvariable mit diskreter Verteilung, die auf \mathbb{N} definiert ist. Ferner sei $P(X = x | \mu, \varphi, \omega)$ die Wahrscheinlichkeit, daß X den nicht-negativen Wert x annimmt. X ist Zero Inflated Verallgemeinert Poisson-verteilt (ZIGP-verteilt) mit Parametern μ , φ und ω , falls

$$\begin{aligned}
 P(X = x | \mu, \varphi, \omega) &= \begin{cases} \omega + (1 - \omega) p^*(0 | \mu, \varphi) & x = 0 \\ (1 - \omega) p^*(x | \mu, \varphi) & x = 1, 2, \dots \end{cases} \\
 &= \mathbb{1}_{\{x=0\}} [\omega + (1 - \omega) e^{-\frac{\mu}{\varphi}}] \\
 &\quad + \mathbb{1}_{\{x>0\}} [(1 - \omega) \frac{\mu(\mu + (\varphi - 1)x)^{x-1}}{x!} \varphi^{-x} e^{-\frac{1}{\varphi}(\mu + (\varphi - 1)x)}]
 \end{aligned}$$

Dabei gelten dieselben Einschränkungen an μ und φ wie in Def. 1.1. Man schreibt $X \sim \text{ZIGP}(\mu, \varphi, \omega)$.

Mit Hilfe von Satz 1.4 lassen sich Erwartungswert und Varianz bestimmen.

Satz 1.6 (Erwartungswert und Varianz der ZIGP-Verteilung).

Sei $X^* \sim \text{GP}(\mu, \varphi)$ generalisiert Poisson verteilt. Für Zufallsvariable, die zero-inflated generalisiert Poisson verteilt sind, gilt dann:

$$\begin{aligned}
 X \sim \text{ZIGP}(\mu, \varphi, \omega) &\Rightarrow E(X) = (1 - \omega)E(X^*) = (1 - \omega)\mu \\
 \text{Var}(X) &= (1 - \omega)\text{Var}(X^*) + \omega(1 - \omega)[E(X^*)]^2 \\
 &= (1 - \omega)\varphi^2\mu + \omega(1 - \omega)\mu^2 \\
 &= (1 - \omega)\mu \cdot [\varphi^2 + \mu\omega] = E(X) \cdot [\varphi^2 + \mu\omega]
 \end{aligned}$$

Beweis. Man benutze Definition 1.1 und Satz 1.4. □

Aus der letzten Darstellung der Varianz kann man sehen, daß ein Nullenüberschuß zu Überdispersion führt: die Varianz erhöht sich, falls $\omega > 0$.

1.1.2 Erzeugung von Zufallsvariablen

Nun möchte man Zufallsvariablen erzeugen, die einer $ZIGP(\mu, \varphi, \omega)$ -Verteilung folgen. Statistikpakete wie "R" oder "S-Plus" erlauben lediglich die Erzeugung von (Pseudo-) Zufallszahlen der gängigsten Verteilungen, nicht jedoch für $ZIGP$.

Einige Möglichkeiten zur Erzeugung von Zufallsvariablen beliebiger Verteilung werden bei Famoye [12] vorgestellt, hier soll aber nur die Invertierungsmethode (s. Famoye [12, S. 222]) genutzt werden. Diese basiert auf einem nicht verteilungsspezifischen Verfahren für stetige Zufallsvariablen, der Wahrscheinlichkeitsintegraltransformation (engl. "probability integral transformation"), s. z.B. Kendall und Buckland [21, S. 227]. Sie wurde jedoch auf den diskreten Fall übertragen. Benötigt wird lediglich ein Zufallszahlengenerator, der gleichverteilte Zufallsvariablen erzeugt.

Satz 1.7 (Invertierungsmethode).

Sei $U \sim Unif(0, 1)$ gleichverteilte Zufallsvariable auf $[0, 1]$ und $F(t)$ die Verteilungsfunktion von $ZIGP(\mu, \varphi, \omega)$. Dann ist x Realisierung einer Zufallsvariablen mit $X \sim ZIGP(\mu, \varphi, \omega)$, sofern sie die Ungleichung

$$F(x - 1) < U \leq F(x) \quad (1.8)$$

erfüllt.

Beweis. Die Verteilungsfunktion $F(X)$ von $ZIGP(\mu, \varphi, \omega)$ hat die Form

$$F(X) = P(X \leq x) = \sum_{i=0}^x P_i(\mu, \varphi).$$

Für $U \sim Unif(0, 1)$ gilt

$$\begin{aligned} P(F(x - 1) < U \leq F(x)) &= \int_{F(x-1)}^{F(x)} 1 \, du \\ &= F(x) - F(x - 1) \\ &= P_x(\mu, \varphi). \end{aligned}$$

Damit ist x Realisierung einer $ZIGP$ -verteilten Zufallsvariablen X , wenn sie Ungleichung (1.8) erfüllt. \square

Für die Invertierungsmethode muß also eine Zufallsvariable $U \sim Unif(0, 1)$ erzeugt werden und dann ein geeignetes x gefunden werden, das (1.8) erfüllt. Dazu wird die linke Ungleichung sequentiell für aufsteigende i – beginnend mit $i = 0$ – überprüft, bis sie verletzt wird. Das erste i , das die linke Ungleichung verletzt, erfüllt jedoch noch die rechte Ungleichung, und entspricht daher dem gewünschten x .

Famoye [12] schlägt vor, zur Überprüfung der Ungleichung eine rekurrente Beziehung zwischen den Eintrittswahrscheinlichkeiten der Verallgemeinerten Poissonverteilung zu verwenden. Allerdings verwendet er dort die ursprünglich von Consul und Jain [6] eingeführte Parametrisierung der Verteilung; ein Nullenüberschuß wird in dieser Verteilung

ebenfalls nicht berücksichtigt. In Anlehnung an diese Idee wird $P(X = x)$ aber durch einen Korrekturterm aus $P(X = x - 1)$ berechnet:

$$\frac{P(X = x)}{P(X = x - 1)} = \frac{(\mu + (\varphi - 1)x)^{x-1}}{(\mu + (\varphi - 1)(x - 1))^{x-2}} \frac{1}{x \varphi} \exp\left(\frac{1 - \varphi}{\varphi}\right)$$

Dies ist jedoch erst ab $x = 1$ möglich, da die ZIGP-Wahrscheinlichkeitsfunktion an der Stelle $x = 0$ eine andere Gestalt als für $x \geq 1$ hat (s. Definition 1.5). Daher wird ein Zwischenschritt für $P_{ZIGP}(X = 1)$ nötig. Der Algorithmus ist als Pseudo-Code verfaßt. Hochgestellte Indizes wie z.B. in $x^{(j)}$ stehen für den Iterationsschritt j . Wahrscheinlichkeiten $P_{ZIGP}(X = x)$ der $ZIGP(\mu_i, \varphi, \omega_i)$ -Verteilung werden als Werte $p^{(j)}$ gespeichert. Die Verteilungsfunktion dieser diskreten Verteilung ist als $P(X \leq x) = \sum_{z=0}^x P_{ZIGP}(Z = z)$ gegeben. Daher werden die Punkt-Wahrscheinlichkeiten $p^{(i)}$ zur Wahrscheinlichkeitsfunktion $s^{(j)} = \sum_{i=0}^j p^{(i)}$ aufsummiert.

Algorithmus:

(1) Initialisierung:

(a) $x^{(0)} \leftarrow 0$

(b) $p^{(0)} \leftarrow P_{ZIGP}(X = x^{(0)})$

(c) $s^{(0)} \leftarrow p^{(0)}$

(d) Erzeuge eine Realisation u von $U \sim Unif(0, 1)$

(2) Zwischenschritt für $P_{ZIGP}(X = 1)$:

(a) Wenn $u > s^{(0)}$, führe 2 (b)-(d) aus. Sonst: gehe zu (4)!

(b) $x^{(1)} \leftarrow 1$

(c) $p^{(1)} \leftarrow P_{ZIGP}(X = x^{(1)})$

(d) $s^{(1)} \leftarrow s^{(0)} + p^{(1)}$

(3) Iteration:

(a) Für $j = 2, 3, \dots$ gilt: Solange $u > s^{(j-1)}$, wiederhole 3 (b)-(d)!

(b) $x^{(j)} \leftarrow x^{(j-1)} + 1$

(c) $p^{(j)} \leftarrow \frac{(\mu + (\varphi - 1)x^{(j)})^{x^{(j)} - 1}}{(\mu + (\varphi - 1)(x^{(j)} - 1))^{x^{(j)} - 2}} \frac{1}{x^{(j)} \varphi} \exp\left(\frac{1 - \varphi}{\varphi}\right) \cdot p^{(j-1)}$

(d) $s^{(j)} \leftarrow s^{(j-1)} + p^{(j)}$

(4) Ausgabe: $x^{(k)}$ (das ist das erste k , das $u > s^{(k-1)}$ verletzt)

Die Software dazu ist in Anhang A abgedruckt. Famoye [12, S. 227] vergleicht die Invertierungsmethode mit zwei weiteren Methoden: der "Branching" Methode und dem "Rejection" Algorithmus. Das Bild ist jedoch uneinheitlich: die benötigte Zeit zur Erzeugung von 1000 Zufallsvariablen hängt stark von der Parametrisierung ab. Je nach Wahl

der Methode ergeben sich bei Variation des Dispersionsparameters sogar gegenläufige Ergebnisse beim Zeitaufwand. Ein Algorithmus, der stets die geringste Zeit benötigt, ist somit nicht auszumachen.

Die Geschwindigkeit zur Erzeugung auch großer Datenvektoren war mit der in diesem Abschnitt vorgestellten Invertierungsmethode jedoch für alle Simulationsläufe sehr hoch.

1.2 Konstruktion eines $ZIGP(\mu_i, \varphi, \omega_i)$ – Regressionsmodells

Die $ZIGP(\mu, \varphi, \omega)$ -Verteilung möchte man nun zur Konstruktion eines Regressionsmodells nutzen. Man hat eine bestimmte Menge von Beobachtungen Y_i , $i = 1, \dots, n$, dies können z.B. die Anzahl der Kfz-Schäden sein, die ein Haftpflichtversicherer für sein Policenportfolio $i = 1, \dots, n$ beobachtet. Erklärt werden sollen diese in einem Regressionsmodell durch Einflußgrößen x_{i1}, \dots, x_{ik} , z.B. sei x_{i1} das Geschlecht des Fahrers bzw. der Fahrerin in Police i ; x_{i2} die Anzahl der "schadenfreien" Versicherungsjahre.

Möchte man ein Regressionsmodell anwenden, so wird $\mathbf{Y} = (Y_1, \dots, Y_n)^t$ als Zielvariable (abhängige Variable, Response) und $\mathbf{X}_j = (x_{1j}, \dots, x_{nj})^t$, $j = 1, \dots, k$ als j-te Kovariable (unabhängige Variable) verwendet. Diese werden zur Designmatrix (auch Modellmatrix) zusammengefaßt:

$$\mathbf{X} = \begin{pmatrix} x_{11} & \dots & x_{1k} \\ \vdots & & \vdots \\ x_{n1} & \dots & x_{nk} \end{pmatrix} \in \mathbb{R}^{n,k}$$

Es wird vorausgesetzt, daß \mathbf{X} vollen Spaltenrang habe.

Anmerkung: Mit "groß geschriebenen" \mathbf{X}_j sei im Folgenden stets die Kovariable j (also die j -te Spalte der Designmatrix), mit "klein geschriebenen" \mathbf{x}_i^t die i -te Zeile der Designmatrix bezeichnet (also der Vektor der erklärenden Variablen für Beobachtung i).

Jede der Kovariablen wird mit einem Regressionsparameter β_j multipliziert, um den Response möglichst gut über eine Linearkombination der Kovariablen zu erklären. Als linearen Prädiktor bezeichnet man $\boldsymbol{\eta} = \mathbf{X} \boldsymbol{\beta}$. Die Auswahl geeigneter Kovariablen und die Bestimmung eines geeigneten Regressionsparameters $\boldsymbol{\beta}$ ist die Problemstellung, die man mit einem Regressionsmodells bewältigen möchte.

Das Regressionsmodell wird im vorliegenden Fall ähnlich einem generalisierten linearen Modell (GLM) konstruiert. GLM's werden z.B. bei McCullagh und Nelder [26] oder Pruscha [30, S. 255f] beschrieben. Dieses besteht aus einer Zufallskomponente, einer Systematischen Komponente und einer Verknüpfungskomponente. Die Verknüpfungskomponente besteht aus einer Responsefunktion h , die den linearen Prädiktor $\boldsymbol{\eta}$ mit dem Erwartungswert $\boldsymbol{\mu}$ verknüpft, also die systematische sinnvoll mit der stochastischen Komponente verbindet: $h(\boldsymbol{\eta}) = \boldsymbol{\mu}$. Es wird vorausgesetzt, daß h differenzierbar und streng monoton ist. Dann existiert eine Umkehrfunktion $g(\boldsymbol{\mu}) := h^{-1}(\boldsymbol{\mu}) = \boldsymbol{\eta}$; g wird als Linkfunktion bezeichnet.

Der Unterschied zu GLM's besteht allerdings darin, daß die angenommene $ZIGP$ -Verteilung der Responseedaten hier nicht zur exponentiellen Familie gehört (z.B. Pruscha [30, S. 23ff]). Stekeler [35, S. 66] zeigt dies für den Spezialfall der GP-Verteilung. Da Regression auf mehreren Parametern zugelassen wird, gibt es auch mehrere lineare Prädiktoren, also mehrere Designmatrizen und Regressionsparameter. Es wird nicht mehr ausschließlich auf den Erwartungswert, sondern auch auf andere Verteilungsparameter regressiert.

In diesem Kapitel soll Regression sowohl auf Erwartungswert als auch auf den Auswahlparameter zugelassen werden, der Dispersionsparameter φ jedoch noch für alle Beobachtungen konstant geschätzt werden. Die notwendigen Regressionsparameter und φ werden simultan geschätzt.

Für abweichende Beobachtungszeiträume wird eine Exposure-Variable E_i (auch "Offset") eingeführt. Dies entspricht im Beispiel der Anzahl von Versicherungsschäden der Police i der Anzahl der Versicherungsjahre; im Fall ohne individuelle Exposure gilt $E_i = 1 \forall i$.

Zero Inflated GP – Regressionsmodell $ZIGP(\mu_i, \varphi, \omega_i)$

1. *Zufallskomponente:*

$\{Y_i, 1 \leq i \leq n\}$ sind unabhängig mit $Y_i \sim ZIGP(\mu_i, \varphi, \omega_i)$.

2. *Systematische Komponente:*

Die linearen Prädiktoren $\eta_i^\mu(\boldsymbol{\beta}) = \mathbf{x}_i^t \boldsymbol{\beta}$ und $\eta_i^\omega(\boldsymbol{\gamma}) = \mathbf{z}_i^t \boldsymbol{\gamma}$, $i = 1, \dots, n$ beeinflussen den Response Y_i . Dabei sind $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_p)^t$ und $\boldsymbol{\gamma} = (\gamma_0, \gamma_1, \dots, \gamma_q)^t$ unbekannte Regressionsparameter. Die Matrizen $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^t$ und $\mathbf{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_n)^t$ werden als Designmatrizen bezeichnet.

3. *Parametrische Linkfunktionen:*

Die linearen Prädiktoren $\eta_i^\mu(\boldsymbol{\beta})$ und $\eta_i^\omega(\boldsymbol{\gamma})$ werden mit Erwartungswert $\mu_i(\boldsymbol{\beta})$ und Auswahlparameter $\omega_i(\boldsymbol{\gamma})$, $i = 1, \dots, n$ wie folgt verknüpft. Der Dispersionsparameter φ wird als konstant angekommen. Ferner seien E_i Exposurevariablen mit $E_i > 0 \forall i$.

(i) *Erwartungswertebene*

$$\begin{aligned} E(Y_i | \boldsymbol{\beta}) = \mu_i(\boldsymbol{\beta}) &:= E_i e^{\mathbf{x}_i^t \boldsymbol{\beta}} = e^{\mathbf{x}_i^t \boldsymbol{\beta} + \log(E_i)} > 0 \\ \Leftrightarrow \eta_i^\mu(\boldsymbol{\beta}) &= \log(\mu_i(\boldsymbol{\beta})) - \log(E_i) \quad (\text{Log-Link}), \end{aligned} \quad (1.9)$$

(ii) *Überdispersionsebene*

$$\varphi \quad \text{konstant mit } \varphi > 1, \quad (1.10)$$

(iii) *Nullenüberschußebene*

$$\begin{aligned} \omega_i(\boldsymbol{\gamma}) &:= \frac{e^{\mathbf{z}_i^t \boldsymbol{\gamma}}}{1 + e^{\mathbf{z}_i^t \boldsymbol{\gamma}}} \in (0, 1) \\ \Leftrightarrow \eta_i^\omega(\boldsymbol{\gamma}) &= \log\left(\frac{\omega_i(\boldsymbol{\gamma})}{1 - \omega_i(\boldsymbol{\gamma})}\right) \quad (\text{Logit-Link}). \end{aligned} \quad (1.11)$$

Um die Berechnungen übersichtlicher zu gestalten, werden einige Notationen eingeführt:

$$\boldsymbol{\delta} := (\boldsymbol{\beta}^t, \varphi, \boldsymbol{\gamma}^t)^t,$$

$$\begin{aligned}\mu_i(\boldsymbol{\beta}) &:= e^{\mathbf{x}_i^t \boldsymbol{\beta} + \log(E_i)}, \\ k_i(\boldsymbol{\gamma}) &:= e^{\mathbf{z}_i^t \boldsymbol{\gamma}}, \\ \omega_i(\boldsymbol{\gamma}) &:= \frac{k_i(\boldsymbol{\gamma})}{1 + k_i(\boldsymbol{\gamma})}\end{aligned}$$

und

$$P_i^0(\boldsymbol{\delta}) := \exp\left(-\frac{1}{\varphi} e^{\mathbf{x}_i^t \boldsymbol{\beta} + \log(E_i)}\right).$$

Mit Formel (1.1) gilt dann für eine GP-verteilte Zufallsvariable $Y^* \sim GP(\mu_i(\boldsymbol{\beta}), \varphi)$

$$P(Y^* = 0 | \mu_i(\boldsymbol{\beta}), \varphi) = e^{-\frac{\mu_i(\boldsymbol{\beta})}{\varphi}}$$

und damit

$$\begin{aligned}P(Y_i = y_i) &= \begin{cases} \omega_i(\boldsymbol{\gamma}) + (1 - \omega_i(\boldsymbol{\gamma}))e^{-\frac{\mu_i(\boldsymbol{\beta})}{\varphi}} & \text{falls } y_i = 0 \\ (1 - \omega_i(\boldsymbol{\gamma}))P(y_i | \mu_i(\boldsymbol{\beta}), \varphi) & y_i = 1, 2, \dots \end{cases} \\ &= \mathbb{1}_{\{y_i=0\}}[\omega_i(\boldsymbol{\gamma}) + (1 - \omega_i(\boldsymbol{\gamma}))e^{-\frac{\mu_i(\boldsymbol{\beta})}{\varphi}}] \\ &\quad + \mathbb{1}_{\{y_i>0\}}[(1 - \omega_i(\boldsymbol{\gamma}))\frac{\mu_i(\boldsymbol{\beta})(\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i)^{y_i-1}}{y_i!}\varphi^{-y_i}e^{-\frac{1}{\varphi}(\mu_i(\boldsymbol{\beta}) + (\varphi-1)y_i)}].\end{aligned}$$

1.2.1 ML-Schätzer der Regressionsparameter $\boldsymbol{\delta} = (\boldsymbol{\beta}^t, \boldsymbol{\alpha}^t, \boldsymbol{\gamma}^t)^t$

Die Regressionsparameter $\boldsymbol{\delta}$ werden mit einem Maximum-Likelihood-Ansatz geschätzt. Die Likelihoodfunktion betrachtet die gemeinsame Wahrscheinlichkeitsfunktion $P(\mathbf{Y} | \boldsymbol{\delta})$ nicht als Funktion der Zielvariablen \mathbf{Y} , sondern als Funktion der Parameter $\boldsymbol{\delta}$. Sie ist demnach definiert als

$$L(\boldsymbol{\delta} | \mathbf{Y} = \mathbf{y}) := P(\mathbf{Y} | \boldsymbol{\delta}). \quad (1.12)$$

$$l(\boldsymbol{\delta} | \mathbf{Y} = \mathbf{y}) := \log L(\boldsymbol{\delta} | \mathbf{Y} = \mathbf{y}) \quad (1.13)$$

wird als Loglikelihoodfunktion bezeichnet.

Der Maximum-Likelihood-Schätzer (kurz "ML-Schätzer" oder "MLE") ist derjenige Parameterwert $\hat{\boldsymbol{\delta}}$, der für die Erzeugung der beobachteten Daten \mathbf{Y} am wahrscheinlichsten unter allen zulässigen Parametrisierungen ist. Aufgrund der Unabhängigkeitsannahme für Y_i , $i = 1, \dots, n$ kann man ihn darstellen als

$$\begin{aligned}\hat{\boldsymbol{\delta}} &= \operatorname{argmax}_{\boldsymbol{\delta}} L(\boldsymbol{\delta} | \mathbf{Y} = \mathbf{y}) \\ &= \operatorname{argmax}_{\boldsymbol{\delta}} \prod_{i=1}^n P(Y_i = y_i | \boldsymbol{\delta}) \\ &= \operatorname{argmax}_{\boldsymbol{\delta}} \log L(\boldsymbol{\delta} | \mathbf{Y} = \mathbf{y}) = \operatorname{argmax}_{\boldsymbol{\delta}} l(\boldsymbol{\delta} | \mathbf{Y} = \mathbf{y}) \\ &= \operatorname{argmax}_{\boldsymbol{\delta}} \sum_{i=1}^n \log P(Y_i = y_i | \boldsymbol{\delta}).\end{aligned}$$

Dazu leitet man die Loglikelihoodfunktion partiell nach $\boldsymbol{\delta} = (\boldsymbol{\beta}^t, \varphi, \boldsymbol{\gamma}^t)^t$ ab und setzt die Ableitungen gleich 0. Man erhält die Scoregleichungen $s_n(\boldsymbol{\delta}_n)$; der Maximum Likelihood-Schätzer ist eine Lösung der Scoregleichungen. Diese müssen aber keine Lösung haben und müssen auch nicht eindeutig sein. Die Scoregleichungen können i.d.R. nur numerisch gelöst werden.

1.2.2 Loglikelihoodfunktion der $ZIGP(\mu_i, \varphi, \omega_i)$ –Regression

Da man stets von Beobachtungen $\mathbf{Y} = \mathbf{y}$ ausgeht, wird im Folgenden für die Loglikelihoodfunktion lediglich kurz $l(\boldsymbol{\delta})$ geschrieben. Die Loglikelihoodfunktion aus nach Formel (1.13) wird nun für das $ZIGP(\mu_i, \varphi, \omega_i)$ –Regressionsmodell explizit bestimmt.

Satz 1.8 (Loglikelihoodfunktion).

Die Loglikelihoodfunktion des oben beschriebenen $ZIGP(\mu_i, \varphi, \omega_i)$ –Modelles lautet

$$\begin{aligned} l(\boldsymbol{\delta}) &= \sum_{i=1}^n \mathbb{1}_{\{y_i=0\}} [\log(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta})) - \log(1 + k_i(\boldsymbol{\gamma}))] \\ &\quad + \mathbb{1}_{\{y_i>0\}} \left[-\log(1 + k_i(\boldsymbol{\gamma})) + \log(E_i) + \mathbf{x}_i^t \boldsymbol{\beta} + (y_i - 1) \log(\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i) \right. \\ &\quad \left. - \log(y_i!) - y_i \log \varphi - \frac{1}{\varphi}(\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i) \right] \end{aligned}$$

Beweis.

Für Beobachtungen $\mathbf{y} = (y_1, \dots, y_n)^t$ ergibt sich die Likelihoodfunktion zu $L(\boldsymbol{\delta}) = L(\boldsymbol{\delta} \mid \mathbf{y}) = L(\mathbf{Y} = \mathbf{y} \mid \boldsymbol{\delta}) = \prod_{i=1}^n P(Y_i = y_i \mid \boldsymbol{\delta})$.

Und somit ist der Loglikelihood

$$\begin{aligned} l(\boldsymbol{\delta}) &= \log L(\boldsymbol{\delta}) = \sum_{i=1}^n \mathbb{1}_{\{y_i=0\}} \log [\omega_i(\boldsymbol{\gamma}) + (1 - \omega_i(\boldsymbol{\gamma}))P_i^0(\boldsymbol{\delta})] \\ &\quad + \mathbb{1}_{\{y_i>0\}} \left[\log(1 - \omega_i(\boldsymbol{\gamma})) + \log \mu_i(\boldsymbol{\beta}) + (y_i - 1) \log(\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i) \right. \\ &\quad \left. - \log(y_i!) - y_i \log \varphi - \frac{1}{\varphi}(\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i) \right] \\ &= \sum_{i=1}^n \mathbb{1}_{\{y_i=0\}} \log \left[\frac{k_i(\boldsymbol{\gamma})}{1 + k_i(\boldsymbol{\gamma})} + \frac{1}{1 + k_i(\boldsymbol{\gamma})} P_i^0(\boldsymbol{\delta}) \right] \\ &\quad + \mathbb{1}_{\{y_i>0\}} \left[\log \left(\frac{1}{1 + k_i(\boldsymbol{\gamma})} \right) + \log(E_i) + \mathbf{x}_i^t \boldsymbol{\beta} + (y_i - 1) \log(\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i) \right. \\ &\quad \left. - \log(y_i!) - y_i \log \varphi - \frac{1}{\varphi}(\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i) \right] \\ &= \sum_{i=1}^n \mathbb{1}_{\{y_i=0\}} [\log(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta})) - \log(1 + k_i(\boldsymbol{\gamma}))] \end{aligned}$$

$$\begin{aligned}
& + \mathbb{1}_{\{y_i > 0\}} \left[-\log(1 + k_i(\gamma)) + \log(E_i) + \mathbf{x}_i^t \boldsymbol{\beta} + (y_i - 1) \log(\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i) \right. \\
& \left. - \log(y_i!) - y_i \log \varphi - \frac{1}{\varphi}(\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i) \right].
\end{aligned}$$

□

1.2.3 Score-Vektor der ZIGP(μ_i, φ, ω_i)-Regression

Der Score-Vektor werde mit $\mathbf{s}_n(\boldsymbol{\delta}) := \left(\frac{\partial l(\boldsymbol{\delta})}{\partial \boldsymbol{\beta}}, \frac{\partial l(\boldsymbol{\delta})}{\partial \varphi}, \frac{\partial l(\boldsymbol{\delta})}{\partial \boldsymbol{\gamma}} \right)^t$ bezeichnet, stellt also den Gradienten der Log Likelihood - Funktion bezüglich $\boldsymbol{\delta}$ dar. Zur Berechnung des Maximum Likelihood - Schätzers in der Software ist eine explizite Angabe des Score-Vektors von Vorteil, da die Konvergenzgeschwindigkeit der verwendeten Optimierungsroutine deutlich erhöht wird.

Satz 1.9 (Score-Vektor).

Der Score-Vektor des oben beschriebenen Modells lautet

$$\begin{aligned}
\frac{\partial}{\partial \beta_j} l(\boldsymbol{\delta}) &= \sum_{i=1}^n x_{ij} \left(\mathbb{1}_{\{y_i=0\}} \left[\frac{-\frac{1}{\varphi} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta})}{k_i(\gamma) + P_i^0(\boldsymbol{\delta})} \right] \right. \\
&\quad \left. + \mathbb{1}_{\{y_i>0\}} \left[1 + (y_i - 1) \frac{\mu_i(\boldsymbol{\beta})}{\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i} - \frac{1}{\varphi} \mu_i(\boldsymbol{\beta}) \right] \right), \\
j &= 0, \dots, p,
\end{aligned} \tag{1.14}$$

$$\begin{aligned}
\frac{\partial}{\partial \varphi} l(\boldsymbol{\delta}) &= \sum_{i=1}^n \mathbb{1}_{\{y_i=0\}} \left[\frac{\frac{1}{\varphi^2} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta})}{k_i(\gamma) + P_i^0(\boldsymbol{\delta})} \right] \\
&\quad + \mathbb{1}_{\{y_i>0\}} \left[\frac{(y_i - 1)y_i}{\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i} - \frac{y_i}{\varphi} + \frac{\mu_i(\boldsymbol{\beta}) - y_i}{\varphi^2} \right],
\end{aligned} \tag{1.15}$$

$$\begin{aligned}
\frac{\partial}{\partial \gamma_j} l(\boldsymbol{\delta}) &= \sum_{i=1}^n z_{ij} k_i(\gamma) \left(\mathbb{1}_{\{y_i=0\}} \left[\frac{1}{k_i(\gamma) + P_i^0(\boldsymbol{\delta})} \right] - \frac{1}{1 + k_i(\gamma)} \right), \\
j &= 0, \dots, q.
\end{aligned} \tag{1.16}$$

Beweis.

Man leitet die Loglikelihood Funktion nach $\boldsymbol{\beta}$, φ und $\boldsymbol{\gamma}$ ab.

$$\begin{aligned}
\frac{\partial}{\partial \beta_j} l(\boldsymbol{\delta}) &= \sum_{i=1}^n \mathbb{1}_{\{y_i=0\}} \left[\frac{P_i^0(\boldsymbol{\delta}) \left(-\frac{1}{\varphi}\right) \mu_i(\boldsymbol{\beta}) x_{ij}}{k_i(\gamma) + P_i^0(\boldsymbol{\delta})} \right] \\
&\quad + \mathbb{1}_{\{y_i>0\}} \left[x_{ij} + (y_i - 1) \frac{\mu_i(\boldsymbol{\beta}) x_{ij}}{\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i} - \frac{1}{\varphi} \mu_i(\boldsymbol{\beta}) x_{ij} \right] \\
&= \sum_{i=1}^n x_{ij} \left(\mathbb{1}_{\{y_i=0\}} \left[\frac{-\frac{1}{\varphi} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta})}{k_i(\gamma) + P_i^0(\boldsymbol{\delta})} \right] \right. \\
&\quad \left. + \mathbb{1}_{\{y_i>0\}} \left[1 + (y_i - 1) \frac{\mu_i(\boldsymbol{\beta})}{\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i} - \frac{1}{\varphi} \mu_i(\boldsymbol{\beta}) \right] \right),
\end{aligned}$$

$$\begin{aligned}
j &= 0, \dots, p, \\
\frac{\partial}{\partial \varphi} l(\boldsymbol{\delta}) &= \sum_{i=1}^n \mathbb{1}_{\{y_i=0\}} \left[\frac{P_i^0(\boldsymbol{\delta}) \frac{1}{\varphi^2} \mu_i(\boldsymbol{\beta})}{k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta})} \right] \\
&\quad + \mathbb{1}_{\{y_i>0\}} \left[\frac{(y_i - 1)y_i}{\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i} - \frac{y_i}{\varphi} + \frac{1}{\varphi^2} \mu_i(\boldsymbol{\beta}) - \frac{y_i}{\varphi^2} \right] \\
&= \sum_{i=1}^n \mathbb{1}_{\{y_i=0\}} \left[\frac{\frac{1}{\varphi^2} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta})}{k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta})} \right] \\
&\quad + \mathbb{1}_{\{y_i>0\}} \left[\frac{(y_i - 1)y_i}{\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i} - \frac{y_i}{\varphi} + \frac{\mu_i(\boldsymbol{\beta}) - y_i}{\varphi^2} \right], \\
\frac{\partial}{\partial \gamma_j} l(\boldsymbol{\delta}) &= \sum_{i=1}^n \mathbb{1}_{\{y_i=0\}} \left[\frac{k_i(\boldsymbol{\gamma}) z_{ij}}{k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta})} - \frac{k_i(\boldsymbol{\gamma}) z_{ij}}{1 + k_i(\boldsymbol{\gamma})} \right] - \mathbb{1}_{\{y_i>0\}} \left[\frac{k_i(\boldsymbol{\gamma}) z_{ij}}{1 + k_i(\boldsymbol{\gamma})} \right] \\
&= \sum_{i=1}^n z_{ij} k_i(\boldsymbol{\gamma}) \left(\mathbb{1}_{\{y_i=0\}} \left[\frac{1}{k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta})} \right] - \frac{1}{1 + k_i(\boldsymbol{\gamma})} \right), \\
j &= 0, \dots, q.
\end{aligned}$$

□

1.2.4 Hessematrix der ZIGP(μ_i, φ, ω_i)–Regression

Um Aussagen über Varianz und Asymptotik der ML-Schätzer treffen zu können, interessiert man sich auch für die Hessematrix der Loglikelihoodfunktion. Die Hessematrix

$$\mathcal{H}(\boldsymbol{\delta}) = \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\delta} \partial \boldsymbol{\delta}^t}$$

wird auch als beobachtete Fisher-Informationsmatrix \mathbf{I}_{obs} bezeichnet. Ist sie an der Stelle $\hat{\boldsymbol{\delta}}$ negativ definit, so liegt eine lokale Maximumstelle der Log Likelihood - Funktion vor. Ist sie für alle Parameterwerte $\boldsymbol{\delta}$ negativ definit, so ist $\hat{\boldsymbol{\delta}}$ sogar eindeutiger Maximum-Likelihood - Schätzer.

In generalisierten linearen Modellen mit kanonischen Linkfunktionen ist die Hessematrix deterministisch und negativ definit (s. Pruscha [30, S. 260]). Im vorliegenden Fall jedoch ergibt sich eine Hessematrix mit zufälligen Einträgen, für die zudem negative Definitheit nicht bewiesen werden kann. Daß die gefundene Lösung tatsächlich ein globales Maximum der Likelihoodgleichung darstellt, kann damit nicht sichergestellt werden.

Die Hessematrix im betrachteten Regressionsmodell kann partitioniert werden als

$$\mathcal{H}(\boldsymbol{\delta}) = \mathbf{I}_{obs} = \begin{pmatrix} \mathcal{H}_{\beta\beta}(\boldsymbol{\delta}) & \mathcal{H}_{\beta\varphi}(\boldsymbol{\delta}) & \mathcal{H}_{\beta\gamma}(\boldsymbol{\delta}) \\ \mathcal{H}_{\varphi\beta}(\boldsymbol{\delta}) & \mathcal{H}_{\varphi\varphi}(\boldsymbol{\delta}) & \mathcal{H}_{\varphi\gamma}(\boldsymbol{\delta}) \\ \mathcal{H}_{\gamma\beta}(\boldsymbol{\delta}) & \mathcal{H}_{\gamma\varphi}(\boldsymbol{\delta}) & \mathcal{H}_{\gamma\gamma}(\boldsymbol{\delta}) \end{pmatrix}$$

und hat die Einträge

$$\mathcal{H}_{\beta\beta}(\boldsymbol{\delta}) = \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^t} \in \mathbb{R}^{p \times p},$$

$$\begin{aligned}
\mathcal{H}_{\varphi\varphi}(\boldsymbol{\delta}) &= \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \varphi^2} \in \mathbb{R}, \\
\mathcal{H}_{\gamma\gamma}(\boldsymbol{\delta}) &= \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \gamma \partial \gamma^t} \in \mathbb{R}^{q \times q}, \\
\mathcal{H}_{\beta\varphi}(\boldsymbol{\delta}) = \mathcal{H}_{\varphi\beta}(\boldsymbol{\delta})^t &= \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \beta^t \partial \varphi} \in \mathbb{R}^{p \times 1}, \\
\mathcal{H}_{\beta\gamma}(\boldsymbol{\delta}) = \mathcal{H}_{\gamma\beta}(\boldsymbol{\delta})^t &= \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \beta \partial \gamma^t} \in \mathbb{R}^{p \times q} \\
\text{und} \\
\mathcal{H}_{\varphi\gamma}(\boldsymbol{\delta}) = \mathcal{H}_{\gamma\varphi}(\boldsymbol{\delta})^t &= \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \varphi \partial \gamma^t} \in \mathbb{R}^{1 \times q}.
\end{aligned}$$

Zur Berechnung der Einträge der Hessematrix wird ein Lemma benötigt.

Lemma 1.10.

Sei $g(x_i, \boldsymbol{\beta}, \varphi) = -\frac{1}{\varphi} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta})$. Dann gilt

$$\frac{\partial}{\partial \beta_m} g(x_i, \boldsymbol{\beta}, \varphi) = x_{im} \frac{\mu_i(\boldsymbol{\beta}) - \varphi}{\varphi^2} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta}).$$

Beweis.

$$\begin{aligned}
\frac{\partial}{\partial \beta_m} g(x_i, \boldsymbol{\beta}, \varphi) &= -\frac{1}{\varphi} \left[P_i^0(\boldsymbol{\delta}) \left(-\frac{1}{\varphi}\right) (\mu_i(\boldsymbol{\beta}))^2 x_{im} + P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta}) x_{im} \right] \\
&= -\frac{1}{\varphi} x_{im} \left[P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta}) \left(1 - \frac{\mu_i(\boldsymbol{\beta})}{\varphi}\right) \right] \\
&= x_{im} \frac{\mu_i(\boldsymbol{\beta}) - \varphi}{\varphi^2} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta})
\end{aligned}$$

□

Damit können nun die partiellen Ableitungen des Score-Vektors berechnet werden.

$$\begin{aligned}
\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \beta_l \partial \beta_m} &= \sum_{i=1}^n x_{il} \left(\mathbb{1}_{\{y_i=0\}} \left[\frac{\frac{\partial}{\partial \beta_m} g(x_i, \boldsymbol{\beta}, \varphi) (k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))^2} \right. \right. \\
&\quad \left. \left. - \frac{g(x_i, \boldsymbol{\beta}, \varphi) P_i^0(\boldsymbol{\delta}) \left(-\frac{1}{\varphi}\right) \mu_i(\boldsymbol{\beta}) x_{im}}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))^2} \right] + \mathbb{1}_{\{y_i>0\}} [(y_i - 1) \right. \\
&\quad \left. \times \frac{\mu_i(\boldsymbol{\beta}) x_{im} (\mu_i(\boldsymbol{\beta}) + (\varphi - 1) y_i) - (\mu_i(\boldsymbol{\beta}))^2 x_{im}}{(\mu_i(\boldsymbol{\beta}) + (\varphi - 1) y_i)^2} - \frac{1}{\varphi} \mu_i(\boldsymbol{\beta}) x_{im} \right] \Bigg) \\
&\stackrel{\text{Lemma 1.10}}{=} \sum_{i=1}^n x_{il} x_{im} \mu_i(\boldsymbol{\beta}) \left(\mathbb{1}_{\{y_i=0\}} \left[\frac{\frac{\mu_i(\boldsymbol{\beta}) - \varphi}{\varphi^2} P_i^0(\boldsymbol{\delta}) (k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))^2} \right. \right.
\end{aligned}$$

$$\begin{aligned}
& + \frac{g(x_i, \beta, \varphi) \frac{1}{\varphi} P_i^0(\delta)}{(k_i(\gamma) + P_i^0(\delta))^2} \Big] + \mathbb{1}_{\{y_i > 0\}} \left[\frac{(\varphi - 1)(y_i - 1)y_i}{(\mu_i(\beta) + (\varphi - 1)y_i)^2} - \frac{1}{\varphi} \right] \Big) \\
= & \sum_{i=1}^n x_{il} x_{im} \mu_i(\beta) \left(\mathbb{1}_{\{y_i = 0\}} \left[\frac{-\frac{1}{\varphi} (P_i^0(\delta))^2 + \frac{\mu_i(\beta) - \varphi}{\varphi^2} P_i^0(\delta) k_i(\gamma)}{(k_i(\gamma) + P_i^0(\delta))^2} \right] \right. \\
& \left. + \mathbb{1}_{\{y_i > 0\}} \left[\frac{(\varphi - 1)(y_i - 1)y_i}{(\mu_i(\beta) + (\varphi - 1)y_i)^2} - \frac{1}{\varphi} \right] \right), \quad l, m = 0, \dots, p, \\
\frac{\partial^2 l(\delta)}{\partial \beta_l \partial \varphi} = & \sum_{i=1}^n x_{il} \left(\mathbb{1}_{\{y_i = 0\}} \right. \\
& \times \left[\frac{(\frac{1}{\varphi^2} P_i^0(\delta) (\mu_i(\beta))^2 (-\frac{1}{\varphi}) + P_i^0(\delta) \frac{1}{\varphi^2} \mu_i(\beta)) (k_i(\gamma) + P_i^0(\delta))}{(k_i(\gamma) + P_i^0(\delta))^2} \right. \\
& \left. - \frac{(P_i^0(\delta) (-\frac{1}{\varphi}) \mu_i(\beta)) (\frac{1}{\varphi^2} \mu_i(\beta) P_i^0(\delta))}{(k_i(\gamma) + P_i^0(\delta))^2} \right] \\
& \left. + \mathbb{1}_{\{y_i > 0\}} \left[\frac{(y_i - 1)(-\mu_i(\beta) y_i)}{(\mu_i(\beta) + (\varphi - 1)y_i)^2} + \frac{1}{\varphi^2} \mu_i(\beta) \right] \right) \\
= & \sum_{i=1}^n x_{il} \mu_i(\beta) \left(\mathbb{1}_{\{y_i = 0\}} \left[\frac{(-\frac{1}{\varphi^3} P_i^0(\delta) \mu_i(\beta) + \frac{1}{\varphi^2} P_i^0(\delta)) (k_i(\gamma) + P_i^0(\delta))}{(k_i(\gamma) + P_i^0(\delta))^2} \right. \right. \\
& \left. \left. + \frac{\frac{1}{\varphi^3} (P_i^0(\delta))^2 \mu_i(\beta)}{(k_i(\gamma) + P_i^0(\delta))^2} \right] + \mathbb{1}_{\{y_i > 0\}} \left[\frac{-y_i(y_i - 1)}{(\mu_i(\beta) + (\varphi - 1)y_i)^2} + \frac{1}{\varphi^2} \right] \right) \\
= & \sum_{i=1}^n x_{il} \mu_i(\beta) \left(\mathbb{1}_{\{y_i = 0\}} \right. \\
& \times \frac{-\frac{1}{\varphi^3} P_i^0(\delta) \mu_i(\beta) k_i(\gamma) + \frac{1}{\varphi^2} P_i^0(\delta) (k_i(\gamma) + P_i^0(\delta))}{(k_i(\gamma) + P_i^0(\delta))^2} \\
& \left. - \mathbb{1}_{\{y_i > 0\}} \left[\frac{y_i(y_i - 1)}{(\mu_i(\beta) + (\varphi - 1)y_i)^2} - \frac{1}{\varphi^2} \right] \right), \quad l = 0, \dots, p, \\
\frac{\partial^2 l(\delta)}{\partial \beta_l \partial \gamma_r} = & \sum_{i=1}^n x_{il} z_{ir} \mathbb{1}_{\{y_i = 0\}} \left[\frac{\frac{1}{\varphi} P_i^0(\delta) \mu_i(\beta) k_i(\gamma)}{(k_i(\gamma) + P_i^0(\delta))^2} \right], \quad l = 0, \dots, p; \quad r = 0, \dots, q, \\
\frac{\partial^2 l(\delta)}{\partial \varphi^2} = & \sum_{i=1}^n \mathbb{1}_{\{y_i = 0\}} \left[\frac{(\frac{1}{\varphi^4} P_i^0(\delta) (\mu_i(\beta))^2 - \frac{2}{\varphi^3} P_i^0(\delta) \mu_i(\beta)) (k_i(\gamma) + P_i^0(\delta))}{(k_i(\gamma) + P_i^0(\delta))^2} \right. \\
& \left. - \frac{\frac{1}{\varphi^4} (P_i^0(\delta))^2 (\mu_i(\beta))^2}{(k_i(\gamma) + P_i^0(\delta))^2} \right] \\
& + \mathbb{1}_{\{y_i > 0\}} \left[\frac{-y_i^2(y_i - 1)}{(\mu_i(\beta) + (\varphi - 1)y_i)^2} + \frac{y_i}{\varphi^2} - 2 \frac{\mu_i(\beta) - y_i}{\varphi^3} \right]
\end{aligned}$$

$$\begin{aligned}
&= \sum_{i=1}^n \mathbb{1}_{\{y_i=0\}} \left[\frac{\frac{1}{\varphi^4} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta})^2 k_i(\boldsymbol{\gamma}) - \frac{2}{\varphi^3} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta}) (k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))^2} \right] \\
&\quad + \mathbb{1}_{\{y_i>0\}} \left[\frac{-y_i^2(y_i-1)}{(\mu_i(\boldsymbol{\beta}) + (\varphi-1)y_i)^2} + \frac{y_i}{\varphi^2} - 2 \frac{\mu_i(\boldsymbol{\beta}) - y_i}{\varphi^3} \right], \\
\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \varphi \partial \gamma_s} &= \sum_{i=1}^n z_{is} \mathbb{1}_{\{y_i=0\}} \left[\frac{-\frac{1}{\varphi^2} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta}) k_i(\boldsymbol{\gamma})}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))^2} \right], \quad s = 0, \dots, q, \\
\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \gamma_j \partial \gamma_t} &= \sum_{i=1}^n z_{ij} \left(\mathbb{1}_{\{y_i=0\}} \left[\frac{k_i(\boldsymbol{\gamma}) z_{it} (k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta})) - (k_i(\boldsymbol{\gamma}))^2 z_{it}}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))^2} \right] \right. \\
&\quad \left. - \frac{k_i(\boldsymbol{\gamma}) z_{it} (1 + k_i(\boldsymbol{\gamma})) - (k_i(\boldsymbol{\gamma}))^2 z_{it}}{(1 + k_i(\boldsymbol{\gamma}))^2} \right) \\
&= \sum_{i=1}^n z_{ij} z_{it} k_i(\boldsymbol{\gamma}) \left(\mathbb{1}_{\{y_i=0\}} \left[\frac{P_i^0(\boldsymbol{\delta})}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))^2} \right] - \frac{1}{(1 + k_i(\boldsymbol{\gamma}))^2} \right), \\
&\quad j, t = 0, \dots, q.
\end{aligned}$$

1.2.5 Erwartete Fisher-Information der $ZIGP(\mu_i, \varphi, \omega_i)$ -Regression

Definition 1.11 (Erwartete Fisher - Informationsmatrix).

Der negative Erwartungswert der Hessematrix

$$\mathbf{F}_n(\boldsymbol{\delta}) := -E \left(\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\delta} \partial \boldsymbol{\delta}^t} \right)$$

wird als (erwartete) Fisher - Informationsmatrix (oder kurz "Fisher-Information") bezeichnet. Man schreibt $\mathbf{H}_n(\boldsymbol{\delta}) = -\boldsymbol{\mathcal{H}}_n(\boldsymbol{\delta})$ und $\mathbf{F}_n(\boldsymbol{\delta}) = E(\mathbf{H}_n(\boldsymbol{\delta}))$.

Es wird vorausgesetzt, daß die *Regularitätsbedingungen* erfüllt sind. Eine genaue Ausführung geben Bickel und Freedman [2]. Dabei bedarf man besonders der Vertauschbarkeit von Differenziation und Summation

$$\frac{\partial}{\partial \boldsymbol{\delta}} \left(\sum_{y=0}^{\infty} L(\boldsymbol{\delta} | \mathbf{Y} = \mathbf{y}) \right) = \sum_{y=0}^{\infty} \frac{\partial}{\partial \boldsymbol{\delta}} L(\boldsymbol{\delta} | \mathbf{Y} = \mathbf{y}). \quad (1.17)$$

Dann läßt sich zeigen, daß

$$\begin{aligned}
E \left(\frac{\partial}{\partial \boldsymbol{\delta}} l(\boldsymbol{\delta} | \mathbf{Y}) \right) &= \sum_{y=0}^{\infty} \frac{\partial}{\partial \boldsymbol{\delta}} \left(l(\boldsymbol{\delta} | \mathbf{Y} = \mathbf{y}) \right) P(y, \boldsymbol{\delta}) \\
&= \sum_{y=0}^{\infty} \frac{P(y, \boldsymbol{\delta})}{L(\boldsymbol{\delta} | \mathbf{Y} = \mathbf{y})} \frac{\partial}{\partial \boldsymbol{\delta}} \left(L(\boldsymbol{\delta} | \mathbf{Y} = \mathbf{y}) \right) \\
&= \sum_{y=0}^{\infty} \frac{\partial}{\partial \boldsymbol{\delta}} L(\boldsymbol{\delta} | \mathbf{Y} = \mathbf{y}) \stackrel{(1.17)}{=} \underbrace{\frac{\partial}{\partial \boldsymbol{\delta}} \left(\sum_{y=0}^{\infty} L(\boldsymbol{\delta} | \mathbf{Y} = \mathbf{y}) \right)}_{\equiv 1} = 0.
\end{aligned}$$

Die Fisher-Information wird nach Evans und Rosenthal [9, S. 343] zur Schätzung der Varianz der Koeffizientenschätzer herangezogen:

$$\hat{Var}(\hat{\boldsymbol{\delta}}_i) = [\mathbf{F}_n(\hat{\boldsymbol{\delta}})^{-1}]_{ii}. \quad (1.18)$$

Die Fisher-Information benötigt man jedoch auch für Aussagen über die Asymptotik der Maximum-Likelihood-Schätzer: unter bestimmten Voraussetzungen kann man zeigen, daß die MLE asymptotisch normalverteilt sind, d.h.

$$\mathbf{F}_n(\boldsymbol{\delta}_0)^{t/2}(\hat{\boldsymbol{\delta}}_n - \boldsymbol{\delta}_0) \xrightarrow{D} N_p(\mathbf{0}, \mathbf{I}_{p+q+3}), \text{ wenn } n \rightarrow \infty.$$

Dabei ist $\boldsymbol{\delta}_0$ der wahre, jedoch unbekannte, Parametervektor. Diese Aussage wird allerdings erst in Abschnitt 2.3 für den allgemein gültigen Fall einer $ZIGP(\mu_i, \varphi_i, \omega_i)$ -Regression gezeigt. Setzt man $\varphi_i = \varphi$ konstant, so gilt die Aussage damit auch für $ZIGP(\mu_i, \varphi, \omega_i)$ -Modelle. Eine graphische Überprüfung der asymptotischen Normalität mit Hilfe von QQ-Plots folgt in Abschnitt 1.4.

Die erwartete Fisher-Informationsmatrix hat für das besprochene Regressionsmodell die Form

$$\mathbf{F}_n(\boldsymbol{\delta}) = -E\mathcal{H}_n(\boldsymbol{\delta}) = -E \begin{pmatrix} \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^t} & \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\beta}^t \partial \varphi} & \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\gamma}^t} \\ \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \varphi \partial \boldsymbol{\beta}} & \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \varphi^2} & \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \varphi \partial \boldsymbol{\gamma}} \\ \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\gamma} \partial \boldsymbol{\beta}^t} & \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\gamma}^t \partial \varphi} & \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\gamma} \partial \boldsymbol{\gamma}^t} \end{pmatrix}.$$

Um den Erwartungswert der Hessematrix zu berechnen, sind einige Hilfssätze vonnöten.

Lemma 1.12 (Erwartungswert der Indikatorfunktion).

Für Zufallsvariablen $Y_i \sim ZIGP(\mu_i, \varphi, \omega_i)$ und Linkfunktionen (1.9) und (1.11) gilt

$$E(\mathbb{1}_{\{y_i=0\}}) = a_i \quad i = 1, \dots, n \quad (1.19)$$

und

$$E(\mathbb{1}_{\{y_i>0\}}) = 1 - a_i \quad i = 1, \dots, n \quad (1.20)$$

mit

$$a_i = \frac{k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta})}{1 + k_i(\boldsymbol{\gamma})}. \quad (1.21)$$

Beweis.

$$E(\mathbb{1}_{\{y_i=0\}}) = 1 \cdot P(Y_i = 0) + 0 \cdot P(Y_i > 0) = \omega_i + (1 - \omega_i)P_i^0(\boldsymbol{\delta}) = \frac{k_i(\boldsymbol{\gamma})}{1 + k_i(\boldsymbol{\gamma})} + \frac{1}{1 + k_i(\boldsymbol{\gamma})}P_i^0(\boldsymbol{\delta}). \quad \square$$

Lemma 1.13.

Sei $Y \sim ZIGP(\mu, \varphi, \omega)$. Dann gelten

$$E\left(\frac{Y(Y-1)}{[\mu + (\varphi-1)Y]^2}\right) = (1-\omega) \frac{\mu}{\varphi^2(\mu-2+2\varphi)} \quad (1.22)$$

$$E\left(\frac{Y^2(Y-1)}{[\mu + (\varphi-1)Y]^2}\right) = (1-\omega) \frac{\mu(\mu+2\varphi)}{\varphi^2(\mu-2+2\varphi)}. \quad (1.23)$$

Beweis.

Der Beweis ist angelehnt an Stekler [35, S. 81]. Dort ist jedoch $Y \sim GP(\mu, \varphi)$, deshalb taucht hier zusätzlich der Term $(1 - \omega)$ auf.

$$\begin{aligned}
E\left(\frac{Y(Y-1)}{[\mu + (\varphi - 1)Y]^2}\right) &= \sum_{y=0}^{\infty} \frac{y(y-1)}{[\mu + (\varphi - 1)y]^2} P_y \\
&= \sum_{y=2}^{\infty} \frac{(1-\omega)y(y-1)}{[\mu + (\varphi - 1)y]^2} \cdot \frac{\mu[\mu + (\varphi - 1)y]^{y-1}}{y!} \varphi^{-y} e^{-\frac{1}{\varphi}[\mu + (\varphi - 1)y]} \\
&= (1-\omega) \sum_{y=2}^{\infty} \frac{\mu[\mu + (\varphi - 1)y]^{y-3}}{(y-2)!} \varphi^{-y} e^{-\frac{1}{\varphi}[\mu + (\varphi - 1)y]} \\
&= (1-\omega) \frac{\mu}{\varphi^2} \sum_{y=2}^{\infty} \frac{[\mu + 2(\varphi - 1) + (\varphi - 1)(y-2)]^{y-3}}{(y-2)!} \\
&\quad \times \varphi^{-(y-2)} e^{-\frac{1}{\varphi}[\mu + 2(\varphi - 1) + (\varphi - 1)(y-2)]} \\
&\quad \text{(substituiere } \tilde{\mu} := \mu + 2(\varphi - 1) \text{ und } \tilde{y} := y - 2) \\
&= (1-\omega) \frac{\mu}{\varphi^2} \sum_{\tilde{y}=0}^{\infty} \frac{[\tilde{\mu} + (\varphi - 1)\tilde{y}]^{\tilde{y}-1}}{\tilde{y}!} \varphi^{-\tilde{y}} e^{-\frac{1}{\varphi}[\tilde{\mu} + (\varphi - 1)\tilde{y}]} \\
&= (1-\omega) \frac{\mu}{\varphi^2 \tilde{\mu}} \sum_{\tilde{y}=0}^{\infty} \frac{\tilde{\mu}[\tilde{\mu} + (\varphi - 1)\tilde{y}]^{\tilde{y}-1}}{\tilde{y}!} \varphi^{-\tilde{y}} e^{-\frac{1}{\varphi}[\tilde{\mu} + (\varphi - 1)\tilde{y}]} \\
&= (1-\omega) \frac{\mu}{\varphi^2 \tilde{\mu}} \sum_{\tilde{y}=0}^{\infty} P_{GP}(\tilde{y} | \tilde{\mu}, \varphi) = \frac{(1-\omega)\mu}{\varphi^2(\mu - 2 + 2\varphi)} \quad \text{und} \\
E\left(\frac{Y^2(Y-1)}{[\mu + (\varphi - 1)Y]^2}\right) &= \sum_{y=0}^{\infty} \frac{y^2(y-1)}{[\mu + (\varphi - 1)y]^2} P_y \\
&= \sum_{y=2}^{\infty} \frac{(1-\omega)y^2(y-1)}{[\mu + (\varphi - 1)y]^2} \frac{\mu[\mu + (\varphi - 1)y]^{y-1}}{y!} \varphi^{-y} e^{-\frac{1}{\varphi}[\mu + (\varphi - 1)y]} \\
&\quad \text{(substituiere } \tilde{\mu} := \mu + 2(\varphi - 1) \text{ und } \tilde{y} := y - 2) \\
&= (1-\omega) \frac{\mu}{\varphi^2 \tilde{\mu}} \sum_{\tilde{y}=0}^{\infty} y \frac{\tilde{\mu}[\tilde{\mu} + (\varphi - 1)\tilde{y}]^{\tilde{y}-1}}{\tilde{y}!} \varphi^{-\tilde{y}} e^{-\frac{1}{\varphi}[\tilde{\mu} + (\varphi - 1)\tilde{y}]} \\
&= (1-\omega) \frac{\mu}{\varphi^2 \tilde{\mu}} \sum_{\tilde{y}=0}^{\infty} (\tilde{y} + 2) \frac{\tilde{\mu}[\tilde{\mu} + (\varphi - 1)\tilde{y}]^{\tilde{y}-1}}{\tilde{y}!} \varphi^{-\tilde{y}} e^{-\frac{1}{\varphi}[\tilde{\mu} + (\varphi - 1)\tilde{y}]} \\
&= (1-\omega) \frac{\mu}{\varphi^2 \tilde{\mu}} \sum_{\tilde{y}=0}^{\infty} (\tilde{y} + 2) P_{GP}(\tilde{y} | \tilde{\mu}, \varphi) \\
&= (1-\omega) \frac{\mu}{\varphi^2 \tilde{\mu}} E((\tilde{y} + 2) | \tilde{\mu}, \varphi) \\
&= (1-\omega) \frac{\mu}{\varphi^2 \tilde{\mu}} (\tilde{\mu} + 2) = (1-\omega) \frac{\mu(\mu - 2 + 2\varphi + 2)}{\varphi^2(\mu - 2 + 2\varphi)}
\end{aligned}$$

$$= (1 - \omega) \frac{\mu(\mu + 2\varphi)}{\varphi^2(\mu - 2 + 2\varphi)}.$$

□

Mit diesen Lemmas kann man nun die erwartete Fisher-Informationsmatrix berechnen. Sei a_i definiert wie in (1.21).

$$\begin{aligned}
-E \left(\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \beta_l \partial \beta_m} \right) &= - \sum_{i=1}^n x_{il} x_{im} \mu_i(\boldsymbol{\beta}) \left(E \left[\mathbb{1}_{\{y_i=0\}} \right] \frac{-\frac{1}{\varphi} (P_i^0(\boldsymbol{\delta}))^2 + \frac{\mu_i(\boldsymbol{\beta}) - \varphi}{\varphi^2} P_i^0(\boldsymbol{\delta}) k_i(\boldsymbol{\gamma})}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))^2} \right. \\
&\quad \left. + E \left[\mathbb{1}_{\{y_i>0\}} \frac{(\varphi - 1)(y_i - 1)y_i}{(\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i)^2} \right] - E \left[\mathbb{1}_{\{y_i>0\}} \frac{1}{\varphi} \right] \right) \\
&= - \sum_{i=1}^n x_{il} x_{im} \mu_i(\boldsymbol{\beta}) \left(a_i \frac{-\frac{1}{\varphi} (P_i^0(\boldsymbol{\delta}))^2 + \frac{\mu_i(\boldsymbol{\beta}) - \varphi}{\varphi^2} P_i^0(\boldsymbol{\delta}) k_i(\boldsymbol{\gamma})}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))^2} \right. \\
&\quad \left. + E \left[\frac{(\varphi - 1)(y_i - 1)y_i}{(\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i)^2} \right] - \frac{1 - a_i}{\varphi} \right) \\
&\stackrel{(1.22)}{=} - \sum_{i=1}^n x_{il} x_{im} \mu_i(\boldsymbol{\beta}) \left(\frac{-\frac{1}{\varphi} (P_i^0(\boldsymbol{\delta}))^2 + \frac{\mu_i(\boldsymbol{\beta}) - \varphi}{\varphi^2} P_i^0(\boldsymbol{\delta}) k_i(\boldsymbol{\gamma})}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))(1 + k_i(\boldsymbol{\gamma}))} \right. \\
&\quad \left. + \frac{(\varphi - 1)\mu_i(\boldsymbol{\beta})}{\varphi^2(\mu_i(\boldsymbol{\beta}) - 2 + 2\varphi)(1 + k_i(\boldsymbol{\gamma}))} - \frac{1 - a_i}{\varphi} \right), \\
&\quad l, m = 0, \dots, p, \\
-E \left(\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \beta_l \partial \varphi} \right) &= - \sum_{i=1}^n x_{il} \mu_i(\boldsymbol{\beta}) \\
&\quad \times \left(a_i \frac{-\frac{1}{\varphi^3} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta}) k_i(\boldsymbol{\gamma}) + \frac{1}{\varphi^2} P_i^0(\boldsymbol{\delta}) (k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))^2} \right. \\
&\quad \left. - E \left[\mathbb{1}_{\{y_i>0\}} \frac{y_i(y_i - 1)}{(\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i)^2} \right] + E \left[\mathbb{1}_{\{y_i>0\}} \right] \frac{1}{\varphi^2} \right) \\
&\stackrel{(1.22)}{=} - \sum_{i=1}^n x_{il} \mu_i(\boldsymbol{\beta}) \\
&\quad \times \left(\frac{-\frac{1}{\varphi^3} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta}) k_i(\boldsymbol{\gamma}) + \frac{1}{\varphi^2} P_i^0(\boldsymbol{\delta}) (k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))(1 + k_i(\boldsymbol{\gamma}))} \right. \\
&\quad \left. - \frac{\mu_i(\boldsymbol{\beta})}{\varphi^2(\mu_i(\boldsymbol{\beta}) - 2 + 2\varphi)(1 + k_i(\boldsymbol{\gamma}))} + \frac{1 - a_i}{\varphi^2} \right), \quad l = 0, \dots, p, \\
-E \left(\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \beta_l \partial \gamma_r} \right) &= - \sum_{i=1}^n x_{il} z_{ir} a_i \frac{\frac{1}{\varphi} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta}) k_i(\boldsymbol{\gamma})}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))^2}
\end{aligned}$$

$$\begin{aligned}
&= - \sum_{i=1}^n x_{il} z_{ir} \frac{\frac{1}{\varphi} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta}) k_i(\boldsymbol{\gamma})}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))(1 + k_i(\boldsymbol{\gamma}))}, \\
&\quad l = 0, \dots, p \text{ und } r = 0, \dots, q, \\
-E \left(\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \varphi^2} \right) &= - \sum_{i=1}^n E \left(\mathbb{1}_{\{y_i=0\}} \right) \\
&\quad \times \frac{\frac{1}{\varphi^4} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta})^2 k_i(\boldsymbol{\gamma}) - \frac{2}{\varphi^3} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta}) (k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))^2} \\
&\quad + E \left[\mathbb{1}_{\{y_i>0\}} \frac{-y_i^2(y_i - 1)}{(\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i)^2} \right] \\
&\quad + E \left[\mathbb{1}_{\{y_i>0\}} \left(\frac{y_i}{\varphi^2} - 2 \frac{\mu_i(\boldsymbol{\beta}) - y_i}{\varphi^3} \right) \right] \\
&\stackrel{(1.23)}{=} - \sum_{i=1}^n a_i \frac{\frac{1}{\varphi^4} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta})^2 k_i(\boldsymbol{\gamma}) - \frac{2}{\varphi^3} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta}) (k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))^2} \\
&\quad - \frac{\mu_i(\boldsymbol{\beta})(\mu_i(\boldsymbol{\beta}) + 2\varphi)}{\varphi^2(\mu_i(\boldsymbol{\beta}) - 2 + 2\varphi)(1 + k_i(\boldsymbol{\gamma}))} + E(y_i) \left(\frac{1}{\varphi^2} + \frac{2}{\varphi^3} \right) \\
&\quad - 2E \left(\mathbb{1}_{\{y_i>0\}} \right) \frac{\mu_i(\boldsymbol{\beta})}{\varphi^3} \\
&= - \sum_{i=1}^n \frac{\frac{1}{\varphi^4} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta})^2 k_i(\boldsymbol{\gamma}) - \frac{2}{\varphi^3} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta}) (k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))(1 + k_i(\boldsymbol{\gamma}))} \\
&\quad - \frac{\mu_i(\boldsymbol{\beta})(\mu_i(\boldsymbol{\beta}) + 2\varphi)}{\varphi^2(\mu_i(\boldsymbol{\beta}) - 2 + 2\varphi)(1 + k_i(\boldsymbol{\gamma}))} + \frac{\mu_i(\boldsymbol{\beta})}{1 + k_i(\boldsymbol{\gamma})} \left(\frac{1}{\varphi^2} + \frac{2}{\varphi^3} \right) \\
&\quad - 2(1 - a_i) \frac{\mu_i(\boldsymbol{\beta})}{\varphi^3}, \\
-E \left(\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \varphi \partial \gamma_s} \right) &= - \sum_{i=1}^n z_{is} \left[\frac{-\frac{1}{\varphi^2} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta}) k_i(\boldsymbol{\gamma})}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))(1 + k_i(\boldsymbol{\gamma}))} \right], \quad s = 0, \dots, q, \\
-E \left(\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \gamma_j \partial \gamma_t} \right) &= - \sum_{i=1}^n z_{ij} z_{it} k_i(\boldsymbol{\gamma}) \left(\frac{P_i^0(\boldsymbol{\delta})}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))(1 + k_i(\boldsymbol{\gamma}))} - \frac{1}{(1 + k_i(\boldsymbol{\gamma}))^2} \right), \\
&\quad j, t = 0, \dots, q.
\end{aligned}$$

1.3 Anmerkungen zur Software für die Modellanpassung und -analyse

In der "R"-Software wird eine Minimierung von $-l(\boldsymbol{\delta})$ mit dem Befehl *optim* vorgenommen. Ein Problem dabei ist, einen Startwert zu finden, an dem die Log Likelihood - Funktion gültig ist. Für $\boldsymbol{\beta}$ kann man die Koeffizienten aus der Anpassung eines Poisson-GLM verwenden. Für den Nullenüberschuß ist dies jedoch nicht möglich. Hier lautet die Aufgabe, $\gamma_0, \dots, \gamma_q$ jeweils auf ganz \mathbb{R} jeweils so zu wählen, daß die Log Likelihood-Funktion $l(\boldsymbol{\delta}, \mathbf{Y}) = \sum_{i=1}^n l_i(\boldsymbol{\delta}, \mathbf{Y}_i)$ in "R" für jeden Summanden $l_i(\boldsymbol{\delta}, \mathbf{Y}_i)$ einen endlichen Wert

zurückgibt. Da "R" aber nur beschränkte Exponenten zuläßt und den Wert bei Überschreiten dieser Grenze als 0, $-\infty$ oder ∞ definiert, kann nicht mit beliebigen Startwerten gearbeitet werden. Bei der Untersuchung größerer Datensätze mit beliebigen Startwerten wurde diese Voraussetzung fast durchwegs verletzt.

Zur Lösung dieses Problems wird mit einem Anlauf-Schätzverfahren gearbeitet, das zunächst ein $ZIGP(\mu_i, \varphi, \omega)$ -Modell (mit festem φ und ω) anpaßt:

a) Schätze $\delta^* = (\beta^t, \varphi^*, \omega^*)^t$ in einem $ZIGP(\mu_i, \varphi^*, \omega^*)$ -Modell mit Startwerten β_0 aus einem Poissonansatz, $\varphi_0^* \leftarrow 1.2$ und $\omega_0^* \leftarrow 0.2$. Man erhält eine Schätzung $\hat{\omega}^*$, also eine *durchschnittliche* Zero Inflation.

b) Sei die Linkfunktion für ω gegeben durch $\eta_i^\omega(\gamma) = g^\omega(\omega_i)$. Nun möchte man die Startwerte γ so bestimmen, daß

$$\|\eta_i^\omega(\gamma) - \mathbf{1}_n \cdot g^\omega(\hat{\omega}^*)\|_2 \rightarrow \min_{\gamma}.$$

Dabei ist $\mathbf{1}_n = (1, \dots, 1)^t$. Diese Problemstellung entspricht der Anpassung eines Linearen Modells für γ . Es lautet $\mathbf{Y}^\omega := \mathbf{1}_n \cdot g^\omega(\hat{\omega}^*) \sim \mathbf{Z}\gamma + \varepsilon$, wobei $\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)^t$ ein Fehlervektor mit $\varepsilon_i \sim N(0, \sigma^2)$. In "R" löst man dieses mittels:

```
rechte.seite <- rep(log(omega.star)-log(1-omega.star),n)      # LM for gamma
out<-lm(rechte.seite ~ Z-1)
gamma.start <- out$coefficients
```

Mit diesen Startwerten arbeitet die Software zuverlässig, Abbrüche aufgrund unzulässiger Startwerte traten damit nicht mehr auf. Es ist zu beachten, daß für die Ausführung der Software auch Funktionen kompiliert werden müssen, die zum Modell $ZIGP(\mu_i, \varphi, \omega)$ gehören:

- "mle.zigp.full.like",
- "log.likelihood.zigp.full",
- "fit.zigp1" und
- "summary.zigp1" .

Diese wurden von Stekeler [35] entwickelt, eigenen sich jedoch erst nach einigen Korrekturen für den Anlaufschritt. Daher wurden lauffähige Funktionen an die $ZIGP(\mu_i, \varphi, \omega_i)$ - Software auf der CD angehängt. Die Software kann in Anhang A angesehen werden. Dort werden neben den Funktionen auch Hilfe-Beschreibungen ähnlich zur Hilfe für "R"-Methoden angegeben.

1.4 Simulationsstudie zur Untersuchung der Asymptotik der ML-Schätzer

Im Folgenden werden die Maximum Likelihood - Schätzer untersucht, dabei soll überprüft werden, wie sich der

- Stichprobenumfang n
- Range der μ_i
- Dispersionsparameter φ
- Range der ω_i

auf die Güte der Schätzer auswirkt. Zunächst werden zwei Größen hergeleitet, die zur Beurteilung von Schätzern herangezogen werden können.

Definition 1.14 (Verzerrung und Erwartungstreue eines Schätzers).

Betrachtet man den Parameter θ , so gibt

$$b(\theta, \hat{\theta}) = E(\hat{\theta}) - \theta$$

die Verzerrung (engl. Bias) des Schätzers $\hat{\theta}$ an. Stimmen Erwartungswert des Schätzers und der Wert des zu schätzenden Parameters überein, d.h.

$$b(\theta, \hat{\theta}) = 0 \quad \Leftrightarrow \quad E(\hat{\theta}) = \theta$$

so bezeichnet man den Schätzer $\hat{\theta}$ als unverzerrt oder erwartungstreu.

Für die Genauigkeit eines Schätzers ist jedoch auch seine Varianz ausschlaggebend. Eine Statistik, die beide Größen berücksichtigt, ist die Mittlere Quadratische Abweichung.

Definition 1.15 (Mittlere Quadratische Abweichung).

Die Mittlere Quadratische Abweichung (engl. Mean Squared Error (MSE)) ist bestimmt durch

$$MSE = R(\theta, \hat{\theta}) = E([\hat{\theta} - \theta]^2).$$

Formt man diese Darstellung noch um, so läßt sich die Mittlere Quadratische Abweichung als Summe aus der Varianz und der quadrierten Verzerrung von $\hat{\theta}$ darstellen:

$$\begin{aligned} R(\theta, \hat{\theta}) &= E([\hat{\theta} - \theta]^2) = E([\hat{\theta} - E(\hat{\theta}) + E(\hat{\theta}) - \theta]^2) \\ &= E([\hat{\theta} - E(\hat{\theta})]^2) + \underbrace{2E([\hat{\theta} - E(\hat{\theta})][E(\hat{\theta}) - \theta])}_{=0} + \underbrace{E([E(\hat{\theta}) - \theta]^2)}_{\text{konstant}} \\ &= E([\hat{\theta} - E(\hat{\theta})]^2) + E([E(\hat{\theta}) - \theta]^2) \\ &= \text{Var}(\hat{\theta}) + b^2(\hat{\theta}, \theta). \end{aligned}$$

Aus letzter Darstellung für den MSE erkennt man, daß die Varianz eines Schätzers als Maß für dessen Güte nur für unverzerrte Schätzer sinnvoll ist.

Möchte man die Güte der Maximum Likelihood - Schätzer messen, scheitert die Anwendung von Bias oder Varianz des Schätzers daran, daß die Verteilung des Schätzers

unbekannt ist. Daher arbeitet man mit asymptotischen Schätzern für Bias und Varianz, nämlich

$$\hat{b}(\theta) = \hat{\theta} - \theta = \frac{1}{N} \sum_{i=1}^N \hat{\theta}_i - \theta.$$

N ist dabei die Anzahl der erzeugten Datensets, d.h. die Anzahl der Durchläufe, in denen jeweils ein Vektor von n Beobachtungen erzeugt und ein Modell darauf angepaßt wurde. Es wurde mit $N = 100$ gearbeitet. Anstelle von $Var(\hat{\theta})$ verwendet man die erwartungstreue und konsistente Stichprobenvarianz s^2 (s. z.B. Fahrmeir und Tutz [11, S. 369]).

Auch die Varianz des MSE ist interessant. Zu dessen Berechnung schlägt Stekeler [35, S. 126] vor, die Abschätzung

$$Var(\hat{R}) = \frac{1}{N}(m_4 - 4\theta m_3 + 4\theta^2 m_2 - m_2^2 + 4\theta m_1 m_2 - 4\theta^2 m_2)$$

zu verwenden. Dabei ist m_k der k -te Momentenschätzer von $\hat{\theta}$; $m_k = \frac{1}{N} \sum_{i=1}^N \theta_i^k$ schätzt also $\mu_k = E(\hat{\theta}^k)$.

Nun interessiert man sich dafür, wie die Schätzer des vorliegenden Regressionsmodells konvergieren. Wie in Abschnitt 2.3 gezeigt wird, sind die *Maximum Likelihood - Schätzer asymptotisch normalverteilt*. In diesem Abschnitt wird die asymptotische Normalität der ML-Schätzer unter einigen Voraussetzungen für den von beiden Modellen allgemeineren Fall einer $ZIGP(\mu_i, \varphi_i, \omega_i)$ -Regression nachgewiesen. Damit ist die Aussage mit $\varphi_i = \varphi \forall i$ auch auf $ZIGP(\mu_i, \varphi, \omega_i)$ -Modelle übertragbar.

Wenn man nun weiß, daß eine Asymptotik der Koeffizientenschätzer gegen die Normalverteilung vorliegt, so kann man mit Hilfe des Mean Squared Error die Konvergenzgeschwindigkeit überprüfen: je geringer der MSE eines Parameters, desto schneller konvergiert er gegen den wahren Wert. Dazu werden nun für Stichprobengrößen $n = 200$ und $n = 1000$ Daten verschiedener Szenarios erzeugt. Als Designmatrizen werden zweispaltige Matrizen erzeugt: Für den Erwartungswert wählt man $\mu_i = \exp\{\beta_0 + x_{i1}\beta_1\}$ mit x_{i1} äquidistant steigend auf $[-1, 1] \quad \forall i$. Damit steigen auch die μ_i mit Beobachtung i an. Ferner sei $\omega_i = \exp\{\gamma_0 + z_{i1}\gamma_1\}/(1 + \exp\{\gamma_0 + z_{i1}\gamma_1\})$ mit z_{i1} äquidistant fallend auf $[-1, 1] \quad \forall i$. Dies ergibt wiederum steigende ω_i .

Dabei werden β und γ so gewählt, daß sowohl "kleine" $\mu_i \in [2, 5]$ als auch "große" $\mu_i \in [2, 15]$ erzeugt werden. Entsprechend gibt es "kleine" $\omega_i \in [0.1, 0.25]$ und "große" $\omega_i \in [0.1, 0.5]$.

Parametrisierung	Range
$\beta = (1.15, 0.46)^t$	$\mu_i \in [2, 5]$
$\beta = (1.7, 1)^t$	$\mu_i \in [2, 15]$
$\gamma = (-1.65, -0.55)^t$	$\omega_i \in [0.1, 0.25]$
$\gamma = (-1.10, -1.1)^t$	$\omega_i \in [0.1, 0.5]$

Tabelle 1.1: Range von μ und ω in je zwei Settings

Der jeweils größere Parameterraum enthält den kleineren. Für den Dispersionsparameter wird je ein Setting mit $\varphi = 1.03$, $\varphi = 1.25$ und $\varphi = 3$ ausgeführt.

Die Ergebnisse der Simulationsläufe in Tabelle 1.2 enthalten Schätzer für die Parameter und als Fehlerstatistik den "Mean Squared Error". In Klammern sind jeweils die geschätzten Standardabweichungen der Statistiken angegeben. Man erkennt, daß MSE und Varianzen bei γ_0 und γ_1 am größten sind, diese Koeffizienten demnach sehr langsam konvergieren. Man erhält den geschätzten Bias $\hat{b}(\theta, \hat{\theta})$ aus der Tabelle, indem man den Wert der Spalte "ML-Schätzer" von dem der Spalte "Wahrer Wert" subtrahiert. Die QQ-Plots können in Abb. 1.1 bis Abb. 1.4 gefunden werden. Man beachte, daß die Punkte der QQ-Plots zur besseren Visualisierung durch Linien verbunden wurden. Gestrichelte Linien stehen für $n=200$, durchgezogene für $n=1000$. Eine detaillierte Einführung zu QQ-Plots findet man unter Faraway [14, S. 58-60]. Mit ihrer Hilfe kann man die asymptotische Normalität verifizieren. Sie plotten zentrierte und normierte Daten gegen ihre Quantile bezüglich der Standardnormalverteilung. Sind die Daten annähernd normalverteilt, so liegen die entstehenden Punkte nahe der Winkelhalbierenden des ersten und dritten Quadranten. Im folgenden wird von *Verschlechterung* gesprochen, wenn zwei QQ-Plots zeigen, daß die Linien im zweiten Plot nicht mehr so nahe oder geradlinig entlang der Winkelhalbierenden verlaufen. Eine *Verbesserung* sei der gegenteilige Fall.

Zuletzt wird zu jedem betrachteten mittleren ML-Schätzer $\hat{\delta}_j$ angegeben, ob dieser signifikant verschieden von Null ist. Es gelte $\hat{\delta}_j \sim N(\delta_j, \sigma)$, wobei σ durch s nach Formel (1.18) aus der Fisher-Information geschätzt wird. In Tabelle 1.2 kann man s in der Spalte *MSE* in Klammern finden. Damit kann man nun einen Wald-Test durchführen: $H_0 : \delta_j = 0$ gegen $H_1 : \text{nicht } H_0$. Da $N = 100$ Simulationen durchgeführt wurden, kann anstelle der t -Verteilung bedenkenlos die Standardnormalverteilung verwendet werden. Unter H_0 gilt dann approximativ $\frac{\hat{\delta}_j}{\sqrt{1/Ns}} \sim N(0, 1)$. Als Signifikanzniveau wird $\alpha = 5\%$ gewählt, womit H_0 abgelehnt wird, wenn

$$|T| := \left| 10 \frac{\hat{\delta}_j}{s} \right| > z_{1-\frac{\alpha}{2}} = 1.96. \quad (1.24)$$

Es konnte jedoch kein Setting gefunden werden, in dem $|T| \leq 1.96$, das spricht dafür, daß die Anzahl der Simulationsläufe mit $N = 100$ ausreichend groß gewählt wurde. Wichtigen Zahlen wurde in Text und Tabelle dieselbe Farbe gegeben, um die Suche in der Tabelle zu erleichtern.

Erhöhung des Stichprobenumfangs n

Erhöht man den Stichprobenumfang von $n = 200$ auf $n = 1000$, so verbessern sich diese Werte erwartungsgemäß. So sinkt für $\mu_i \in [2, 5]$, $\varphi = 1.25$ und $\omega_i \in [0.1, 0.5]$ der MSE für φ von **0.010** auf **0.002**. Auch die Verzerrung nimmt ab. So sinkt der Bias betragsmäßig von $\hat{b}(\theta, \hat{\theta}) = 1.25 - 1.223 = 0.027$ für $n = 200$ auf $\hat{b}(\theta, \hat{\theta}) = 1.25 - 1.248 = 0.002$ für $n = 1000$. Die Verbesserung kann auch im QQ-Plot in Abb. 1.2 (8) beim Vergleich der gestrichelten mit der durchgezogenen Linie gesehen werden. Es wird weiter untersucht, ob der mittlere ML-Schätzer signifikant von der Null abweicht. Dazu wird er wie dargestellt durch seine geschätzte Standardabweichung geteilt. Damit erhält man für $n = 200$ einen Quotienten von $10 \frac{1.223}{0.096} = 127.4$. Der Koeffizient ist demnach signifikant verschieden von Null. Für $n = 1000$ ergibt sich $10 \frac{1.248}{0.043} = 290.2$.

Erhöhung der Erwartungswerte μ_i

Im Szenario mit höheren Erwartungswerten sind die Schätzungen aller Parameter genauer. Dies ist einleuchtend, da geringere Erwartungswerte mehr Nullen erzeugen, und diese nur schwer ihrem Ursprung (also geringe Erwartungswerte, hohe Überdispersion oder hoher Nullenüberschuß) zugeordnet werden können. Dazu vergleiche man etwa für $\varphi = 3$, $\omega_i \in [0.1, 0.5]$, $n = 200$ den MSE von β_1 . Dieser sinkt bei größerem μ_i -Range um etwa 60% von **0.082** auf **0.033**. Teststatistiken nach Formel (1.24) sind $T = 10 \frac{0.403}{0.281} = 14.3$ und $T = 10 \frac{1.000}{0.182} = 54.9$. Der geschätzte Bias $\hat{b}(\theta, \hat{\theta})$ sinkt ebenfalls von 0.057 auf 0. Visuell ist die Verbesserung in den QQ-Plots beim Vergleich von Abb. 1.2 (12) mit 1.4 (12) sichtbar.

Erhöhung des Dispersionsparameters φ

Je größer φ , desto schlechter sind auch die Schätzungen. Dies leuchtet ein, da bei zunehmender Überdispersion die Varianz der Daten $Var(Y_i) = E(Y_i) \cdot [\varphi^2 + \mu_i \omega_i]$ und damit ihre Streuung wächst. Dies erschwert die Schätzung der Koeffizienten. Als Beispiel kann eine Erhöhung von $\varphi = 1.03$ auf $\varphi = 3$ betrachtet werden. Dort verzehnfacht sich der MSE von γ_1 im Setting $\mu_i \in [2, 5]$, $\omega_i \in [0.1, 0.25]$ für $n = 200$ von **0.205** ($T = 10 \frac{-0.587}{0.451} = -13.0$) auf **2.172** ($T = 10 \frac{-0.306}{1.453} = -2.1$), für $n = 1000$ sogar von **0.041** ($T = 10 \frac{-0.567}{0.201} = -28.2$) auf **0.422** ($T = 10 \frac{-0.601}{0.648} = -9.27$). Auch die Verzerrung $\hat{b}(\theta, \hat{\theta})$ steigt dem Betrage nach von 0.037 auf -0.244 bzw. im zweiten von Fall 0.017 auf 0.051. Die Verschlechterung wird auch in den QQ-Plots widerspiegelt, dazu vergleiche man Abb. 1.1 (5) mit (15).

Erhöhung des Nullenüberschuß ω_i

Eine Erhöhung im Nullenüberschuß ω_i erhöht wieder die Streuung der Daten und verschlechtert damit die Ergebnisse. Die Änderung des MSE fällt jedoch gering aus: so steigt er für $\varphi = 1.03$, $\mu_i \in [2, 15]$, β_1 und $n = 200$ lediglich von **0.004** ($T = 10 \frac{1.000}{0.064} = 156.3$) auf **0.005** ($T = 10 \frac{0.992}{0.069} = 143.8$), wobei $\hat{b}(\theta, \hat{\theta})$ von 0 auf 0.008 steigt. Dies wird in Abb. 1.3 (2) und Abb. 1.4 (2) deutlich, in denen eine Verschlechterung festzustellen ist. Dagegen fallen die MSE-Statistiken von γ_0 und γ_1 geringer aus. Das rührt daher, daß bei höherem Nullenüberschuß auch γ_j besser schätzbar sind. Vergleicht man für $\varphi = 1.25$, $\mu_i \in [2, 15]$, $n = 200$ die MSE-Statistiken für $\omega_i \in [0.1, 0.25]$ mit $\omega_i \in [0.1, 0.5]$, so stellt man fest: Für γ_0 sinkt der MSE von **0.243** ($T = 10 \frac{-3.333}{0.491} = -67.9$) auf **0.190** ($T = 10 \frac{-2.308}{0.422} = -54.7$), das entspricht einem Minus von 22%. Für γ_1 sinkt er von **0.193** ($T = 10 \frac{-0.584}{0.438} = -13.3$) auf **0.156** ($T = 10 \frac{-1.203}{0.381} = -31.6$), das sind 20% weniger. Der Einfluß des Nullenüberschuß auf die Güte der übrigen Parameter ist gering. Die Verzerrung steigt im ersten Fall von 0.033 auf 0.108, im zweiten Fall von 0.034 auf 0.103. Die QQ-Plots in Abb. 1.3 (9)-(10) und Abb. 1.4 (9)-(10) zeigen jedoch eher eine Verschlechterung.

Zusammenfassend kann man sagen, daß hohe Dispersion und hoher Nullenüberschuß die Konvergenzgeschwindigkeit der Schätzer verringern. Auch niedrige Erwartungswerte sind ungünstig. Eine Erhöhung des Stichprobenumfangs n erhöht jedoch wie erwartet die Konvergenzgeschwindigkeit.

Zuletzt ist noch festzustellen, daß der Graph im Falle $\varphi = 1.03$ häufig waagrecht nach links abknickt. Dies liegt daran, daß der wahre Wert von φ sehr nahe an 1 liegt und Schätzungen durch 1 nach unten beschränkt sind.

	$\varphi = 1.03$						$\varphi = 1.25$						$\varphi = 3$						
	Parameter	Wahrer Wert	n	ML-Schätzer		MSE	Wahrer Wert	ML-Schätzer		MSE	Wahrer Wert	ML-Schätzer		MSE	Wahrer Wert	ML-Schätzer		MSE	
$\mu_i \in [2, 5]$ $\omega_i \in [0.1, 0.25]$	β_0	1.15	200	1.153	(0.052)	0.003 ($8 \cdot 10^{-8}$)	1.15	1.151	(0.066)	0.004 ($3 \cdot 10^{-7}$)	1.15	1.173	(0.202)	0.041 ($1 \cdot 10^{-5}$)		s. Abb. 1.1			
			1000	1.150	(0.023)	0.001 ($5 \cdot 10^{-9}$)		1.142	(0.029)	0.001 ($1 \cdot 10^{-8}$)		1.139	(0.090)	0.008 ($5 \cdot 10^{-7}$)					
	β_1	0.46	200	0.449	(0.086)	0.008 ($6 \cdot 10^{-7}$)	0.46	0.460	(0.109)	0.012 ($3 \cdot 10^{-6}$)	0.46	0.408	(0.295)	0.090 (0.0002)					
			1000	0.459	(0.039)	0.001 ($2 \cdot 10^{-8}$)		0.467	(0.049)	0.002 ($1 \cdot 10^{-7}$)		0.446	(0.132)	0.018 ($7 \cdot 10^{-6}$)					
	φ	1.03	200	1.040	(0.067)	0.005 ($9 \cdot 10^{-7}$)	1.25	1.245	(0.090)	0.008 ($9 \cdot 10^{-7}$)	3	2.882	(0.425)	0.195 (0.0006)					
			1000	1.031	(0.030)	0.001 ($6 \cdot 10^{-9}$)		1.248	(0.040)	0.002 ($4 \cdot 10^{-8}$)		2.961	(0.190)	0.038 ($2 \cdot 10^{-5}$)					
	γ_0	-3.3	200	-3.383	(0.539)	0.298 (0.004)	-3.3	-3.476	(0.654)	0.458 (0.03)	-3.3	-3.407	(1.467)	2.162 (0.07)					
			1000	-3.354	(0.241)	0.061 ($8 \cdot 10^{-5}$)		-3.370	(0.292)	0.090 (0.0003)		-3.266	(0.656)	0.431 (0.004)					
	γ_1	-0.55	200	-0.587	(0.451)	0.205 (0.002)	-0.55	-0.656	(0.523)	0.285 (0.009)	-0.55	-0.306	(1.453)	2.172 (0.05)					
			1000	-0.567	(0.201)	0.041 ($3 \cdot 10^{-5}$)		-0.581	(0.233)	0.055 (0.0001)		-0.601	(0.648)	0.422 (0.003)					
$\omega_i \in [0.1, 0.5]$	β_0	1.15	200	1.145	(0.054)	0.003 ($2 \cdot 10^{-7}$)	1.15	1.148	(0.069)	0.005 ($2 \cdot 10^{-7}$)	1.15	1.136	(0.204)	0.042 ($3 \cdot 10^{-5}$)		s. Abb. 1.2			
			1000	1.148	(0.024)	0.001 ($5 \cdot 10^{-9}$)		1.151	(0.031)	0.001 ($1 \cdot 10^{-8}$)		1.139	(0.091)	0.008 ($1 \cdot 10^{-6}$)					
	β_1	0.46	200	0.461	(0.093)	0.009 ($2 \cdot 10^{-6}$)	0.46	0.453	(0.117)	0.014 ($2 \cdot 10^{-6}$)	0.46	0.403	(0.281)	0.082 (0.0002)					
			1000	0.461	(0.042)	0.002 ($5 \cdot 10^{-8}$)		0.456	(0.052)	0.003 ($7 \cdot 10^{-8}$)		0.448	(0.126)	0.016 ($4 \cdot 10^{-6}$)					
	φ	1.03	200	1.045	(0.071)	0.005 ($6 \cdot 10^{-7}$)	1.25	1.223	(0.096)	0.010 ($1 \cdot 10^{-6}$)	3	2.869	(0.453)	0.222 (0.0007)					
			1000	1.033	(0.032)	0.001 ($1 \cdot 10^{-8}$)		1.248	(0.043)	0.002 ($6 \cdot 10^{-8}$)		2.971	(0.202)	0.042 ($3 \cdot 10^{-5}$)					
	γ_0	-2.2	200	-2.264	(0.454)	0.210 (0.0009)	-2.2	-2.186	(0.535)	0.286 (0.001)	-2.2	-2.482	(1.523)	2.400 (0.2)					
			1000	-2.229	(0.203)	0.042 ($4 \cdot 10^{-5}$)		-2.206	(0.239)	0.057 ($6 \cdot 10^{-5}$)		-2.300	(0.681)	0.474 (0.004)					
	γ_1	-1.1	200	-1.165	(0.391)	0.158 (0.0007)	-1.1	-1.037	(0.445)	0.202 (0.0003)	-1.1	-1.133	(1.015)	1.031 (0.03)					
			1000	-1.133	(0.175)	0.032 ($3 \cdot 10^{-5}$)		-1.117	(0.199)	0.040 ($2 \cdot 10^{-5}$)		-1.133	(0.452)	0.206 (0.0008)					
$\mu_i \in [2, 15]$ $\omega_i \in [0.1, 0.25]$	β_0	1.7	200	1.700	(0.038)	0.001 ($4 \cdot 10^{-8}$)	1.7	1.703	(0.047)	0.002 ($9 \cdot 10^{-8}$)	1.7	1.726	(0.125)	0.016 ($6 \cdot 10^{-6}$)		s. Abb. 1.3			
			1000	1.700	(0.017)	0.000 ($1 \cdot 10^{-9}$)		1.696	(0.021)	0.000 ($3 \cdot 10^{-9}$)		1.705	(0.056)	0.003 ($1 \cdot 10^{-7}$)					
	β_1	1	200	1.000	(0.064)	0.004 ($3 \cdot 10^{-7}$)	1	0.996	(0.079)	0.006 ($6 \cdot 10^{-7}$)	1	0.976	(0.178)	0.032 ($2 \cdot 10^{-6}$)					
			1000	1.001	(0.028)	0.001 ($2 \cdot 10^{-8}$)		1.002	(0.035)	0.001 ($3 \cdot 10^{-8}$)		1.001	(0.080)	0.006 ($8 \cdot 10^{-7}$)					
	φ	1.03	200	1.038	(0.060)	0.004 ($5 \cdot 10^{-8}$)	1.25	1.239	(0.078)	0.006 ($6 \cdot 10^{-7}$)	3	2.942	(0.319)	0.105 (0.0002)					
			1000	1.031	(0.027)	0.001 ($5 \cdot 10^{-9}$)		1.250	(0.035)	0.001 ($3 \cdot 10^{-8}$)		3.013	(0.143)	0.020 ($7 \cdot 10^{-6}$)					
	γ_0	-3.3	200	-3.374	(0.450)	0.208 (0.001)	-3.3	-3.333	(0.491)	0.243 (0.008)	-3.3	-3.394	(1.111)	1.243 (0.04)					
			1000	-3.329	(0.201)	0.041 ($5 \cdot 10^{-5}$)		-3.336	(0.220)	0.050 ($8 \cdot 10^{-5}$)		-3.306	(0.496)	0.246 (0.0009)					
	γ_1	-0.55	200	-0.564	(0.402)	0.162 (0.0004)	-0.55	-0.584	(0.438)	0.193 (0.002)	-0.55	-0.616	(0.842)	0.714 (0.02)					
			1000	-0.581	(0.179)	0.033 ($2 \cdot 10^{-5}$)		-0.562	(0.195)	0.038 ($3 \cdot 10^{-5}$)		-0.570	(0.375)	0.141 (0.0004)					
$\omega_i \in [0.1, 0.5]$	β_0	1.7	200	1.697	(0.040)	0.002 ($7 \cdot 10^{-8}$)	1.7	1.698	(0.050)	0.002 ($6 \cdot 10^{-8}$)	1.7	1.692	(0.128)	0.016 ($5 \cdot 10^{-6}$)		s. Abb. 1.4			
			1000	1.702	(0.018)	0.000 ($1 \cdot 10^{-9}$)		1.700	(0.022)	0.000 ($4 \cdot 10^{-9}$)		1.703	(0.057)	0.003 ($2 \cdot 10^{-7}$)					
	β_1	1	200	0.992	(0.069)	0.005 ($6 \cdot 10^{-7}$)	1	0.997	(0.085)	0.007 ($9 \cdot 10^{-7}$)	1	1.000	(0.182)	0.033 ($2 \cdot 10^{-6}$)					
			1000	1.000	(0.031)	0.001 ($8 \cdot 10^{-9}$)		1.001	(0.038)	0.001 ($3 \cdot 10^{-8}$)		0.993	(0.081)	0.007 ($6 \cdot 10^{-7}$)					
	φ	1.03	200	1.042	(0.065)	0.004 ($2 \cdot 10^{-7}$)	1.25	1.236	(0.084)	0.007 ($2 \cdot 10^{-6}$)	3	2.931	(0.347)	0.125 (0.0003)					
			1000	1.033	(0.029)	0.001 ($5 \cdot 10^{-9}$)		1.242	(0.038)	0.001 ($5 \cdot 10^{-8}$)		2.969	(0.155)	0.025 ($8 \cdot 10^{-6}$)					
	γ_0	-2.2	200	-2.208	(0.391)	0.153 (0.0004)	-2.2	-2.308	(0.422)	0.190 (0.002)	-2.2	-2.316	(0.836)	0.713 (0.01)					
			1000	-2.199	(0.175)	0.031 ($1 \cdot 10^{-5}$)		-2.188	(0.189)	0.036 ($1 \cdot 10^{-5}$)		-2.140	(0.374)	0.143 (0.0004)					
	γ_1	-1.1	200	-1.107	(0.354)	0.125 (0.0003)	-1.1	-1.203	(0.381)	0.156 (0.0008)	-1.1	-1.228	(0.664)	0.457 (0.006)					
			1000	-1.105	(0.158)	0.025 ($1 \cdot 10^{-5}$)		-1.088	(0.170)	0.029 ($1 \cdot 10^{-5}$)		-1.069	(0.296)	0.088 (0.0001)					

Tabelle 1.2: Mittlere MLE-Schätzungen und MSE-Statistiken für $ZIGP(\mu_i, \varphi, \omega_i)$ in mehreren Parameter-Settings basierend auf $N = 100$ Datensätzen; geschätzte Standardabweichungen der Statistiken in Klammern.

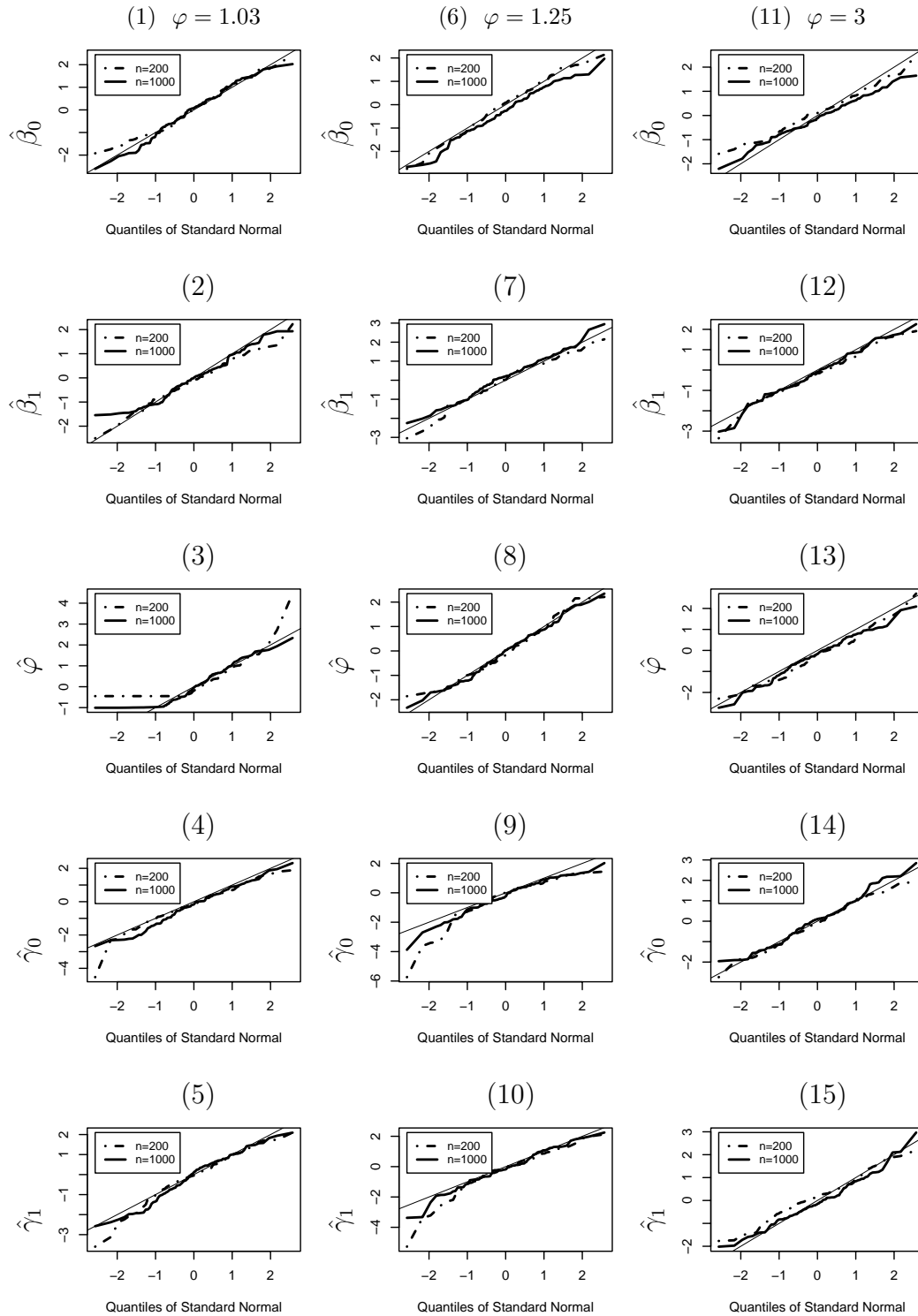


Abbildung 1.1: QQ-Plots zentrierter und normierter ML Schätzer basierend auf $N = 100$ Datensätzen im $ZIGP(\mu_i, \varphi, \omega_i)$ -Modell mit $\mu_i \in [2, 5]$ und $\omega_i \in [0.1, 0.25]$.

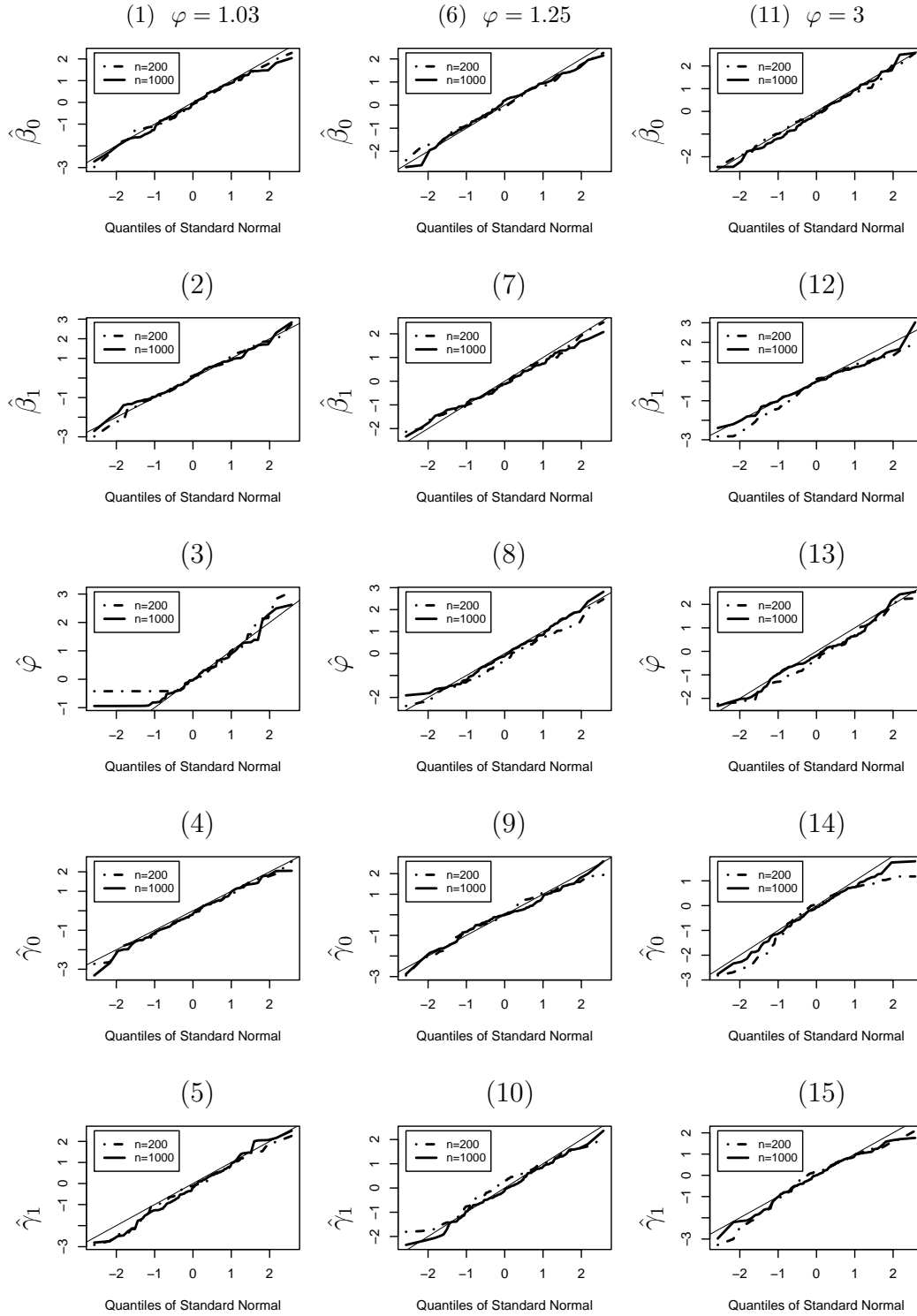


Abbildung 1.2: QQ-Plots zentrierter und normierter ML Schätzer basierend auf $N = 100$ Datensätzen im $ZIGP(\mu_i, \varphi, \omega_i)$ -Modell mit $\mu_i \in [2, 5]$ und $\omega_i \in [0.1, 0.5]$.

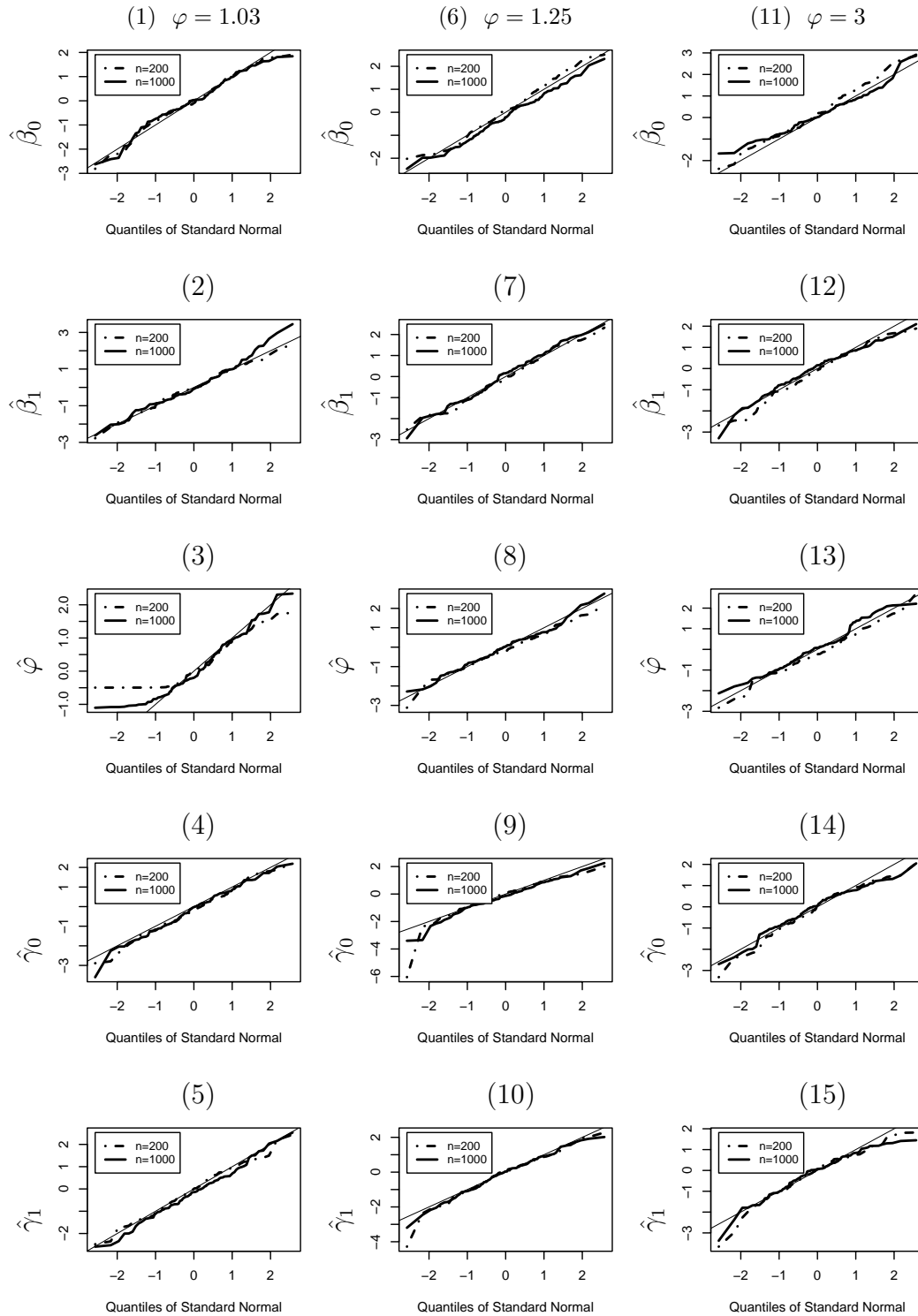


Abbildung 1.3: QQ-Plots zentrierter und normierter ML Schätzer basierend auf $N = 100$ Datensätzen im $ZIGP(\mu_i, \varphi, \omega_i)$ -Modell mit $\mu_i \in [2, 15]$ und $\omega_i \in [0.1, 0.25]$.

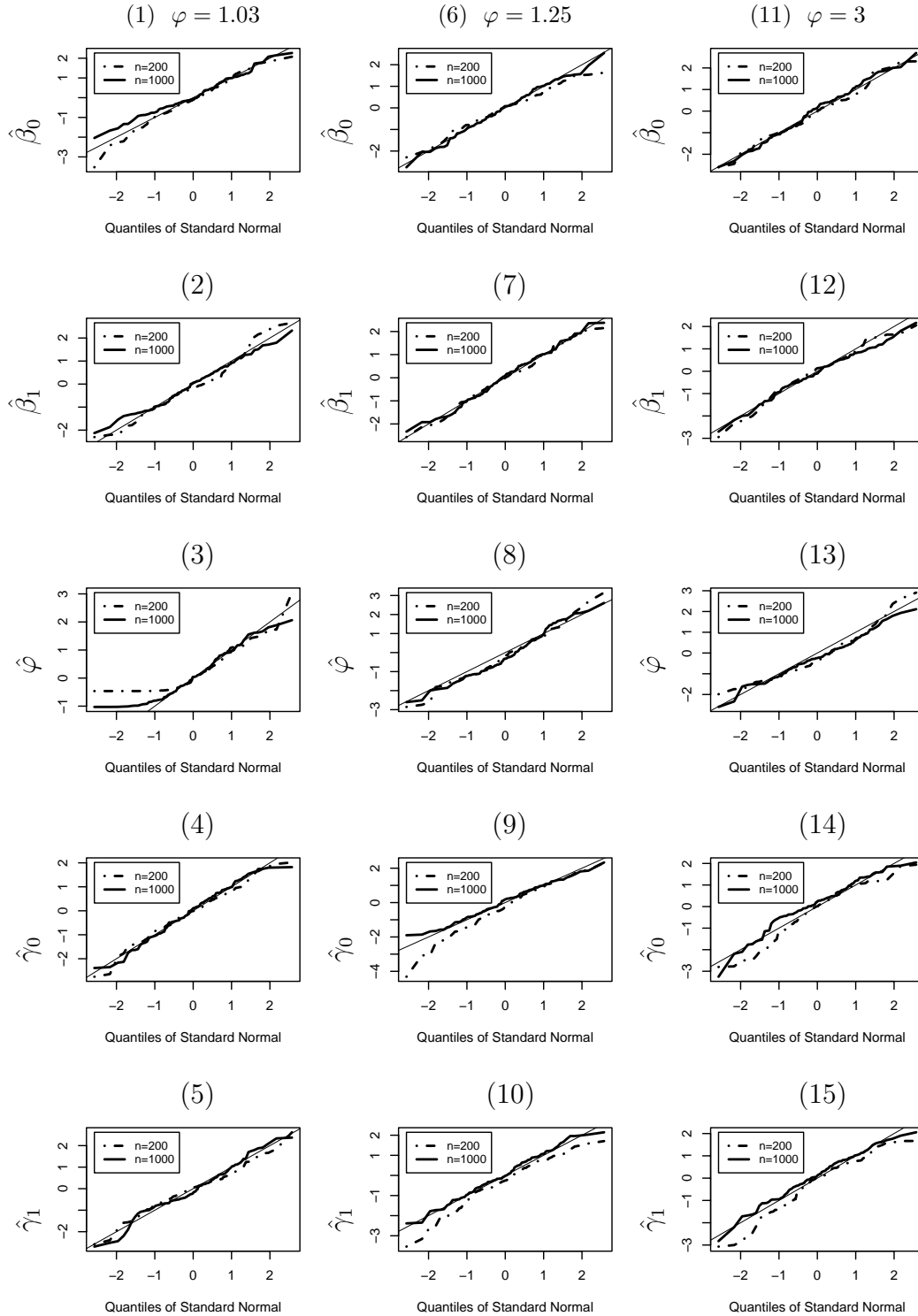


Abbildung 1.4: QQ-Plots zentrierter und normierter ML Schätzer basierend auf $N = 100$ Datensätzen im $ZIGP(\mu_i, \varphi, \omega_i)$ -Modell mit $\mu_i \in [2, 15]$ und $\omega_i \in [0.1, 0.5]$.

Kapitel 2

ZIGP - Modell mit Regression auf μ , φ und ω

2.1 Definition und Eigenschaften des $ZIGP(\mu_i, \varphi_i, \omega_i)$ – Regressionsmodells

Das Regressionsmodell aus Kapitel 1 wird nun um die Regression auf den Dispersionsparameter φ erweitert. Davon verspricht man sich einen Zugewinn an Flexibilität, da nun untersucht werden kann, was die Überdispersion beeinflusst. Variable Überdispersion verspricht gegenüber konstanter Überdispersion zudem, eine höhere Anpassungsgüte zu ermöglichen. Als Verknüpfungsfunktion für φ_i wird ein (modifizierter) Loglink verwendet. Das Regressionsmodell sieht dann wie folgt aus.

Zero Inflated GP – Regressionsmodell $ZIGP(\mu_i, \varphi_i, \omega_i)$

1. *Zufallskomponente:*

$\{Y_i, 1 \leq i \leq n\}$ sind unabhängig mit $Y_i \sim ZIGP(\mu_i, \varphi_i, \omega_i)$.

2. *Systematische Komponente:*

Drei linearen Prädiktoren $\eta_i^\mu(\boldsymbol{\beta}) = \mathbf{x}_i^t \boldsymbol{\beta}$, $\eta_i^\varphi(\boldsymbol{\alpha}) = \mathbf{w}_i^t \boldsymbol{\alpha}$ und $\eta_i^\omega(\boldsymbol{\gamma}) = \mathbf{z}_i^t \boldsymbol{\gamma}$, $i = 1, \dots, n$ beeinflussen den Response Y_i . Seien $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_p)^t$, $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \dots, \alpha_r)^t$ und $\boldsymbol{\gamma} = (\gamma_0, \gamma_1, \dots, \gamma_q)$ unbekannte Regressionsparameter. Die Matrizen $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^t$, $\mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_n)^t$ und $\mathbf{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_n)^t$ werden als Designmatrizen bezeichnet.

3. *Parametrische Linkfunktionen:*

Die linearen Prädiktoren $\eta_i^\mu(\boldsymbol{\beta})$, $\eta_i^\varphi(\boldsymbol{\alpha})$ und $\eta_i^\omega(\boldsymbol{\gamma})$ werden mit $\mu_i(\boldsymbol{\beta})$, $\varphi_i(\boldsymbol{\alpha})$ und $\omega_i(\boldsymbol{\gamma})$, $i = 1, \dots, n$ wie folgt verknüpft. Wieder seien E_i Exposurevariablen mit $E_i > 0 \forall i$.

(i) *Erwartungswertebene*

$$\begin{aligned} E(Y_i | \boldsymbol{\beta}) = \mu_i(\boldsymbol{\beta}) &:= E_i e^{\mathbf{x}_i^t \boldsymbol{\beta}} = e^{\mathbf{x}_i^t \boldsymbol{\beta} + \log(E_i)} > 0 \\ \Leftrightarrow \eta_i^\mu(\boldsymbol{\beta}) &= \log(\mu_i(\boldsymbol{\beta})) - \log(E_i) \quad (\text{Log-Link}), \end{aligned} \quad (2.1)$$

(ii) *Überdispersionsebene*

$$\begin{aligned}\varphi_i(\boldsymbol{\alpha}) &:= 1 + e^{\mathbf{w}_i^t \boldsymbol{\alpha}} > 1 \\ \Leftrightarrow \eta_i^\varphi(\boldsymbol{\alpha}) &= \log(\varphi_i(\boldsymbol{\alpha}) - 1), \quad (\text{modif. Log-Link}),\end{aligned}\quad (2.2)$$

(iii) *Nullenüberschußebene*

$$\begin{aligned}\omega_i(\boldsymbol{\gamma}) &:= \frac{e^{\mathbf{z}_i^t \boldsymbol{\gamma}}}{1 + e^{\mathbf{z}_i^t \boldsymbol{\gamma}}} \in (0, 1) \\ \Leftrightarrow \eta_i^\omega(\boldsymbol{\gamma}) &= \log\left(\frac{\omega_i(\boldsymbol{\gamma})}{1 - \omega_i(\boldsymbol{\gamma})}\right) \quad (\text{Logit-Link}).\end{aligned}\quad (2.3)$$

In Analogie zum vorangegangenen Kapitel wird der Parametervektor mit $\boldsymbol{\delta}$ bezeichnet, nur daß er nun anstelle von φ dessen Regressionsparameter $\boldsymbol{\alpha}$ enthält. Weitere Notationen sind:

$$\begin{aligned}\boldsymbol{\delta} &:= (\boldsymbol{\beta}^t, \boldsymbol{\alpha}^t, \boldsymbol{\gamma}^t)^t, \\ \mu_i(\boldsymbol{\beta}) &:= e^{\mathbf{x}_i^t \boldsymbol{\beta} + \log(E_i)}, \\ b_i(\boldsymbol{\alpha}) &:= e^{\mathbf{w}_i^t \boldsymbol{\alpha}}, \\ \varphi_i(\boldsymbol{\alpha}) &:= 1 + b_i(\boldsymbol{\alpha}), \\ k_i(\boldsymbol{\gamma}) &:= e^{\mathbf{z}_i^t \boldsymbol{\gamma}}, \\ \omega_i(\boldsymbol{\gamma}) &:= \frac{k_i(\boldsymbol{\gamma})}{1 + k_i(\boldsymbol{\gamma})}\end{aligned}$$

und

$$P_i^0(\boldsymbol{\delta}) := \exp\left(-\frac{E_i \cdot e^{\mathbf{x}_i^t \boldsymbol{\beta}}}{1 + e^{\mathbf{w}_i^t \boldsymbol{\alpha}}}\right). \quad (2.4)$$

2.1.1 Loglikelihoodfunktion der ZIGP($\mu_i, \varphi_i, \omega_i$)–Regression

Zur Schätzung der Regressionsparameter benötigt man zunächst die Loglikelihoodfunktion. Diese muß nun aufgrund der Regression auf den Dispersionsparameter neu berechnet werden. Sie lautet:

Satz 2.1 (Loglikelihood-Funktion).

Die Loglikelihood-Funktion im ZIGP($\mu_i, \varphi_i, \omega_i$)-Modell lautet:

$$\begin{aligned}l(\boldsymbol{\delta}) &= \sum_{i=1}^n \mathbb{1}_{\{y_i=0\}} \left[\log(e^{\mathbf{z}_i^t \boldsymbol{\gamma}} + \exp\{-\frac{E_i \cdot e^{\mathbf{x}_i^t \boldsymbol{\beta}}}{1 + e^{\mathbf{w}_i^t \boldsymbol{\alpha}}}\}) - \log(1 + e^{\mathbf{z}_i^t \boldsymbol{\gamma}}) \right] \\ &\quad + \mathbb{1}_{\{y_i>0\}} \left[-\log(1 + e^{\mathbf{z}_i^t \boldsymbol{\gamma}}) + \log(E_i) + \mathbf{x}_i^t \boldsymbol{\beta} + (y_i - 1) \log(E_i e^{\mathbf{x}_i^t \boldsymbol{\beta}} + e^{\mathbf{w}_i^t \boldsymbol{\alpha}} y_i) \right. \\ &\quad \left. - \log(y_i!) - y_i \log(1 + e^{\mathbf{w}_i^t \boldsymbol{\alpha}}) - \frac{E_i e^{\mathbf{x}_i^t \boldsymbol{\beta}} + e^{\mathbf{w}_i^t \boldsymbol{\alpha}} y_i}{1 + e^{\mathbf{w}_i^t \boldsymbol{\alpha}}} \right].\end{aligned}$$

Beweis.

Für Beobachtungen $\mathbf{y} = (y_1, \dots, y_n)^t$ ergibt sich die Likelihoodfunktion zu $L(\boldsymbol{\delta}) = L(\boldsymbol{\delta} \mid \mathbf{y}) = L(\mathbf{Y} = \mathbf{y} \mid \boldsymbol{\delta}) = \prod_{i=1}^n P(Y_i = y_i \mid \boldsymbol{\delta})$.

Und somit ist die Loglikelihood-Funktion

$$\begin{aligned}
 l(\boldsymbol{\delta}) &= \log L(\boldsymbol{\delta}) = \sum_{i=1}^n \mathbb{1}_{\{y_i=0\}} \log [\omega_i(\boldsymbol{\gamma}) + (1 - \omega_i(\boldsymbol{\gamma}))P_i^0(\boldsymbol{\delta})] \\
 &\quad + \mathbb{1}_{\{y_i>0\}} \left[\log(1 - \omega_i(\boldsymbol{\gamma})) + \log \mu_i(\boldsymbol{\beta}) + (y_i - 1) \log(\mu_i(\boldsymbol{\beta}) + (\varphi_i(\boldsymbol{\alpha}) - 1)y_i) \right. \\
 &\quad \left. - \log(y_i!) - y_i \log \varphi_i(\boldsymbol{\alpha}) - \frac{1}{\varphi_i(\boldsymbol{\alpha})} (\mu_i(\boldsymbol{\beta}) + (\varphi_i(\boldsymbol{\alpha}) - 1)y_i) \right] \\
 &= \sum_{i=1}^n \mathbb{1}_{\{y_i=0\}} \left[\log(e^{z_i^t \boldsymbol{\gamma}} + \exp\{-\frac{E_i \cdot e^{x_i^t \boldsymbol{\beta}}}{1 + e^{w_i^t \boldsymbol{\alpha}}}\}) - \log(1 + e^{z_i^t \boldsymbol{\gamma}}) \right] \\
 &\quad + \mathbb{1}_{\{y_i>0\}} \left[-\log(1 + e^{z_i^t \boldsymbol{\gamma}}) + \log(E_i) + x_i^t \boldsymbol{\beta} + (y_i - 1) \log(E_i e^{x_i^t \boldsymbol{\beta}} + e^{w_i^t \boldsymbol{\alpha}} y_i) \right. \\
 &\quad \left. - \log(y_i!) - y_i \log(1 + e^{w_i^t \boldsymbol{\alpha}}) - \frac{E_i e^{x_i^t \boldsymbol{\beta}} + e^{w_i^t \boldsymbol{\alpha}} y_i}{1 + e^{w_i^t \boldsymbol{\alpha}}} \right].
 \end{aligned}$$

□

2.1.2 Score-Vektor der $ZIGP(\mu_i, \varphi_i, \omega_i)$ -Regression

Der Score-Vektor werde mit $\mathbf{s}_n(\boldsymbol{\delta}) := \left(\frac{\partial l(\boldsymbol{\delta})}{\partial \boldsymbol{\beta}}, \frac{\partial l(\boldsymbol{\delta})}{\partial \boldsymbol{\alpha}}, \frac{\partial l(\boldsymbol{\delta})}{\partial \boldsymbol{\gamma}} \right)^t$ bezeichnet.

Satz 2.2 (Score-Vektor).

Im oben beschriebenen Modell sind die Einträge des Score-Vektors gegeben durch

$$\begin{aligned}
 \frac{\partial}{\partial \beta_j} l(\boldsymbol{\delta}) &= \sum_{i=1}^n x_{ij} \left(\mathbb{1}_{\{y_i=0\}} \frac{-\frac{1}{\varphi_i(\boldsymbol{\alpha})} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta})}{k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta})} \right. \\
 &\quad \left. + \mathbb{1}_{\{y_i>0\}} \left[1 + \frac{(y_i - 1) \mu_i(\boldsymbol{\beta})}{\mu_i(\boldsymbol{\beta}) + b_i(\boldsymbol{\alpha}) y_i} - \frac{1}{\varphi_i(\boldsymbol{\alpha})} \mu_i(\boldsymbol{\beta}) \right] \right), \quad j = 0, \dots, p, \\
 \frac{\partial}{\partial \alpha_j} l(\boldsymbol{\delta}) &= \sum_{i=1}^n w_{ij} b_i(\boldsymbol{\alpha}) \left(\mathbb{1}_{\{y_i=0\}} \frac{P_i^0(\boldsymbol{\delta}) \frac{\mu_i(\boldsymbol{\beta})}{\varphi_i(\boldsymbol{\alpha})^2}}{k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta})} + \mathbb{1}_{\{y_i>0\}} \left[\frac{(y_i - 1) y_i}{\mu_i(\boldsymbol{\beta}) + b_i(\boldsymbol{\alpha}) y_i} \right. \right. \\
 &\quad \left. \left. - \frac{y_i}{\varphi_i(\boldsymbol{\alpha})} + \frac{\mu_i(\boldsymbol{\beta}) - y_i}{\varphi_i(\boldsymbol{\alpha})^2} \right] \right), \quad j = 0, \dots, r, \\
 \frac{\partial}{\partial \gamma_j} l(\boldsymbol{\delta}) &= \sum_{i=1}^n z_{ij} k_i(\boldsymbol{\gamma}) \left(\mathbb{1}_{\{y_i=0\}} \frac{1}{k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta})} - \frac{1}{1 + k_i(\boldsymbol{\gamma})} \right), \quad j = 0, \dots, q.
 \end{aligned}$$

Beweis.

Die Terme $\frac{\partial}{\partial \beta_j} l(\boldsymbol{\delta})$ und $\frac{\partial}{\partial \gamma_j} l(\boldsymbol{\delta})$ erhält man direkt aus Satz 1.9, indem man φ durch $\varphi_i(\boldsymbol{\alpha})$ ersetzt. Die partielle Ableitung $\frac{\partial}{\partial \alpha_j} l(\boldsymbol{\delta})$ läßt sich mittels

$$\frac{\partial l(\boldsymbol{\delta})}{\partial \alpha_j} = \frac{\partial l(\boldsymbol{\delta})}{\partial \varphi_i(\boldsymbol{\alpha})} \cdot \frac{\partial \varphi_i(\boldsymbol{\alpha})}{\partial \alpha_j}$$

berechnen. □

2.1.3 Hessematrix der ZIGP($\mu_i, \varphi_i, \omega_i$)–Regression

Da der zusammengesetzte Parametervektor nun $\boldsymbol{\delta} = (\boldsymbol{\beta}^t, \boldsymbol{\alpha}^t, \boldsymbol{\gamma}^t)^t$ lautet, kann die Hessematrix nun partitioniert werden als

$$\mathcal{H}(\boldsymbol{\delta}) = \mathbf{I}_{obs} = \begin{pmatrix} \mathcal{H}_{\beta\beta}(\boldsymbol{\delta}) & \mathcal{H}_{\beta\alpha}(\boldsymbol{\delta}) & \mathcal{H}_{\beta\gamma}(\boldsymbol{\delta}) \\ \mathcal{H}_{\alpha\beta}(\boldsymbol{\delta}) & \mathcal{H}_{\alpha\alpha}(\boldsymbol{\delta}) & \mathcal{H}_{\alpha\gamma}(\boldsymbol{\delta}) \\ \mathcal{H}_{\gamma\beta}(\boldsymbol{\delta}) & \mathcal{H}_{\gamma\alpha}(\boldsymbol{\delta}) & \mathcal{H}_{\gamma\gamma}(\boldsymbol{\delta}) \end{pmatrix},$$

wobei

$$\mathcal{H}_{\beta\beta}(\boldsymbol{\delta}) = \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^t} \in \mathbb{R}^{p \times p},$$

$$\mathcal{H}_{\alpha\alpha}(\boldsymbol{\delta}) = \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\alpha} \partial \boldsymbol{\alpha}^t} \in \mathbb{R}^{r \times r},$$

$$\mathcal{H}_{\gamma\gamma}(\boldsymbol{\delta}) = \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\gamma} \partial \boldsymbol{\gamma}^t} \in \mathbb{R}^{q \times q},$$

$$\mathcal{H}_{\beta\alpha}(\boldsymbol{\delta}) = \mathcal{H}_{\alpha\beta}(\boldsymbol{\delta})^t = \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\alpha}^t} \in \mathbb{R}^{p \times r},$$

$$\mathcal{H}_{\beta\gamma}(\boldsymbol{\delta}) = \mathcal{H}_{\gamma\beta}(\boldsymbol{\delta})^t = \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\gamma}^t} \in \mathbb{R}^{p \times q}$$

und

$$\mathcal{H}_{\alpha\gamma}(\boldsymbol{\delta}) = \mathcal{H}_{\gamma\alpha}(\boldsymbol{\delta})^t = \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\alpha} \partial \boldsymbol{\gamma}^t} \in \mathbb{R}^{r \times q}.$$

Die benötigten partiellen Ableitungen lassen sich auf das in Kapitel 1 entwickelte Modell zurückführen. Mit Anwendung der Kettenregel auf $\frac{\partial}{\partial \alpha_l} \left(\frac{\partial l(\boldsymbol{\delta})}{\partial \alpha_j} \right)$ mit $\frac{\partial l(\boldsymbol{\delta})}{\partial \alpha_j} = \frac{\partial l(\boldsymbol{\delta})}{\partial \varphi_i(\boldsymbol{\alpha})} \cdot \frac{\partial \varphi_i(\boldsymbol{\alpha})}{\partial \alpha_j}$ gilt für partielle Ableitungen der Log-Likelihoodfunktion, daß

$$\frac{\partial}{\partial \alpha_l} \left(\frac{\partial l(\boldsymbol{\delta})}{\partial \alpha_j} \right) = \frac{\partial^2 l(\boldsymbol{\delta})}{\partial^2 \varphi_i(\boldsymbol{\alpha})} \cdot \left(\frac{\partial \varphi_i(\boldsymbol{\alpha})}{\partial \alpha_j} \right)^2 + \frac{\partial l(\boldsymbol{\delta})}{\partial \varphi_i(\boldsymbol{\alpha})} \cdot \frac{\partial^2 \varphi_i(\boldsymbol{\alpha})}{\partial \alpha_j \partial \alpha_l}.$$

Anmerkungen:

Aus Kapitel 1 unter Austauschen " $\varphi \leftrightarrow \varphi_i(\boldsymbol{\alpha})$ " sowie durch Ableiten der Linkfunktionen erhält man:

- $\frac{\partial^2 l(\boldsymbol{\delta})}{\partial^2 \varphi_i(\boldsymbol{\alpha})} = \sum_{i=1}^n \mathbb{1}_{\{y_i=0\}} \left[\frac{\frac{1}{\varphi_i(\boldsymbol{\alpha})^4} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta})^2 k_i(\boldsymbol{\gamma}) - \frac{2}{\varphi_i(\boldsymbol{\alpha})^3} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta}) (k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))^2} \right]$
 $+ \mathbb{1}_{\{y_i>0\}} \left[\frac{-y_i^2(y_i-1)}{(\mu_i(\boldsymbol{\beta}) + b_i(\boldsymbol{\alpha})y_i)^2} + \frac{y_i}{\varphi_i(\boldsymbol{\alpha})^2} - 2\frac{\mu_i(\boldsymbol{\beta}) - y_i}{\varphi_i(\boldsymbol{\alpha})^3} \right],$
- $\frac{\partial \varphi_i(\boldsymbol{\alpha})}{\partial \alpha_j} = b_i(\boldsymbol{\alpha}) w_{ij},$
- $\frac{\partial l(\boldsymbol{\delta})}{\partial \varphi_i(\boldsymbol{\alpha})} = \sum_{i=1}^n \mathbb{1}_{\{y_i=0\}} \left[\frac{\frac{1}{\varphi_i(\boldsymbol{\alpha})^2} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta})}{k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta})} \right] + \mathbb{1}_{\{y_i>0\}} \left[\frac{(y_i-1)y_i}{\mu_i(\boldsymbol{\beta}) + b_i(\boldsymbol{\alpha})y_i} - \frac{y_i}{\varphi_i(\boldsymbol{\alpha})} + \frac{\mu_i(\boldsymbol{\beta}) - y_i}{\varphi_i(\boldsymbol{\alpha})^2} \right],$
- $\frac{\partial^2 \varphi_i(\boldsymbol{\alpha})}{\partial \alpha_j \partial \alpha_l} = b_i(\boldsymbol{\alpha}) w_{ij} w_{il}.$

Dann sind die Einträge der Hessematrix

$$\begin{aligned} \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \beta_l \partial \beta_m} &= \sum_{i=1}^n x_{il} x_{im} \mu_i(\boldsymbol{\beta}) \left(\mathbb{1}_{\{y_i=0\}} \left[\frac{-\frac{1}{\varphi_i(\boldsymbol{\alpha})} (P_i^0(\boldsymbol{\delta}))^2 + \frac{\mu_i(\boldsymbol{\beta}) - \varphi_i(\boldsymbol{\alpha})}{\varphi_i(\boldsymbol{\alpha})^2} P_i^0(\boldsymbol{\delta}) k_i(\boldsymbol{\gamma})}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))^2} \right] \right. \\ &\quad \left. + \mathbb{1}_{\{y_i>0\}} \left[\frac{(\varphi_i(\boldsymbol{\alpha}) - 1)(y_i - 1)y_i}{(\mu_i(\boldsymbol{\beta}) + (\varphi_i(\boldsymbol{\alpha}) - 1)y_i)^2} - \frac{1}{\varphi_i(\boldsymbol{\alpha})} \right] \right), \quad l, m = 0, \dots, p, \\ \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \beta_l \partial \alpha_j} &= \sum_{i=1}^n x_{il} \left(\mathbb{1}_{\{y_i=0\}} \right. \\ &\quad \times \left[\frac{\left[-P_i^0(\boldsymbol{\delta}) \frac{\mu_i(\boldsymbol{\beta}) b_i(\boldsymbol{\alpha}) w_{ij}}{\varphi_i(\boldsymbol{\alpha})^3} + P_i^0(\boldsymbol{\delta}) \frac{b_i(\boldsymbol{\alpha}) w_{ij}}{\varphi_i(\boldsymbol{\alpha})^2} \right] (k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))^2} \right. \\ &\quad \left. + \frac{P_i^0(\boldsymbol{\delta}) \frac{1}{\varphi_i(\boldsymbol{\alpha})} \left[P_i^0(\boldsymbol{\delta}) \frac{\mu_i(\boldsymbol{\beta}) b_i(\boldsymbol{\alpha}) w_{ij}}{\varphi_i(\boldsymbol{\alpha})^2} \right]}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))^2} \mu_i(\boldsymbol{\beta}) \right] \\ &\quad \left. + \mathbb{1}_{\{y_i>0\}} \left[\frac{-y_i(y_i - 1) \mu_i(\boldsymbol{\beta}) b_i(\boldsymbol{\alpha}) w_{ij}}{(\mu_i(\boldsymbol{\beta}) + b_i(\boldsymbol{\alpha}) y_i)^2} + \frac{\mu_i(\boldsymbol{\beta}) b_i(\boldsymbol{\alpha}) w_{ij}}{\varphi_i(\boldsymbol{\alpha})^2} \right] \right) \\ &= \sum_{i=1}^n x_{il} w_{ij} \mu_i(\boldsymbol{\beta}) b_i(\boldsymbol{\alpha}) \left(\mathbb{1}_{\{y_i=0\}} \right. \\ &\quad \times \frac{-P_i^0(\boldsymbol{\delta}) \frac{\mu_i(\boldsymbol{\beta})}{\varphi_i(\boldsymbol{\alpha})^3} k_i(\boldsymbol{\gamma}) + \frac{P_i^0(\boldsymbol{\delta})}{\varphi_i(\boldsymbol{\alpha})^2} (k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))^2} \\ &\quad \left. + \mathbb{1}_{\{y_i>0\}} \left[-\frac{y_i(y_i - 1)}{(\mu_i(\boldsymbol{\beta}) + b_i(\boldsymbol{\alpha}) y_i)^2} + \frac{1}{\varphi_i(\boldsymbol{\alpha})^2} \right] \right), \quad l = 0, \dots, p; \quad j = 0, \dots, r, \\ \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \beta_l \partial \gamma_m} &= \sum_{i=1}^n x_{il} z_{im} \mathbb{1}_{\{y_i=0\}} \left[\frac{\frac{1}{\varphi_i(\boldsymbol{\alpha})} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta}) k_i(\boldsymbol{\gamma})}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))^2} \right], \quad l = 0, \dots, p; \quad m = 0, \dots, q, \\ \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \alpha_l \partial \alpha_j} &= \sum_{i=1}^n b_i(\boldsymbol{\alpha})^2 w_{ij} w_{il} \left(\mathbb{1}_{\{y_i=0\}} \right. \end{aligned}$$

$$\begin{aligned}
& \times \left[\frac{\frac{1}{\varphi_i(\alpha)^4} P_i^0(\delta) \mu_i(\beta)^2 k_i(\gamma) - \frac{2}{\varphi_i(\alpha)^3} P_i^0(\delta) \mu_i(\beta) (k_i(\gamma) + P_i^0(\delta))}{(k_i(\gamma) + P_i^0(\delta))^2} \right] \\
& + \mathbb{1}_{\{y_i > 0\}} \left[\frac{-y_i^2(y_i - 1)}{(\mu_i(\beta) + b_i(\alpha)y_i)^2} + \frac{y_i}{\varphi_i(\alpha)^2} - 2 \frac{\mu_i(\beta) - y_i}{\varphi_i(\alpha)^3} \right] \\
& + \sum_{i=1}^n b_i(\alpha) w_{ij} w_{il} \left(\mathbb{1}_{\{y_i=0\}} \left[\frac{\frac{1}{\varphi_i(\alpha)^2} P_i^0(\delta) \mu_i(\beta)}{k_i(\gamma) + P_i^0(\delta)} \right] \right. \\
& \left. + \mathbb{1}_{\{y_i > 0\}} \left[\frac{(y_i - 1)y_i}{\mu_i(\beta) + b_i(\alpha)y_i} - \frac{y_i}{\varphi_i(\alpha)} + \frac{\mu_i(\beta) - y_i}{\varphi_i(\alpha)^2} \right] \right) \\
& = \sum_{i=1}^n b_i(\alpha) w_{il} w_{ij} \left(\mathbb{1}_{\{y_i=0\}} \frac{P_i^0(\delta) \mu_i(\beta)}{(k_i(\gamma) + P_i^0(\delta))^2} \right. \\
& \left[\frac{\mu_i(\beta) b_i(\alpha) k_i(\gamma)}{\varphi_i(\alpha)^4} + \left(\frac{1}{\varphi_i(\alpha)^2} - 2 \frac{b_i(\alpha)}{\varphi_i(\alpha)^3} \right) (k_i(\gamma) + P_i^0(\delta)) \right] \\
& + \mathbb{1}_{\{y_i > 0\}} \left[\frac{(y_i - 1)y_i(\mu_i(\beta) + b_i(\alpha)y_i) - y_i^2(y_i - 1)b_i(\alpha)}{(\mu_i(\beta) + b_i(\alpha)y_i)^2} - \frac{y_i}{\varphi_i(\alpha)} \right. \\
& \left. + \frac{y_i b_i(\alpha)}{\varphi_i(\alpha)^2} + \frac{\mu_i(\beta) - y_i}{\varphi_i(\alpha)^2} - 2 \frac{\mu_i(\beta) b_i(\alpha) - y_i b_i(\alpha)}{\varphi_i(\alpha)^3} \right] \Bigg) \\
& = \sum_{i=1}^n b_i(\alpha) w_{il} w_{ij} \left(\mathbb{1}_{\{y_i=0\}} \frac{P_i^0(\delta) \mu_i(\beta)}{(k_i(\gamma) + P_i^0(\delta))^2} \right. \\
& \left[\frac{\mu_i(\beta) b_i(\alpha) k_i(\gamma)}{\varphi_i(\alpha)^4} + \left(\frac{1}{\varphi_i(\alpha)^2} - 2 \frac{b_i(\alpha)}{\varphi_i(\alpha)^3} \right) (k_i(\gamma) + P_i^0(\delta)) \right] \\
& + \mathbb{1}_{\{y_i > 0\}} \left[\frac{(y_i - 1)y_i \mu_i(\beta)}{(\mu_i(\beta) + b_i(\alpha)y_i)^2} + y_i \left[\frac{-\varphi_i(\alpha) + b_i(\alpha) - 1}{\varphi_i(\alpha)^2} \right. \right. \\
& \left. \left. + \frac{2b_i(\alpha)}{\varphi_i(\alpha)^3} \right] + \frac{\mu_i(\beta) \varphi_i(\alpha) - 2\mu_i(\beta) b_i(\alpha)}{\varphi_i(\alpha)^3} \right] \Bigg) \\
& = \sum_{i=1}^n b_i(\alpha) w_{il} w_{ij} \left(\mathbb{1}_{\{y_i=0\}} \frac{P_i^0(\delta) \mu_i(\beta)}{(k_i(\gamma) + P_i^0(\delta))^2} \right. \\
& \left[\frac{\mu_i(\beta) b_i(\alpha) k_i(\gamma)}{\varphi_i(\alpha)^4} + \left(\frac{1}{\varphi_i(\alpha)^2} - 2 \frac{b_i(\alpha)}{\varphi_i(\alpha)^3} \right) (k_i(\gamma) + P_i^0(\delta)) \right] \\
& + \mathbb{1}_{\{y_i > 0\}} \left[\frac{(y_i - 1)y_i \mu_i(\beta)}{(\mu_i(\beta) + b_i(\alpha)y_i)^2} + 2y_i \left(\frac{-1}{\varphi_i(\alpha)^2} + \frac{b_i(\alpha)}{\varphi_i(\alpha)^3} \right) \right. \\
& \left. + \frac{1}{\varphi_i(\alpha)^3} [\mu_i(\beta) \varphi_i(\alpha) - 2\mu_i(\beta) b_i(\alpha)] \right] \Bigg), \quad l, j = 0, \dots, r, \\
\frac{\partial^2 l(\delta)}{\partial \alpha_l \partial \gamma_s} & = \sum_{i=1}^n w_{il} z_{is} b_i(\alpha) \mathbb{1}_{\{y_i=0\}} \frac{-P_i^0(\delta) \frac{1}{\varphi_i(\alpha)^2} \mu_i(\beta) k_i(\gamma)}{(k_i(\gamma) + P_i^0(\delta))^2},
\end{aligned}$$

$$l = 0, \dots, r \text{ und } s = 0, \dots, q,$$

$$\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \gamma_j \partial \gamma_t} = \sum_{i=1}^n z_{ij} z_{it} k_i(\gamma) \left(\mathbb{1}_{\{y_i=0\}} \left[\frac{P_i^0(\boldsymbol{\delta})}{(k_i(\gamma) + P_i^0(\boldsymbol{\delta}))^2} \right] - \frac{1}{(1 + k_i(\gamma))^2} \right),$$

$$j, t = 0, \dots, q.$$

2.1.4 Erwartete Fisher-Information der $ZIGP(\mu_i, \varphi_i, \omega_i)$ -Regression

Wie in Abschnitt 1.2.5 dargestellt, kann die Fisherinformation zur Schätzung der Varianz der ML-Schätzer verwendet werden. Sie wird als negativer Erwartungswert der Hessematrix $\mathbf{F}_n(\boldsymbol{\delta}) = -E\mathcal{H}_n(\boldsymbol{\delta})$ berechnet und hat die Struktur:

$$\mathbf{F}_n(\boldsymbol{\delta}) = -E\mathcal{H}_n(\boldsymbol{\delta}) = -E \begin{pmatrix} \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^t} & \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\alpha}^t} & \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\gamma}^t} \\ \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\alpha} \partial \boldsymbol{\beta}^t} & \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\alpha} \partial \boldsymbol{\alpha}^t} & \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\alpha} \partial \boldsymbol{\gamma}^t} \\ \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\gamma} \partial \boldsymbol{\beta}^t} & \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\gamma} \partial \boldsymbol{\alpha}^t} & \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\gamma} \partial \boldsymbol{\gamma}^t} \end{pmatrix}.$$

Sei a_i definiert wie in (1.21), wobei jedoch $P_i^0(\boldsymbol{\delta})$ wie in (2.4) mit $\varphi_i(\boldsymbol{\alpha})$ statt φ gegeben ist. Dann erhält man:

$$\begin{aligned} -E \left(\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \beta_l \partial \beta_m} \right) &= - \sum_{i=1}^n x_{il} x_{im} \mu_i(\boldsymbol{\beta}) \left(\frac{-\frac{1}{\varphi_i(\boldsymbol{\alpha})} (P_i^0(\boldsymbol{\delta}))^2 + \frac{\mu_i(\boldsymbol{\beta}) - \varphi_i(\boldsymbol{\alpha})}{\varphi_i(\boldsymbol{\alpha})^2} P_i^0(\boldsymbol{\delta}) k_i(\gamma)}{(k_i(\gamma) + P_i^0(\boldsymbol{\delta}))(1 + k_i(\gamma))} \right. \\ &\quad \left. + \frac{b_i(\boldsymbol{\alpha}) \mu_i(\boldsymbol{\beta})}{\varphi_i(\boldsymbol{\alpha})^2 (\mu_i(\boldsymbol{\beta}) - 2 + 2\varphi_i(\boldsymbol{\alpha}))(1 + k_i(\gamma))} - \frac{1 - a_i}{\varphi_i(\boldsymbol{\alpha})} \right), \\ &\quad l, m = 0, \dots, p, \\ -E \left(\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \beta_l \partial \alpha_j} \right) &= - \sum_{i=1}^n x_{il} w_{ij} \mu_i(\boldsymbol{\beta}) b_i(\boldsymbol{\alpha}) \left(\frac{-P_i^0(\boldsymbol{\delta}) \frac{\mu_i(\boldsymbol{\beta})}{\varphi_i(\boldsymbol{\alpha})^3} k_i(\gamma) + \frac{P_i^0(\boldsymbol{\delta})}{\varphi_i(\boldsymbol{\alpha})^2} (k_i(\gamma) + P_i^0(\boldsymbol{\delta}))}{(k_i(\gamma) + P_i^0(\boldsymbol{\delta}))(1 + k_i(\gamma))} \right. \\ &\quad \left. - \frac{\mu_i(\boldsymbol{\beta})}{\varphi_i(\boldsymbol{\alpha})^2 (\mu_i(\boldsymbol{\beta}) - 2 + 2\varphi_i(\boldsymbol{\alpha}))(1 + k_i(\gamma))} + \frac{1 - a_i}{\varphi_i(\boldsymbol{\alpha})^2} \right), \\ &\quad l = 0, \dots, p \text{ und } j = 0, \dots, r, \\ -E \left(\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \beta_l \partial \gamma_m} \right) &= - \sum_{i=1}^n x_{il} z_{im} \frac{\frac{1}{\varphi_i(\boldsymbol{\alpha})} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta}) k_i(\gamma)}{(k_i(\gamma) + P_i^0(\boldsymbol{\delta}))(1 + k_i(\gamma))}, \\ &\quad l = 0, \dots, p \text{ und } m = 0, \dots, q, \\ -E \left(\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \alpha_l \partial \alpha_j} \right) &= - \sum_{i=1}^n b_i(\boldsymbol{\alpha}) w_{il} w_{ij} \left(\frac{P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta})}{(k_i(\gamma) + P_i^0(\boldsymbol{\delta}))(1 + k_i(\gamma))} \right. \\ &\quad \times \left[\frac{\mu_i(\boldsymbol{\beta}) b_i(\boldsymbol{\alpha}) k_i(\gamma)}{\varphi_i(\boldsymbol{\alpha})^4} + \left(\frac{1}{\varphi_i(\boldsymbol{\alpha})^2} - 2 \frac{b_i(\boldsymbol{\alpha})}{\varphi_i(\boldsymbol{\alpha})^3} \right) (k_i(\gamma) + P_i^0(\boldsymbol{\delta})) \right] \\ &\quad \left. + \frac{\mu_i(\boldsymbol{\beta})^2}{\varphi_i(\boldsymbol{\alpha})^2 (\mu_i(\boldsymbol{\beta}) - 2 + 2\varphi_i(\boldsymbol{\alpha}))(1 + k_i(\gamma))} + \frac{\mu_i(\boldsymbol{\beta})}{1 + k_i(\gamma)} \right) \end{aligned}$$

$$\begin{aligned}
& \times \left[\frac{-2}{\varphi_i(\boldsymbol{\alpha})^2} + \frac{\varphi_i(\boldsymbol{\alpha}) - P_i^0(\boldsymbol{\delta})(1 - b_i(\boldsymbol{\alpha}))}{\varphi_i(\boldsymbol{\alpha})^3} \right] \Bigg), \\
& l, j = 0, \dots, r, \\
-E \left(\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \alpha_l \partial \gamma_s} \right) &= - \sum_{i=1}^n w_{il} z_{is} b_i(\boldsymbol{\alpha}) \frac{-P_i^0(\boldsymbol{\delta}) \frac{1}{\varphi_i(\boldsymbol{\alpha})^2} \mu_i(\boldsymbol{\beta}) k_i(\boldsymbol{\gamma})}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))(1 + k_i(\boldsymbol{\gamma}))}, \\
& l = 0, \dots, r \text{ und } s = 0, \dots, q, \\
-E \left(\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \gamma_j \partial \gamma_t} \right) &= - \sum_{i=1}^n z_{ij} z_{it} k_i(\boldsymbol{\gamma}) \left(\frac{P_i^0(\boldsymbol{\delta})}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))(1 + k_i(\boldsymbol{\gamma}))} - \frac{1}{(1 + k_i(\boldsymbol{\gamma}))^2} \right), \\
& j, t = 0, \dots, q.
\end{aligned}$$

2.2 Anmerkungen zur Software für die Modellanpassung und -analyse

Auch zur Berechnung der ML-Schätzer und zur Anpassung des $ZIGP(\mu_i, \varphi_i, \omega_i)$ -Regressionsmodells wird eine Software erstellt. Sie ist mit Hilfebeschreibungen und Beispielen in Anhang B angefügt. Um einen gültigen Startwert zu finden, der einen endlichen Wert der Loglikelihoodfunktion unter "R" zurückgibt, muß analog zu Kapitel 1 nun auch für die Regressionsparameter der Überdispersion $\boldsymbol{\alpha}$ ein Lineares Modell angepaßt werden. Die Suche nach Startwerten für $\boldsymbol{\gamma}$ bleibt gleich.

a) Schätze $\boldsymbol{\delta}^* = (\boldsymbol{\beta}^t, \varphi^*, \omega^*)^t$ in einem $ZIGP(\mu_i, \varphi^*, \omega^*)$ -Modell mit Startwerten $\boldsymbol{\beta}_0$ aus einem Poissonansatz, $\varphi_0^* \leftarrow 1.2$ und $\omega_0^* \leftarrow 0.2$. Man erhält Schätzungen $\hat{\varphi}^*$ und $\hat{\omega}^*$, also Schätzungen der *durchschnittlichen* Überdispersion und Zero Inflation.

b) Seien die Linkfunktionen für $\boldsymbol{\varphi}$ und $\boldsymbol{\omega}$ gegeben durch $\eta_i^\varphi(\boldsymbol{\alpha}) = g^\varphi(\varphi_i)$ und $\eta_i^\omega(\boldsymbol{\gamma}) = g^\omega(\omega_i)$. Nun möchte man die Startwerte $\boldsymbol{\alpha}$ und $\boldsymbol{\gamma}$ so bestimmen, daß

$$\|\eta_i^\varphi(\boldsymbol{\alpha}) - \mathbf{1}_n \cdot g^\varphi(\hat{\varphi}^*)\|_2 \rightarrow \min_{\boldsymbol{\alpha}} \quad \text{und} \quad \|\eta_i^\omega(\boldsymbol{\gamma}) - \mathbf{1}_n \cdot g^\omega(\hat{\omega}^*)\|_2 \rightarrow \min_{\boldsymbol{\gamma}}.$$

Diese Problemstellung entspricht der Anpassung zweier Linearer Modelle für $\boldsymbol{\alpha}$ und $\boldsymbol{\gamma}$. Sie lauten $\mathbf{Y}^\varphi := \mathbf{1}_n \cdot g^\varphi(\hat{\varphi}^*) \sim \mathbf{W}\boldsymbol{\alpha} + \boldsymbol{\varepsilon}^\varphi$, wobei $\boldsymbol{\varepsilon}^\varphi = (\varepsilon_1^\varphi, \dots, \varepsilon_n^\varphi)^t$ ein Fehlervektor mit $\varepsilon_i^\varphi \sim N(0, \sigma_\varphi^2)$. Das zweite Lineare Modell ist $\mathbf{Y}^\omega := \mathbf{1}_n \cdot g^\omega(\hat{\omega}^*) \sim \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\varepsilon}^\omega$, wobei $\boldsymbol{\varepsilon}^\omega = (\varepsilon_1^\omega, \dots, \varepsilon_n^\omega)^t$ mit $\varepsilon_i^\omega \sim N(0, \sigma_\omega^2)$. In "R" löst man diese mit:

```

rechte.seite <- rep(log(phi.star - 1),n)                # LM for alpha
out<-lm(rechte.seite ~ W-1)
alpha.start <- out$coefficients
rechte.seite <- rep(log(omega.star)-log(1-omega.star),n) # LM for gamma
out<-lm(rechte.seite ~ Z-1)
gamma.start <- out$coefficients

```

Für ein Summary ähnlich dem *glm*-Summary unter "R" wird ein Wald-Test zum Test der Variablen auf Signifikanz erstellt. Dazu wird auch die Erwartete Fisher-Informationsmatrix berechnet.

2.3 Konsistenz und Konvergenz

Der folgende Abschnitt beschäftigt sich mit Eigenschaften von Schätzfolgen, der Konsistenz und Konvergenz. Er beginnt mit Notationen und Voraussetzungen an das Theorem. Der Beweis des Theorems ist in drei Abschnitte unterteilt. Zunächst wird die *asymptotische Existenz*, dann die *schwache Konsistenz* und zuletzt die *asymptotische Normalität* gezeigt. Diese drei Eigenschaften werden in Theorem 2.3 erklärt.

2.3.1 Notationen und Voraussetzungen

Zunächst werden folgende Notationen festgelegt:

Sei allgemein $\boldsymbol{\delta} := (\boldsymbol{\beta}^t, \boldsymbol{\alpha}^t, \boldsymbol{\gamma}^t)^t$. Der Vektor $\boldsymbol{\delta}_0 := (\boldsymbol{\beta}_0^t, \boldsymbol{\alpha}_0^t, \boldsymbol{\gamma}_0^t)^t$ enthalte die wahren (jedoch unbekannten) Parameterwerte des Modells. Darüber hinaus bezeichne $\hat{\boldsymbol{\delta}}_n$ die Nullstellen der Scoregleichungen. Die (erwartete) Fisher-Informationsmatrix ausgewertet an der Stelle $\boldsymbol{\delta}$ sei $\mathbf{F}_n(\boldsymbol{\delta})$. Der Scorevektor werde mit $\mathbf{s}_n(\boldsymbol{\delta})$ bezeichnet. Man stellt fest, daß man $\mathbf{s}_n(\boldsymbol{\delta})$ als Summe unabhängiger Zufallsvariablen darstellen kann, nämlich

$$\mathbf{s}_n(\boldsymbol{\delta}) = \sum_{i=1}^n \mathbf{s}_n(Y_i, \boldsymbol{\delta}), \quad (2.5)$$

wobei

$$\mathbf{s}_n(Y_i, \boldsymbol{\delta}) = \begin{pmatrix} z_0(Y_i, \boldsymbol{\delta}) \\ \vdots \\ z_{p+r+q+1}(Y_i, \boldsymbol{\delta}) \\ z_{p+r+q+2}(Y_i, \boldsymbol{\delta}) \end{pmatrix}. \quad (2.6)$$

Die Einträge lauten

$$\begin{aligned} z_l(Y_i, \boldsymbol{\delta}) &= x_{ij} \left(\mathbb{1}_{\{y_i=0\}} \left[\frac{-\frac{1}{\varphi_i(\boldsymbol{\alpha})} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta})}{k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta})} \right] \right. \\ &\quad \left. + \mathbb{1}_{\{y_i>0\}} \left[1 + \frac{(y_i - 1) \mu_i(\boldsymbol{\beta})}{\mu_i(\boldsymbol{\beta}) + b_i(\boldsymbol{\alpha}) y_i} - \frac{1}{\varphi_i(\boldsymbol{\alpha})} \mu_i(\boldsymbol{\beta}) \right] \right) \end{aligned}$$

für $l = 0, \dots, p$,

$$\begin{aligned} z_l(Y_i, \boldsymbol{\delta}) &= w_{ij} b_i(\boldsymbol{\alpha}) \left(\mathbb{1}_{\{y_i=0\}} \left[\frac{P_i^0(\boldsymbol{\delta}) \frac{\mu_i(\boldsymbol{\beta})}{\varphi_i(\boldsymbol{\alpha})^2}}{k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta})} \right] + \mathbb{1}_{\{y_i>0\}} \left[\frac{(y_i - 1) y_i}{\mu_i(\boldsymbol{\beta}) + b_i(\boldsymbol{\alpha}) y_i} \right. \right. \\ &\quad \left. \left. - \frac{y_i}{\varphi_i(\boldsymbol{\alpha})} + \frac{\mu_i(\boldsymbol{\beta}) - y_i}{\varphi_i(\boldsymbol{\alpha})^2} \right] \right) \end{aligned}$$

für $l = p + 1, \dots, p + r + 1$,
und

$$z_l(Y_i, \boldsymbol{\delta}) = z_{ij} k_i(\boldsymbol{\gamma}) \left(\mathbb{1}_{\{y_i=0\}} \left[\frac{1}{k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta})} \right] - \frac{1}{1 + k_i(\boldsymbol{\gamma})} \right)$$

für $l=p+r+2, \dots, p+r+q+2$.

Analog zu Fahrmeir und Kaufmann [10] soll die Cholesky-Quadratwurzel verwendet werden, um den Maximum Likelihood Schätzer zu normieren. Die linke Cholesky Quadratwurzel $\mathbf{A}^{1/2}$ einer positiv definiten Matrix \mathbf{A} ist diejenige eindeutige untere Dreiecksmatrix mit positiven Diagonalelementen, die $\mathbf{A}^{1/2}(\mathbf{A}^{1/2})^t = \mathbf{A}$ erfüllt. Zur Vereinfachung setzt man $\mathbf{A}^{t/2} := (\mathbf{A}^{1/2})^t$, $\mathbf{A}^{-1/2} := (\mathbf{A}^{1/2})^{-1}$ und $\mathbf{A}^{-t/2} := (\mathbf{A}^{t/2})^{-1}$. Ferner bezeichne $\lambda_{\max}(\mathbf{A})$ und $\lambda_{\min}(\mathbf{A})$ den größten bzw kleinsten Eigenwert von \mathbf{A} . Für Vektoren wird die L^2 -Norm $\|\cdot\|_2$ verwendet, für Matrizen die Spektralnorm:

$$\|\mathbf{A}\|_2 = \lambda_{\max}(\mathbf{A}^t \mathbf{A})^{1/2} = \sup_{\|u\|_2=1} \|\mathbf{A}u\|_2,$$

Anmerkung: Eine Matrixnorm $\|\mathbf{A}\|$ heißt *kompatibel* mit einer Vektornorm $\|\mathbf{x}\|$, falls gilt:

$$\|\mathbf{A} \cdot \mathbf{x}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{x}\| \quad (2.7)$$

Die oben eingeführte $\|\cdot\|_2$ -Matrixnorm ist kompatibel zur $\|\cdot\|_2$ -Vektornorm, da

$$\begin{aligned} \|\mathbf{A}\| &= \sup_{\|u\|_2=1} \|\mathbf{A} \cdot u\| = \sup_{u \neq 0} \frac{\|\mathbf{A} \cdot u\|}{\|u\|} \geq \frac{\|\mathbf{A} \cdot u\|}{\|u\|} \\ \Leftrightarrow \|\mathbf{A} \cdot u\| &\leq \|\mathbf{A}\| \cdot \|u\|. \end{aligned}$$

Darüber hinaus wird eine Umgebung des wahren Parametervektors δ_0

$$N_n(\varepsilon) := \{\delta : \|\mathbf{F}_n^{t/2}(\delta_0)(\delta - \delta_0)\| \leq \varepsilon\}$$

für $\varepsilon > 0$ definiert. Ferner gilt

$$\partial N_n(\varepsilon) = \{\delta : \|\mathbf{F}_n^{t/2}(\delta_0)(\delta - \delta_0)\| = \varepsilon\}.$$

Zur Vereinfachung werden die Argumente δ_0 , β_0 , α_0 und γ_0 weggelassen. Demnach vereinfacht sich $\mu_i(\beta_0)$ zu μ_i , $\varphi_i(\alpha_0)$ zu φ_i , $k_i(\gamma_0)$ zu k_i , $P_i^0(\delta_0)$ zu P_i^0 usw.. Wie oben definiert ist auch hier $\mathcal{H}_n(\delta)$ die Hessematrix, $\mathbf{H}_n(\delta) := -\mathcal{H}_n(\delta)$ die beobachtete und $\mathbf{F}_n(\delta) := E\mathbf{H}_n(\delta)$ die erwartete Fisher-Informationsmatrix. Außerdem sei $\mathbf{V}_n(\delta) := \mathbf{F}_n^{-1/2} \mathbf{H}_n(\delta) \mathbf{F}_n^{-t/2}$ für $n \in \mathbb{N}$ die normierte Fisher-Information. Konstanten werden mit C bezeichnet und können Indizes enthalten. Sie können von δ_0 abhängen, aber nicht von n . Zulässige Mengen für β , α und γ sind B , A und G .

2.3.2 Konsistenz und Konvergenz

Für das beschriebene ZIGP-Modell mit den angeführten Linkfunktionen werden für Theorem 2.3 fünf Annahmen gemacht.

(A1) (Divergenz)

$$\frac{n}{\lambda_{\min}(\mathbf{F}_n)} \leq C_1 \quad \forall n \geq 1,$$

wobei C_1 eine positive Konstante ist,

(A2) (Kompakte Regressoren)

$\{\mathbf{x}_n, n \geq 1\} \subset K_x$, wobei $K_x \subset \mathbb{R}^{p+1}$ eine kompakte Menge ist,
 $\{\mathbf{w}_n, n \geq 1\} \subset K_w$, wobei $K_w \subset \mathbb{R}^{r+1}$ eine kompakte Menge ist und
 $\{\mathbf{z}_n, n \geq 1\} \subset K_z$, wobei $K_z \subset \mathbb{R}^{q+1}$ ebenfalls eine kompakte Menge ist.

(A3) ($ZIGP(\mu_i, \varphi_i, \omega_i)$ -Regressionsmodell mit beschränkten Parametern)

Es liege das $ZIGP(\mu_i, \varphi_i, \omega_i)$ -Regressionsmodell aus Abschnitt 2.1 mit beschränkten Parametern $\mu_i(\boldsymbol{\delta}_0) > 0$, $\varphi_i(\boldsymbol{\delta}_0) > 1$ und $\omega_i(\boldsymbol{\delta}_0) \in (0, 1)$ vor. Dabei sei $\boldsymbol{\delta}_0$ ein Punkt aus der Menge $B \times A \times G$, wobei $B \subset \mathbb{R}^{p+1}$, $A \subset \mathbb{R}^{r+1}$ und $G \subset \mathbb{R}^{q+1}$ offene Mengen seien.

(A4) (Konvergenz und Stetigkeit)

$\max_{\boldsymbol{\delta} \in N_n(\varepsilon)} \|\mathbf{V}_n(\boldsymbol{\delta}) - \mathbf{I}_{p+r+q+3}\| \xrightarrow{P} 0 \quad \forall \varepsilon > 0$, wobei $\mathbf{V}_n(\boldsymbol{\delta}) := \mathbf{F}_n^{-1/2} \mathbf{H}_n(\boldsymbol{\delta}) \mathbf{F}_n^{-t/2}$ die normierte Fisher-Information ist.

(A5) (Beschränktheit $\mathbf{H}_n(\boldsymbol{\delta})$ von unten)

$P_{\boldsymbol{\delta}_0}\{\mathbf{H}_n(\boldsymbol{\delta}) - c\mathbf{F}(\boldsymbol{\delta}_0) \text{ ist positiv semidefinit für alle } \boldsymbol{\delta} \in N_n(\varepsilon)\} \rightarrow 1$

Ist (A1) erfüllt, dann gilt $\lambda_{\min}(\mathbf{F}_n) \rightarrow \infty$ schneller als $n \rightarrow \infty$. Damit divergiert die Fisher-Information. Annahme (A4) bedeutet, daß die negative Hessematrix stetig in Wahrscheinlichkeit gegen ihren Erwartungswert konvergiert. Die letzte Annahme (A5) steht für untere Beschränktheit der negativen Hessematrix durch die Fisher-Information. Das folgende Theorem enthält die Hauptaussage des gesamten Abschnittes 2.3.

Theorem 2.3 (Konsistenz und Konvergenz).

Unter den Annahmen (A1) - (A3) existiert eine Folge von Zufallsvariablen $\hat{\boldsymbol{\delta}}_n$, so daß

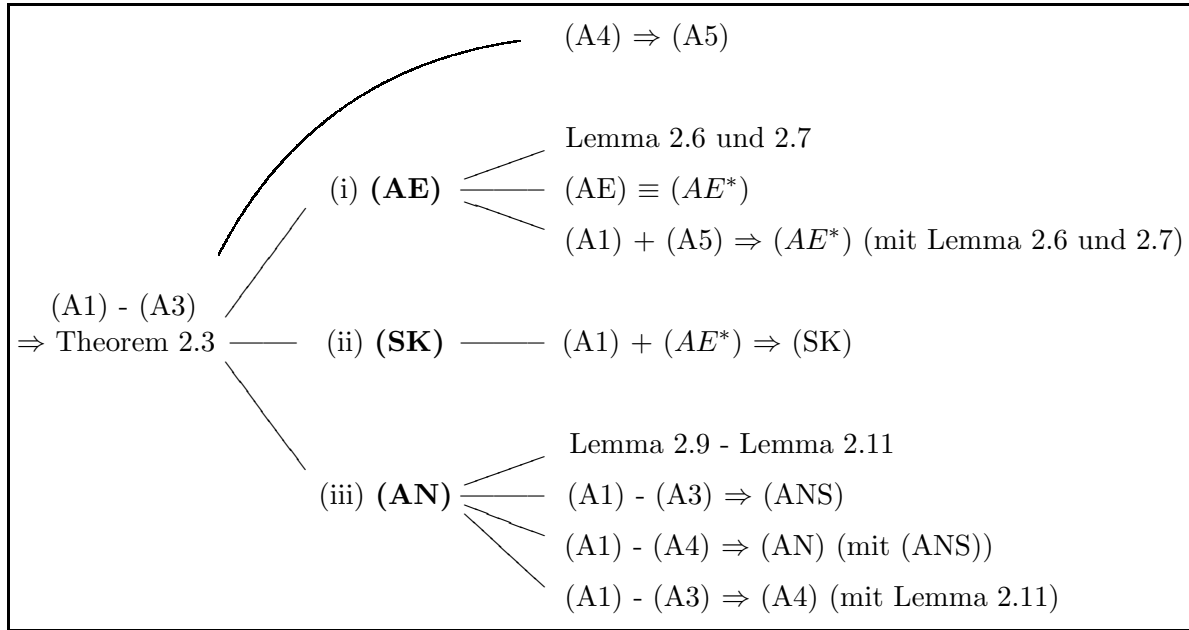
$$(i) \quad P(s_n(\hat{\boldsymbol{\delta}}_n) = 0) \rightarrow 1, \text{ wenn } n \rightarrow \infty \text{ (asymptotische Existenz (AE))},$$

$$(ii) \quad \hat{\boldsymbol{\delta}}_n \xrightarrow{P} \boldsymbol{\delta}_0, \text{ wenn } n \rightarrow \infty \text{ (schwache Konsistenz (SK))},$$

$$(iii) \quad \mathbf{F}_n^{t/2}(\hat{\boldsymbol{\delta}}_n - \boldsymbol{\delta}_0) \xrightarrow{D} N_p(\mathbf{0}, \mathbf{I}_{p+r+q+3}), \text{ wenn } n \rightarrow \infty \text{ (asymptotische Normalität (AN))}.$$

Es fällt auf, daß als Voraussetzungen nur (A1)–(A3) gewählt werden. Wie gezeigt werden wird, sind (A4) und (A5) redundant. Sie dienen jedoch der Vereinfachung und guten Strukturierung des Beweises. Der Beweis des Theorems folgt dem Beweis in Fahrmeir und Kaufmann [10] und Czado und Min [7]. Der gesamte Beweis wird für den Fall kompakter Regressoren entwickelt (dies ist Annahme (A2)); Fahrmeir und Kaufmann [10, S. 349] betrachten jedoch auch den Fall stochastischer Regressoren mit beschränktem Range.

Zur Übersicht des Beweisaufbaus dient folgendes Struktogramm, das ausschnittsweise immer wieder im Beweis auftaucht. Dort wird dann jeweils die Aussage umrahmt, die als nächstes gezeigt wird.



Begonnen wird damit zu zeigen, daß (A5) bereits von (A4) impliziert wird. Anschließend beginnt man damit, Theorem 2.3 zu beweisen. Man beginnt mit der asymptotischen Existenz (AE). Die beiden kurzen Lemmas 2.6 und 2.7 werden später verwendet, um (AE*) zu beweisen. Dann wird gezeigt, daß mit (AE*) eine äquivalente Darstellung der asymptotischen Äquivalenz (AE) vorliegt.

Auch der Beweis der schwachen Konsistenz (SK) verwendet die Darstellung (AE*). Sowohl für den Beweis der Asymptotischen Existenz als auch für die schwache Konsistenz werden jeweils nur die Annahmen (A1) und (A5) benötigt. Man benötigt hier also noch keine kompakten Regressoren.

Zuletzt wird die asymptotische Normalität der ML-Schätzer (AN) gezeigt. Dafür werden als Vorbereitung die Lemmas 2.9 - 2.11 gezeigt. Die asymptotische Normalität der Scoregleichungen werde mit (ANS) bezeichnet. Diese wird unter den Annahmen (A1) - (A4) bewiesen. Während der Beweis von (AN) zunächst ebenfalls unter den Annahmen (A1) - (A4) geführt wird, wird danach vorführt, daß man mit Hilfe von Lemma 2.9 zeigen kann, daß sich (A4) bereits aus (A1) - (A3) ergibt.

Man beginnt damit zu zeigen, daß (A5) aus (A4) folgt.

Satz 2.4 ("(A4) ⇒ (A5)").

Konvergiert die normierte Fisher-Information in Wahrscheinlichkeit gegen die Einheitsmatrix, d.h.

$$\max_{\delta \in N_n(\varepsilon)} \|\mathbf{V}_n(\delta) - \mathbf{I}_{p+r+q+3}\| \xrightarrow{P} 0 \quad \forall \varepsilon > 0 \quad (A4),$$

so gilt

$$P_{\delta_0} \{ \mathbf{H}_n(\delta) - c\mathbf{F}(\delta_0) \text{ ist positiv semidefinit für alle } \delta \in N_n(\varepsilon) \} \rightarrow 1 \quad (A5).$$

Beweis. Fahrmeir und Kaufmann [10, S. 360] behaupten, daß (A4) ⇒ (A5). Einen Beweis dieser Aussage geben sie jedoch nicht. Annahme (A4) lautet, daß für alle $\eta > 0$ gilt:

$$P \left(\omega : \max_{\delta \in N_n(\varepsilon)} (\|\mathbf{V}_n(\delta) - \mathbf{I}_{p+r+q+3}\|_2) \leq \eta \right) \rightarrow 1$$

$$\begin{aligned}
&\Leftrightarrow P\left(\omega : \max_{\delta \in N_n(\varepsilon)} \max_{\mathbf{x}: \|\mathbf{x}\|=1} \left| \mathbf{x}^t \left(\mathbf{F}_n^{-1/2} \mathbf{H}_n(\delta) \mathbf{F}_n^{-t/2} - \mathbf{F}_n^{-1/2} \mathbf{F}_n \mathbf{F}_n^{-t/2} \right) \mathbf{x} \right| \leq \eta \right) \rightarrow 1 \\
&\hspace{15em} (\mathbf{x} \in \mathbb{R}^{p+r+q+3}) \\
&\Leftrightarrow P\left(\omega : \max_{\delta \in N_n(\varepsilon)} \max_{\mathbf{x}: \|\mathbf{x}\|=1} \left| \mathbf{x}^t \mathbf{F}_n^{-1/2} (\mathbf{H}_n(\delta) - \mathbf{F}_n) \mathbf{F}_n^{-t/2} \mathbf{x} \right| \leq \eta \underbrace{\|\mathbf{x}\|^2}_{=1} \right) \rightarrow 1 \\
&\hspace{15em} (\text{substituiere } \mathbf{u} := \mathbf{F}_n^{-t/2} \mathbf{x}, \|\mathbf{x}\| = 1) \\
&P(\omega : \max_{\delta \in N_n(\varepsilon)} |\mathbf{u}^t (\mathbf{H}_n(\delta) - \mathbf{F}_n) \mathbf{u}| \leq \eta \mathbf{u}^t \mathbf{F}_n \mathbf{u}) \rightarrow 1.
\end{aligned}$$

Nun gilt aber

$$\begin{aligned}
&P(\omega : \max_{\delta \in N_n(\varepsilon)} |\mathbf{u}^t (\mathbf{H}_n(\delta) - \mathbf{F}_n) \mathbf{u}| \leq \eta \underbrace{\mathbf{u}^t \mathbf{F}_n \mathbf{u}}_{\text{unabh. von } \delta}) \\
&= P(\omega : |\mathbf{u}^t (\mathbf{H}_n(\delta) - \mathbf{F}_n) \mathbf{u}| \leq \eta \mathbf{u}^t \mathbf{F}_n \mathbf{u} \quad \forall \delta \in N_n(\varepsilon), \forall \eta > 0) \\
&= P(\omega : -\eta \mathbf{u}^t \mathbf{F}_n \mathbf{u} \leq \mathbf{u}^t \mathbf{H}_n(\delta) \mathbf{u} - \mathbf{u}^t \mathbf{F}_n \mathbf{u} \leq \eta \mathbf{u}^t \mathbf{F}_n \mathbf{u} \quad \forall \delta \in N_n(\varepsilon), \forall \eta > 0) \\
&= P(\omega : (1 - \eta) \mathbf{u}^t \mathbf{F}_n \mathbf{u} \leq \mathbf{u}^t \mathbf{H}_n(\delta) \mathbf{u} \leq (1 + \eta) \mathbf{u}^t \mathbf{F}_n \mathbf{u} \quad \forall \delta \in N_n(\varepsilon), \forall \eta > 0).
\end{aligned}$$

Die rechte Ungleichung ist uninteressant, da sie für beliebige $\eta > 0$ immer erfüllt ist. Für die Ungleichung links kann man nun für ein bestimmtes η eine Konstante $c := 1 - \eta$ wählen. Diese Ersetzung schwächt die Ungleichung ab, womit sie von mehr ω erfüllt wird. Insgesamt gilt also die Abschätzung

$$\begin{aligned}
&P(\omega : (1 - \eta) \mathbf{u}^t \mathbf{F}_n \mathbf{u} \leq \mathbf{u}^t \mathbf{H}_n(\delta) \mathbf{u} \leq (1 + \eta) \mathbf{u}^t \mathbf{F}_n \mathbf{u} \quad \forall \delta \in N_n(\varepsilon), \forall \eta > 0) \\
&\leq P(\omega : \mathbf{u}^t \mathbf{H}_n(\delta) \mathbf{u} \geq c \mathbf{u}^t \mathbf{F}_n \mathbf{u} \quad \forall \delta \in N_n(\varepsilon)) \leq 1.
\end{aligned}$$

Mit der Konvergenz von $P(\omega : \max_{\delta \in N_n(\varepsilon)} (\|\mathbf{V}_n(\delta) - \mathbf{I}_{p+r+q+3}\|_2) \leq \eta) \rightarrow 1$ muß dann auch

$$P(\omega : \mathbf{u}^t \mathbf{H}_n(\delta) \mathbf{u} \geq c \mathbf{u}^t \mathbf{F}_n \mathbf{u} \quad \forall \delta \in N_n(\varepsilon)) \rightarrow 1$$

gelten. Damit ist (A5) gezeigt. □

2.3.3 Beweis der asymptotischen Existenz und der schwachen Konsistenz der Lösungen der Scoregleichungen

Zuerst soll Aussage (i) gezeigt werden, die im folgenden Satz noch einmal wiederholt wird.

Satz 2.5 (Asymptotische Existenz).

Es gelten die Annahmen (A1) und (A5). Dann gibt es eine Folge von Zufallsvariablen $\hat{\delta}_n$, die Lösungen der Score-Gleichungen $\mathbf{s}_n(\delta) = \mathbf{0}$ sind, so daß

$$(i) \quad P(\mathbf{s}_n(\hat{\delta}_n) = \mathbf{0}) \rightarrow 1 \quad (\text{asymptotische Existenz}).$$

Für den Beweis von Satz 2.5 werden zunächst die beiden Lemmas 2.6 und 2.7 benötigt.

Lemma 2.6 und 2.7	
(i) (AE)	$(AE) \equiv (AE^*)$ $(A1) + (A5) \Rightarrow (AE^*)$ (mit Lemma 2.6 und 2.7)

Lemma 2.6 (Abschätzung der quadr. Form $\boldsymbol{\lambda}^t \mathbf{F}_n^{-1/2} \mathbf{H}_n(\tilde{\boldsymbol{\delta}}) \mathbf{F}_n^{-t/2} \boldsymbol{\lambda}$).

Wenn $\mathbf{H}_n(\boldsymbol{\delta}) - c\mathbf{F}_n$ f.s. positiv semidefinit ist für alle $\boldsymbol{\delta} \in N_n(\varepsilon)$ (d.h. f.s. $\mathbf{x}^t(\mathbf{H}_n(\boldsymbol{\delta}) - c\mathbf{F}_n)\mathbf{x} \geq 0$ für alle $\mathbf{x} \in \mathbb{R}^p$, $\mathbf{x} \neq 0$), dann gilt mit $\boldsymbol{\lambda} = \frac{1}{\varepsilon} \mathbf{F}_n^{t/2}(\boldsymbol{\delta} - \boldsymbol{\delta}_0)$

$$\boldsymbol{\lambda}^t \mathbf{F}_n^{-1/2} \mathbf{H}_n(\tilde{\boldsymbol{\delta}}) \mathbf{F}_n^{-t/2} \boldsymbol{\lambda} \geq c \quad \text{f.s.}$$

Annahme (A5) fordert, daß positive Definitheit in Wahrscheinlichkeit gegeben ist.

Beweis.

$$\begin{aligned}
\boldsymbol{\lambda}^t \mathbf{F}_n^{-1/2} \mathbf{H}_n(\tilde{\boldsymbol{\delta}}) \mathbf{F}_n^{-t/2} \boldsymbol{\lambda} &= \frac{1}{\varepsilon^2} (\boldsymbol{\delta} - \boldsymbol{\delta}_0)^t \mathbf{H}_n(\tilde{\boldsymbol{\delta}}) (\boldsymbol{\delta} - \boldsymbol{\delta}_0) \\
&\geq \frac{c}{\varepsilon^2} (\boldsymbol{\delta} - \boldsymbol{\delta}_0)^t \cdot \mathbf{F}_n \cdot (\boldsymbol{\delta} - \boldsymbol{\delta}_0) \\
&= \frac{c}{\varepsilon^2} (\boldsymbol{\delta} - \boldsymbol{\delta}_0)^t \cdot \mathbf{F}_n^{1/2} \mathbf{F}_n^{t/2} \cdot (\boldsymbol{\delta} - \boldsymbol{\delta}_0) = \frac{c}{\varepsilon^2} \|\mathbf{F}_n^{t/2}(\boldsymbol{\delta} - \boldsymbol{\delta}_0)\|^2 \\
(\text{ mit } \boldsymbol{\delta} \in \partial N_n(\varepsilon)) &= \frac{c}{\varepsilon^2} \cdot \varepsilon^2 \\
&= c.
\end{aligned}$$

□

Lemma 2.7 (Lancaster).

Sei \mathbf{s}_n ein Zufallsvektor mit $E\mathbf{s}_n = \mathbf{0}$ und $\text{Cov}(\mathbf{s}_n) = \mathbf{F}_n$. Ferner sei eine \mathbf{A} eine symmetrische Matrix mit $\mathbf{A} := \mathbf{F}_n^{-t}$ und $\text{tr}(\cdot)$ bezeichne ihre Spur. Dann gilt $E(\mathbf{s}_n^t \mathbf{A} \mathbf{s}_n) = \text{tr}(\mathbf{A} \mathbf{F}_n)$.

Beweis. Der Beweis kann bei Seber [33, S. 13] gefunden werden. □

Mit Hilfe dieser beiden Lemmas kann man nun an den Beweis von Satz 2.5 herangehen.

Beweis. [Satz 2.5]

Nun leitet man eine äquivalente Darstellung (AE^*) der asymptotischen Existenz der Lösungen der Scoregleichungen her. Diese Darstellung (AE^*) kann man danach dann leichter als (AE) zeigen.

Lemma 2.6 und 2.7	
(i) (AE)	$(AE) \equiv (AE^*)$ $(A1) + (A5) \Rightarrow (AE^*)$ (mit Lemma 2.6 und 2.7)

Man definiere Zufallsvariablen $\hat{\boldsymbol{\delta}}_n$ als Nullstellen der Score-Funktionen (wenn sie existieren), d.h. $\mathbf{s}_n(\hat{\boldsymbol{\delta}}_n) = 0$, und andernfalls als beliebige Konstanten aus der zulässigen Menge $B \times A \times G$, aus der auch $\boldsymbol{\delta}$ stammt.

Für jedes n und jedes ε bedeutet das Ereignis

$$l_n(\boldsymbol{\delta}) - l_n(\boldsymbol{\delta}_0) < 0 \quad \text{für alle} \quad \boldsymbol{\delta} \in \partial N_n(\varepsilon),$$

daß es mindestens ein lokales Maximum von $l_n(\boldsymbol{\delta})$ in $N_n(\varepsilon)$ gibt (beachte, daß das Ereignis $\boldsymbol{\delta}_0 \in N_n(\varepsilon)$ und ein lokales Maximum noch nicht notwendigerweise ein globales Maximum ergeben).

Um die *erste Aussage (i)* zu beweisen, genügt es zu zeigen, daß für jedes $\tilde{\eta} > 0$ ein $\varepsilon > 0$ und ein n_1 existieren, so daß

$$P \{l_n(\boldsymbol{\delta}) - l_n(\boldsymbol{\delta}_0) < 0 \text{ für alle } \boldsymbol{\delta} \in \partial N_n(\varepsilon)\} \geq 1 - \tilde{\eta} \quad (\mathbf{AE}^*)$$

oder äquivalent dazu

$$P \left\{ \bigcap_{\boldsymbol{\delta} \in \partial N_n(\varepsilon)} \{\omega : l_n(\boldsymbol{\delta}) - l_n(\boldsymbol{\delta}_0) < 0\} \right\} \geq 1 - \tilde{\eta} \quad (2.8)$$

für alle $n \geq n_1$, da $P \{s_n(\hat{\boldsymbol{\delta}}_n) = 0\} \geq P \left\{ \bigcap_{\boldsymbol{\delta} \in \partial N_n(\varepsilon)} \{\omega : l_n(\boldsymbol{\delta}) - l_n(\boldsymbol{\delta}_0) < 0\} \right\}$. (Es kann mehrere lokale Maxima mit $s_n(\hat{\boldsymbol{\delta}}_n) = 0$ geben.)

Jetzt kann man an den Beweis von (AE^*) herangehen. Ist (AE^*) gezeigt, so ist auch (AE) bewiesen.

(i) (AE)	\swarrow \longrightarrow \searrow	Lemma 2.6 und 2.7 $(AE) \equiv (AE^*)$
		$(A1) + (A5) \Rightarrow (AE^*)$ (mit Lemma 2.6 und 2.7)

Sei nun $\boldsymbol{\lambda} = \frac{1}{\varepsilon} \mathbf{F}_n^{t/2}(\boldsymbol{\delta} - \boldsymbol{\delta}_0)$, dann ist die Taylorentwicklung des Loglikelihood

$$l_n(\boldsymbol{\delta}) - l_n(\boldsymbol{\delta}_0) = \varepsilon \boldsymbol{\lambda}^t \mathbf{F}_n^{-1/2} \mathbf{s}_n - \frac{\varepsilon^2}{2} \boldsymbol{\lambda}^t \mathbf{F}_n^{-1/2} \mathbf{H}_n(\tilde{\boldsymbol{\delta}}) \mathbf{F}_n^{-t/2} \boldsymbol{\lambda},$$

wobei $\tilde{\boldsymbol{\delta}}$ zwischen $\boldsymbol{\delta}$ und $\boldsymbol{\delta}_0$ liegt.

Man beachte, daß für $\boldsymbol{\delta} \in \partial N_n(\varepsilon)$ $\boldsymbol{\lambda}^t \boldsymbol{\lambda} = \frac{\|\mathbf{F}_n^{t/2}(\boldsymbol{\delta} - \boldsymbol{\delta}_0)\|^2}{\varepsilon^2} = 1$ gilt. Benutzt man nun $\|\mathbf{F}_n^{-1/2} \mathbf{s}_n\| := \sup_{\|\boldsymbol{\lambda}\|=1} \boldsymbol{\lambda}^t \mathbf{F}_n^{-1/2} \mathbf{s}_n$, so erkennt man, daß es zu zeigen genügt, daß für jedes $\tilde{\eta} = \eta_1 + \eta_2 > 0$ gilt:

$$P \left\{ \bigcap_{\boldsymbol{\lambda}: \|\boldsymbol{\lambda}\|=1} \left\{ \omega : \varepsilon \|\mathbf{F}_n^{-1/2} \mathbf{s}_n\| < \frac{\varepsilon^2}{2} \boldsymbol{\lambda}^t \mathbf{F}_n^{-1/2} \mathbf{H}_n(\tilde{\boldsymbol{\delta}}) \mathbf{F}_n^{-t/2} \boldsymbol{\lambda} \right\} \right\} \geq 1 - \tilde{\eta}$$

oder äquivalent dazu

$$\Delta_1(n) := P \left\{ \bigcup_{\boldsymbol{\lambda}: \|\boldsymbol{\lambda}\|=1} \left\{ \omega : \varepsilon \|\mathbf{F}_n^{-1/2} \mathbf{s}_n\| \geq \frac{\varepsilon^2}{2} \boldsymbol{\lambda}^t \mathbf{F}_n^{-1/2} \mathbf{H}_n(\tilde{\boldsymbol{\delta}}) \mathbf{F}_n^{-t/2} \boldsymbol{\lambda} \right\} \right\} < \tilde{\eta}.$$

Erklärung: Sind A_λ die Ereignisse in $P(\bigcap_{\lambda: \|\lambda\|=1} A_\lambda)$, dann gilt $P(\bigcup_\lambda A_\lambda) = 1 - P(\bigcap_\lambda A_\lambda^C)$. Sei nun B_n das Ereignis im Argument der Wahrscheinlichkeit von (A5), d.h.

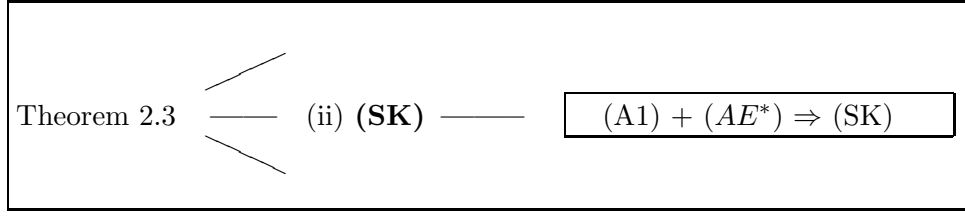
$$B_n = \{\omega : \mathbf{H}_n(\boldsymbol{\delta}) - c\mathbf{F}_n \text{ positiv semidefinit für alle } \boldsymbol{\delta} \in N_n(\varepsilon)\}. \quad (2.9)$$

Einfache Argumentation ergibt

$$\begin{aligned} \Delta_1(n) &= P \left\{ \bigcup_{\lambda: \|\lambda\|=1} \left\{ \omega : \varepsilon \|\mathbf{F}_n^{-1/2} \mathbf{s}_n\| \geq \frac{\varepsilon^2}{2} \boldsymbol{\lambda}^t \mathbf{F}_n^{-1/2} \mathbf{H}_n(\tilde{\boldsymbol{\delta}}) \mathbf{F}_n^{-t/2} \boldsymbol{\lambda} \right\} \right\} \\ &= P \left\{ \bigcup_{\lambda: \|\lambda\|=1} \left\{ \omega : \|\mathbf{F}_n^{-1/2} \mathbf{s}_n\| \geq \frac{\varepsilon}{2} \boldsymbol{\lambda}^t \mathbf{F}_n^{-1/2} \mathbf{H}_n(\tilde{\boldsymbol{\delta}}) \mathbf{F}_n^{-t/2} \boldsymbol{\lambda} \right\} \right\} \\ &= P \left\{ \left(\bigcup_{\lambda: \|\lambda\|=1} \left\{ \omega : \|\mathbf{F}_n^{-1/2} \mathbf{s}_n\| \geq \underbrace{\frac{\varepsilon}{2} \boldsymbol{\lambda}^t \mathbf{F}_n^{-1/2} \mathbf{H}_n(\tilde{\boldsymbol{\delta}}) \mathbf{F}_n^{-t/2} \boldsymbol{\lambda}}_{\geq c} \right\} \right) \cap B_n \right\} \\ &\quad + P \left\{ \left(\bigcup_{\lambda: \|\lambda\|=1} \left\{ \omega : \|\mathbf{F}_n^{-1/2} \mathbf{s}_n\| \geq \underbrace{\frac{\varepsilon}{2} \boldsymbol{\lambda}^t \mathbf{F}_n^{-1/2} \mathbf{H}_n(\tilde{\boldsymbol{\delta}}) \mathbf{F}_n^{-t/2} \boldsymbol{\lambda}}_{\geq c} \right\} \right) \cap B_n^C \right\} \\ &\quad \text{(mit Lemma 2.6 mit (A5) für hinreichend großes } n) \\ &\leq P \left\{ \bigcup_{\lambda: \|\lambda\|=1} \left(\left\{ \omega : \|\mathbf{F}_n^{-1/2} \mathbf{s}_n\| \geq \frac{\varepsilon c}{2} \right\} \cap B_n \right) \right\} + \eta_1 \\ &\quad \text{(der zweite Term wird durch } \eta_1 \leq \tilde{\eta} \text{ abgeschätzt)} \\ &\leq P \left\{ \|\mathbf{F}_n^{-1/2} \mathbf{s}_n\|^2 \geq \frac{c^2 \varepsilon^2}{4} \right\} + \eta_1 \\ &\quad \text{(Chebyshev-Ungleichung: } P(\|X\| \geq z) \leq \frac{E\|X\|^2}{z^2} \text{ und} \\ &\quad P(\|X\|^2 \geq z^2) = P(\|X\| \geq z), \text{ da hier } \|X\|, z \geq 0) \\ &\leq \frac{4E\|\mathbf{F}_n^{-1/2} \mathbf{s}_n\|^2}{\varepsilon^2 c^2} + \eta_1 \\ &\quad \text{(mit Lemma 2.7: Theorem nach Lancaster: } E\|\mathbf{F}_n^{-1/2} \mathbf{s}_n\|^2 = \text{tr}(\mathbf{F}_n^{-1} \mathbf{F}_n) \\ &= \frac{4(p+r+q+3)}{\varepsilon^2 c^2} + \eta_1 \\ &\quad \left(\text{für } \varepsilon = \sqrt{\frac{4(p+r+q+3)}{\eta_2 c^2}} \right) \\ &= \eta_2 + \eta_1 = \tilde{\eta}. \end{aligned}$$

Somit ist $(\mathbf{A}\mathbf{E}^*)$ bewiesen. □

Der Beweis der schwachen Konsistenz in Aussage (ii) ist kurz. Als Voraussetzung benötigt man (A1) und die Darstellung $(\mathbf{A}\mathbf{E}^*)$.



Noch einmal zur Erinnerung:

Satz 2.8 (Schwache Konsistenz).

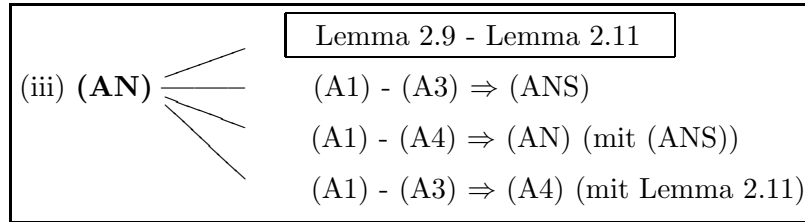
Es gelte die Annahme (A1). Dann gibt es eine Folge von Zufallsvariablen $\hat{\delta}_n$, die Lösungen der Score-Gleichungen $\mathbf{s}_n(\delta) = \mathbf{0}$ sind, so daß

$$(ii) \quad \hat{\delta}_n \xrightarrow{P} \delta_0, \text{ wenn } n \rightarrow \infty \quad (\text{schwache Konsistenz}).$$

Beweis. Man kann sich überlegen: da die Annahme (A1) impliziert, daß die Umgebungen $N_n(\varepsilon)$ gegen δ_0 konvergieren und wiederum $\hat{\delta} \in N_n(\varepsilon)$ ist, folgt aus (\mathbf{AE}^*) auch die zweite Aussage (ii) des Theorems. Schließlich konvergiert $N_n(\varepsilon)$ gegen δ_0 , da mit $n \rightarrow \infty$ nach (A1) auch $\lambda_{\min}(\mathbf{F}_n)$ wächst ($\frac{n}{c_1} \leq \lambda_{\min}(\mathbf{F}_n)$). Wegen $\lambda_{\min}(\mathbf{F}_n) \cdot \|\hat{\delta}_n - \delta_0\|^2 \leq (\hat{\delta}_n - \delta_0)^t \mathbf{F}_n (\hat{\delta}_n - \delta_0) = \|\mathbf{F}_n^{t/2}(\hat{\delta}_n - \delta_0)\|^2 < \varepsilon^2$ f.s. muß mit $n \rightarrow \infty$ wegen der letzten Ungleichung f.s. $\hat{\delta} \xrightarrow{P} \delta_0$ gelten. \square

2.3.4 Beweis der asymptotischen Normalität der Scorevektoren \mathbf{s}_n

Zunächst werden zwei Lemmas gezeigt, die für den Beweis der asymptotischen Normalität (AN) benötigt werden.



Lemma 2.9.

Sei $Y \sim ZIGP(\mu, \varphi, \omega)$ und $\tilde{Y} \sim GP(\mu, \varphi)$, dann gilt für jede meßbare Funktion u

$$E_{\delta}(I_{\{Y>0\}} u(Y)) = (1 - \omega) E_{\delta} \left(u(\tilde{Y}) \right) - u(0)(1 - \omega) \exp \left(-\frac{\mu}{\varphi} \right) \quad (2.10)$$

sowie

$$E_{\delta}(I_{\{Y=0\}} u(Y)) = u(0) \left(\omega + (1 - \omega) e^{-\frac{\mu}{\varphi}} \right). \quad (2.11)$$

Wenn ferner u auf $[0, \infty)$ nicht-negativ ist, dann gilt die Abschätzung

$$E_{\delta}(I_{\{Y>0\}} u(Y)) \leq (1 - \omega) E_{\delta} \left(u(\tilde{Y}) \right). \quad (2.12)$$

Beweis.

$$\begin{aligned}
E_{\delta}(I_{\{Y>0\}}u(Y)) &= \sum_{y=1}^{\infty} u(y)P_{ZIGP}(y) \\
&= (1-\omega) E\left(u(\tilde{Y})\right) - u(0)(1-\omega) P_{GP}(0) \\
&= (1-\omega) E\left(u(\tilde{Y})\right) - u(0)(1-\omega) \exp\left(-\frac{\mu}{\varphi}\right)
\end{aligned}$$

Die zweite Aussage ist offensichtlich. Die Abschätzung (2.12) folgt aus der ersten Aussage, indem man den letzten Term wegläßt. \square

Lemma 2.10.

Sei $\tilde{Y}_i \sim GP(\mu_i, \varphi_i)$, $i = 1, \dots, n$, eine Folge von Zufallsvariablen. Dann gilt unter (A2) und (A3), daß für jedes endliche $k > 0$

$$\begin{aligned}
\max_{i=1, \dots, n} E\left(\frac{1}{(\mu_i + (\varphi_i - 1)\tilde{Y}_i)^k}\right) &\leq C_1 \\
\max_{i=1, \dots, n} E(\tilde{Y}_i^k) &\leq C_2,
\end{aligned}$$

wobei C_1 und C_2 positive Konstanten sind, die nur von k und δ_0 abhängen.

Beweis.

Nach (A3) gilt $\varphi_i \in (1, \infty) \ \forall i$. Somit kann man schließen, daß

$$E\left(\frac{1}{(\mu_i + (\varphi_i - 1)\tilde{Y}_i)^k}\right) \leq \frac{1}{\mu_i^k}. \quad (2.13)$$

Nun folgt, daß

$$\begin{aligned}
\max_{i=1, \dots, n} \frac{1}{\mu_i^k} &= \max_{i=1, \dots, n} \frac{1}{\exp(k\mathbf{x}_i^t \boldsymbol{\beta}_0)} \\
&\leq \max_{\mathbf{x} \in K_x} \frac{1}{\exp(k\mathbf{x}^t \boldsymbol{\beta}_0)} \\
&\stackrel{(A2)}{\leq} C_1(\boldsymbol{\beta}_0, k),
\end{aligned}$$

da K_x unter (A2) ein Kompaktum ist und $\exp(k\mathbf{x}^t \boldsymbol{\beta}_0)$ eine stetige Funktion von \mathbf{x} . Es sollte auch darauf hingewiesen werden, daß $C_1(\boldsymbol{\beta}_0, k)$ stetig in Bezug auf $\boldsymbol{\beta}_0$ und wohldefiniert für alle $\boldsymbol{\beta}_0 \in B$ ist.

Nun zur zweiten Ungleichung des Lemmas. Zunächst wird die GP-Verteilung umparametrisiert, indem die Parameter $\theta_i := \frac{\mu_i}{\varphi_i}$ und $\lambda_i := \frac{\varphi_i - 1}{\varphi_i}$, $i = 1, \dots, n$ verwendet werden.

Consul und Shenton [34] haben folgende Rekursionsformel für die nicht-zentralen Momente der $GP(\theta_i, \lambda_i)$ -Verteilung gezeigt:

$$(1 - \lambda_i)m_{i,k+1} = \theta_i m_{i,k} + \theta_i \frac{\partial m_{i,k}}{\partial \theta_i} + \lambda_i \frac{\partial m_{i,k}}{\partial \lambda_i}, \quad k = 0, 1, 2, \dots, \quad (2.14)$$

wobei $m_{i,k} := E(\tilde{Y}_i^k)$, also $m_{i,1} = \sum_{\tilde{y}=0}^{\infty} \tilde{y} P(\mu_i, \varphi_i)$, $m_{i,2} = \sum_{\tilde{y}=0}^{\infty} \tilde{y}^2 P(\mu_i, \varphi_i)$, etc. Löst man diese Rekursionsformel für festes k nach $m_{i,k}$ auf, so stellt man fest, daß $m_{i,k}$ ein Polynom in θ_i , λ_i und $1/(1 - \lambda_i)$ ist. Somit ist $m_{i,k}$ auch stetig in (μ_i, φ_i) . Es folgt nun, daß

$$\begin{aligned} \max_{i=1,\dots,n} E(\tilde{Y}_i^k) &= \max_{i=1,\dots,n} m_{i,k}(\theta_i, \lambda_i) \\ &= \max_{i=1,\dots,n} m_{i,k} \left(\frac{\mu_i}{\varphi_i}, \frac{\varphi_i - 1}{\varphi_i} \right) \\ &\leq \max_{\mathbf{x} \in K_x, \mathbf{w} \in K_w} m_k \left(\frac{e^{\mathbf{x}^t \boldsymbol{\beta}_0}}{1 + e^{\mathbf{w}^t \boldsymbol{\alpha}_0}}, \frac{e^{\mathbf{w}^t \boldsymbol{\alpha}_0}}{1 + e^{\mathbf{w}^t \boldsymbol{\alpha}_0}} \right) \\ &\leq C_2(\boldsymbol{\delta}_0) \end{aligned}$$

wobei $m_k := E(\tilde{Y}^k)$ und $Y \sim GP(e^{\mathbf{x}^t \boldsymbol{\beta}_0}, 1 + e^{\mathbf{w}^t \boldsymbol{\alpha}_0})$. Man kann leicht sehen, daß $C_2(\boldsymbol{\delta}_0)$ stetig in $\boldsymbol{\delta}_0$ und wohldefiniert für alle $\boldsymbol{\delta}_0 \in K_{\boldsymbol{\delta}}$ ist. \square

Lemma 2.11.

Sei $Q_k(y)$ ein Polynom beschränkter Ordnung k ($k \in \mathbb{N}$), dessen Koeffizienten positive stetige Funktionen von \mathbf{x} , $\boldsymbol{\delta}$ und $\boldsymbol{\delta}_0$ sind.

Ferner sei $Y_i \sim ZIGP \left(\exp(\mathbf{x}_i^t \boldsymbol{\beta}_0), 1 + e^{\mathbf{w}_i^t \boldsymbol{\alpha}_0}, \frac{\exp(\mathbf{z}_i^t \boldsymbol{\gamma}_0)}{1 + \exp(\mathbf{z}_i^t \boldsymbol{\gamma}_0)} \right)$ für $i = 1, \dots, n$. Sind (A1) bis (A3) erfüllt, so gilt

$$\max_{\boldsymbol{\delta} \in N_n(\varepsilon)} \max_{i=1,\dots,n} E(\mathbb{1}_{\{y_i > 0\}} Q_k(Y_i)) < C, \quad (2.15)$$

wobei C eine positive Konstante ist, die von k und $\boldsymbol{\delta}_0$ abhängt.

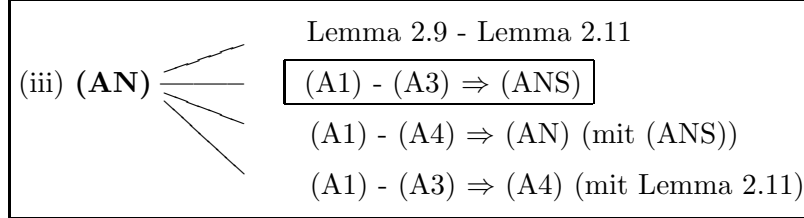
Beweis.

Man beachte, daß unter (A1) die Umgebung $N_n(\varepsilon)$ für jedes $n \in \mathbb{N}$ ein Kompaktum ist und für jedes $\varepsilon > 0$ mit $n \rightarrow \infty$ gegen $\boldsymbol{\delta}_0$ schrumpft. Unter Benutzung der Lemmas 2.9 und 2.10, folgt nun

$$\begin{aligned} \max_{\boldsymbol{\delta} \in N_n(\varepsilon)} \max_{i=1,\dots,n} E(\mathbb{1}_{\{y_i > 0\}} Q_k(Y_i)) &\stackrel{\text{Lemma 2.9}}{\leq} \max_{\boldsymbol{\delta} \in N_n(\varepsilon)} \max_{i=1,\dots,n} \left(\frac{1}{1 + e^{\mathbf{z}_i^t \boldsymbol{\gamma}}} \right) E \left(Q_k(\tilde{Y}_i) \right) \\ &\stackrel{(A2)}{\leq} \max_{\boldsymbol{\delta} \in N_1(\varepsilon)} \max_{\mathbf{x} \in K_x, \mathbf{z} \in K_z} \left(\frac{1}{1 + e^{\mathbf{z}^t \boldsymbol{\gamma}}} \right) E \left(Q_k(\tilde{Y}) \right) \\ &\stackrel{\text{Lemma 2.10}}{\leq} C, \end{aligned}$$

wobei $\tilde{Y}_i \sim GP(\exp(\mathbf{x}_i^t \boldsymbol{\beta}_0), 1 + e^{\mathbf{w}_i^t \boldsymbol{\alpha}_0})$ und $\tilde{Y} \sim GP(\exp(\mathbf{x}^t \boldsymbol{\beta}_0), 1 + e^{\mathbf{w}^t \boldsymbol{\alpha}_0})$. \square

Man kann zeigen, daß auch die Scoregleichung asymptotisch normalverteilt sind, das ist Bedingung (ANS). Diese Eigenschaft sowie Lemma 2.9, 2.10 und (ANS)) wird später für den Beweis der asymptotischen Normalität der *Lösungen* der Scoregleichungen verwendet.



Satz 2.12 (Asymptotische Normalität der Scoregleichungen).

Unter den Annahmen (A1) bis (A3) gilt

$$\mathbf{F}_n^{-1/2} \mathbf{s}_n \xrightarrow{D} N_{p+r+q+3}(\mathbf{0}, \mathbf{I}_{p+r+q+3}), \quad \text{wenn } n \rightarrow \infty.$$

Hierbei ist $N_{p+r+q+3}(\mathbf{0}, \mathbf{I}_{p+r+q+3})$ die $(p+r+q+3)$ - dimensionale Normalverteilung mit Erwartungsvektor $\mathbf{0}$ und Kovarianzmatrix $\mathbf{I}_{p+r+q+3}$.

Beweis.

Mit Hilfe des **Cramer-Wald-Instruments** (z.B. Gut [16, S. 246]) ist es hinreichend zu zeigen, daß eine Linearkombination $\mathbf{a}^t \mathbf{F}_n^{-1/2} \mathbf{s}_n$ in Verteilung gegen $N(0, \mathbf{a}^t \mathbf{a})$ für beliebigen Vektor $\mathbf{a} \in \mathbb{R}^{p+r+q+3}$, $\mathbf{a} \neq \mathbf{0}$, konvergiert. O.B.d.A. setzt man $\|\mathbf{a}\| = 1$.

Da \mathbf{s}_n als Summe unabhängiger Zufallsvariablen geschrieben werden kann, nämlich $\mathbf{s}_n = \sum_{i=1}^n \mathbf{s}_{ni}$, wobei $\mathbf{s}_{ni} = (s_{0,i}, \dots, s_{p,i}, s_{p+1,i}, s_{p+2,i}, \dots, s_{p+r+q+2,i})^t$ mit $s_{k,i} := s_{k,i}(\boldsymbol{\delta}_0) = \frac{\partial l}{\partial \delta_k} |_{\boldsymbol{\delta}=\boldsymbol{\delta}_0}$, wie in (1.14), (1.15) und (1.16) für $k = 0, \dots, (p+r+q+2)$ und $i = 1, \dots, n$ definiert. Ferner werden unabhängige Zufallsvariablen $\xi_{in} := \mathbf{a}^t \mathbf{F}_n^{-1/2} \mathbf{s}_{ni}$ definiert. Da $E(\xi_{in}) = 0$ und $Var(\sum_{i=1}^n \xi_{in}) = 1$, genügt es nach dem **Zentralen Grenzwertsatz im \mathbb{R}^d** zu zeigen, daß die **Lyapunov-Bedingung** erfüllt ist (beides zu finden in Müller [27, S. 507f]), d.h.

$$L_s := \sum_{i=1}^n E|\xi_{in}|^s \xrightarrow{n \rightarrow \infty} 0, \quad \text{für beliebiges } s > 2,$$

hier z.B. für $s = 3$.

Da $\|\mathbf{F}_n^{-1/2}\|^2 = \lambda_{\max}(\mathbf{F}_n^{-1}) = 1/\lambda_{\min}(\mathbf{F}_n)$, folgt aus der Kompatibilitätsbedingung (2.7) und (A1), daß

$$\begin{aligned} L_3 &\leq \sum_{i=1}^n E \left(\|\mathbf{a}^t\|^3 \left\| \mathbf{F}_n^{-1/2} \right\|^3 \|\mathbf{s}_{ni}\|^3 \right) \\ &\stackrel{(A1)}{\leq} \left(\frac{C_1}{n} \right)^{(3/2)} \sum_{i=1}^n E \|\mathbf{s}_{ni}\|^3 \\ &\leq \frac{C}{\sqrt{n}} \max_{i=1, \dots, n} E \|\mathbf{s}_{ni}\|^3. \end{aligned}$$

Es gilt $\frac{C}{\sqrt{n}} \max_{i=1, \dots, n} E \|\mathbf{s}_{ni}\|^3 \xrightarrow{n \rightarrow \infty} 0$, falls $\max_{i=1, \dots, n} E \|\mathbf{s}_{ni}\|^3$ beschränkt.

Dazu nutzt man eine Erweiterung der c_r -Ungleichung

$$E \left| \sum_{i=1}^m \zeta_i \right|^k \leq m^{k-1} \sum_{i=1}^m E |\zeta_i|^k \quad (k > 1, k \in \mathbb{R}), \quad (2.16)$$

die für m beliebige Zufallsvariable ζ_1, \dots, ζ_m gilt. Damit ergibt sich

$$E\|\mathbf{s}_{ni}\|^3 \leq C \left(E|s_{o,i}|^3 + \dots + E|s_{p+r+q+1,i}|^3 + E|s_{p+r+q+2,i}|^3 \right).$$

Somit bleibt zu zeigen, daß auch das Maximum der Summanden $\max_{i=1, \dots, n} E|s_{r,i}|^3$ in n gleichmäßig beschränkt ist für $l = 0, \dots, p+r+q+2$. Dies wird nur für den Fall $l = 0, \dots, p$ gezeigt, die Fälle $l = p+1, \dots, p+r+q+2$ können analog behandelt werden. O.B.d.A setzt man $l = p$. Benutzt man (2.16) mit $m = 2$, so erhält man

$$\begin{aligned} \max_{i=1, \dots, n} E|s_{p,i}|^3 &\leq 2^2 \max_{i=1, \dots, n} E \left(\left| x_{ip} \mathbb{1}_{\{y_i=0\}} \left[\frac{-\frac{1}{\varphi_i} P_i^0 \mu_i}{k_i + P_i^0} \right] \right|^3 \right) \\ &\quad + 2^2 \max_{i=1, \dots, n} E \left(\left| x_{ip} \mathbb{1}_{\{y_i>0\}} \left[1 + \frac{(y_i - 1)\mu_i}{\mu_i + (\varphi_i - 1)y_i} - \frac{\mu_i}{\varphi_i} \right] \right|^3 \right) \\ &=: 4A_p(\boldsymbol{\delta}_0) + 4B_p(\boldsymbol{\delta}_0). \end{aligned}$$

Nun bleibt noch zu zeigen, daß $A_p(\boldsymbol{\delta}_0) < C_2$ und $B_p(\boldsymbol{\delta}_0) < C_3$, wobei C_2 und C_3 Konstanten sind, die nur von $\boldsymbol{\delta}_0$ abhängen.

Für die erste Abschätzung gilt

$$\begin{aligned} A_p(\boldsymbol{\delta}_0) &\leq \max_{i=1, \dots, n} \|\mathbf{x}\| E \left(\mathbb{1}_{\{y_i=0\}} \left| \frac{-\frac{1}{\varphi_0} P_i^0 \mu_i}{k_i + P_i^0} \right|^3 \right) \tag{2.17} \\ &\stackrel{\text{Lemma 2.9}}{\leq} \max_{u(Y)=|\cdot|^3} \|\mathbf{x}\| \left| \frac{-\frac{1}{\varphi_i} P_i^0 \mu_i}{k_i + P_i^0} \right|^3 (\omega_i + (1 - \omega_i) P_i^0) \\ &\leq \max_{\mathbf{x} \in K_x, \mathbf{w} \in K_w, \mathbf{z} \in K_z} \|\mathbf{x}\| \left| \frac{-\frac{1}{1+e^{\mathbf{w}^t \boldsymbol{\alpha}_0}} \exp\left\{-\frac{e^{\mathbf{x}^t \boldsymbol{\beta}_0}}{1+e^{\mathbf{w}^t \boldsymbol{\alpha}_0}}\right\} e^{\mathbf{x}^t \boldsymbol{\beta}_0}}{e^{\mathbf{z}^t \boldsymbol{\gamma}_0} + \exp\left\{-\frac{e^{\mathbf{x}^t \boldsymbol{\beta}_0}}{1+e^{\mathbf{w}^t \boldsymbol{\alpha}_0}}\right\}} \right|^3 \\ &\quad \left(\frac{e^{\mathbf{z}^t \boldsymbol{\gamma}_0} + \exp\left\{-\frac{e^{\mathbf{x}^t \boldsymbol{\beta}_0}}{1+e^{\mathbf{w}^t \boldsymbol{\alpha}_0}}\right\}}{1 + e^{\mathbf{z}^t \boldsymbol{\gamma}_0}} \right) \\ &\stackrel{(A2)}{\leq} C_2(\boldsymbol{\delta}_0), \end{aligned}$$

wobei sich (2.17) aufgrund von $|x_{ip}| \leq (\sum_{r=0}^p x_{ir}^2)^{1/2} = \|\mathbf{x}_i\|_2$ ergibt.

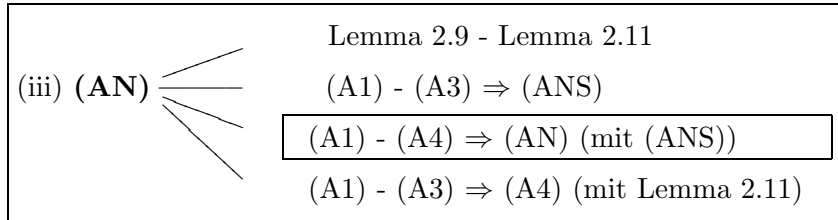
Nun zu $B_p(\boldsymbol{\delta}_0)$. Mit Lemma 2.9, Ungleichung (2.16), der Cauchy-Schwarz-Ungleichung (CSU) sowie Lemma 2.10 erhält man

$$\begin{aligned} B_p(\boldsymbol{\delta}_0) &\stackrel{\text{Lemma 2.9}}{\leq} \max_{u(Y)=|\cdot|^3} E \left((1 - \omega_i) |x_{ir}|^3 \cdot \left| 1 + \frac{\mu_i(\tilde{Y}_i - 1)}{\mu_i + (\varphi_0 - 1)\tilde{Y}_i} - \frac{\mu_i}{\varphi_i} \right|^3 \right) \\ &\stackrel{(2.16)}{\leq} \max_{\mathbf{x} \in K_x, \mathbf{w} \in K_w, \mathbf{z} \in K_z} \frac{1}{1 + e^{\mathbf{z}^t \boldsymbol{\gamma}_0}} \|\mathbf{x}\|^3 \end{aligned}$$

$$\begin{aligned}
& \times \left(1^3 + E \left| \frac{e^{\mathbf{x}^t \boldsymbol{\beta}_0} (\tilde{Y} - 1)}{e^{\mathbf{x}^t \boldsymbol{\beta}_0} + e^{\mathbf{w}^t \boldsymbol{\alpha}_0} \tilde{Y}} \right|^3 + \left(\frac{e^{\mathbf{x}^t \boldsymbol{\beta}_0}}{1 + e^{\mathbf{w}^t \boldsymbol{\alpha}_0}} \right)^3 \right) \\
& \stackrel{\text{Lemma 2.10}}{\leq} C_1(\boldsymbol{\delta}_0) + C_2(\boldsymbol{\delta}_0) \max_{\mathbf{x} \in K_x, \mathbf{w} \in K_w, \mathbf{z} \in K_z} E \left| \tilde{Y} - 1 \right|^3 \\
& \stackrel{\text{CSU}}{\leq} C_1(\boldsymbol{\delta}_0) + C_2(\boldsymbol{\delta}_0) \max_{\mathbf{x} \in K_x, \mathbf{w} \in K_w, \mathbf{z} \in K_z} \sqrt{E \left(\tilde{Y} - 1 \right)^6} \\
& \stackrel{(A2)}{\leq} C_3(\boldsymbol{\delta}_0),
\end{aligned}$$

wobei $\tilde{Y}_i \sim GP(\mu_i, \varphi_i)$ für $i = 1, \dots, n$ und $\tilde{Y} \sim GP(e^{\mathbf{x}^t \boldsymbol{\beta}_0}, 1 + e^{\mathbf{w}^t \boldsymbol{\alpha}_0})$. \square

2.3.5 Beweis der asymptotischen Normalität der ML-Schätzer $\hat{\boldsymbol{\delta}}_n$



Die asymptotische Normalität der Lösungen der Scoregleichungen wird aus Gründen der Übersichtlichkeit zunächst unter den Voraussetzungen (A1) – (A4) gezeigt. Später wird dann nachgewiesen, daß sich (A4) aus (A1) – (A3) ergibt. Der folgende Satz ruft noch einmal die Aussage der asymptotischen Normalität von Lösungen der Scoregleichungen (iii) ins Gedächtnis.

Satz 2.13 (Asymptotische Normalität).

Unter den Annahmen (A1)–(A4) gilt

$$\mathbf{F}_n^{t/2}(\hat{\boldsymbol{\delta}}_n - \boldsymbol{\delta}_0) \sim N_p(\mathbf{0}, \mathbf{I}_{p+r+q+3}),$$

wobei $\hat{\boldsymbol{\delta}}_n$ eine Lösung der Score-Gleichungen $\mathbf{s}_n(\boldsymbol{\delta}) = \mathbf{0}$ ist.

Beweis.

Zur Vereinfachung wird im Folgenden das Argument $\boldsymbol{\delta}_0$ in $\mathbf{s}_n(\boldsymbol{\delta}_0)$, $\mathbf{F}_n(\boldsymbol{\delta}_0)$, $P_{\boldsymbol{\delta}_0}$ etc. ausgelassen. Der Beweis folgt Fahrmeir und Kaufmann [10, S. 353] sowie Czado und Min [7]. Aus der Mittelwerttheorie für vektorielle Funktionen (z.B. Heuser [19, S. 278]) erhält man

$$\mathbf{s}_n(\boldsymbol{\delta}_0) = \left[\int_0^1 \mathbf{H}_n \left(\boldsymbol{\delta}_0 + t(\hat{\boldsymbol{\delta}}_n - \boldsymbol{\delta}_0) \right) dt \right] (\hat{\boldsymbol{\delta}}_n - \boldsymbol{\delta}_0)$$

wobei Integration elementweise verstanden werden muß, und

$$\mathbf{F}_n^{-1/2} \mathbf{s}_n(\boldsymbol{\delta}_0) = \left[\int_0^1 \mathbf{F}_n^{-1/2} \mathbf{H}_n \left(\boldsymbol{\delta}_0 + t(\hat{\boldsymbol{\delta}}_n - \boldsymbol{\delta}_0) \right) \mathbf{F}_n^{-t/2} dt \right] \mathbf{F}_n^{t/2} (\hat{\boldsymbol{\delta}}_n - \boldsymbol{\delta}_0).$$

Man kann die letzte Gleichung nun in der folgenden Form umschreiben:

$$\begin{aligned}
\mathbf{F}_n^{-1/2} \mathbf{s}_n(\boldsymbol{\delta}_0) &= \left[\int_0^1 \mathbf{F}_n^{-1/2} \mathbf{H}_n \left(\boldsymbol{\delta}_0 + t(\hat{\boldsymbol{\delta}}_n - \boldsymbol{\delta}_0) \right) \mathbf{F}_n^{-t/2} dt \right] \cdot \mathbf{F}_n^{t/2} (\hat{\boldsymbol{\delta}}_n - \boldsymbol{\delta}_0) \\
&= \left[\int_0^1 \left\{ \mathbf{F}_n^{-1/2} \mathbf{H}_n \left(\boldsymbol{\delta}_0 + t(\hat{\boldsymbol{\delta}}_n - \boldsymbol{\delta}_0) \right) \mathbf{F}_n^{-t/2} - \mathbf{I} \right\} dt \right] \cdot \mathbf{F}_n^{t/2} (\hat{\boldsymbol{\delta}}_n - \boldsymbol{\delta}_0) \\
&\quad + \mathbf{F}_n^{t/2} (\hat{\boldsymbol{\delta}}_n - \boldsymbol{\delta}_0)
\end{aligned} \tag{2.18}$$

Nimmt man an, daß $\hat{\boldsymbol{\delta}}_n \in N_n(\varepsilon)$, so erhält man aus Fahrmeir und Kaufmanns [10] Annahme (\mathbf{N}^*) (hier (A4)), daß

$$P \left\{ \left\| \int_0^1 \mathbf{F}_n^{-1/2} \mathbf{H}_n \left(\boldsymbol{\delta}_0 + t(\hat{\boldsymbol{\delta}}_n - \boldsymbol{\delta}_0) \right) \mathbf{F}_n^{-t/2} dt - \mathbf{I} \right\| \geq \varepsilon \right\} \xrightarrow{n \rightarrow \infty} 0, \tag{2.19}$$

wenn $\varepsilon > 0$ beliebig klein sein darf.

Die bereits bewiesene Aussage (\mathbf{AE}^*) impliziert, daß für jedes $\eta > 0$ ein $\varepsilon > 0$ und ein n_1 existieren, so daß

$$P \left(\left\| \mathbf{F}_n^{t/2} (\hat{\boldsymbol{\delta}}_n - \boldsymbol{\delta}_0) \right\| \leq \varepsilon \right) \geq 1 - \eta, \tag{2.20}$$

für alle $n \geq n_1$. Grund: das Ereignis in Ausdruck (\mathbf{AE}^*) bedeutet, daß es eine Lösung $\hat{\boldsymbol{\delta}}_n$ der Scoregleichungen $\mathbf{s}_n(\boldsymbol{\delta}) = \mathbf{0}$ gibt, die zu $N_n(\varepsilon)$ gehört.

Damit kann ε so gewählt werden, daß die Wahrscheinlichkeit des Ereignisses

$$B_n = \left\{ \omega : \left\| \int_0^1 \mathbf{F}_n^{-1/2} \mathbf{H}_n \left(\boldsymbol{\delta}_0 + t(\hat{\boldsymbol{\delta}}_n - \boldsymbol{\delta}_0) \right) \mathbf{F}_n^{-t/2} dt - \mathbf{I} \right\| \geq \varepsilon \right\}$$

beliebig nahe an 0 liegt ((A4) spricht lediglich von Konvergenz in Wahrscheinlichkeit).

Allgemein gilt, daß $\forall \eta > 0 \exists \varepsilon > 0$, so daß

$$\begin{aligned}
P(B_n) &= P \left(B_n ; \left\| \mathbf{F}_n^{T/2} (\hat{\boldsymbol{\delta}}_n - \boldsymbol{\delta}_0) \right\| \leq \varepsilon \right) + P \left(B_n ; \left\| \mathbf{F}_n^{t/2} (\hat{\boldsymbol{\delta}}_n - \boldsymbol{\delta}_0) \right\| \geq \varepsilon \right) \\
&\quad \left(\text{mit (2.19), da } \hat{\boldsymbol{\delta}}_n \in N_n(\varepsilon) \right) \\
&= o(1) + P \left(B_n ; \left\| \mathbf{F}_n^{t/2} (\hat{\boldsymbol{\delta}}_n - \boldsymbol{\delta}_0) \right\| \geq \varepsilon \right) \\
&\leq o(1) + P \left(\left\| \mathbf{F}_n^{t/2} (\hat{\boldsymbol{\delta}}_n - \boldsymbol{\delta}_0) \right\| \geq \varepsilon \right) \\
(\text{mit (2.20)}) &\leq o(1) + \eta.
\end{aligned}$$

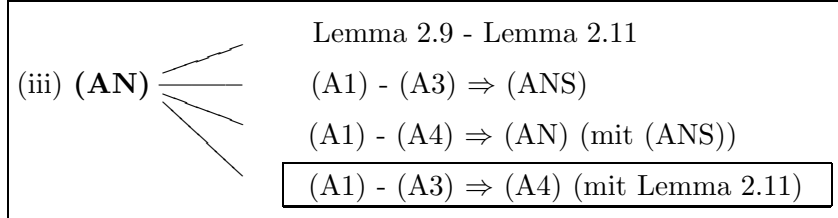
Somit konvergiert

$$\int_0^1 \mathbf{F}_n^{-1/2} \mathbf{H}_n \left(\boldsymbol{\delta}_0 + t(\hat{\boldsymbol{\delta}}_n - \boldsymbol{\delta}_0) \right) \mathbf{F}_n^{-t/2} dt \xrightarrow{P} \mathbf{I},$$

und mit dem **Stetigkeitssatz** (z.B. Gut [16, S. 238]) angewandt auf Formel (2.18) und dem Satz von Slutsky (z.B. Fisz [15, S. 209]) folgt die Aussage aus der asymptotischen Normalität von $\mathbf{F}_n^{-1/2} \mathbf{s}_n(\boldsymbol{\delta}_0)$ (ANS).

□

Nun möchte man beweisen, daß die asymptotische Normalität der Lösungen der Scoregleichungen aus Satz 2.13 sogar unter (A1) – (A3) gilt. In Satz 2.14 wird diese Aussage noch einmal formuliert. Annahme (A4) ist in der Praxis schwer zu verifizieren, daher lohnt es sich zu bewiesen, daß sich Annahme (A4) bereits aus (A1) – (A3) ergibt. Der Beweis benötigt außerdem Lemma 2.9.



Satz 2.14 (Asymptotische Normalität).

Unter den Annahmen (A1) – (A3) gilt

$$\mathbf{F}_n^{t/2}(\hat{\boldsymbol{\delta}}_n - \boldsymbol{\delta}_0) \sim N_p(\mathbf{0}, \mathbf{I}_{p+r+q+3}),$$

wobei $\hat{\boldsymbol{\delta}}_n$ eine Lösung der Score-Gleichungen $\mathbf{s}_n(\boldsymbol{\delta}) = \mathbf{0}$ ist.

Beweis.

Mit Satz 2.13 genügt es zu zeigen, daß (A4) aus (A1) – (A3) folgt.

Es gilt P-f.s.

$$\begin{aligned}
 \|\mathbf{V}_n(\boldsymbol{\delta}) - \mathbf{I}_{p+q+3}\| &= \|\mathbf{F}_n^{-1/2}[\mathbf{H}_n(\boldsymbol{\delta}) - \mathbf{F}_n]\mathbf{F}_n^{-t/2}\| \\
 &\leq \frac{1}{\lambda_{\min}(\mathbf{F}_n)} \|\mathbf{H}_n(\boldsymbol{\delta}) - \mathbf{F}_n\| \\
 &\stackrel{(A1)}{\leq} \frac{C}{n} \|\mathbf{H}_n(\boldsymbol{\delta}) - \mathbf{F}_n\| \\
 &\leq C \left\| \frac{1}{n}(\mathbf{H}_n(\boldsymbol{\delta}) - E \mathbf{H}_n(\boldsymbol{\delta})) \right\| + C \left\| \frac{1}{n}(E \mathbf{H}_n(\boldsymbol{\delta}) - \mathbf{F}_n) \right\|.
 \end{aligned}$$

Es bleibt also zu zeigen, daß

$$\max_{\boldsymbol{\delta} \in N_n(\varepsilon)} \left\| \frac{1}{n}(\mathbf{H}_n(\boldsymbol{\delta}) - E \mathbf{H}_n(\boldsymbol{\delta})) \right\| \xrightarrow{P} 0 \quad (2.21)$$

und

$$\max_{\boldsymbol{\delta} \in N_n(\varepsilon)} \left\| \frac{1}{n}(E \mathbf{H}_n(\boldsymbol{\delta}) - \mathbf{F}_n) \right\| \rightarrow 0 \quad (2.22)$$

gelten.

Um (2.21) zu zeigen genügt es nachzuweisen, daß das Maximum über $\boldsymbol{\delta} \in N_n(\varepsilon)$ des absoluten Wertes des Elementes (r, s) der Zufallsmatrix $[\mathbf{H}_n(\boldsymbol{\delta}) - E \mathbf{H}_n(\boldsymbol{\delta})]/n$ in Wahrscheinlichkeit gegen Null konvergiert, d.h.

$$\max_{\boldsymbol{\delta} \in N_n(\varepsilon)} \frac{|h_{rs}(\boldsymbol{\delta}) - E h_{rs}(\boldsymbol{\delta})|}{n} \xrightarrow{P} 0.$$

Es sei noch einmal betont, daß die Hessematrix sechs verschiedene Arten von Einträgen hat. Hier wird die angesprochene Konvergenz für den Fall $0 \leq l, m \leq p$ gezeigt. Die verbleibenden Fälle können ähnlich behandelt werden. O.B.d.A wird hier nur gezeigt, daß

$$\max_{\boldsymbol{\delta} \in N_n(\varepsilon)} \left| \frac{1}{n} (h_{p,p}(\boldsymbol{\delta}) - E h_{p,p}(\boldsymbol{\delta})) \right| \xrightarrow{P} 0. \quad (2.23)$$

Sei $Z_i := \mathbb{1}_{\{y_i > 0\}} Y_i (Y_i - 1)$, $U_i(\boldsymbol{\beta}, \boldsymbol{\alpha}) := \mu_i(\boldsymbol{\beta}) + (\varphi_i(\boldsymbol{\alpha}) - 1) Y_i$, $q_{i,p}(\boldsymbol{\delta}) := x_{ip}^2 \mu_i(\boldsymbol{\beta}) (\varphi_i(\boldsymbol{\alpha}) - 1)$ und $v_{i,p}(\boldsymbol{\delta}) := x_{ip}^2 \mu_i(\boldsymbol{\beta}) \frac{-\frac{1}{\varphi_i(\boldsymbol{\alpha})} P_i^0(\boldsymbol{\delta})^2 + \frac{\mu_i(\boldsymbol{\delta}) - \varphi_i(\boldsymbol{\alpha})}{\varphi_i(\boldsymbol{\alpha})^2} P_i^0(\boldsymbol{\delta}) k_i(\boldsymbol{\gamma})}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))^2}$.

Man erinnere sich, daß

$$\begin{aligned} h_{p,p}(\boldsymbol{\delta}) = & \sum_{i=1}^n x_{ip}^2 \mu_i(\boldsymbol{\beta}) \left(\mathbb{1}_{\{y_i=0\}} \left[\frac{-\frac{1}{\varphi_i(\boldsymbol{\alpha})} P_i^0(\boldsymbol{\delta})^2 + \frac{\mu_i(\boldsymbol{\delta}) - \varphi_i(\boldsymbol{\alpha})}{\varphi_i(\boldsymbol{\alpha})^2} P_i^0(\boldsymbol{\delta}) k_i(\boldsymbol{\gamma})}{(k_i(\boldsymbol{\gamma}) + P_i^0(\boldsymbol{\delta}))^2} \right] \right. \\ & \left. + \mathbb{1}_{\{y_i>0\}} \left[-\frac{1}{\varphi_i(\boldsymbol{\alpha})} + \frac{(\varphi_i(\boldsymbol{\alpha}) - 1)(Y_i - 1)Y_i}{(\mu_i(\boldsymbol{\beta}) + (\varphi_i(\boldsymbol{\alpha}) - 1)Y_i)^2} \right] \right). \end{aligned}$$

Nun kann man leicht sehen, daß sich (2.23) aus den folgenden drei Bedingungen ergibt:

$$\begin{aligned} & \max_{\boldsymbol{\delta} \in N_n(\varepsilon)} \left| \frac{1}{n} \sum_{i=1}^n v_{i,p}(\boldsymbol{\delta}) (\mathbb{1}_{\{y_i=0\}} - E(\mathbb{1}_{\{y_i=0\}})) \right| \xrightarrow{P} 0, \\ & \max_{\boldsymbol{\delta} \in N_n(\varepsilon)} \left| \frac{1}{n} \sum_{i=1}^n \frac{q_{i,p}(\boldsymbol{\delta})}{\varphi_i(\boldsymbol{\alpha})(\varphi_i(\boldsymbol{\alpha}) - 1)} (\mathbb{1}_{\{y_i>0\}} - E(\mathbb{1}_{\{y_i>0\}})) \right| \xrightarrow{P} 0, \\ & \max_{\boldsymbol{\delta} \in N_n(\varepsilon)} \left| \frac{1}{n} \sum_{i=1}^n q_{i,p}(\boldsymbol{\delta}) \left[\frac{Z_i}{[U_i(\boldsymbol{\beta}, \boldsymbol{\alpha})]^2} - E \left(\frac{Z_i}{[U_i(\boldsymbol{\beta}, \boldsymbol{\alpha})]^2} \right) \right] \right| \xrightarrow{P} 0. \end{aligned} \quad (2.24)$$

Da die Terme ähnliche Struktur aufweisen, wird hier nur die Gültigkeit der letzten Gleichung aufgezeigt. Noch einmal: die Argumente $\boldsymbol{\delta}_0$, $\boldsymbol{\beta}_0$ und $\boldsymbol{\alpha}_0$ werden stets weggelassen. Die rechte Seite von (2.24) kann durch die Summe $A_n + D_n - B_n$ nach oben beschränkt werden, wobei

$$\begin{aligned} A_n &= \max_{\boldsymbol{\delta} \in N_n(\varepsilon)} \left| \frac{1}{n} \sum_{i=1}^n q_{i,p}(\boldsymbol{\delta}) \left[\frac{Z_i}{[U_i(\boldsymbol{\beta}, \boldsymbol{\alpha})]^2} - \frac{Z_i}{U_i^2} \right] \right|, \\ B_n &= \max_{\boldsymbol{\delta} \in N_n(\varepsilon)} \left| \frac{1}{n} \sum_{i=1}^n q_{i,p}(\boldsymbol{\delta}) \left[E \left(\frac{Z_i}{[U_i(\boldsymbol{\beta}, \boldsymbol{\alpha})]^2} \right) - E \left(\frac{Z_i}{U_i^2} \right) \right] \right|, \\ D_n &= \max_{\boldsymbol{\delta} \in N_n(\varepsilon)} \left| \frac{1}{n} \sum_{i=1}^n q_{i,p}(\boldsymbol{\delta}) \left[\frac{Z_i}{U_i^2} - E \left(\frac{Z_i}{U_i^2} \right) \right] \right|. \end{aligned}$$

Für A_n erhält man folgende Schranken:

$$A_n = \max_{\boldsymbol{\delta} \in N_n(\varepsilon)} \left| \frac{1}{n} \sum_{i=1}^n q_{i,p}(\boldsymbol{\delta}) \left[\frac{Z_i(U_i^2 - [U_i(\boldsymbol{\beta}, \boldsymbol{\alpha})]^2)}{[U_i(\boldsymbol{\beta}, \boldsymbol{\alpha})]^2 U_i^2} \right] \right|$$

$$\begin{aligned}
&\leq \max_{\boldsymbol{\delta} \in N_n(\varepsilon)} \frac{1}{n} \sum_{i=1}^n \frac{|q_{i,p}(\boldsymbol{\delta})Z_i| |U_i(\boldsymbol{\beta}, \boldsymbol{\alpha}) + U_i|}{\underbrace{[\mu_i^2(\boldsymbol{\beta}) + 2\mu_i(\boldsymbol{\beta})(\varphi_i(\boldsymbol{\alpha}) - 1)Y_i + (\varphi_i(\boldsymbol{\alpha}) - 1)^2Y_i^2]}_{>0, \text{ da } "Y_i > 0" \text{ durch } Z_i \text{ sichergestellt ist}}} \\
&\quad \times \frac{|U_i(\boldsymbol{\beta}, \boldsymbol{\alpha}) - U_i|}{\underbrace{[\mu_i^2 + 2\mu_i(\varphi_i - 1)Y_i + (\varphi_i - 1)^2Y_i^2]}_{>0}} \\
&\leq \max_{\boldsymbol{\delta} \in N_n(\varepsilon)} \frac{1}{n} \sum_{i=1}^n \frac{|q_{i,p}(\boldsymbol{\delta})Z_i|}{\mu_i^2(\boldsymbol{\beta})\mu_i^2} \cdot |U_i(\boldsymbol{\beta}, \boldsymbol{\alpha}) + U_i| \underbrace{|\mu_i(\boldsymbol{\beta}) - \mu_i + (\varphi_i(\boldsymbol{\alpha}) - \varphi_i)Y_i|}_{U_i(\boldsymbol{\beta}, \boldsymbol{\alpha}) - U_i} \\
&\stackrel{(*)}{\leq} \max_{\boldsymbol{\delta} \in N_n(\varepsilon)} \frac{1}{n} \sum_{i=1}^n \frac{|q_{i,p}(\boldsymbol{\delta})Z_i|}{\mu_i^2(\boldsymbol{\beta})\mu_i^2} \cdot |(Y_i + 1)(\mu_i(\boldsymbol{\beta}) + \mu_i + \varphi_i(\boldsymbol{\alpha}) + \varphi_i - 2)| \\
&\quad \times |\mu_i(\boldsymbol{\beta}) - \mu_i + (\varphi_i(\boldsymbol{\alpha}) - \varphi_i)Y_i| \\
&\leq \frac{C_1}{n} \left(\sum_{i=1}^n Z_i(Y_i + 1) \right) \max_{\boldsymbol{\delta} \in N_n(\varepsilon)} \max_{\mathbf{x} \in K_x} |\exp(\mathbf{x}^t \boldsymbol{\beta}) - \exp(\mathbf{x}^t \boldsymbol{\beta}_0)| \\
&\quad + \frac{C_1}{n} \left(\sum_{i=1}^n Z_i Y_i(Y_i + 1) \right) \max_{\boldsymbol{\delta} \in N_n(\varepsilon)} \max_{\mathbf{w} \in K_w} |\exp(\mathbf{w}^t \boldsymbol{\alpha}) - \exp(\mathbf{w}^t \boldsymbol{\alpha}_0)| \\
&=: AB_n + AC_n.
\end{aligned}$$

(Dabei ergibt sich $(*)$ durch

$$\begin{aligned}
|U_i(\boldsymbol{\beta}, \boldsymbol{\alpha}) + U_i| &= |\mu_i(\boldsymbol{\beta}) + \mu_i + (\varphi_i(\boldsymbol{\alpha}) + \varphi_i - 2)Y_i| \\
&\leq |\mu_i(\boldsymbol{\beta}) + \mu_i + (\varphi_i(\boldsymbol{\alpha}) + \varphi_i - 2)(Y_i + 1)| \\
&= \underbrace{(Y_i + 1)}_{\geq 1} \left| \frac{\mu_i(\boldsymbol{\beta}) + \mu_i}{Y_i + 1} + \varphi_i(\boldsymbol{\alpha}) + \varphi_i - 2 \right| \\
&\leq |(Y_i + 1)(\mu_i(\boldsymbol{\beta}) + \mu_i + \varphi_i(\boldsymbol{\alpha}) + \varphi_i - 2)| \quad .)
\end{aligned}$$

Man kann sehen, daß

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n Z_i(Y_i + 1) \xrightarrow{P} \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n E(Z_i(Y_i + 1)),$$

was aufgrund von Lemma 2.11 beschränkt ist.

Ferner ist die Funktion $\max_{\mathbf{x} \in K_x} |\exp(\mathbf{x}^t \boldsymbol{\beta}) - \exp(\mathbf{x}^t \boldsymbol{\beta}_0)|$ stetig in $\boldsymbol{\beta}$ und hat den Wert Null in $\boldsymbol{\beta} = \boldsymbol{\beta}_0$. Daher ergibt sich die Konvergenz von AB_n gegen Null in Wahrscheinlichkeit für $n \rightarrow \infty$.

Die Konvergenz von $AC_n \xrightarrow{P} 0$ kann analog gezeigt werden.

Zurück zu B_n . Mit ähnlichen Argumenten kann man die Konvergenz $B_n \xrightarrow{P} 0$ zeigen. Um $D_n \xrightarrow{P} 0$ zu beweisen, kann man nachweisen, daß die Funktion $\max_{i=1, \dots, n} |q_{i,p}(\boldsymbol{\delta}) - q_{i,p}(\boldsymbol{\delta}_0)|$ von oben her durch folgende in $\boldsymbol{\delta}$ stetige Funktion beschränkt werden kann, die ebenfalls

in $\delta = \delta_0$ den Wert Null annimmt:

$$C \cdot \max_{\mathbf{x} \in K_x, \mathbf{w} \in K_w} |e^{\mathbf{x}^t \boldsymbol{\beta}} \cdot e^{\mathbf{w}^t \boldsymbol{\alpha}} - e^{\mathbf{x}^t \boldsymbol{\beta}_0} \cdot e^{\mathbf{w}^t \boldsymbol{\alpha}_0}|$$

Das gewünschte Ergebnis folgt nun aus dem Gesetz der Großen Zahlen.

Es bleibt noch (2.22) zu zeigen. Hier soll wieder das Element (l, m) mit $0 \leq l, m \leq p$ herausgegriffen werden, also

$$\max_{\delta \in N_n(\varepsilon)} \left| \frac{[E \mathbf{H}_n(\delta) - \mathbf{F}_n]_{lm}}{n} \right| \rightarrow 0. \quad (2.25)$$

Darüber hinaus beschränkt man sich erneut auf den Fall $l = m = p$. Man kann sehen, daß (2.25) von den nächsten drei Bedingungen impliziert wird:

$$\max_{\delta \in N_n(\varepsilon)} \left| \frac{1}{n} \sum_{i=1}^n (v_{i,p}(\delta) - v_{i,p}) E(\mathbb{1}_{\{y_i=0\}}) \right| \rightarrow 0, \quad (2.26)$$

$$\max_{\delta \in N_n(\varepsilon)} \left| \frac{1}{n} \sum_{i=1}^n \left(\frac{q_{i,p}(\delta)}{\varphi_i(\boldsymbol{\alpha})(\varphi_i(\boldsymbol{\alpha}) - 1)} - \frac{q_{i,p}}{\varphi_i(\varphi_i - 1)} \right) E(\mathbb{1}_{\{y_i>0\}}) \right| \rightarrow 0, \quad (2.27)$$

$$\max_{\delta \in N_n(\varepsilon)} \left| \frac{1}{n} \sum_{i=1}^n q_{i,p}(\delta) E \left(\frac{Z_i}{[U_i(\boldsymbol{\beta}, \boldsymbol{\alpha})]^2} \right) - q_{i,p} E \left(\frac{Z_i}{U_i^2} \right) \right| \rightarrow 0. \quad (2.28)$$

Nun kann dieselbe Technik auf (2.26) bis (2.28) angewandt werden, die zum Beweis von (2.24) verwendet wurde. \square

2.4 Simulationsstudie zur Untersuchung der Asymptotik der ML-Schätzer

In Abschnitt 2.3 wurde gezeigt, daß die ML-Schätzer unter gewissen Voraussetzungen asymptotisch normalverteilt sind. Diese Aussage wird nun genutzt, um graphisch und mit Hilfe von Fehlerstatistiken wie dem MSE zu überprüfen, wie schnell die Schätzer konvergieren. Dazu werden erneut für den Stichprobenumfang $n = 200$ und $n = 1000$ Daten verschiedener Szenarien erzeugt. Die Designmatrizen werden so gewählt: Für den Erwartungswert wird $\mu_i = \exp\{\beta_0 + x_{i1}\beta_1\}$ mit x_{i1} äquidistant steigend auf $[-1, 1] \ \forall i$ verwendet. Damit steigen auch die μ_i mit Beobachtung i an. Für die Überdispersion nimmt man $\varphi_i = 1 + \exp\{\alpha_0 + w_{i1}\alpha_1\}$ mit w_{i1} äquidistant steigend auf $[-1, 1]$. Das ergibt steigende φ_i .

Und zuletzt wird $\omega_i = \exp\{\gamma_0 + z_{i1}\gamma_1\} / (1 + \exp\{\gamma_0 + z_{i1}\gamma_1\})$ mit z_{i1} äquidistant fallend auf $[-1, 1] \ \forall i$ gesetzt, das ergibt steigende ω_i . Wie bisher werden $\mu_i \in [2, 5]$ und $\mu_i \in [2, 15]$ erzeugt; entsprechend $\varphi_i \in [1.2, 1.5]$ und $\varphi_i \in [1.2, 3]$ sowie $\omega_i \in [0.1, 0.25]$ gegenüber $\omega_i \in [0.1, 0.5]$.

Die tabellarischen Ergebnisse der Simulationsläufe aus Tabelle 2.2 enthalten Schätzer für die Parameter und als Fehlerstatistik den "Mean Squared Error". Dieser wurde schon

Parametrisierung	Range
$\beta = (1.15, 0.46)^t$	$\mu_i \in [2, 5]$
$\beta = (1.7, 1)^t$	$\mu_i \in [2, 15]$
$\alpha = (-1.15, 0.46)^t$	$\varphi_i \in [1.2, 1.5]$
$\alpha = (-0.46, 1.15)^t$	$\varphi_i \in [1.2, 3]$
$\gamma = (-1.65, -0.55)^t$	$\omega_i \in [0.1, 0.25]$
$\gamma = (-1.10, -1.1)^t$	$\omega_i \in [0.1, 0.5]$

Tabelle 2.1: Range von μ , φ und ω in je zwei Settings

in Definition 1.15 eingeführt. In Klammern sind jeweils die geschätzten Standardabweichungen der Statistiken angegeben. Man erkennt, daß MSE und Varianzen bei γ_0 und γ_1 am größten sind, diese Werte also auch in der $ZIGP(\mu_i, \varphi_i, \omega_i)$ -Regression sehr langsam gegen ihren wahren Wert konvergieren. Auch hier wird wie in Kapitel 1 ein Wald-Test nach Formel (1.24) durchgeführt und damit überprüft, ob die mittleren ML-Schätzer signifikant verschieden von Null sind. Es konnte jedoch kein Setting gefunden werden, in dem $|T| \leq 1.96$, das spricht dafür, daß die Anzahl der Simulationsläufe mit $N = 100$ ausreichend groß gewählt wurde. Wichtigen Zahlen wurde in Text und Tabelle dieselbe Farbe gegeben, um die Suche in der Tabelle zu erleichtern.

Erhöhung des Stichprobenumfangs n

Erhöht man den Stichprobenumfang von $n = 200$ auf $n = 1000$, so verbessern sich diese Werte erwartungsgemäß. So sinkt für $\mu_i \in [2, 5]$, $\varphi_i \in [1.2, 1.25]$ und $\omega_i \in [0.1, 0.5]$ der MSE für α_0 von **0.170** ($T = 10 \frac{-1.380}{0.342} = -40.4$) auf **0.025** ($T = 10 \frac{-1.187}{0.153} = -77.6$). Auch der geschätzte Bias $\hat{b}(\theta, \hat{\theta})$ sinkt von $0.23 = -1.15 - (-1.380)$ auf $0.037 = -1.15 - (-1.187)$. Dies spiegelt sich auch in Abb. 2.2 (3) wider.

Erhöhung der Erwartungswerte μ_i

Werden höhere Erwartungswerte erzeugt, sind die Schätzungen aller Parameter genauer. Dies ist einleuchtend, da ein geringer Erwartungswert die Wahrscheinlichkeit von Nullen erhöht. Diese sind nur schwer ihrem Ursprung (also geringe Erwartungswerte, hohe Überdispersion oder hoher Nullenüberschuß) zuzuordnen. Dazu vergleiche man etwa für $\varphi_i \in [1.2, 3]$, $\omega_i \in [0.1, 0.5]$, $n = 200$ den MSE von β_1 . Dieser sinkt bei größerem μ_i -Range um etwa die Hälfte von **0.032** ($T = 10 \frac{0.440}{0.178} = 24.7$) auf **0.014** ($T = 10 \frac{0.998}{0.119} = 83.9$). Die Verzerrung nimmt ebenfalls ab, sie sinkt von 0.02 auf 0.002. Dies entspricht auch der Verbesserung in den QQ-Plots in Abb. 2.2 (8) mit Abb. 2.4 (8).

Erhöhung der Dispersionsparameter φ_i

Je größer φ_i , desto schlechter sind auch die Schätzungen. Dies leuchtet ein, da bei zunehmender Überdispersion die Streuung der Daten wächst und damit eine Schätzung der Koeffizienten schwerer wird. Eine Ausnahme dabei bildet die Schätzgüte der α_i ; dies macht jedoch Sinn, da für größere φ_i auch deren Regressoren besser zu schätzen sind. Als Beispiel kann eine Erhöhung von $\varphi_i \in [1.2, 1.5]$ auf $\varphi_i \in [1.2, 3]$ betrachtet werden. Dort sinkt der MSE im Setting $\mu_i \in [2, 5]$, $\omega_i \in [0.1, 0.25]$ für α_0 und $n = 200$ leicht von **0.116** ($T = 10 \frac{-1.246}{0.328} = -38.0$) auf **0.084** ($T = 10 \frac{-0.588}{0.261} = -22.5$), dies entspricht einem Minus

von etwa 28%. Der geschätzte Bias $\hat{b}(\theta, \hat{\theta})$ steigt an, er wächst von 0.096 auf 0.128. Eine Verschlechterung ist auch in den QQ-Plots in Abb. 2.1 (3) mit (9) zu sehen.

Erhöhung des Nullenüberschuß ω_i

Eine Erhöhung im Nullenüberschuß ω_i erhöht wieder die Streuung der Daten und verschlechtert damit die Ergebnisse. Die Änderung des MSE fällt jedoch gering aus: so steigt er für $\varphi_i \in [1.2, 1.5]$, $\mu_i \in [2, 15]$, β_1 und $n = 200$ lediglich von 0.007 ($T = 10 \frac{1.005}{0.083} = 12.1$) auf 0.008 ($T = 10 \frac{0.990}{0.090} = 110$). Die Verzerrung steigt in diesem Fall betragsmäßig von -0.005 auf 0.01. Auch die QQ-Plots verschlechtern sich, siehe dazu Abb. 2.3 (2) und Abb. 2.4 (2). Dagegen fallen die MSE-Statistiken von γ_0 und γ_1 geringer aus. Das rührt daher, daß bei höherem Nullenüberschuß die Regressionsparameter γ_j besser schätzbar sind. Vergleicht man die MSE-Statistiken für $\varphi_i \in [1.2, 1.5]$, $\mu_i \in [2, 15]$, $n = 200$ für $\omega_i \in [0.1, 0.25]$ mit $\omega_i \in [0.1, 0.5]$, so stellt man fest: Für γ_0 sinkt der MSE von 0.261 ($T = 10 \frac{-3.377}{0.505} = -66.9$) auf 0.186 ($T = 10 \frac{-2.190}{0.431} = -50.8$), das sind etwa 29% weniger. Der Bias fällt von 0.077 auf -0.01 . Die QQ-Plots verbessern sich von Abb. 2.3 (5) auf Abb. 2.4 (5). Der Einfluß des Nullenüberschuß auf die Güte der übrigen Parameter ist gering.

	$\varphi_i \in [1.2, 1.5] \quad \forall i$						$\varphi_i \in [1.2, 3] \quad \forall i$								
	Parameter	Wahrer Wert	n	ML-Schätzer		MSE	Parameter	Wahrer Wert	n	ML-Schätzer		MSE			
$\mu_i \in [2, 5]$	β_0	1.15	200	1.147	(0.070)	0.005	$(5 \cdot 10^{-7})$	β_0	1.15	200	1.144	(0.096)	0.009	$(1 \cdot 10^{-6})$	s. Abb. 2.1
			1000	1.146	(0.031)	0.001	$(2 \cdot 10^{-8})$				1.151	(0.043)	0.002	$(4 \cdot 10^{-8})$	
	β_1	0.46	200	0.461	(0.118)	0.014	$(3 \cdot 10^{-6})$	β_1	0.46	200	0.455	(0.168)	0.028	$(8 \cdot 10^{-6})$	
			1000	0.459	(0.053)	0.003	$(1 \cdot 10^{-7})$				0.456	(0.075)	0.006	$(5 \cdot 10^{-7})$	
	α_0	-1.15	200	-1.246	(0.328)	<i>0.116</i>	(0.0005)	α_0	-0.46	200	-0.588	(0.261)	<i>0.084</i>	(0.0002)	
			1000	-1.180	(0.147)	0.022	$(3 \cdot 10^{-5})$				-0.476	(0.117)	0.014	$(6 \cdot 10^{-6})$	
$\omega_i \in [0.1, 0.25]$	α_1	0.46	200	0.454	(0.545)	0.297	(0.002)	α_1	1.15	200	1.220	(0.458)	0.215	(0.001)	
			1000	0.495	(0.243)	0.060	(0.0001)				1.144	(0.204)	0.042	$(4 \cdot 10^{-5})$	
	γ_0	-3.3	200	-3.322	(0.695)	0.484	(0.003)	γ_0	-3.3	200	-3.312	(0.854)	0.730	(0.005)	
			1000	-3.311	(0.311)	0.097	(0.0001)				-3.302	(0.382)	0.146	(0.0002)	
	γ_1	-0.55	200	-0.661	(0.576)	0.345	(0.002)	γ_1	-0.55	200	-0.535	(0.759)	0.576	(0.01)	
			1000	-0.556	(0.257)	0.066	$(6 \cdot 10^{-5})$				1000	-0.604	(0.339)	0.118	(0.0004)
$\omega_i \in [0.1, 0.5]$	β_0	1.15	200	1.147	(0.074)	0.005	$(5 \cdot 10^{-7})$	β_0	1.15	200	1.115	(0.104)	0.012	$(5 \cdot 10^{-6})$	s. Abb. 2.2
			1000	1.148	(0.033)	0.001	$(2 \cdot 10^{-8})$			1000	1.143	(0.046)	0.002	$(4 \cdot 10^{-8})$	
	β_1	0.46	200	0.444	(0.126)	0.016	$(1 \cdot 10^{-5})$	β_1	0.46	200	0.440	(0.178)	<i>0.032</i>	$(3 \cdot 10^{-5})$	
			1000	0.461	(0.056)	0.003	$(2 \cdot 10^{-7})$				0.457	(0.080)	0.006	$(9 \cdot 10^{-7})$	
	α_0	-1.15	200	-1.380	(0.342)	<i>0.170</i>	(0.001)	α_0	-0.46	200	-0.616	(0.273)	0.099	(0.001)	
			1000	-1.187	(0.153)	<i>0.025</i>	$(1 \cdot 10^{-5})$				-0.475	(0.122)	0.015	$(4 \cdot 10^{-6})$	
	α_1	0.46	200	0.557	(0.581)	0.347	(0.006)	α_1	1.15	200	1.266	(0.495)	0.258	(0.006)	
1000			0.470	(0.259)	0.067	(0.0001)	1.125				(0.221)	0.049	$(6 \cdot 10^{-5})$		
γ_0	-2.2	200	-2.208	(0.561)	0.314	(0.003)	γ_0	-2.2	200	-2.351	(0.667)	0.467	(0.007)		
		1000	-2.219	(0.251)	0.063	(0.0001)				-2.244	(0.298)	0.091	(0.0001)		
γ_1	-1.1	200	-1.097	(0.477)	0.228	(0.002)	γ_1	-1.1	200	-1.135	(0.587)	0.345	(0.004)		
			1000	-1.104	(0.213)	0.045	$(7 \cdot 10^{-5})$				1000	-1.092	(0.262)	0.069	(0.0001)
$\mu_i \in [2, 15]$	β_0	1.7	200	1.699	(0.049)	0.002	$(6 \cdot 10^{-8})$	β_0	1.7	200	1.697	(0.061)	0.004	$(3 \cdot 10^{-7})$	s. Abb. 2.3
			1000	1.698	(0.022)	0.000	$(4 \cdot 10^{-9})$				1.701	(0.028)	0.001	$(1 \cdot 10^{-8})$	
	β_1	1	200	1.005	(0.083)	<i>0.007</i>	$(8 \cdot 10^{-7})$	β_1	1	200	0.988	(0.111)	0.012	$(2 \cdot 10^{-6})$	
			1000	1.003	(0.037)	0.001	$(4 \cdot 10^{-8})$				0.999	(0.050)	0.002	$(7 \cdot 10^{-8})$	
	α_0	-1.15	200	-1.325	(0.283)	0.111	(0.0006)	α_0	-0.46	200	-0.509	(0.218)	0.050	$(5 \cdot 10^{-5})$	
			1000	-1.180	(0.127)	0.017	$(9 \cdot 10^{-6})$				-0.479	(0.097)	0.010	$(4 \cdot 10^{-6})$	
$\omega_i \in [0.1, 0.25]$	α_1	0.46	200	0.484	(0.474)	0.226	(0.001)	α_1	1.15	200	1.132	(0.356)	0.127	(0.0002)	
			1000	0.452	(0.212)	0.045	$(5 \cdot 10^{-5})$				1.199	(0.159)	0.028	$(2 \cdot 10^{-5})$	
	γ_0	-3.3	200	-3.377	(0.505)	<i>0.261</i>	(0.01)	γ_0	-3.3	200	-3.355	(0.545)	0.300	(0.01)	
			1000	-3.331	(0.226)	0.052	$(5 \cdot 10^{-5})$				-3.282	(0.244)	0.060	(0.0001)	
γ_1	-0.55	200	-0.571	(0.448)	0.201	(0.004)	γ_1	-0.55	200	-0.599	(0.472)	0.225	(0.003)		
			1000	-0.582	(0.200)	0.041	$(3 \cdot 10^{-5})$				1000	-0.521	(0.210)	0.045	$(4 \cdot 10^{-5})$
$\omega_i \in [0.1, 0.5]$	β_0	1.7	200	1.697	(0.052)	0.003	$(1 \cdot 10^{-7})$	β_0	1.7	200	1.683	(0.066)	0.005	$(4 \cdot 10^{-7})$	s. Abb. 2.4
			1000	1.702	(0.023)	0.001	$(1 \cdot 10^{-8})$			1000	1.701	(0.029)	0.001	$(2 \cdot 10^{-8})$	
	β_1	1	200	0.990	(0.090)	<i>0.008</i>	$(1 \cdot 10^{-6})$	β_1	1	200	0.998	(0.119)	<i>0.014</i>	$(4 \cdot 10^{-6})$	
			1000	1.003	(0.040)	0.002	$(4 \cdot 10^{-8})$				0.996	(0.053)	0.003	$(3 \cdot 10^{-7})$	
	α_0	-1.15	200	-1.224	(0.297)	0.094	(0.0001)	α_0	-0.46	200	-0.509	(0.227)	0.054	$(4 \cdot 10^{-5})$	
			1000	-1.176	(0.133)	0.018	$(6 \cdot 10^{-6})$				-0.468	(0.102)	0.010	$(2 \cdot 10^{-6})$	
	α_1	0.46	200	0.492	(0.510)	0.261	(0.003)	α_1	1.15	200	1.138	(0.384)	0.148	(0.001)	
1000			0.455	(0.228)	0.052	$(8 \cdot 10^{-5})$	1.150				(0.171)	0.029	$(2 \cdot 10^{-5})$		
γ_0	-2.2	200	-2.190	(0.431)	<i>0.186</i>	(0.0003)	γ_0	-2.2	200	-2.259	(0.458)	0.213	(0.01)		
		1000	-2.193	(0.193)	0.037	$(2 \cdot 10^{-5})$				-2.194	(0.205)	0.042	$(4 \cdot 10^{-5})$		
γ_1	-1.1	200	-1.119	(0.389)	0.152	(0.0004)	γ_1	-1.1	200	-1.086	(0.406)	0.165	(0.002)		
			1000	-1.097	(0.173)	0.030	$(2 \cdot 10^{-5})$				1000	-1.100	(0.181)	0.033	$(2 \cdot 10^{-5})$

Tabelle 2.2: Mittlere MLE-Schätzungen und MSE-Statistiken für $ZIGP(\mu_i, \varphi_i, \omega_i)$ in mehreren Parameter-Settings basierend auf $N = 100$ Datensätzen; geschätzte Standardabweichungen der Statistiken in Klammern.

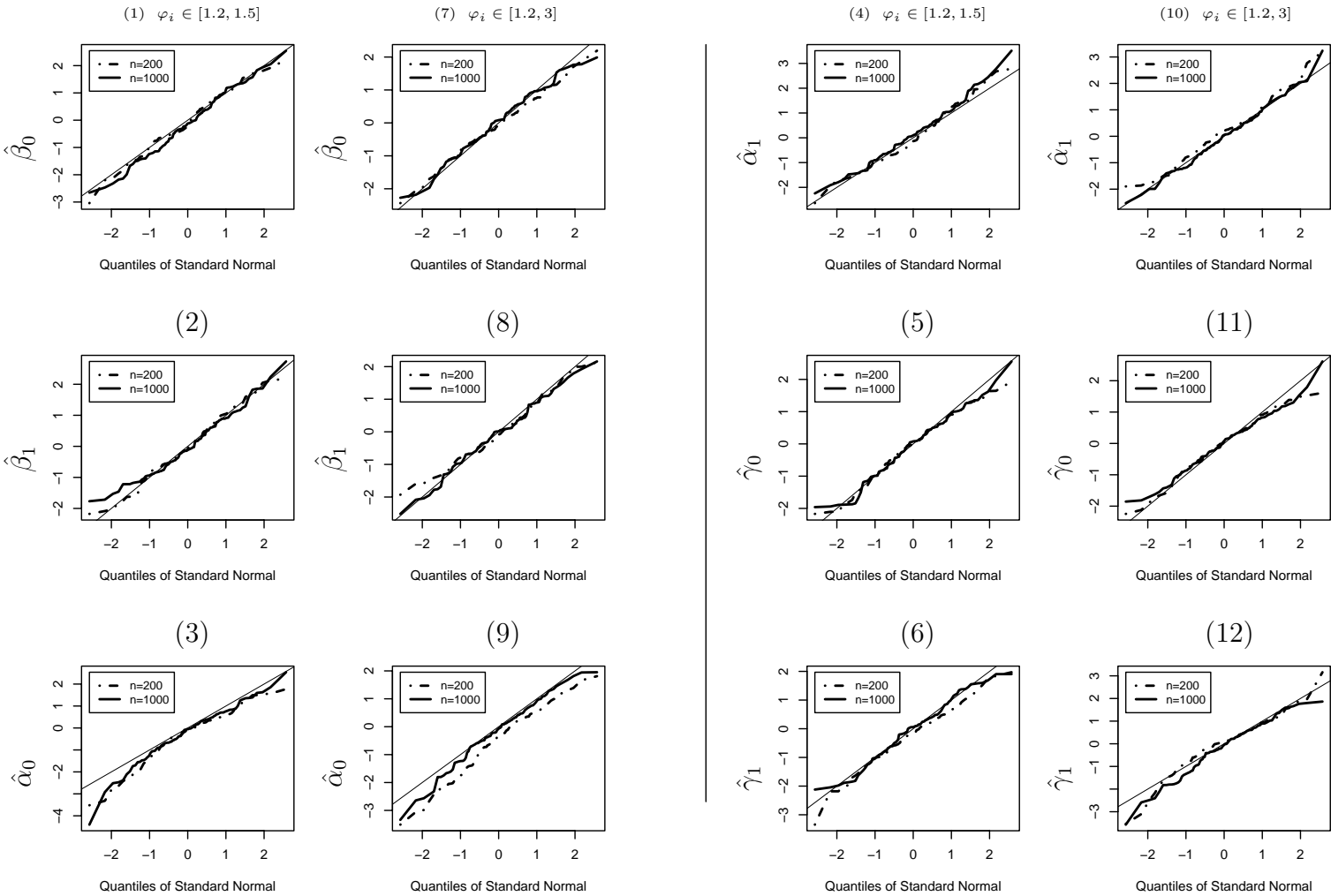
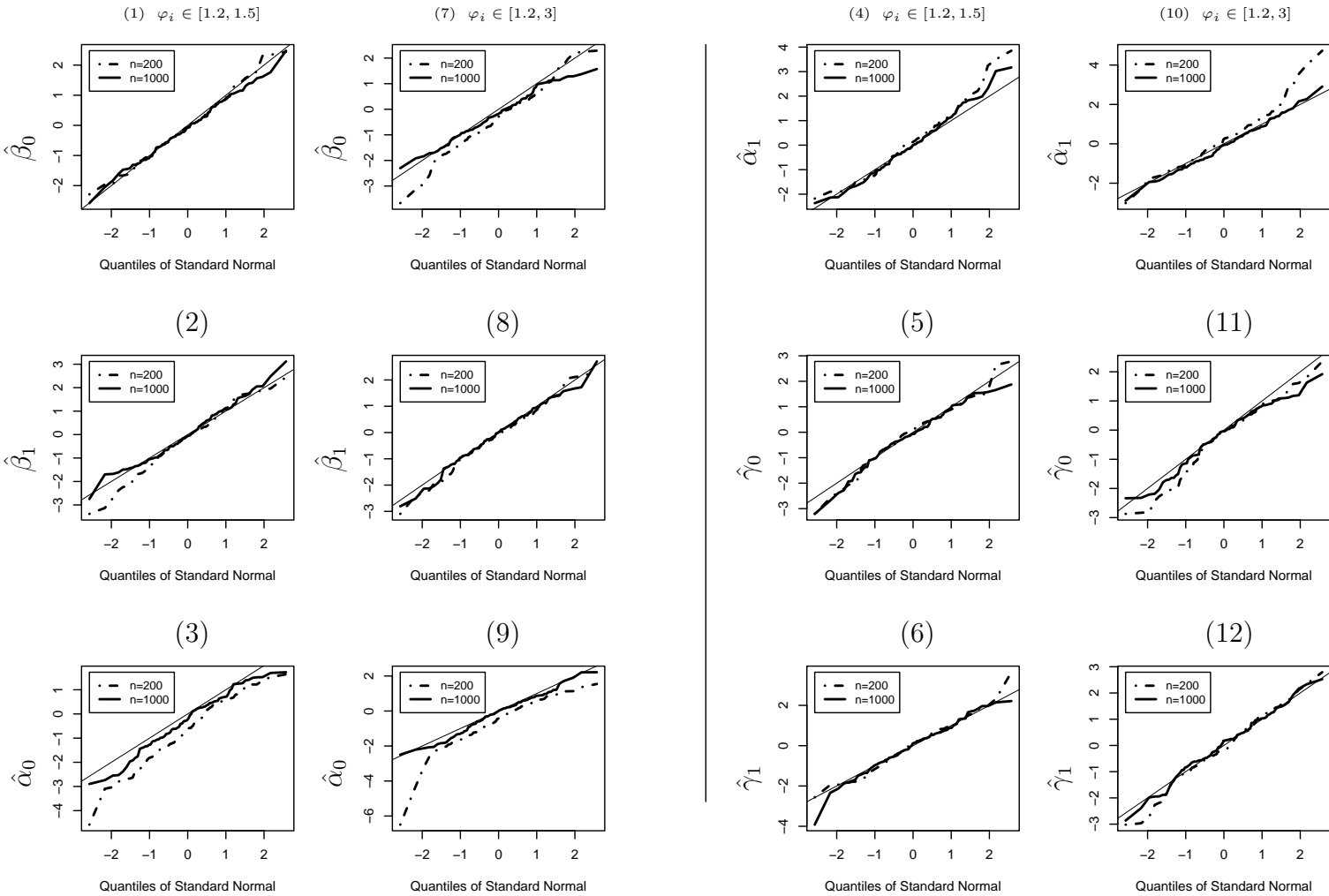


Abbildung 2.1: QQ-Plots zentrierter und normierter ML Schätzer basierend auf $N = 100$ Datensätzen im $ZIGP(\mu_i, \varphi_i, \omega_i)$ -Modell mit $\mu_i \in [2, 5]$ und $\omega_i \in [0.1, 0.25]$.



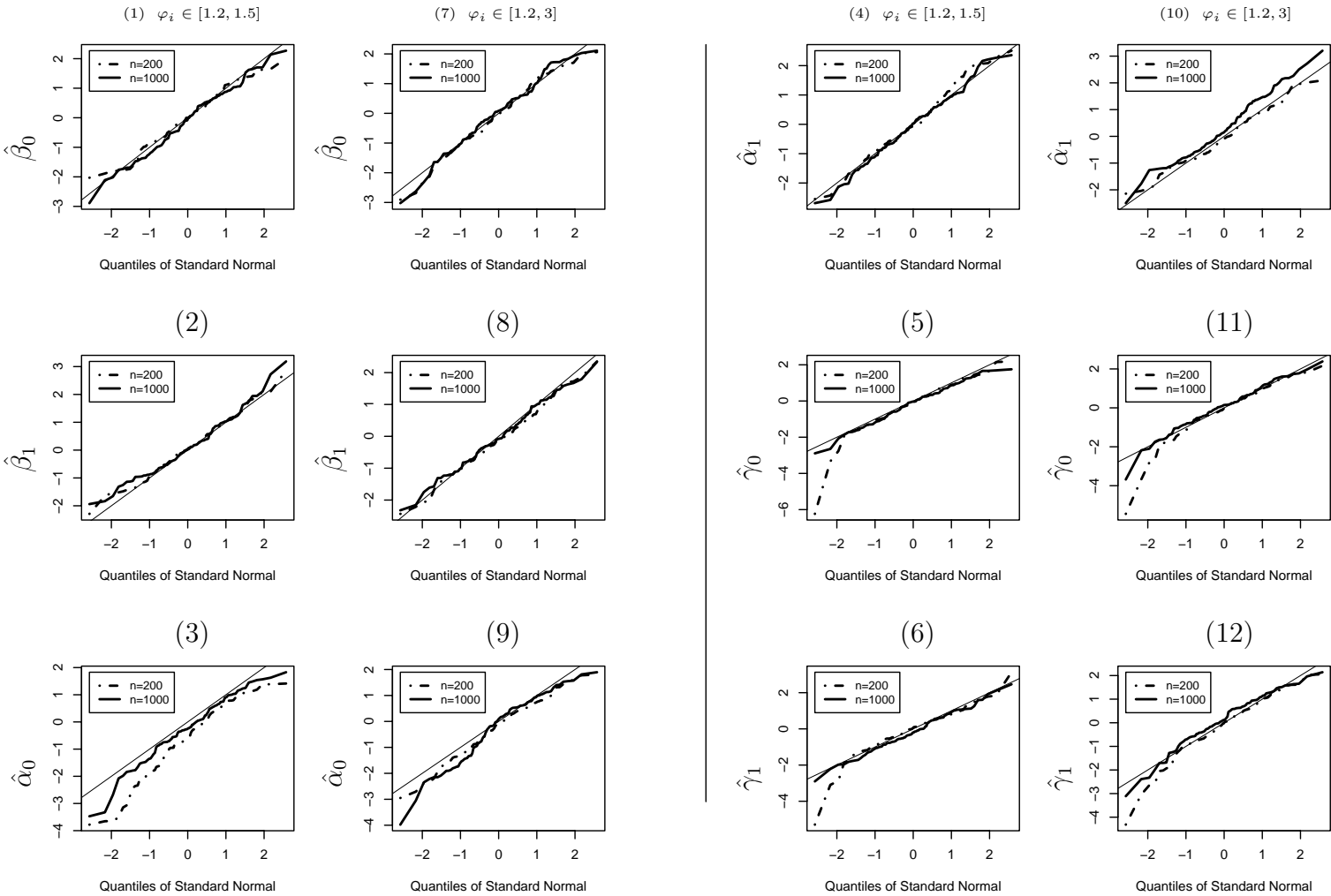


Abbildung 2.3: QQ-Plots zentrierter und normierter ML Schätzer basierend auf $N = 100$ Datensätzen im $ZIGP(\mu_i, \varphi_i, \omega_i)$ -Modell mit $\mu_i \in [2, 15]$ und $\omega_i \in [0.1, 0.25]$.

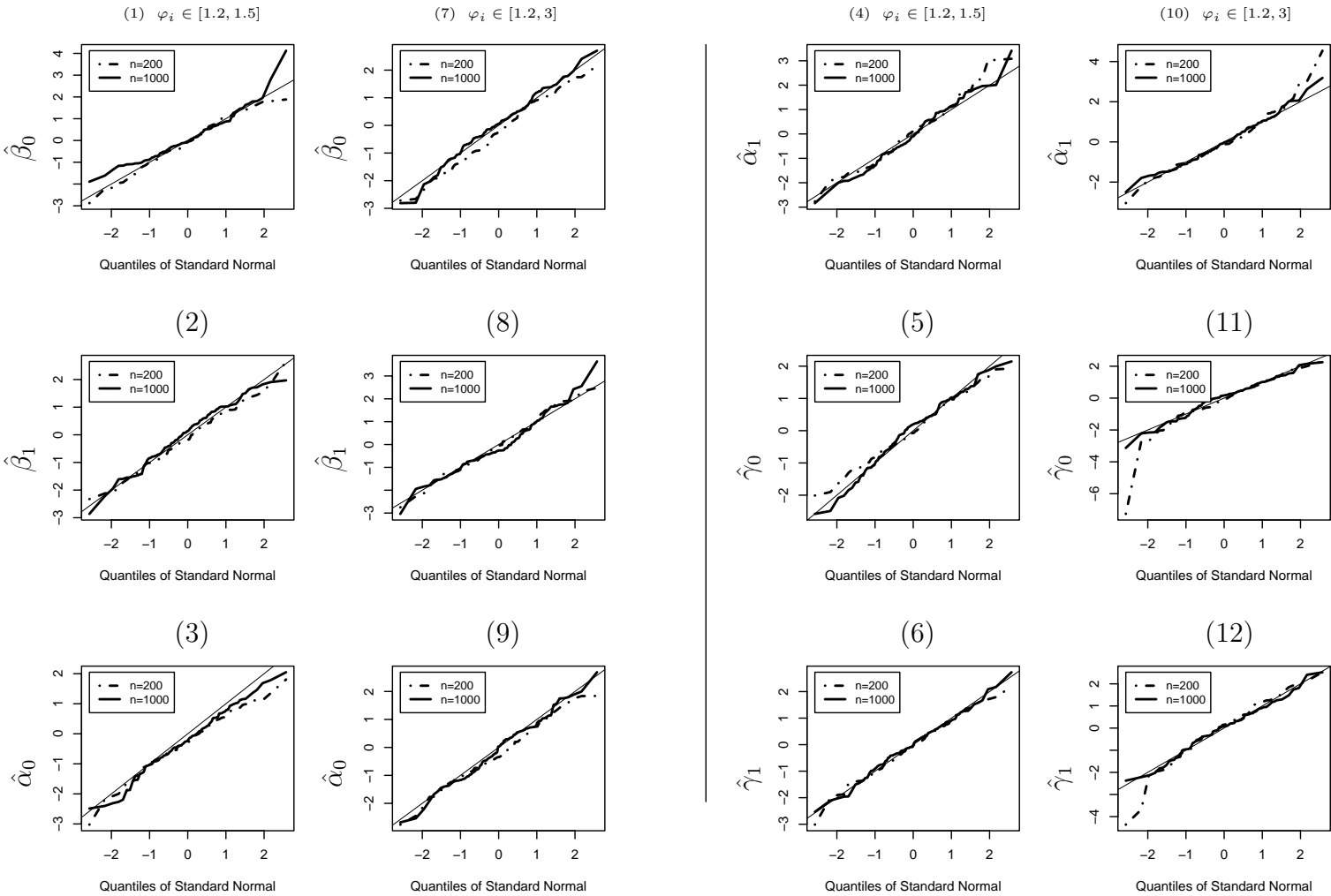


Abbildung 2.4: QQ-Plots zentrierter und normierter ML Schätzer basierend auf $N = 100$ Datensätzen im $ZIGP(\mu_i, \varphi_i, \omega_i)$ -Modell mit $\mu_i \in [2, 15]$ und $\omega_i \in [0.1, 0.5]$.

Kapitel 3

Zero Modified Generalized Poisson - Modelle

3.1 Einführung

Zero Modified Poisson - Modelle erleichtern Hypothesentests auf den Nullenüberschuß und wurden bereits von Dietz und Böhning [8] untersucht: oft möchte man Hypothesen wie $H_0 : \omega = 0$ gegen $H_1 : \text{Nicht } H_0$ testen. Es kann sich jedoch um einen einseitigen Ablehnungsbereich der Form $H_1 : \omega > 0$ oder aber um einen zweiseitigen Ablehnungsbereich $H_1 : \omega \neq 0$ handeln. Dies spielt eine entscheidende Rolle für die Verteilung der Teststatistik. Für Score-Tests (z.B. Verbeke und Molenberghs [36]) seien T_S Teststatistiken des Score-Tests, dann gilt bei zweiseitigen Tests approximativ: $P(T_S > c \mid H_0) = P(\chi_1^2 > c)$. Ist der Ablehnungsbereich jedoch einseitig, so liegt die Nullhypothese *auf dem Rand des Parameterraums* und die approximative Verteilung der Teststatistik "verkompliziert" sich zu $P(T_S > c \mid H_0) = \frac{1}{2}P(\chi_1^2 > c) + \frac{1}{2}P(\chi_0^2 > c)$ (s. Verbeke und Molenberghs [36, S. 256]). Dabei ist χ_0^2 die Verteilung mit der gesamten Wahrscheinlichkeitsmasse bei 0.

Nun möchte man Regression auf den Nullenüberschußparameter ω zulassen. Verwendet man dafür Logit-Link, so gilt sogar $\omega_i(\gamma) \in (0, 1)$, womit die Null gar nicht im Parameterraum von $\omega_i(\gamma)$ enthalten ist. Damit ist ein Test auf Nullenüberschuß von vornherein nicht möglich. Durch Regression auf ω kann man sich Fragen nähern, die bei festem ZI-Parameter nicht beantwortet werden können. Beispiel: Eine Stichprobe von n Beobachtungen läßt sich in k Gruppen unterteilen. Ridout et al. [32] etwa untersuchen die Anzahl von Wurzeln, die Sprößlinge von Apfelbäumen bei unterschiedlicher Lichteinstrahlung und Düngerzugabe entwickeln. Für die Düngerdosis gibt es vier Gruppen. Eine Fragestellung kann nun lauten: bei welcher dieser Gruppen ist ein ZI-Parameter signifikant verschieden von Null? Wie man diese Fragestellung untersuchen kann, wird später beantwortet.

Eine Lösung für das Problem, daß beim Logit-Link $\omega_i(\gamma) \in (0, 1)$ die Null nicht enthält, ist, den Parameterraum der ω_i so festzulegen, daß auch der Fall einer "Zero Deflation" mit Auswahlparameter $\omega_i < 0$ zulässig ist. Dazu müssen ω_i hier jedoch als Verteilungsparameter und nicht als Wahrscheinlichkeiten betrachtet werden.

Wird ω_i aber zu klein gewählt, wird $P(Y_i = 0) = \omega_i + (1 - \omega_i)P_i^0(\delta) \leq 0$, und damit unzulässig. Die Fragestellung lautet also: *wie groß muß (ein negatives) ω_i mindestens sein, damit $P(Y_i = 0) > 0$?*

Um eine untere Schranke $\kappa_\omega^u < 0$ für $\omega_i \in (\kappa_\omega^u, 1)$ zu finden, werden beschränkte Werte $\mu_i \in (0, \kappa_\mu^o)$ und $\varphi_i \in (1, \infty)$ vorausgesetzt. Konkret bedeutet dies für φ_i , daß der (seltenen) Fall der Unter- oder Äquidispersion ausgeschlossen wird. Für μ_i kann eine obere Schranke etwa ein Erfahrungswert sein: im Beispiel der Anzahl von Versicherungsschäden kann ein Erfahrungswert sein, daß pro Kfz-Police und Versicherungsjahr i.d.R. 10 Versicherungsschäden nicht überschritten werden.

Satz 3.1 (Untere Schranke für ω_i).

Für feste und bekannte Schranken $\mu_i \in (0, \kappa_\mu^o)$, $\varphi_i \in (1, \infty) \forall i$ gilt $P(Y_i = 0) = \omega_i + (1 - \omega_i)e^{-\frac{\mu_i}{\varphi_i}} > 0 \forall i$, falls

$$\omega_i > \frac{e^{-\kappa_\mu^o}}{e^{-\kappa_\mu^o} - 1} =: \kappa_\omega^u(\kappa_\mu^o) \quad \forall i \quad (3.1)$$

mit $\kappa_\omega^u(\kappa_\mu^o) < 0$.

Beweis.

Da $\varphi_i > 1 \forall i$ und $1 - e^{-\frac{\mu_i}{\varphi_i}} > 0 \forall \mu_i \in (0, \kappa_\mu^o)$, gilt:

$$\begin{aligned} P(Y_i = 0) = \omega_i + (1 - \omega_i)e^{-\frac{\mu_i}{\varphi_i}} &> 0 && \forall \mu_i \in (0, \kappa_\mu^o) \quad \forall \varphi_i \in (1, \infty) \\ \Leftrightarrow \omega_i(1 - e^{-\frac{\mu_i}{\varphi_i}}) &> -e^{-\frac{\mu_i}{\varphi_i}} && \forall \mu_i \in (0, \kappa_\mu^o) \quad \forall \varphi_i \in (1, \infty) \\ \Leftrightarrow \omega_i &> \frac{e^{-\frac{\mu_i}{\varphi_i}}}{e^{-\frac{\mu_i}{\varphi_i}} - 1} && \forall \mu_i \in (0, \kappa_\mu^o) \quad \forall \varphi_i \in (1, \infty) \\ \Leftrightarrow \omega_i &> \max_j \left(\frac{e^{-\frac{\mu_j}{\varphi_j}}}{e^{-\frac{\mu_j}{\varphi_j}} - 1} \right) \geq \frac{e^{-\frac{\kappa_\mu^o}{1}}}{e^{-\frac{\kappa_\mu^o}{1}} - 1}. && (3.2) \end{aligned}$$

Für Formel (3.2) beachte man: sei $g(u) := \frac{e^{-u}}{e^{-u}-1}$. Dann ist $g'(u) = \frac{e^{-u}}{(e^{-u}-1)^2} > 0$. Daher nimmt $g(u)$ sein Supremum am maximalen zulässigen Wert von u an. Auf den obigen Fall angewendet muß μ_i also maximal und φ_i minimal werden.

Zum Schluß gilt noch: $\kappa_\mu^o > 0 \Rightarrow e^{-\kappa_\mu^o} < 1 \Rightarrow \kappa_\omega^u(\kappa_\mu^o) < 0$. □

Nun möchte man eine geeignete Linkfunktion für ω_i finden, so daß $\omega_i = g(\mathbf{z}_i^t \boldsymbol{\gamma}) \in (\kappa_\omega^u(\kappa_\mu^o), 1)$. Sie soll

$$g: \mathbb{R} \rightarrow (\kappa_\omega^u(\kappa_\mu^o), 1), \quad \kappa_\omega^u(\kappa_\mu^o) < 0 \quad (3.3)$$

$$g \text{ monoton steigend} \quad (3.4)$$

erfüllen.

Satz 3.2 (Linkfunktion für ω_i).

Die Funktion

$$\omega_i(\boldsymbol{\gamma}) = g(\mathbf{z}_i, \boldsymbol{\gamma}) := \frac{\kappa_\omega^u(\kappa_\mu^o) + e^{\mathbf{z}_i^t \boldsymbol{\gamma}}}{1 + e^{\mathbf{z}_i^t \boldsymbol{\gamma}}}$$

$$\Leftrightarrow \mathbf{z}_i^t \boldsymbol{\gamma} = \log \left(\frac{\omega_i(\boldsymbol{\gamma}) - \kappa_\omega^u(\kappa_\mu^o)}{1 - \omega_i(\boldsymbol{\gamma})} \right) \quad (3.5)$$

erfüllt (3.3) und (3.4).

Beweis.

Mit den Notationen $\eta_i := \mathbf{z}_i^t \boldsymbol{\gamma}$ und $K := \kappa_\omega^u(\kappa_\mu^o)$ läßt sich die Linkfunktion als $g(\eta_i) = \frac{K + e^{\eta_i}}{1 + e^{\eta_i}}$ darstellen. Unter Anwendung der L' Hospitalischen Regel kann man (3.3) zeigen:

$$\begin{aligned} \lim_{\eta_i \rightarrow \infty} g(\eta_i) &= \lim_{\eta_i \rightarrow \infty} \frac{K + e^{\eta_i}}{1 + e^{\eta_i}} = \lim_{\eta_i \rightarrow \infty} \frac{e^{\eta_i}}{e^{\eta_i}} = 1, \\ \lim_{\eta_i \rightarrow -\infty} g(\eta_i) &= K, \\ g'(\eta_i) &= \frac{e^{\eta_i}(1 + e^{\eta_i}) - (K + e^{\eta_i})e^{\eta_i}}{(1 + e^{\eta_i})^2} = \frac{e^{\eta_i}(1 - K)}{(1 + e^{\eta_i})^2} > 0. \end{aligned}$$

Damit ist auch die Monotonie (3.4) gezeigt. \square

Die Linkfunktion sieht für $\kappa_\omega^u(\kappa_\mu^o) = -0.1$ wie in Abb. 3.1 aus.

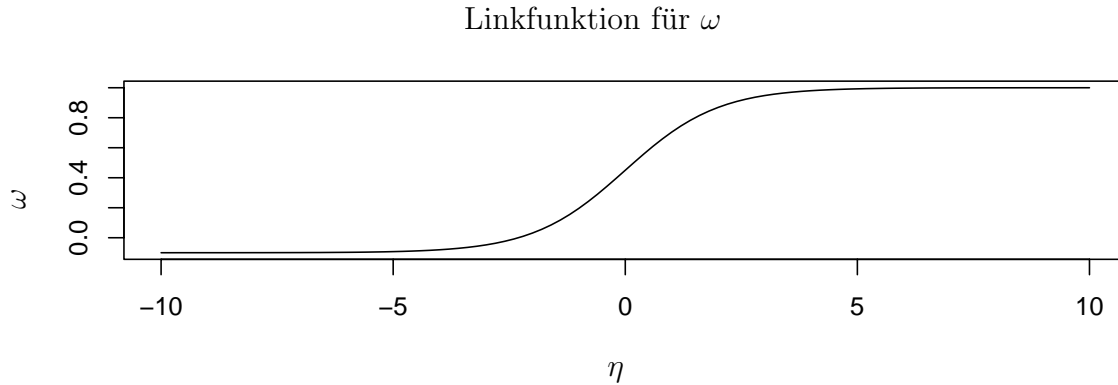


Abbildung 3.1: Linkfunktion für Zero Modified Modelle mit $\kappa_\omega^u(\kappa_\mu^o) = -0.1$

Zuletzt soll noch die Frage beantwortet werden, wie mit Link-Funktion 3.5 auf Nullenüberschuß verschiedener Gruppen getestet werden kann. Man denke wieder an die Anzahl von Wurzeln bei Apfelsprößlingen nach Ridout et al. [32]. Man unterteilt die Sprößlinge in vier Gruppen und behandelt sie mit verschiedenen Düngerdosen, wobei die erste Gruppe als Referenzkategorie diene. Dann sieht die Designmatrix so aus:

$$\mathbf{Z} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

Der Parameter γ_0 ist ein Maß für die Zero Inflation der Referenzkategorie, also von Gruppe 1. Liegt in dieser Gruppe kein Nullenüberschuß vor, so ergibt sich für i aus Gruppe 1, daß $\omega_i = \frac{\kappa_\omega^u(\kappa_\mu^o) + e^{z_i^t \gamma}}{1 + e^{z_i^t \gamma}} = 0 \Leftrightarrow \gamma_0 = \log(-\kappa_\omega^u(\kappa_\mu^o))$. Man testet daher $H_0 : \gamma_0 = -\log(\kappa_\omega^u(\kappa_\mu^o))$ gegen $H_1 : \gamma_0 \neq -\log(\kappa_\omega^u(\kappa_\mu^o))$. Um auf Nullenüberschuß einer Gruppe $j \neq 1$ zu testen, muß die Nullhypothese lauten: $H_0 : \gamma_j = -\gamma_0 + \log(-\kappa_\omega^u(\kappa_\mu^o))$ gegen $H_1 : \gamma_j \neq -\gamma_0 + \log(-\kappa_\omega^u(\kappa_\mu^o))$.

3.2 Definition und Eigenschaften des $ZMGP(\mu_i, \varphi, \omega_i)$ – Regressionsmodells

3.2.1 $ZMGP(\mu_i, \varphi, \omega_i)$ – Regressionsmodell

Das Regressionsmodell aus Kapitel 1 wird nun angepaßt. Man nimmt weiter an, daß die Daten \mathbf{Y} noch immer einer $ZIGP(\mu_i, \varphi, \omega_i)$ -Verteilung folgen. Nun ist aber ω_i mit Linkfunktion (3.5) verknüpft, deshalb schreibt man $Y_i \sim ZMGP(\mu_i, \varphi, \omega_i)$. Das Modell heiße "Zero Modified Generalized Poisson"– oder " $ZMGP(\mu_i, \varphi, \omega_i)$ "–Regressionsmodell. Eine Regression auf den Dispersionsparameter φ wird hier nicht zugelassen.

Zero Modified GP – Regressionsmodell $ZMGP(\mu_i, \varphi, \omega_i)$

1. *Zufallskomponente:*
 $\{Y_i, 1 \leq i \leq n\}$ sind unabhängig mit $Y_i \sim ZMGP(\mu_i, \varphi, \omega_i)$.
2. *Systematische Komponente:*
 Die linearen Prädiktoren $\eta_i^\mu(\boldsymbol{\beta}) = \mathbf{x}_i^t \boldsymbol{\beta}$ und $\eta_i^\omega(\boldsymbol{\gamma}) = \mathbf{z}_i^t \boldsymbol{\gamma}$, $i = 1, \dots, n$ beeinflussen den Response Y_i . Dabei sind $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_p)$ und $\boldsymbol{\gamma} = (\gamma_0, \gamma_1, \dots, \gamma_q)$ unbekannte Regressionsparameter. Die Matrizen $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^t$ und $\mathbf{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_n)^t$ werden als Designmatrizen bezeichnet.
3. *Parametrische Linkfunktionen:*
 Die linearen Prädiktoren $\eta_i^\mu(\boldsymbol{\beta})$ und $\eta_i^\omega(\boldsymbol{\gamma})$ werden mit $\mu_i(\boldsymbol{\beta})$ und $\omega_i(\boldsymbol{\gamma})$, $i = 1, \dots, n$ wie folgt verknüpft. Der Dispersionsparameter φ wird als konstant angekommen. Wieder seien E_i Exposurevariablen mit $E_i > 0 \forall i$.

(i) *Erwartungswertebene*

$$\begin{aligned} E(Y_i | \boldsymbol{\beta}) = \mu_i(\boldsymbol{\beta}) &:= E_i e^{\mathbf{x}_i^t \boldsymbol{\beta}} = e^{\mathbf{x}_i^t \boldsymbol{\beta} + \log(E_i)} > 0 \\ \Leftrightarrow \eta_i^\mu(\boldsymbol{\beta}) &= \log(\mu_i(\boldsymbol{\beta})) - \log(E_i) \quad (\text{Log-Link}), \end{aligned} \quad (3.6)$$

(ii) *Überdispersionsebene*

$$\varphi \quad \text{konstant mit } \varphi > 1, \quad (3.7)$$

(iii) *Nullenüberschußebene*

$$\begin{aligned} \omega_i(\boldsymbol{\gamma}) &:= \frac{K + e^{z_i^t \boldsymbol{\gamma}}}{1 + e^{z_i^t \boldsymbol{\gamma}}} \in (K, 1), \quad K < 0, \\ \Leftrightarrow \eta_i^\omega(\boldsymbol{\gamma}) &= \log\left(\frac{\omega_i(\boldsymbol{\gamma}) - K}{1 - \omega_i(\boldsymbol{\gamma})}\right) \quad (\text{modf. Logit-Link}). \end{aligned} \quad (3.8)$$

Es werden die bereits bekannten und eine neue Notation verwendet.

$$\begin{aligned}\boldsymbol{\delta} &:= (\boldsymbol{\beta}^t, \varphi, \boldsymbol{\gamma}^t)^t, \\ \mu_i(\boldsymbol{\beta}) &:= e^{\mathbf{x}_i^t \boldsymbol{\beta} + \log(E_i)}, \\ k_i(\boldsymbol{\gamma}) &:= e^{\mathbf{z}_i^t \boldsymbol{\gamma}}, \\ P_i^0(\boldsymbol{\delta}) &:= e^{-\frac{E_i}{\varphi} e^{\mathbf{x}_i^t \boldsymbol{\beta}}}\end{aligned}$$

und

$$K := \kappa_{\omega}^u(\kappa_{\mu}^o) \quad \text{wie in Satz 3.1.}$$

3.2.2 Loglikelihoodfunktion der $ZMGP(\mu_i, \varphi, \omega_i)$ -Regression

Satz 3.3 (Loglikelihood-Funktion).

Die Loglikelihoodfunktion im Falle von ZMGP - Modellen lautet

$$\begin{aligned}l(\boldsymbol{\delta}) &= \sum_{i=1}^n \mathbb{1}_{\{y_i=0\}} [\log(k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1 - K)) - \log(1 + k_i(\boldsymbol{\gamma}))] \\ &\quad + \mathbb{1}_{\{y_i>0\}} \left[\log(1 - K) - \log(1 + k_i(\boldsymbol{\gamma})) + \log(E_i) + \mathbf{x}_i^t \boldsymbol{\beta} + (y_i - 1) \log(\mu_i(\boldsymbol{\beta})) \right. \\ &\quad \left. + (\varphi - 1)y_i - \log(y_i!) - y_i \log \varphi - \frac{1}{\varphi}(\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i) \right].\end{aligned}$$

Beweis.

$$\begin{aligned}l(\boldsymbol{\delta}) &= \log L(\boldsymbol{\delta}) = \sum_{i=1}^n \mathbb{1}_{\{y_i=0\}} \log \left[\frac{K + k_i(\boldsymbol{\gamma})}{1 + k_i(\boldsymbol{\gamma})} + \frac{1 - K}{1 + k_i(\boldsymbol{\gamma})} P_i^0(\boldsymbol{\delta}) \right] \\ &\quad + \mathbb{1}_{\{y_i>0\}} \left[\log \left(\frac{1 - K}{1 + k_i(\boldsymbol{\gamma})} \right) + \log(\mu_i(\boldsymbol{\beta})) + (y_i - 1) \log(\mu_i(\boldsymbol{\beta})) + (\varphi - 1)y_i \right. \\ &\quad \left. - \log(y_i!) - y_i \log \varphi - \frac{1}{\varphi}(\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i) \right] \\ &= \sum_{i=1}^n \mathbb{1}_{\{y_i=0\}} [\log(k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1 - K)) - \log(1 + k_i(\boldsymbol{\gamma}))] \\ &\quad + \mathbb{1}_{\{y_i>0\}} \left[\log(1 - K) - \log(1 + k_i(\boldsymbol{\gamma})) + \log(E_i) + \mathbf{x}_i^t \boldsymbol{\beta} + (y_i - 1) \log(\mu_i(\boldsymbol{\beta})) \right. \\ &\quad \left. + (\varphi - 1)y_i - \log(y_i!) - y_i \log \varphi - \frac{1}{\varphi}(\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i) \right].\end{aligned}$$

□

3.2.3 Score-Vektor der $ZMGP(\mu_i, \varphi, \omega_i)$ -Regression

Der Score-Vektor werde mit $\mathbf{s}_n(\boldsymbol{\delta}) := \left(\frac{\partial l(\boldsymbol{\delta})}{\partial \boldsymbol{\beta}}, \frac{\partial l(\boldsymbol{\delta})}{\partial \varphi}, \frac{\partial l(\boldsymbol{\delta})}{\partial \boldsymbol{\gamma}} \right)^t$ bezeichnet. Er lautet

Satz 3.4 (Score-Vektor).

Die Einträge des Score-Vektors lauten

$$\begin{aligned} \frac{\partial l(\boldsymbol{\delta})}{\partial \beta_j} &= \sum_{i=1}^n x_{ij} \left(\mathbb{1}_{\{y_i=0\}} \left[\frac{P_i^0(\boldsymbol{\delta})(-\frac{\mu_i(\boldsymbol{\beta})}{\varphi})(1-K)}{k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1-K)} \right] \right. \\ &\quad \left. + \mathbb{1}_{\{y_i>0\}} \left[1 + (y_i - 1) \frac{\mu_i(\boldsymbol{\beta})}{\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i} - \frac{1}{\varphi} \mu_i(\boldsymbol{\beta}) \right] \right), \\ j &= 0, \dots, p, \end{aligned} \quad (3.9)$$

$$\begin{aligned} \frac{\partial l(\boldsymbol{\delta})}{\partial \varphi} &= \sum_{i=1}^n \mathbb{1}_{\{y_i=0\}} \left[\frac{\frac{1}{\varphi^2} P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta})(1-K)}{k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1-K)} \right] \\ &\quad + \mathbb{1}_{\{y_i>0\}} \left[\frac{(y_i - 1)y_i}{\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i} - \frac{y_i}{\varphi} + \frac{\mu_i(\boldsymbol{\beta}) - y_i}{\varphi^2} \right], \end{aligned} \quad (3.10)$$

$$\begin{aligned} \frac{\partial l(\boldsymbol{\delta})}{\partial \gamma_j} &= \sum_{i=1}^n z_{ij} k_i(\boldsymbol{\gamma}) \left(\mathbb{1}_{\{y_i=0\}} \left[\frac{1}{k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1-K)} \right] - \frac{1}{1 + k_i(\boldsymbol{\gamma})} \right), \\ j &= 0, \dots, q. \end{aligned} \quad (3.11)$$

Beweis.

Analog zu Kapitel 1 werden die partiellen Ableitungen gebildet. Hier lautet der Zähler in der Linkfunktion von $\omega_i(\boldsymbol{\gamma})$ aber nicht mehr $k_i(\boldsymbol{\gamma})$, sondern $K + k_i(\boldsymbol{\gamma})$. \square

Damit kann man die Hessematrix berechnen.

3.2.4 Hessematrix der ZMGP(μ_i, φ, ω_i)–Regression

Im betrachteten Regressionsmodell kann die Hessematrix partitioniert werden als

$$\mathcal{H}(\boldsymbol{\delta}) = \mathbf{I}_{obs} = \begin{pmatrix} \mathcal{H}_{\beta\beta}(\boldsymbol{\delta}) & \mathcal{H}_{\beta\varphi}(\boldsymbol{\delta}) & \mathcal{H}_{\beta\gamma}(\boldsymbol{\delta}) \\ \mathcal{H}_{\varphi\beta}(\boldsymbol{\delta}) & \mathcal{H}_{\varphi\varphi}(\boldsymbol{\delta}) & \mathcal{H}_{\varphi\gamma}(\boldsymbol{\delta}) \\ \mathcal{H}_{\gamma\beta}(\boldsymbol{\delta}) & \mathcal{H}_{\gamma\varphi}(\boldsymbol{\delta}) & \mathcal{H}_{\gamma\gamma}(\boldsymbol{\delta}) \end{pmatrix}.$$

Dabei sind

$$\begin{aligned} \mathcal{H}_{\beta\beta}(\boldsymbol{\delta}) &= \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^t} \in \mathbb{R}^{p \times p}, \\ \mathcal{H}_{\varphi\varphi}(\boldsymbol{\delta}) &= \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \varphi^2} \in \mathbb{R}, \\ \mathcal{H}_{\gamma\gamma}(\boldsymbol{\delta}) &= \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\gamma} \partial \boldsymbol{\gamma}^t} \in \mathbb{R}^{q \times q}, \\ \mathcal{H}_{\beta\varphi}(\boldsymbol{\delta}) = \mathcal{H}_{\varphi\beta}(\boldsymbol{\delta})^t &= \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\beta}^t \partial \varphi} \in \mathbb{R}^{p \times 1}, \\ \mathcal{H}_{\beta\gamma}(\boldsymbol{\delta}) = \mathcal{H}_{\gamma\beta}(\boldsymbol{\delta})^t &= \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\gamma}^t} \in \mathbb{R}^{p \times q} \end{aligned}$$

und

$$\mathcal{H}_{\varphi\gamma}(\delta) = \mathcal{H}_{\gamma\varphi}(\delta)^t = \frac{\partial^2 l(\delta)}{\partial\varphi\partial\gamma^t} \in \mathbb{R}^{1 \times q}.$$

Erneut nutzt man die Hilfsfunktion $g(x_i, \beta, \varphi) = -\frac{1}{\varphi} P_i^0(\delta) \mu_i(\beta)$ aus Lemma 1.10 mit

$$\frac{\partial}{\partial\beta_m} g(x_i, \beta, \varphi) = x_{im} \mu_i(\beta) \frac{\mu_i(\beta) - \varphi}{\varphi^2} P_i^0(\delta).$$

Damit ergibt sich:

$$\begin{aligned} \frac{\partial^2 l(\delta)}{\partial\beta_l \partial\beta_m} &= \sum_{i=1}^n x_{il} \left(\mathbb{1}_{\{y_i=0\}} (1-K) \left[\frac{\frac{\partial}{\partial\beta_m} g(x_i, \beta, \varphi) (k_i(\gamma) + K + P_i^0(\delta)(1-K))}{(k_i(\gamma) + K + P_i^0(\delta)(1-K))^2} \right. \right. \\ &\quad \left. \left. - \frac{g(x_i, \beta, \varphi) P_i^0(\delta) (-\frac{\mu_i(\beta)}{\varphi}) x_{im} (1-K)}{(k_i(\gamma) + K + P_i^0(\delta)(1-K))^2} \right] + \mathbb{1}_{\{y_i>0\}} \right. \\ &\quad \left. \times \left[(y_i - 1) \frac{\mu_i(\beta) x_{im} (\mu_i(\beta) + (\varphi - 1)y_i) - (\mu_i(\beta))^2 x_{im}}{(\mu_i(\beta) + (\varphi - 1)y_i)^2} - \frac{1}{\varphi} \mu_i(\beta) x_{im} \right] \right) \\ &\stackrel{L.1.10}{=} \sum_{i=1}^n x_{il} x_{im} \mu_i(\beta) \left(\mathbb{1}_{\{y_i=0\}} (1-K) \right. \\ &\quad \times \left[\frac{\frac{\mu_i(\beta) - \varphi}{\varphi^2} P_i^0(\delta) (k_i(\gamma) + K + P_i^0(\delta)(1-K))}{(k_i(\gamma) + K + P_i^0(\delta)(1-K))^2} \right. \\ &\quad \left. \left. - \frac{\frac{1}{\varphi^2} \mu_i(\beta) P_i^0(\delta)^2 (1-K)}{(k_i(\gamma) + K + P_i^0(\delta)(1-K))^2} \right] \right. \\ &\quad \left. + \mathbb{1}_{\{y_i>0\}} \left[\frac{(\varphi - 1)(y_i - 1)y_i}{(\mu_i(\beta) + (\varphi - 1)y_i)^2} - \frac{1}{\varphi} \right] \right) \\ &= \sum_{i=1}^n x_{il} x_{im} \mu_i(\beta) \left(\mathbb{1}_{\{y_i=0\}} (1-K) \left[\frac{-\frac{1}{\varphi} P_i^0(\delta)^2 (1-K)}{(k_i(\gamma) + K + P_i^0(\delta)(1-K))^2} \right. \right. \\ &\quad \left. \left. + \frac{\frac{\mu_i(\beta) - \varphi}{\varphi^2} P_i^0(\delta) (k_i(\gamma) + K)}{(k_i(\gamma) + K + P_i^0(\delta)(1-K))^2} \right] \right. \\ &\quad \left. + \mathbb{1}_{\{y_i>0\}} \left[\frac{(\varphi - 1)(y_i - 1)y_i}{(\mu_i(\beta) + (\varphi - 1)y_i)^2} - \frac{1}{\varphi} \right] \right), \quad l, m = 0, \dots, p, \\ \frac{\partial^2 l(\delta)}{\partial\beta_l \partial\varphi} &= \frac{\partial^2}{\partial\varphi \partial\beta_l} l(\delta) = \sum_{i=1}^n \left(\mathbb{1}_{\{y_i=0\}} (1-K) \right. \\ &\quad \times \left[\frac{(-P_i^0(\delta) \frac{\mu_i(\beta)}{\varphi} x_{il} \frac{\mu_i(\beta)}{\varphi^2} + P_i^0(\delta) \frac{\mu_i(\beta)}{\varphi^2} x_{il}) (k_i(\gamma) + K + P_i^0(\delta)(1-K))}{(k_i(\gamma) + K + P_i^0(\delta)(1-K))^2} \right. \\ &\quad \left. \left. + \frac{P_i^0(\delta) \frac{\mu_i(\beta)^2}{\varphi^3} P_i^0(\delta) x_{il} (1-K)}{k_i(\gamma) + K + P_i^0(\delta)(1-K)} \right] \right) \end{aligned}$$

$$\begin{aligned}
& + \mathbb{1}_{\{y_i > 0\}} \left[\frac{-(y_i - 1)y_i \mu_i(\boldsymbol{\beta}) x_{il}}{(\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i)^2} + \frac{\mu_i(\boldsymbol{\beta}) x_{il}}{\varphi^2} \right] \\
& = \sum_{i=1}^n x_{il} \mu_i(\boldsymbol{\beta}) \left(\mathbb{1}_{\{y_i=0\}} (1 - K) \left[\frac{-P_i^0(\boldsymbol{\delta}) \frac{\mu_i(\boldsymbol{\beta})}{\varphi^3} (k_i(\boldsymbol{\gamma}) + K)}{(k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1 - K))^2} \right. \right. \\
& \quad \left. \left. + \frac{P_i^0(\boldsymbol{\delta}) \frac{1}{\varphi^2} (k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1 - K))}{(k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1 - K))^2} \right] \right. \\
& \quad \left. + \mathbb{1}_{\{y_i > 0\}} \left[\frac{-(y_i - 1)y_i}{(\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i)^2} + \frac{1}{\varphi^2} \right] \right), \quad l = 0, \dots, p, \\
\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \beta_l \partial \gamma_r} & = \sum_{i=1}^n x_{il} z_{ir} \mathbb{1}_{\{y_i=0\}} (1 - K) \left[\frac{\frac{\mu_i(\boldsymbol{\beta})}{\varphi} P_i^0(\boldsymbol{\delta}) k_i(\boldsymbol{\gamma})}{(k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1 - K))^2} \right], \\
& \quad l = 0, \dots, p \text{ und } r = 0, \dots, q, \\
\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \varphi^2} & = \sum_{i=1}^n \mathbb{1}_{\{y_i=0\}} (1 - K) \\
& \quad \times \left[\frac{(-\mu_i(\boldsymbol{\beta}) \frac{2}{\varphi^3} P_i^0(\boldsymbol{\delta}) + P_i^0(\boldsymbol{\delta}) \frac{\mu_i(\boldsymbol{\beta})^2}{\varphi^4}) (k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1 - K))}{(k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1 - K))^2} \right. \\
& \quad \left. - \frac{\frac{1}{\varphi^4} P_i^0(\boldsymbol{\delta})^2 \mu_i(\boldsymbol{\beta})^2 (1 - K)}{(k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1 - K))^2} \right] \\
& \quad + \mathbb{1}_{\{y_i > 0\}} \left[\frac{-y_i^2 (y_i - 1)}{(\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i)^2} + \frac{y_i}{\varphi^2} - 2 \frac{\mu_i(\boldsymbol{\beta}) - y_i}{\varphi^3} \right] \\
& = \sum_{i=1}^n \mathbb{1}_{\{y_i=0\}} (1 - K) \\
& \quad \times \frac{-2 \frac{\mu_i(\boldsymbol{\beta}) P_i^0(\boldsymbol{\delta})}{\varphi^3} (k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1 - K)) + \frac{P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta})^2}{\varphi^4} (k_i(\boldsymbol{\gamma}) + K)}{(k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1 - K))^2} \\
& \quad + \mathbb{1}_{\{y_i > 0\}} \left[\frac{-y_i^2 (y_i - 1)}{(\mu_i(\boldsymbol{\beta}) + (\varphi - 1)y_i)^2} + \frac{y_i}{\varphi^2} - 2 \frac{\mu_i(\boldsymbol{\beta}) - y_i}{\varphi^3} \right], \\
\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \varphi \partial \gamma_s} & = \sum_{i=1}^n z_{is} \mathbb{1}_{\{y_i=0\}} \left[\frac{-P_i^0(\boldsymbol{\delta}) \frac{\mu_i(\boldsymbol{\beta})}{\varphi^2} (1 - K) k_i(\boldsymbol{\gamma})}{(k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1 - K))^2} \right], \quad s = 0, \dots, q, \\
\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \gamma_j \partial \gamma_t} & = \sum_{i=1}^n z_{ij} \left(\mathbb{1}_{\{y_i=0\}} \left[\frac{k_i(\boldsymbol{\gamma}) z_{it} (k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1 - K)) - k_i(\boldsymbol{\gamma})^2 z_{it}}{(k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1 - K))^2} \right] \right. \\
& \quad \left. - \frac{k_i(\boldsymbol{\gamma}) z_{it} (1 + k_i(\boldsymbol{\gamma})) - (k_i(\boldsymbol{\gamma}))^2 z_{it}}{(1 + k_i(\boldsymbol{\gamma}))^2} \right) \\
& = \sum_{i=1}^n z_{ij} z_{it} k_i(\boldsymbol{\gamma}) \left(\mathbb{1}_{\{y_i=0\}} \left[\frac{K + P_i^0(\boldsymbol{\delta})(1 - K)}{(k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1 - K))^2} \right] \right.
\end{aligned}$$

$$-\frac{1}{(1+k_i(\gamma))^2}\Bigg), \quad j, t = 0, \dots, q.$$

3.2.5 Erwartete Fisher-Information der $ZMGP(\mu_i, \varphi, \omega_i)$ -Regression

Die erwartete Fisher-Informationsmatrix hat die Struktur

$$\mathbf{F}_n(\boldsymbol{\delta}) = -E\mathcal{H}_n(\boldsymbol{\delta}) = -E \begin{pmatrix} \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^t} & \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\beta}^t \partial \varphi} & \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \boldsymbol{\beta} \partial \gamma^t} \\ \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \varphi \partial \boldsymbol{\beta}} & \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \varphi^2} & \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \varphi \partial \gamma} \\ \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \gamma \partial \boldsymbol{\beta}^t} & \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \gamma^t \partial \varphi} & \frac{\partial^2 l(\boldsymbol{\delta})}{\partial \gamma \partial \gamma^t} \end{pmatrix}.$$

Der Erwartungswert der Indikatorfunktion $E(\mathbb{1}_{\{y_i=0\}})$ ändert sich nun aber gegenüber Kapitel 1.

Lemma 3.5 (Erwartungswert der Indikatorfunktion).

Für Zufallsvariablen $Y_i \sim ZMGP(\mu_i, \varphi, \omega_i)$ und Linkfunktionen (3.6) und (3.8) gilt

$$E(\mathbb{1}_{\{y_i=0\}}) = a_i \quad i = 1, \dots, n \quad (3.12)$$

und

$$E(\mathbb{1}_{\{y_i>0\}}) = 1 - a_i \quad i = 1, \dots, n \quad (3.13)$$

wobei

$$a_i = \frac{k_i(\gamma) + K + P_i^0(\boldsymbol{\delta})(1 - K)}{1 + k_i(\gamma)}. \quad (3.14)$$

Beweis.

$$E(\mathbb{1}_{\{y_i=0\}}) = 1 \cdot P(Y_i = 0) + 0 \cdot P(Y_i > 0) = \omega_i + (1 - \omega_i)P_i^0(\boldsymbol{\delta}) = \frac{K + k_i(\gamma)}{1 + k_i(\gamma)} + \frac{1 - K}{1 + k_i(\gamma)}P_i^0(\boldsymbol{\delta}). \quad \square$$

Sei a_i also definiert wie in (3.14). Damit lauten die Einträge der erwarteten Fisher-Informationsmatrix:

$$\begin{aligned} -E \left(\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \beta_l \partial \beta_m} \right) &= - \sum_{i=1}^n x_{il} x_{im} \mu_i(\boldsymbol{\beta}) \left((1 - K) \right. \\ &\quad \times \left[\frac{-\frac{1}{\varphi} P_i^0(\boldsymbol{\delta})^2 (1 - K)}{(k_i(\gamma) + K + P_i^0(\boldsymbol{\delta})(1 - K))(1 + k_i(\gamma))} \right. \\ &\quad \left. \left. + \frac{\frac{\mu_i(\boldsymbol{\beta}) - \varphi}{\varphi^2} P_i^0(\boldsymbol{\delta})(k_i(\gamma) + K)}{(k_i(\gamma) + K + P_i^0(\boldsymbol{\delta})(1 - K))(1 + k_i(\gamma))} \right] \right. \\ &\quad \left. + \left[\frac{(\varphi - 1)\mu_i(\boldsymbol{\beta})(1 - K)}{\varphi^2(\mu_i(\boldsymbol{\beta}) - 2 + 2\varphi)(1 + k_i(\gamma))} - (1 - a_i)\frac{1}{\varphi} \right] \right), \\ &\quad l, m = 0, \dots, p, \end{aligned}$$

$$\begin{aligned}
-E \left(\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \beta_l \partial \varphi} \right) &= - \sum_{i=1}^n x_{il} \mu_i(\boldsymbol{\beta}) \left((1-K) \left[\frac{-P_i^0(\boldsymbol{\delta}) \frac{\mu_i(\boldsymbol{\beta})}{\varphi^3} (k_i(\boldsymbol{\gamma}) + K)}{(k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1-K))(1+k_i(\boldsymbol{\gamma}))} \right. \right. \\
&\quad \left. \left. + \frac{P_i^0(\boldsymbol{\delta}) \frac{1}{\varphi^2} (k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1-K))}{(k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1-K))(1+k_i(\boldsymbol{\gamma}))} \right] \right. \\
&\quad \left. - \frac{\mu_i(\boldsymbol{\beta})(1-K)}{\varphi^2(\mu_i(\boldsymbol{\beta}) - 2 + 2\varphi)(1-k_i(\boldsymbol{\gamma}))} + (1-a_i) \frac{1}{\varphi^2} \right), \quad l = 0, \dots, p, \\
-E \left(\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \beta_l \partial \gamma_r} \right) &= - \sum_{i=1}^n x_{il} z_{ir} (1-K) \left[\frac{\frac{\mu_i(\boldsymbol{\beta})}{\varphi} P_i^0(\boldsymbol{\delta}) k_i(\boldsymbol{\gamma})}{(k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1-K))(1+k_i(\boldsymbol{\gamma}))} \right], \\
&\quad l = 0, \dots, p \text{ und } r = 0, \dots, q, \\
-E \left(\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \varphi^2} \right) &= - \sum_{i=1}^n (1-K) \\
&\quad \times \frac{-2 \frac{\mu_i(\boldsymbol{\beta}) P_i^0(\boldsymbol{\delta})}{\varphi^3} (k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1-K)) + \frac{P_i^0(\boldsymbol{\delta}) \mu_i(\boldsymbol{\beta})^2}{\varphi^4} (k_i(\boldsymbol{\gamma}) + K)}{(k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1-K))(1+k_i(\boldsymbol{\gamma}))} \\
&\quad - \frac{\mu_i(\boldsymbol{\beta})(\mu_i(\boldsymbol{\beta}) + 2\varphi)(1-K)}{\varphi^2(\mu_i(\boldsymbol{\beta}) - 2 + 2\varphi)(1+k_i(\boldsymbol{\gamma}))} + \frac{\mu_i(\boldsymbol{\beta})(1-K)}{(1+k_i(\boldsymbol{\gamma}))} \left(\frac{1}{\varphi^2} + \frac{2}{\varphi^3} \right) \\
&\quad - 2(1-a_i) \frac{\mu_i(\boldsymbol{\beta})}{\varphi^3}, \\
-E \left(\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \varphi \partial \gamma_s} \right) &= - \sum_{i=1}^n z_{is} \left[\frac{-P_i^0(\boldsymbol{\delta}) \frac{\mu_i(\boldsymbol{\beta})}{\varphi^2} (1-K) k_i(\boldsymbol{\gamma})}{(k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1-K))(1+k_i(\boldsymbol{\gamma}))} \right], \quad s = 0, \dots, q, \\
-E \left(\frac{\partial^2 l(\boldsymbol{\delta})}{\partial \gamma_j \partial \gamma_t} \right) &= - \sum_{i=1}^n z_{ij} z_{it} k_i(\boldsymbol{\gamma}) \left(\left[\frac{K + P_i^0(\boldsymbol{\delta})(1-K)}{(k_i(\boldsymbol{\gamma}) + K + P_i^0(\boldsymbol{\delta})(1-K))(1+k_i(\boldsymbol{\gamma}))} \right] \right. \\
&\quad \left. - \frac{1}{(1+k_i(\boldsymbol{\gamma}))^2} \right), \quad j, t = 0, \dots, q.
\end{aligned}$$

Für ein Summary ähnlich dem *glm*-Summary unter "R" wird ein Wald-Test zum Test der Variablen auf Signifikanz erstellt. Dazu wird die Erwartete Fisherinformation berechnet. Software und Hilfebeschreibungen sind in Anhang C angeführt.

3.3 Simulationsstudie zur Untersuchung der Asymptotik der ML-Schätzer

Im Folgenden wird wieder die Asymptotik der Maximum Likelihood - Schätzer untersucht. Um die Konvergenzgeschwindigkeit zu beurteilen, soll überprüft werden, wie sich eine Veränderung beim Stichprobenumfang n , Range der μ_i , Dispersionsparameter φ und Range der ω_i auf die Güte der Schätzer auswirkt. Dazu werden nun für Stichprobengrößen $n = 200$ und $n = 1000$ Daten verschiedener Szenarios erzeugt. Als Designmatrizen werden zweispaltige Matrizen erzeugt: Es gelte $\mu_i = \exp\{\beta_0 + x_{i1}\beta_1\}$ mit x_{i1} äquidistant steigend

auf $[-1, 1] \quad \forall i$. Damit steigen auch die μ_i mit Beobachtung i an. Darüber hinaus sei $\omega_i = \exp\{\gamma_0 + z_{i1}\gamma_1\} / (1 + \exp\{\gamma_0 + z_{i1}\gamma_1\})$ mit z_{i1} äquidistant fallend auf $[-1, 1] \quad \forall i$. Dies ergibt wiederum steigende ω_i .

Dabei werden β und γ so gewählt, daß sowohl "kleine" $\mu_i \in [2, 5]$ als auch "große" $\mu_i \in [2, 15]$ erzeugt werden. Entsprechend gibt es "kleine" $\omega_i \in [0.1, 0.25]$ und "große" $\omega_i \in [0.1, 0.5]$. Zuletzt wird $\kappa_\mu^o = 300$ gesetzt.

Parametrisierung	Range
$\beta = (1.15, 0.46)^t$	$\mu_i \in [2, 5]$
$\beta = (1.7, 1)^t$	$\mu_i \in [2, 15]$
$\gamma = (-1.65, -0.55)^t$	$\omega_i \in [0.1, 0.25]$
$\gamma = (-1.10, -1.1)^t$	$\omega_i \in [0.1, 0.5]$

Tabelle 3.1: Range von μ und ω in je zwei Settings

Auch hier wird wie in Kapitel 1 ein Wald-Test nach Formel (1.24) durchgeführt und damit überprüft, ob die mittleren ML-Schätzer in Tabelle 3.2 signifikant verschieden von Null sind. Es konnte jedoch kein Setting gefunden werden, in dem $|T| \leq 1.96$, das spricht dafür, daß die Anzahl der Simulationsläufe mit $N = 100$ ausreichend groß gewählt wurde. Wichtigen Zahlen wurde in Text und Tabelle dieselbe Farbe gegeben, um die Suche in der Tabelle zu erleichtern.

Erhöhung des Stichprobenumfangs n

Erhöht man den Stichprobenumfang von $n = 200$ auf $n = 1000$, so verbessern sich diese Werte erwartungsgemäß. So sinkt für $\mu_i \in [2, 5]$, $\varphi = 1.25$ und $\omega_i \in [0.1, 0.5]$ der MSE für φ von **0.010** ($T = 10 \frac{1.225}{0.096} = 127.6$) auf **0.002** ($T = 10 \frac{1.250}{0.043} = 290.7$). Dies stimmt überein mit der Veränderung des geschätzten Bias $\hat{b}(\theta, \hat{\theta})$. Dieser sinkt von $0.025 = 1.25 - 1.225$ auf $0 = 1.25 - 1.25$. Auch der QQ-Plot in Abb. 3.3 (8) verbessert sich.

Erhöhung der Erwartungswerte μ_i

Werden höhere Erwartungswerte erzeugt, sind die Schätzungen aller Parameter genauer. Dies ist einleuchtend, da ein geringerer Erwartungswert die Wahrscheinlichkeit von Nullen erhöht. Diese sind nur schwer ihrem Ursprung (also geringe Erwartungswerte, hohe Überdispersion oder hoher Nullenüberschuß) zuzuordnen. Dazu vergleiche man etwa für $\varphi = 3$, $\omega_i \in [0.1, 0.5]$, $n = 200$ den MSE von β_1 . Dieser sinkt bei größerem μ_i -Range um etwa 60% von **0.080** ($T = 10 \frac{0.425}{0.281} = 15.1$) auf **0.034** ($T = 10 \frac{0.975}{0.182} = 53.6$). Auch die Verzerrung $\hat{b}(\theta, \hat{\theta})$ sinkt wie erwartet, sie fällt von 0.035 auf 0.025. Eine Verbesserung ist auch von Abb. 3.3 (12) mit Abb. 3.5 (12) festzustellen.

Erhöhung des Dispersionsparameters φ

Je größer φ , desto schlechter sind auch die Schätzungen. Dies leuchtet ein, da bei zunehmender Überdispersion die Streuung der Daten wächst und damit eine Schätzung der Koeffizienten schwerer wird. Als Beispiel kann eine Erhöhung von $\varphi = 1.03$ auf $\varphi = 3$ betrachtet werden. Dort verzehnfacht sich der MSE im Setting $\mu_i \in [2, 5]$, $\omega_i \in [0.1, 0.25]$ für $n = 200$ von **0.204** ($T = 10 \frac{-0.584}{0.451} = -12.9$) auf **2.129** ($T = 10 \frac{-0.421}{1.453} = -2.9$), für

$n = 1000$ sogar von **0.041** ($T = 10^{\frac{-0.582}{0.201}} = -29.0$) auf **0.423** ($T = 10^{\frac{-0.487}{0.648}} = -7.5$). Im ersten Fall steigt die Verzerrung betragsmäßig von 0.034 auf -0.129 , im zweiten Fall von 0.032 auf -0.063 . Vergleiche dazu auch die Verschlechterung in Abb. 3.2 (15) gegenüber (5).

Erhöhung des Nullenüberschuß ω_i

Eine Erhöhung im Nullenüberschuß ω_i erhöht wieder die Streuung der Daten und verschlechtert damit die Ergebnisse. Die Änderung des MSE fällt jedoch gering aus: so steigt er für $\varphi = 1.03$, $\mu_i \in [2, 15]$, β_1 und $n = 200$ lediglich von **0.004** ($T = 10^{\frac{1.002}{0.064}} = 156.5$) auf **0.005** ($T = 10^{\frac{1.010}{0.069}} = 146.4$). Auch $\hat{b}(\theta, \hat{\theta})$ steigt betragsmäßig von -0.002 auf -0.01 . Dagegen fallen die MSE-Statistiken von γ_0 und γ_1 geringer aus. Das rührt daher, daß bei höherem Nullenüberschuß die Regressionsparameter γ_j besser schätzbar sind. Vergleicht man die MSE-Statistiken für $\varphi = 1.25$, $\mu_i \in [2, 15]$, $n = 200$ für $\omega_i \in [0.1, 0.25]$ mit $\omega_i \in [0.1, 0.5]$, so stellt man fest: Für γ_0 sinkt der MSE von **0.244** ($T = 10^{\frac{-3.350}{0.491}} = -68.2$) auf **0.178** ($T = 10^{\frac{-2.203}{0.422}} = -52.2$), das entspricht einem Minus von 27%. Der Bias fällt von 0.05 auf 0.003. Eine leichte Verbesserung stellt man auch von Abb. 3.4 (9) zu Abb. 3.5 (9) fest. Der Einfluß des Nullenüberschuß auf die Güte der übrigen Parameter ist gering.

	$\varphi = 1.03$						$\varphi = 1.25$						$\varphi = 3$						
	Parameter	Wahrer Wert	n	ML-Schätzer		MSE	Wahrer Wert	ML-Schätzer		MSE	Wahrer Wert	ML-Schätzer		MSE					
$\mu_i \in [2, 5]$	β_0	1.15	200	1.151	(0.052)	0.003	$(5 \cdot 10^{-8})$	1.15	1.145	(0.066)	0.004	$(6 \cdot 10^{-7})$	1.15	1.167	(0.202)	0.041	$(8 \cdot 10^{-6})$	s. Abb. 3.2	
			1000	1.148	(0.023)	0.001	$(6 \cdot 10^{-9})$												1.152
	β_1	0.46	200	0.466	(0.086)	0.007	$(9 \cdot 10^{-7})$	0.46	0.451	(0.109)	0.012	$(2 \cdot 10^{-6})$	0.46	0.409	(0.295)	0.090	(0.0001)		
			1000	0.463	(0.039)	0.001	$(5 \cdot 10^{-8})$												0.457
	$\omega_i \in [0.1, 0.25]$	φ	1.03	200	1.029	(0.067)	0.004	$(6 \cdot 10^{-8})$	1.25	1.247	(0.090)	0.008	$(2 \cdot 10^{-6})$	3	2.902	(0.425)	0.191		(0.001)
				1000	1.030	(0.030)	0.001	$(9 \cdot 10^{-9})$											
γ_0		-3.3	200	-3.388	(0.539)	0.298	(0.002)	-3.3	-3.443	(0.654)	0.448	(0.06)	-3.3	-3.471	(2.200)	4.868	(0.09)		
			1000	-3.332	(0.241)	0.059	$(8 \cdot 10^{-5})$												-3.316
$\mu_i \in [2, 15]$	β_0	1.15	200	-0.584	(0.451)	<i>0.204</i>	(0.002)	-0.55	-0.637	(0.523)	0.281	(0.009)	-0.55	-0.421	(1.453)	<i>2.129</i>	(0.05)		s. Abb. 3.3
			1000	-0.582	(0.201)	<i>0.041</i>	$(5 \cdot 10^{-5})$												
	β_1	0.46	200	1.140	(0.054)	0.003	$(2 \cdot 10^{-7})$	1.15	1.132	(0.069)	0.005	$(8 \cdot 10^{-7})$	1.15	1.141	(0.204)	0.042	$(4 \cdot 10^{-5})$		
			1000	1.148	(0.024)	0.001	$(5 \cdot 10^{-9})$											1.148	
	$\omega_i \in [0.1, 0.5]$	φ	1.03	200	0.453	(0.093)	0.009	$(3 \cdot 10^{-6})$	0.46	0.462	(0.117)	0.014	$(4 \cdot 10^{-6})$	0.46	0.425	(0.281)	<i>0.080</i>	(0.0003)	
				1000	0.464	(0.042)	0.002	$(6 \cdot 10^{-8})$											
γ_0		-2.2	200	1.030	(0.071)	0.005	$(2 \cdot 10^{-7})$	1.25	1.225	(0.096)	<i>0.010</i>	$(1 \cdot 10^{-6})$	3	2.944	(0.453)	0.208	(0.0004)		
			1000	1.032	(0.032)	0.001	$(1 \cdot 10^{-8})$											1.250	
$\mu_i \in [2, 15]$	β_0	1.7	200	-2.293	(0.454)	0.215	(0.002)	-2.2	-2.280	(0.535)	0.292	(0.004)	-2.2	-2.594	(1.523)	2.476	(0.4)	s. Abb. 3.4	
			1000	-2.220	(0.203)	0.042	$(3 \cdot 10^{-5})$												
	β_1	1	200	-1.156	(0.391)	0.156	(0.0009)	-1.1	-1.142	(0.445)	0.200	(0.0007)	-1.1	-1.331	(1.015)	1.083	(0.08)		
			1000	-1.113	(0.175)	0.031	$(2 \cdot 10^{-5})$												-1.081
	$\omega_i \in [0.1, 0.25]$	φ	1.03	200	-1.694	(0.038)	0.002	$(1 \cdot 10^{-7})$	1.7	1.704	(0.047)	0.002	$(6 \cdot 10^{-8})$	1.7	1.705	(0.125)	0.016		$(5 \cdot 10^{-6})$
				1000	1.700	(0.017)	0.000	$(2 \cdot 10^{-9})$											
γ_0		-3.3	200	1.002	(0.064)	<i>0.004</i>	$(3 \cdot 10^{-7})$	1	0.989	(0.079)	0.006	$(4 \cdot 10^{-7})$	1	0.994	(0.178)	0.032	$(1 \cdot 10^{-5})$		
			1000	0.999	(0.028)	0.001	$(1 \cdot 10^{-8})$												0.999
$\mu_i \in [2, 15]$	β_0	1.7	200	1.037	(0.060)	0.004	$(2 \cdot 10^{-7})$	1.25	1.247	(0.078)	0.006	$(2 \cdot 10^{-6})$	3	3.002	(0.319)	0.102	(0.0002)		s. Abb. 3.5
			1000	1.030	(0.027)	0.001	$(2 \cdot 10^{-9})$												
	γ_0	-3.3	200	-3.437	(0.450)	0.221	(0.002)	-3.3	-3.350	(0.491)	<i>0.244</i>	(0.002)	-3.3	-3.549	(1.111)	1.296	(0.08)		
			1000	-3.272	(0.201)	0.041	$(3 \cdot 10^{-5})$											-3.341	
	$\omega_i \in [0.1, 0.25]$	β_1	1	200	-0.664	(0.402)	0.174	(0.0007)	-0.55	-0.570	(0.438)	0.192	(0.002)	-0.55	-0.688	(0.842)	0.728	(0.02)	
				1000	-0.539	(0.179)	0.032	$(1 \cdot 10^{-5})$											
φ		1.03	200	1.689	(0.040)	0.002	$(7 \cdot 10^{-8})$	1.7	1.699	(0.050)	0.002	$(1 \cdot 10^{-7})$	1.7	1.698	(0.128)	0.016	$(5 \cdot 10^{-6})$		
			1000	1.700	(0.018)	0.000	$(3 \cdot 10^{-9})$											1.696	
$\mu_i \in [2, 15]$	β_1	1	200	1.010	(0.069)	<i>0.005</i>	$(4 \cdot 10^{-7})$	1	0.999	(0.085)	0.007	$(1 \cdot 10^{-6})$	1	0.975	(0.182)	<i>0.034</i>	$(3 \cdot 10^{-5})$		
			1000	0.998	(0.031)	0.001	$(3 \cdot 10^{-8})$											1.004	
	γ_0	-2.2	200	1.037	(0.065)	0.004	$(1 \cdot 10^{-7})$	1.25	1.244	(0.084)	0.007	$(1 \cdot 10^{-6})$	3	2.899	(0.347)	0.130	(0.0003)		
			1000	1.034	(0.029)	0.001	$(1 \cdot 10^{-8})$											1.251	(0.038)
	$\omega_i \in [0.1, 0.25]$	γ_1	-1.1	200	-2.268	(0.391)	0.158	(0.0008)	-2.2	-2.203	(0.422)	<i>0.178</i>	(0.0008)	-2.2	-2.164	(0.836)	0.701	(0.008)	
				1000	-2.211	(0.175)	0.031	$(1 \cdot 10^{-5})$											-2.243
γ_1		-1.1	200	-1.095	(0.354)	0.125	(0.0004)	-1.1	-1.125	(0.381)	0.146	(0.0006)	-1.1	-1.015	(0.664)	0.448	(0.003)		
			1000	-1.112	(0.158)	0.025	$(1 \cdot 10^{-5})$											-1.142	(0.170)

Tabelle 3.2: Mittlere MLE-Schätzungen und MSE-Statistiken für $ZMGP(\mu_i, \varphi, \omega_i)$ in mehreren Parameter-Settings basierend auf $N = 100$ Datensätzen; geschätzte Standardabweichungen der Statistiken in Klammern.

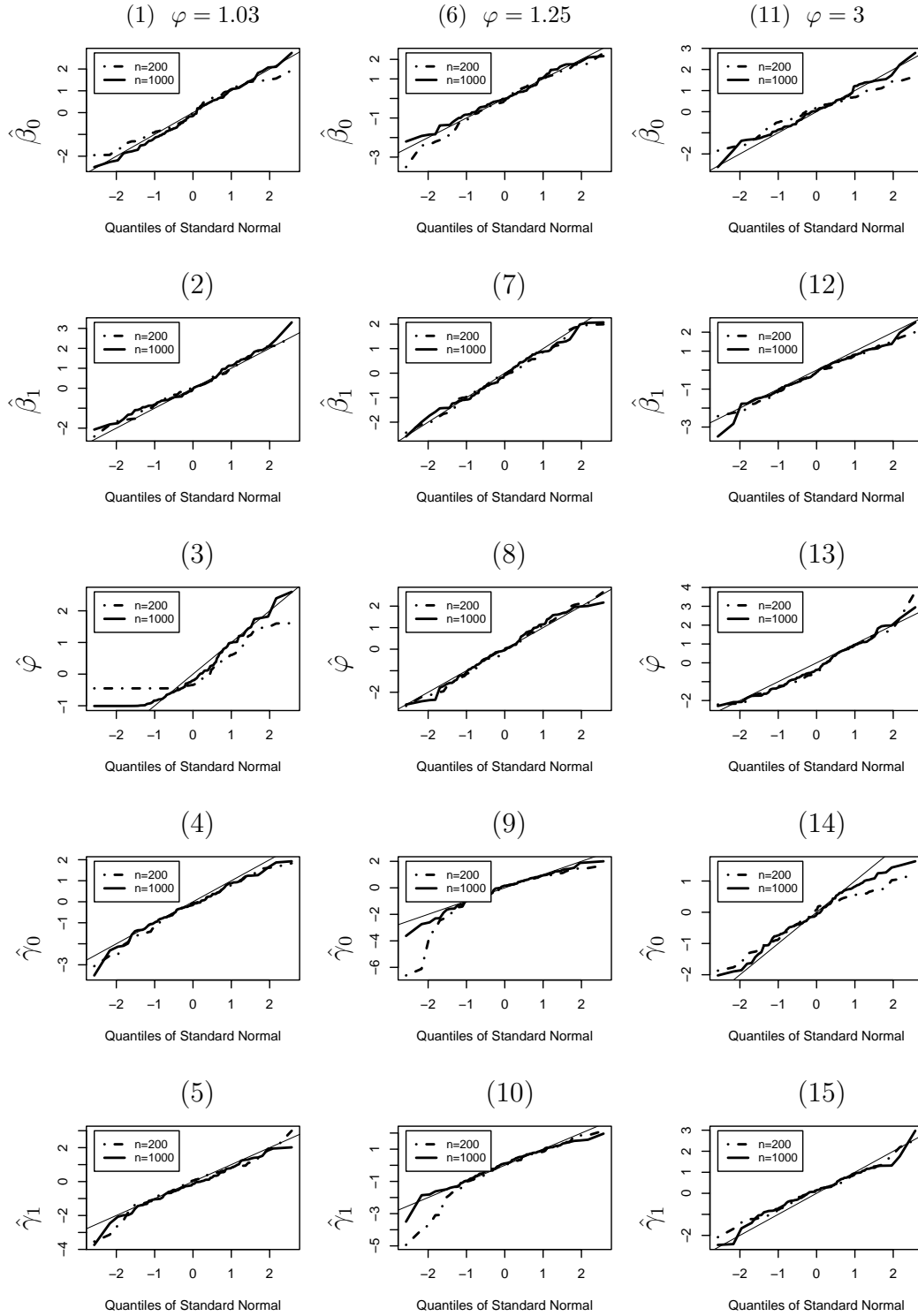


Abbildung 3.2: QQ-Plots zentrierter und normierter ML Schätzer basierend auf $N = 100$ Datensätzen im $ZMGP(\mu_i, \varphi, \omega_i)$ -Modell mit $\mu_i \in [2, 5]$ und $\omega_i \in [0.1, 0.25]$.

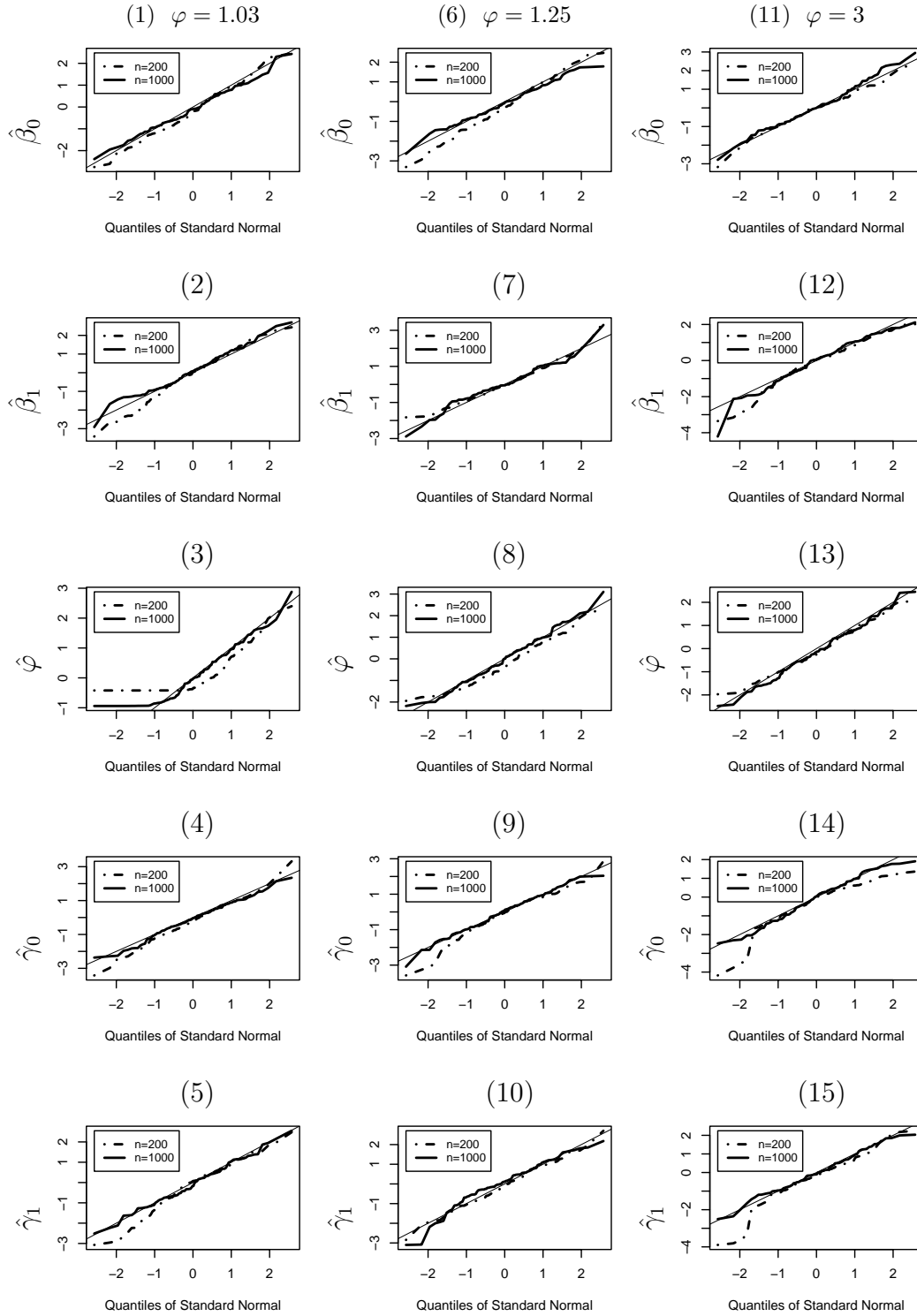


Abbildung 3.3: QQ-Plots zentrierter und normierter ML Schätzer basierend auf $N = 100$ Datensätzen im $ZMGP(\mu_i, \varphi, \omega_i)$ -Modell mit $\mu_i \in [2, 5]$ und $\omega_i \in [0.1, 0.5]$.

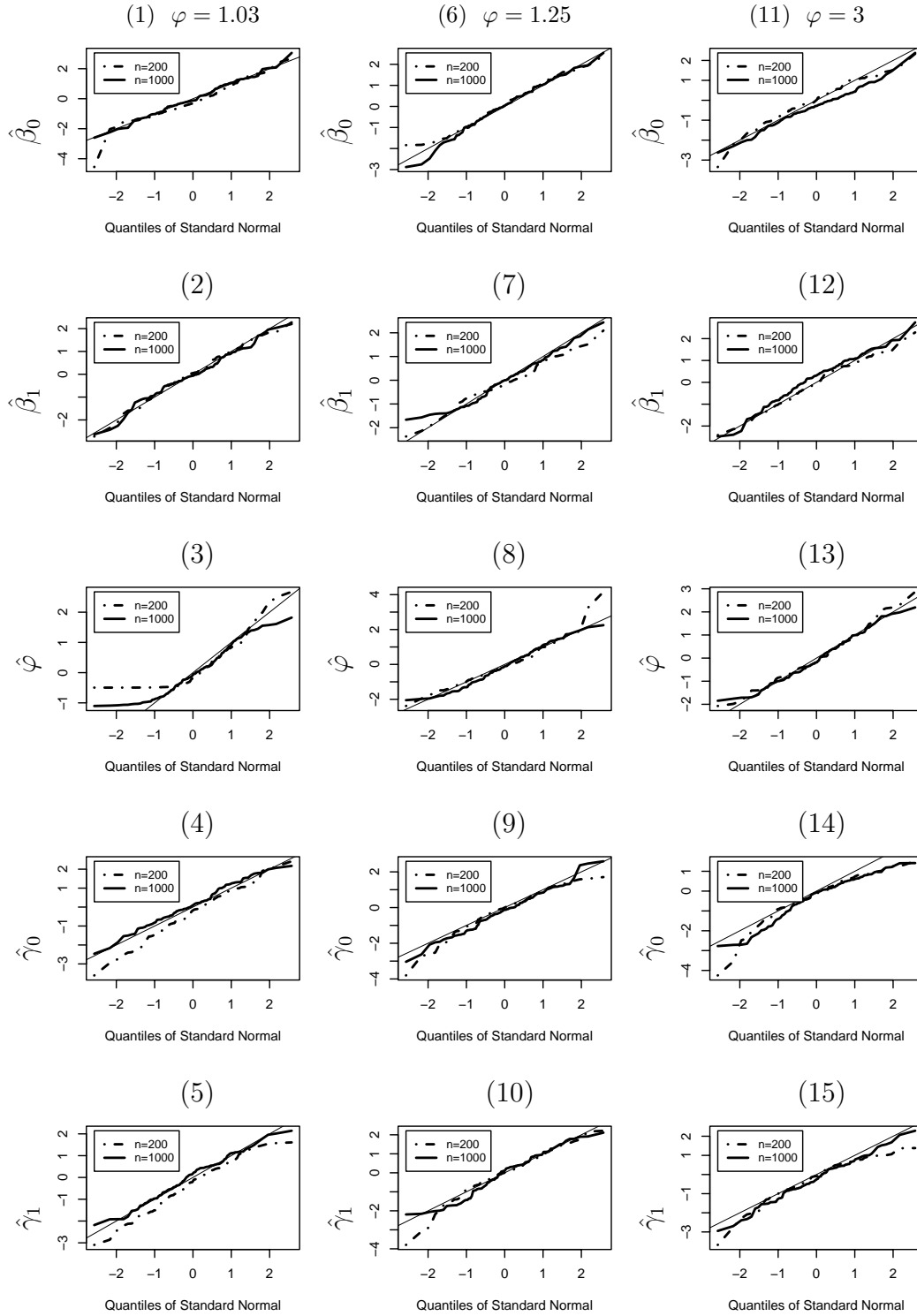


Abbildung 3.4: QQ-Plots zentrierter und normierter ML Schätzer basierend auf $N = 100$ Datensätzen im $ZMGP(\mu_i, \varphi, \omega_i)$ -Modell mit $\mu_i \in [2, 15]$ und $\omega_i \in [0.1, 0.25]$.

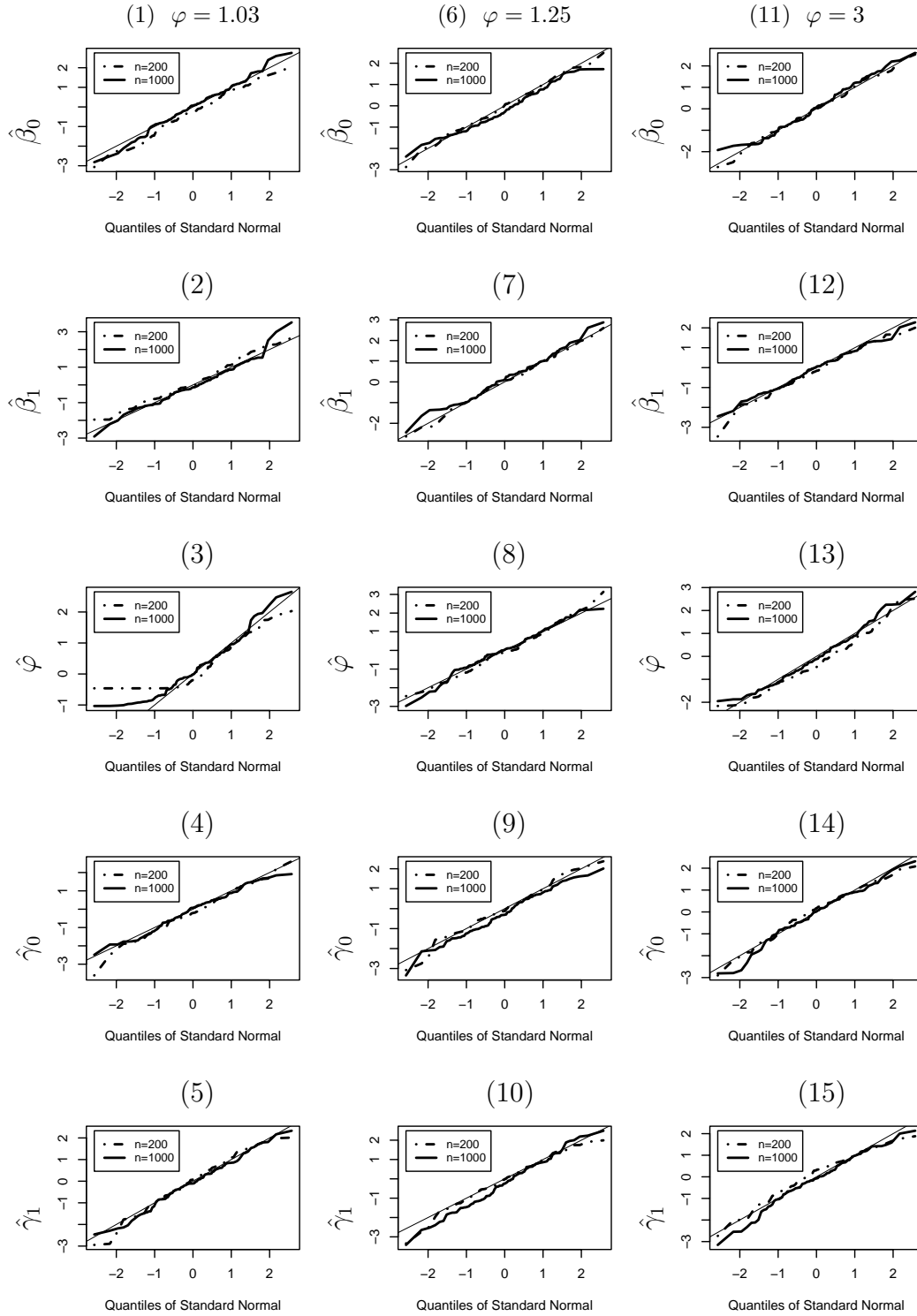


Abbildung 3.5: QQ-Plots zentrierter und normierter ML Schätzer basierend auf $N = 100$ Datensätzen im $ZMGP(\mu_i, \varphi, \omega_i)$ -Modell mit $\mu_i \in [2, 15]$ und $\omega_i \in [0.1, 0.5]$.

Kapitel 4

Anwendung der Modelle zur Patent-Modellierung

In diesem Kapitel sollen die entwickelten Modelle anhand eines Patent-Datensatzes untersucht werden. In Abschnitt 4.1 wird dessen Inhalt beschrieben und eine Fragestellung formuliert. In den darauffolgenden Abschnitten wird eine explorative Datenanalyse für die Modellierung der Erwartungswerte, der Überdispersion und des Nullenüberschuß vorgenommen. Dies stellt sich als begründete Auswahl von Kovariablen und mögliche Transformationen dieser Variablen dar. Transformationen werden jedoch nur für die Modellierung des Erwartungswertes vorgenommen. Abschnitt 4.3 zeigt auf, welche Vergleichskriterien angewandt werden sollen, wenn im nächsten Abschnitt die ausgewählten Kovariablen in verschiedene Modelle eingesetzt werden. Als Abschluß wird eine Interpretation des Modells mit höchster Anpassungsgüte vorgenommen.

4.1 Datenbeschreibung

Der Datensatz enthält Erhebungen des Europäischen Patentamts. Er wurde von Wagner [38] von *INNO-tec* (Institut für Innovationsforschung, Technologiemanagement und Entrepreneurship an der LMU München) untersucht und von ihm um Unternehmensinformationen ergänzt. Wagner beschreibt, wieviele Patente einzelne Unternehmen beim Europäischen Patentamt eingereicht haben und insbesondere, wieviele der Vorgänge davon *über externe Patentanwälte oder alternativ über interne Patentabteilungen* abgewickelt wurden. Wagner paßt anstelle eines ZIGP-Modelles ein Negativ-Binomialmodell für zeitabhängige Daten an. Betrachtet wurden 107 europäische Firmen ($i = 1, \dots, 107$) über einen Zeitraum von acht Jahren ($t = 1, \dots, 8$ entspricht 1993 - 2000). Es sei bereits vorausgeschickt, daß es dazu eine Kovariable namens "time" gibt, die das Jahr angibt. Der Datensatz umfaßt 37 Variablen, davon sind jedoch viele Transformationen anderer Kovariablen oder nur Textfelder. Die Analyse beschränkt sich daher auf die Variablen, die auch in Wagners Regressionsmodelle eingingen. Diese werden im folgenden erklärt.

Y := out.int (Zählraten)

Der Response "out.int" gibt die Gesamtzahl der Patente an, die im Verlauf eines Jahres

von *externen Patentanwälten* beim Europäischen Patentamt eingereicht wurden. Es handelt sich um Zähldaten, die eine Poissonregression nahelegen. Realisationen von "out.int" liegen im Bereich $[0; 953]$, wobei die Null 129 mal vorkommt.

$E := \text{pat.cum.year}$ (Zähldaten)

Die Zahl der Patentanmeldungen im Jahreszeitraum insgesamt wird "pat.cum.year" genannt - und zwar unabhängig davon, wie sie angemeldet wurden. Die Anzahl bewegt sich in $[1; 2053]$, jedes Unternehmen hat also in jedem Jahr mindestens ein Patent angemeldet! Die Variable hat direkten Einfluß auf die Anzahl der ausgelagerten Patentverfahren: je mehr Patente ein Unternehmen insgesamt pro Jahr einreicht, desto höher ist auch die *mögliche* Anzahl an ausgelagerten Patenten. Daher fließt sie als *Exposure* in die Modelle ein und wird in Zukunft als Vektor **E** bezeichnet.

std.5.mean (metrisch)

Diese Kovariable ist ebenfalls metrisch. Sie gibt den empirischen Variationskoeffizienten der Bewerbungen eines Unternehmens an, wobei der Zeitraum der vergangenen fünf Jahre betrachtet wird. Der Variationskoeffizient wird der Varianz vorgezogen, da eine Streuung bei Unternehmen mit kleineren Anmeldungszahlen in absoluten Zahlen einen größeren Unterschied hervorruft. Er dient damit der besseren Vergleichbarkeit. Der Wert wird berechnet als

$$\text{std.5.mean}_{it} := \frac{\sqrt{\frac{1}{5} \sum_{j=t-5}^{t-1} (PA_{ij} - \frac{1}{5} \sum_{j=t-5}^{t-1} PA_{ij})^2}}{\frac{1}{5} \sum_{j=t-5}^{t-1} PA_{ij}}. \quad (4.1)$$

Dabei ist PA_{it} die Anzahl der Patentanmeldungen des Unternehmens i im Jahr t . Die Werte liegen in $[0; 2]$. Wagner [38, S. 15] definiert den Variationskoeffizient jedoch geringfügig anders: im Zähler fehlt unter der Wurzel $\frac{1}{5}$ und der Betrachtungszeitraum ist bei ihm um 1 Jahr nach hinten verschoben. Durch Nachrechnen ergibt sich jedoch, daß die im Datensatz enthaltenen Werte gemäß Formel (4.1) berechnet wurden.

breadth (metrisch)

Ebenfalls metrisch ist "breadth", das mit

$$\text{breadth}_{it} = 1 - \sum_{k=1}^{30} s_{ikt}^2 \in [0; 1] \quad (4.2)$$

angegeben wird. Dabei ist s_{ikt} der prozentuale Anteil der Bewerbungen von Firma i in Jahr t , die dem wissenschaftlichen Gebiet k ($k \in [1; 30]$) zuzuordnen ist. Der "breadth" ist demnach ein Maß dafür, wie breit gestreut die Bewerbungen eines Unternehmens in technologischer Hinsicht sind. Ein hoher Wert bedeutet, daß sich Firma i im Jahr t in vielen Wissenschaftsgebieten engagiert. Die Realisationen liegen in $[0.067; 0.923]$.

emp (metrisch)

Die Zähldaten "emp" sind die Anzahl der Mitarbeiter (MA) eines Unternehmens. Es gilt $\text{emp}_{it} \in [0; 466938]$. Als Einheit wird 1 MA verwendet.

r.d.pat (metrisch)

Die metrische Kovariable "r.d.pat" gibt für das Unternehmen i das Verhältnis aus Ausgaben für Forschung und Entwicklung ("F+E") in Mio. Euro zur Anzahl angemeldeter

Patente an. Es gilt $r.d.pat_{it} \in [0; 465.3]$. Die Kovariable beschreibt damit die durchschnittlichen Entwicklungskosten, die einer Patentanmeldung vorausgingen.

r.d.intens (metrisch)

Ebenfalls metrisch ist "r.d.intens", es setzt die Ausgaben für Forschung und Entwicklung in 1000 Euro in Relation zur Anzahl der Mitarbeiter und ist somit ein Maß für die Forschungsintensität eines Unternehmens. Die Werte liegen in $r.d.intens_{it} \in [0; 215.7]$.

r.d.miss (kategorial)

Eine kategoriale Variable mit Ausprägungen 0 und 1 liegt mit "r.d.miss" vor. Diese Dummyvariable gibt an, ob Angaben zu Forschung und Entwicklung des Unternehmens fehlen (= 1) oder nicht (= 0). Viele Europäische Unternehmen wurden erst ab 2001 gesetzlich verpflichtet, diese Angaben offenzulegen, daher können für alle betrachteten Jahre diese Daten fehlen.

branche (kategorial)

Die Variable "branche" enthält die Werte $\{1, 2, 3, 4, 5, 7\}$, die Branchen-IDs des Unternehmens angeben.

dbranch1, ..., dbranch5, dbranch7 (Dummyvariablen)

Die Branche aus "branche" wird mit den Dummyvariablen "dbranch1", ..., "dbranch5", "dbranch7" angezeigt. Sie geben an, ob das Unternehmen Branche 1, ..., 5 oder 7 angehört (eine Branche 6 existiert nicht). Im einzelnen stehen "dbranch1" für "Chem./ Pharma", "dbranch2" für "Electr./ Telco", "dbranch3" für "Engineering", "dbranch4" für "Cars + Supplier", "dbranch5" ist "Medtech/ Biotech" und "dbranch7" ist "Other" und entspricht den Branchen 6 und 9 in "branche".

Um transparenter mit den Branchen arbeiten zu können, werden die Branchen umbenannt:

```
chem.pha <- dbranch1
elec.tel <- dbranch2
engineer <- dbranch3
car.supp <- dbranch4
med.biot <- dbranch5
other.br <- dbranch7
```

time (kategorial)

Zuletzt gibt "time" das Jahr an. Dabei steht 1 für 1993, 8 für 2000.

Da sich bei Betrachtung des gesamten Datensatzes ($t = 1993, \dots, 2000$) Abhängigkeiten ergeben (die Daten stellen dann für jedes Unternehmen eine Zeitreihe dar), der Response aber unabhängig sein muß, bietet sich eine Betrachtung gruppierter jährlicher Daten an. Da aber mit den betrachteten Regressionsmodellen kein geeignetes Modell vorliegt, um zeitliche Abhängigkeit abzubilden, wird dennoch angenommen, es handele sich um unabhängige Beobachtungen. Da die Beobachtungszeitpunkte jährlich und nicht kürzer sind, kann man davon ausgehen, daß die zeitliche Abhängigkeit gering ist. Die Zeit wird jedoch als Kovariable in die Untersuchung einbezogen. Anstelle der Schreibweise Y_{it} und

E_{it} wird daher von nun an Y_i und E_i mit $i = 1, \dots, 856$ benutzt. Dazu werden die Daten wie folgt aneinandergereiht: $\mathbf{Y} := (Y_{1,1}, Y_{1,2}, \dots, Y_{1,8}, Y_{2,1}, \dots, Y_{2,8}, Y_{3,1}, \dots, Y_{107,8})^t$.

Bemerkung: Um die zeitliche Abhängigkeit in den Daten abzubilden, schlagen Hausmann et al. [17] vor, einen unternehmensspezifischen Parameter hinzuzunehmen:

$$\tilde{\mu}_{it} = \mu_{it}\tilde{\alpha}_i.$$

Dabei ist $\tilde{\alpha}_i$ ein zufälliger firmenspezifischer Effekt, der die Korrelation von $\tilde{\mu}_{it}$ und $\tilde{\mu}_{it'}$, $t \neq t'$ abbildet. Dieser Ansatz führt jedoch zu weitaus komplexeren Modellen und wird hier außer Acht gelassen.

Die genannten Einflüsse und den Response kann man nun graphisch näher betrachten. Der Response wird dabei mit Hilfe eines Histogramms in Abb. 4.1 (1), Exposure und Kovariablen mit Boxplots dargestellt (s. Abb. 4.1 (3)-(10)). Beim Response wurden dabei zur besseren graphischen Darstellung Ausreißer mit $Y_i \geq 300$ herausgenommen: dann sei \mathbf{Y}_2 der Vektor, der aus der Menge $\{Y_i : Y_i < 300\}$ zusammengesetzt sei, wobei 26 Ausreißer $Y_i \geq 300$ weggeworfen werden. Damit vermindert sich das Maximum von $\max_i \{Y_i\} = 953$ auf $\max_i \{Y_{2,i}\} = 261$. Man möchte damit feststellen, ob eine Poissonverteilung für die Anpassung eines Modells geeignet ist. Das Histogramm in Abb. 4.1 (1) für \mathbf{Y}_2 zeigt, daß kleine Werte deutlich überwiegen. Wie man weiß, sind allein 129 Ausprägungen. Für die Untersuchung der durchschnittlichen Outsourcing-Quoten werden die Quotienten $\frac{Y_i}{E_i}$ definiert. Der Vektor $\mathbf{Q} := (Q_1, \dots, Q_{856})^t$ ist in Abb. 4.1 (2) dargestellt. Man sieht, daß jeweils etwa ein Viertel der Beobachtungen weniger als 10% seiner Patentanmeldungen auslagert, etwa ein weiteres Viertel mehr als 90% auslagert. Geringe Auslagerungsraten zwischen 10 und 40% überwiegen deutlich gegenüber höheren Raten.

Die verbleibenden Variablen in Abb. 4.1 (3)-(10) werden mit Boxplots untersucht. Diese Plots zeigen graphisch, wo Minimum, Maximum und die drei Quantile eines Datenvektors liegen. Sie zeigen mit Ausnahme von "breadth" sehr große Ausreißer nach oben. Besonders stark ist die bei der Exposure "pat.cum.year" in Abb. 4.1 (3) zu sehen. Dies heißt jedoch, daß die meisten Unternehmen etwa gleich viele Patente in der Größenordnung von unter 100 eingereicht haben. Die Streuung der Patentanmeldungen "std.5.mean" (s. Abb. 4.1 (4)) und die Mitarbeiterzahl "emp" (s. Abb. 4.1 (6)) weisen ebenfalls Ausreißer nach oben auf, die jedoch nicht so extrem wie bei "pat.cum.year" ausfallen. Die Streuung der wissenschaftlichen Gebiete "breadth" in Abb. 4.1 (5) ist gleichmäßiger verteilt. Der Hauptanteil der Kennzahl liegt zwischen 0.5 und 0.8. Bei "breadth" sind nach Formel (4.2) rechnerisch jedoch nur Werte im Intervall $[0, 1)$ möglich. Bei "r.d.pat" (s. Abb. 4.1 (7)) und "r.d.intens" (s. Abb. 4.1 (9)) überwiegen sehr kleine Werte. Beide Variablen haben etwa 30% Nullen, die hauptsächlich fehlende Angaben darstellen ("r.d.miss" hat bei diesen Beobachtungen den Wert 1, der fehlende Angaben anzeigt). Abb. 4.1 (8) und (9) zeigen Boxplots von $r.d.pat_i \in (0, 40]$ und $r.d.intens_i \in (0, 70]$.

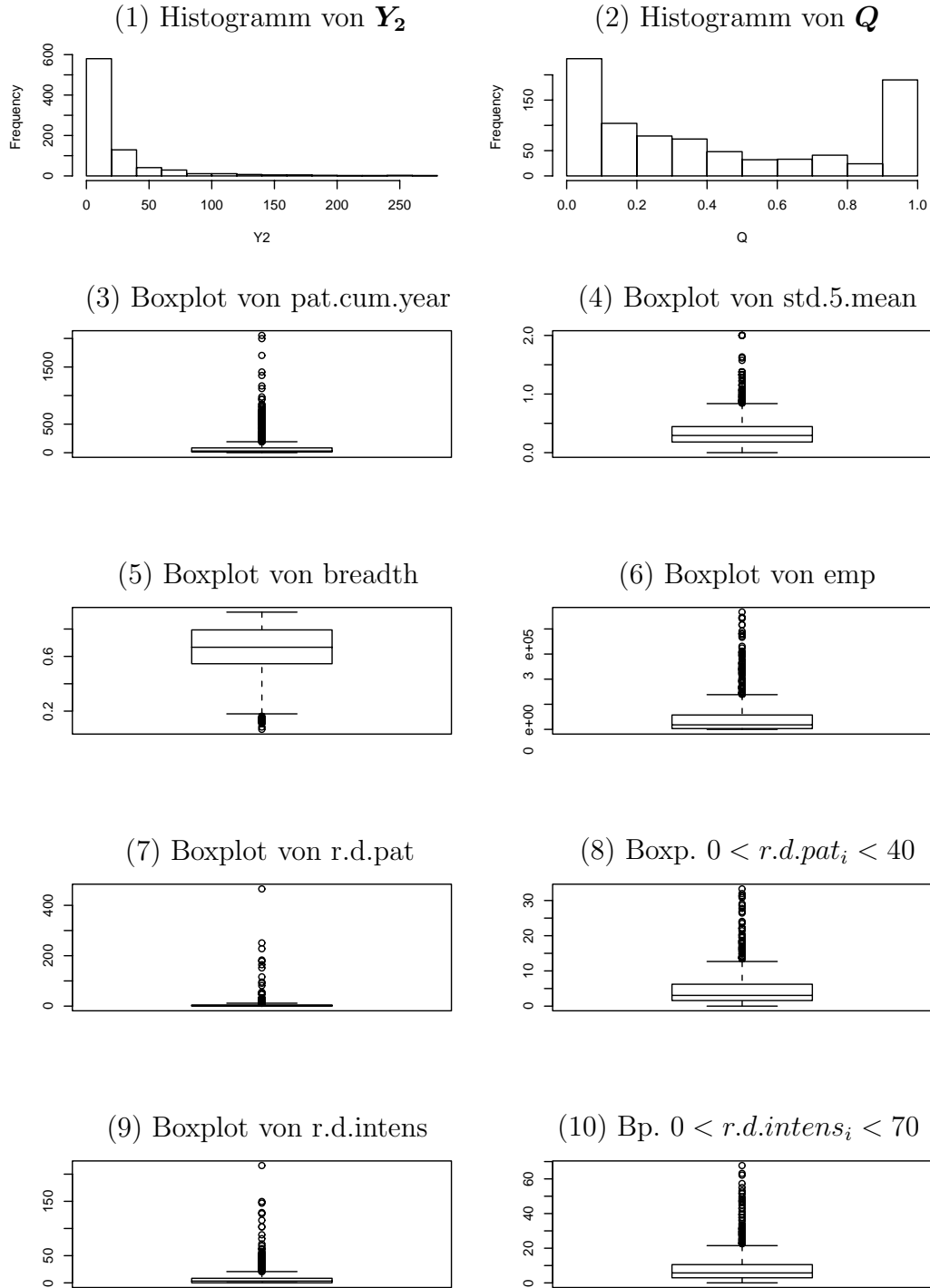


Abbildung 4.1: (1) – Histogramm des Response; (2) – Histogramm des Quotientenvektors Q , $Q_i := \frac{Y_i}{E_i}$; (3)-(10) – Boxplots der metrischen Kovariablen, $i = 1, \dots, 856$.

Variable	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Y	0.00	2.00	9.00	36.07	27.00	953.00
Q	0.00	0.0831	0.3333	0.4305	0.8014	1.00
<i>pat.cum.year</i>	1.00	12.00	30.00	98.21	84.00	2053.00
<i>std.5.mean</i>	0.00	0.1818	0.2925	0.3610	0.4442	2.00
<i>breadth</i>	0.0666	0.5464	0.6667	0.6423	0.7934	0.9233
<i>emp</i>	0	3151	17970	50000	57330	466900
<i>r.d.pat</i>	0.00	0.000	1.823	6.110	4.646	465.3
<i>r.d.intens</i>	0.00	0.000	3.288	7.892	8.360	215.7

Tabelle 4.1: Quantile und Mittelwerte zur Untersuchung metrischer Variablen

Die kategoriale Variable "r.d.miss" wird tabellarisch dargestellt.

	#	%
Angaben F+E ¹ vorhanden: $\{i : r.d.miss_i = 0\}$	602	70.3 %
Angaben F+E fehlen: $\{i : r.d.miss_i = 1\}$	254	29.7 %

Daß der Median in Tabelle 4.1 bei allen Variablen mit Ausnahme von "breadth" kleiner ist als der Mittelwert - die Häufigkeitsverteilung ist rechtsschief - ist ein weiterer Hinweis darauf, daß kleine Werte überwiegen und Ausreißer nach oben bestehen.

4.2 Explorative Datenanalyse

4.2.1 Kovariablen für die Modellierung des Erwartungswertes

In diesem Abschnitt sollen nun Kovariablen für die Modellierung des Erwartungswertes ausgewählt und mögliche Transformationen untersucht werden. Es wird angenommen, die Daten folgten einer Poissonverteilung. Dann gilt, daß $E(Y_i) = E_i \cdot e^{\mathbf{x}_i^t \boldsymbol{\beta}} \Leftrightarrow \log(\mu_i) = \log(E_i) + \mathbf{x}_i^t \boldsymbol{\beta}$ für positive E_i . Diese Gleichung wird durch $\log(Y_i) = \log(E_i) + \mathbf{x}_i^t \boldsymbol{\beta}$ geschätzt, daher sollte sich

$$\log(Y_i) - \log(E_i) \text{ linear in } \mathbf{x}_i$$

verhalten. Da es Beobachtungen mit $Y_i = 0$ gibt, wird mit $\tilde{Y}_i := Y_i + c$, $c = 0.001$ gearbeitet. Es gilt $E_i > 0$ für $i = 1, \dots, 856$.

In Abb. 4.2 bis Abb. 4.6 wird dies nun untersucht. Die ersten drei Graphiken zeigen jeweils (1) den gesamten Datensatz, (2) den Ausschnitt mit abgeschnittenen Werten der Kovariablen und (3) schließlich noch für Werte $Y_i > 0$; letzterer Datenausschnitt wird bei den Transformationen näher betrachtet. Zu den ersten drei Graphen ist jeweils eine mit der "R"-Funktion *lowess()* erstellte Glättungslinie angezeigt für (1) alle Daten in blau gestrichelt, (2) Daten $Y_i = 0$ in lila und (3) Daten $Y_i > 0$ in rot. Danach wird in

(4) diejenige Transformation von (3) gezeigt, die einen möglichst linearen Zusammenhang darstellt und mit der weitergearbeitet wird.

Mit *lowess()* kann man für Daten in Scatterplots eine Glättungslinie erzeugen. Die Prozedur arbeitet mit lokal gewichteter Regression und wurde von Bennett [1] beschrieben. Dazu wird ein Polynom aufgestellt, daß mit Hilfe von gewichteten kleinsten Fehlerquadraten die Daten angepasst wird. Die Gewichte werden in den darauffolgenden Iterationen sukzessive je nach der Größe der Residuen aus dem vorherigen Schritt angepaßt.

In Abb. 4.2 werden mögliche Transformationen von "std.5.mean" untersucht. Ein Plot von $\ln(Y_i + 0.001) - \ln(E_i)$ gegen $x_{i1} = \text{std.5.mean}_i$ kann in Abb. 4.2 (1) gefunden werden. In Abb. 4.2 (2) werden Ausreißer mit $x_{i1} > 1.1$ abgeschnitten, das sind 22 Beobachtungen. Die Nullen in \mathbf{Y} wurden in Abb. 4.2 (3) entfernt. Die verbleibenden Werte haben die Gestalt der Logarithmus-Funktion. Folgende Transformationen wurden versucht: $\log(x_{i1})$, $-\frac{1}{x_{i1}}$, $x_{i1}^{2/3}$ und $x_{i1}^{1/5}$. Den besten linearen Zusammenhang scheint aber $\sqrt{x_{i1}}$ zu haben, diese ist in Abb. 4.2 (4) dargestellt. Für die Transformationen wurde aufgrund der Nullen in "std.5.mean" ein Shift von 0.001 vorgenommen.

Für "breadth" wurden 23 Beobachtungen entfernt: alle Werte $x_{i2} = \text{breadth}_i < 0.2$ wurden aus Abb. 4.3 (1) entfernt. Das Ergebnis ist in Abb. 4.3 (2) dargestellt. Abb. 4.3 (3) zeigt bereits einen guten linearen Zusammenhang der Kovariablen mit $\ln(Y_i + 0.001) - \ln(E_i)$. Doch auch hier wurden Transformationen versucht, im einzelnen waren dies aufgrund der leichten quadratischen Form der *lowess*-Linie in Abb. 4.3 (3) $-x_{i2}^2$, $-x_{i2}^3$ und $-x_{i2}^4$. Dennoch geht "breadth" ohne Transformation in die Regression ein.

Bei "emp" wurden 104 Beobachtungen aus Abb. 4.4 (1) entfernt, so enthält Abb. 4.4 (2) nur noch Werte mit $x_{i3} = \text{emp}_i \leq 120000$. Die Punkte in Abb. 4.4 (3) zeigen Log-Gestalt, daher werden Transformationen $\log(x_{i3})$, $-\frac{1}{x_{i3}}$, $-x_{i3}^{2/3}$ und $-x_{i3}^{1/5}$ versucht. Den besten linearen Eindruck macht jedoch die Transformation $\sqrt{x_{i3}}$ in Abb. 4.4 (4).

Für die Variable "r.d.pat" zeigt Abb. 4.5 (1) sehr große Ausreißer. Um das Zentrum der meisten Datenpunkte genauer betrachten zu können, werden alle Werte ab $x_{i4} = \text{r.d.pat}_i > 20$ abgeschnitten (s. Abb. 4.5 (2)). Abb. 4.5 (3) zeigt erneut Log-Gestalt. Als mögliche Transformationen werden $\log(x_{i4})$, $x_{i4}^{2/3}$, $\sqrt{x_{i4}}$ und $x_{i4}^{1/5}$ in Betracht gezogen. Die Entscheidung fällt jedoch aufgrund der guten linearen Form der Glättungslinie in Abb. 4.5 (4) auf $-\frac{1}{x_{i4}}$. Auch hier wird wegen der Nullen in "r.d.pat" ein Shift der Daten von 0.001 vorgenommen.

Zuletzt wird "r.d.intens" untersucht. Aus Abb. 4.6 (1) werden Ausreißer mit $x_{i5} = \text{r.d.intens}_i > 20$ entfernt. Auch hier zeigt der dritte Scatterplot in Abb. 4.6 (3) eine mögliche Log-Gestalt. Man versucht es mit Transformationen $-\log(x_{i5})$, $\frac{1}{x_{i5}}$, $-x_{i5}^{0.85}$, $-x_{i5}^{2/3}$ und $-\sqrt{x_{i5}}$. In Abb. 4.6 (4) zeigt sich aber, daß eine kubische Transformation der Form $-x_{i5}^3$ die besten Ergebnisse zeigt. Zur Erzeugung der Potenzen von "r.d.intens" bis zum dritten Grad wird die "R"-Funktion *poly()* verwendet. Diese erzeugt mit dem Gram-Schmidtschen Orthogonalisierungsverfahren orthogonale Vektoren bis zum gewünschten Grad. Damit kann untersucht werden, ob lineare, quadratische und kubische Einflüsse von "r.d.intens" auf den Erwartungswert bestehen.

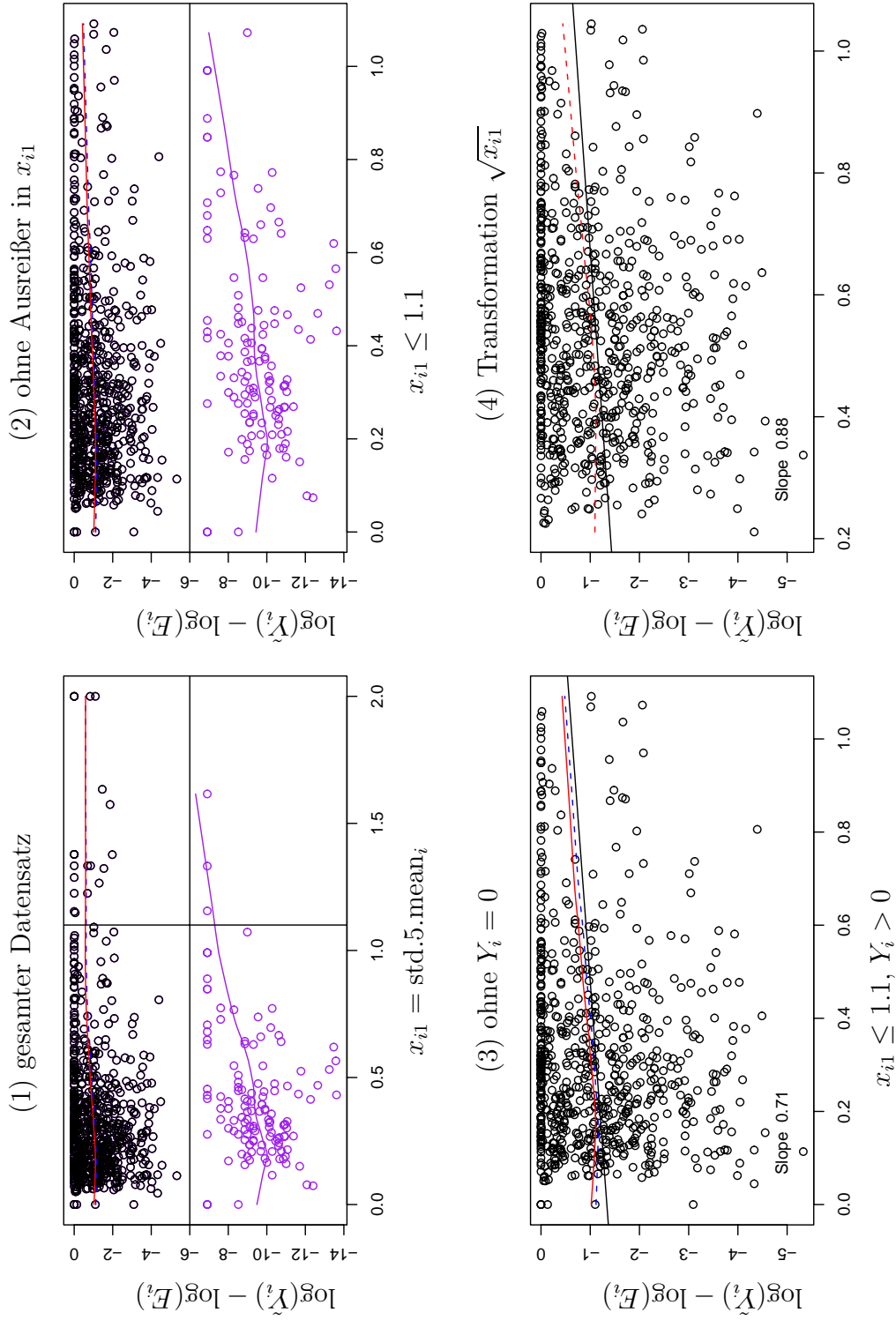


Abbildung 4.2: Plots $\ln(Y_i + 0.001) - \ln(E_i)$ gegen $x_{i1} = \text{std.5.mean}_i$; mit *lowess()* geglättete Daten für alle Daten (blau gestrichelt), Daten $Y_i = 0$ (lila) und $Y_i > 0$ (rot); Transformationen für Datenausschnitte mit $Y_i = 0$, Regressionslinien in schwarz

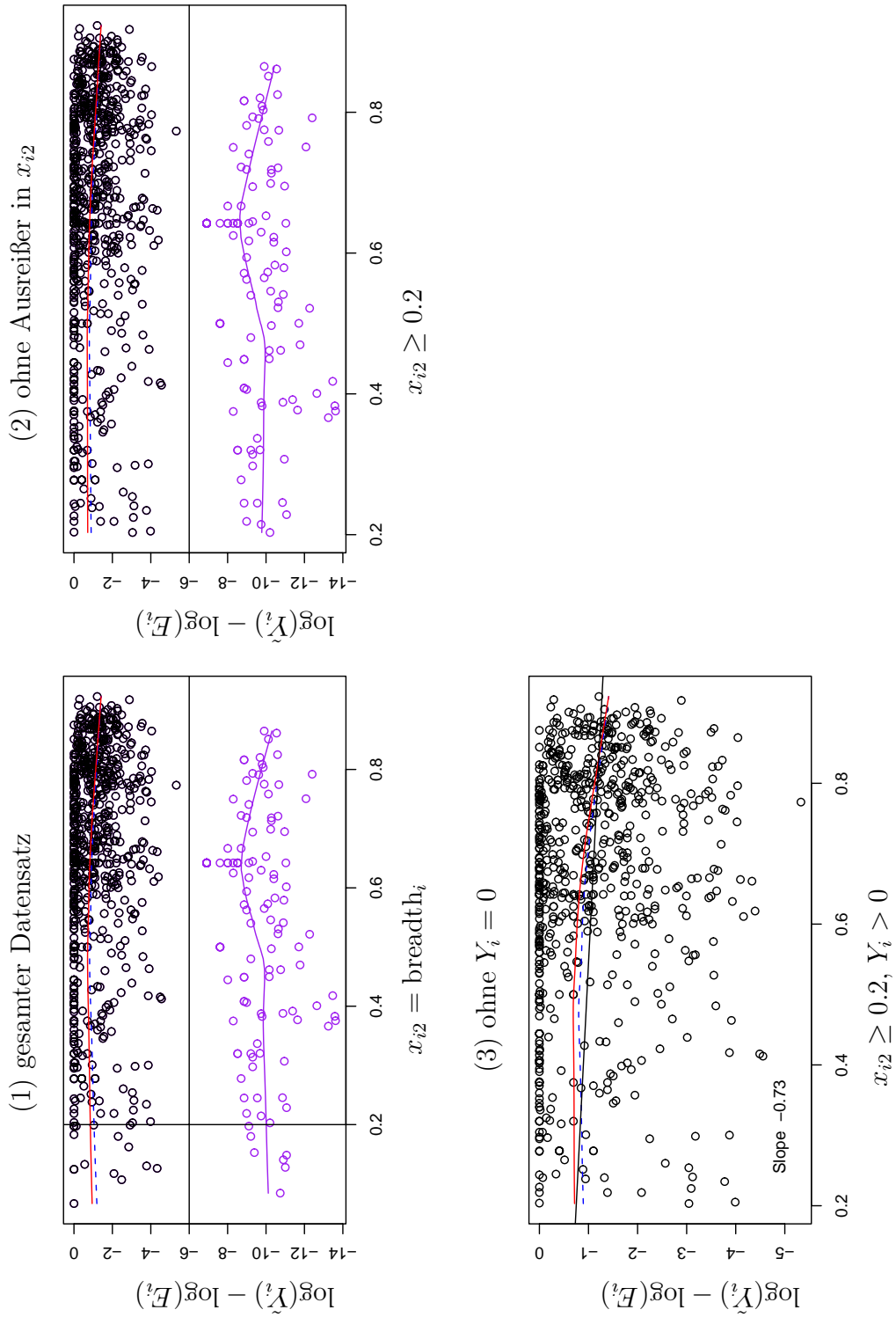


Abbildung 4.3: Plots $\ln(Y_i + 0.001) - \ln(E_i)$ gegen $x_{i2} = \text{breadth}_i$; mit *lowess()* geglättete Daten für alle Daten (blau gestrichelt), Daten $Y_i = 0$ (lila) und $Y_i > 0$ (rot); Transformationen für Datenausschnitte mit $Y_i = 0$, Regressionslinien in schwarz

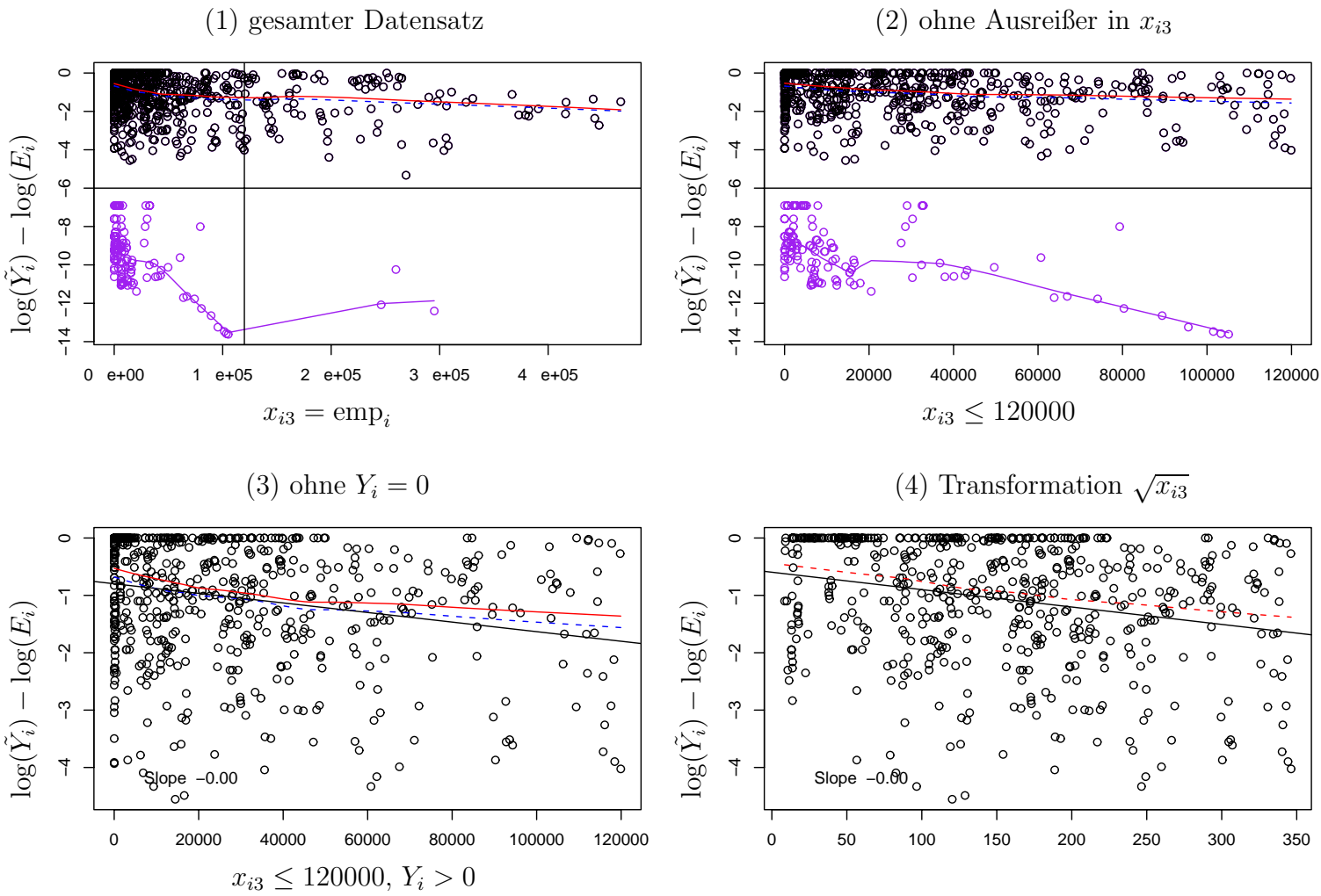


Abbildung 4.4: Plots $\ln(Y_i + 0.001) - \ln(E_i)$ gegen $x_{i3} = \text{emp}_i$; mit *lowess()* geglättete Daten für alle Daten (blau gestrichelt), Daten $Y_i = 0$ (lila) und $Y_i > 0$ (rot); Transformationen für Datenausschnitte mit $Y_i = 0$, Regressionslinien in schwarz

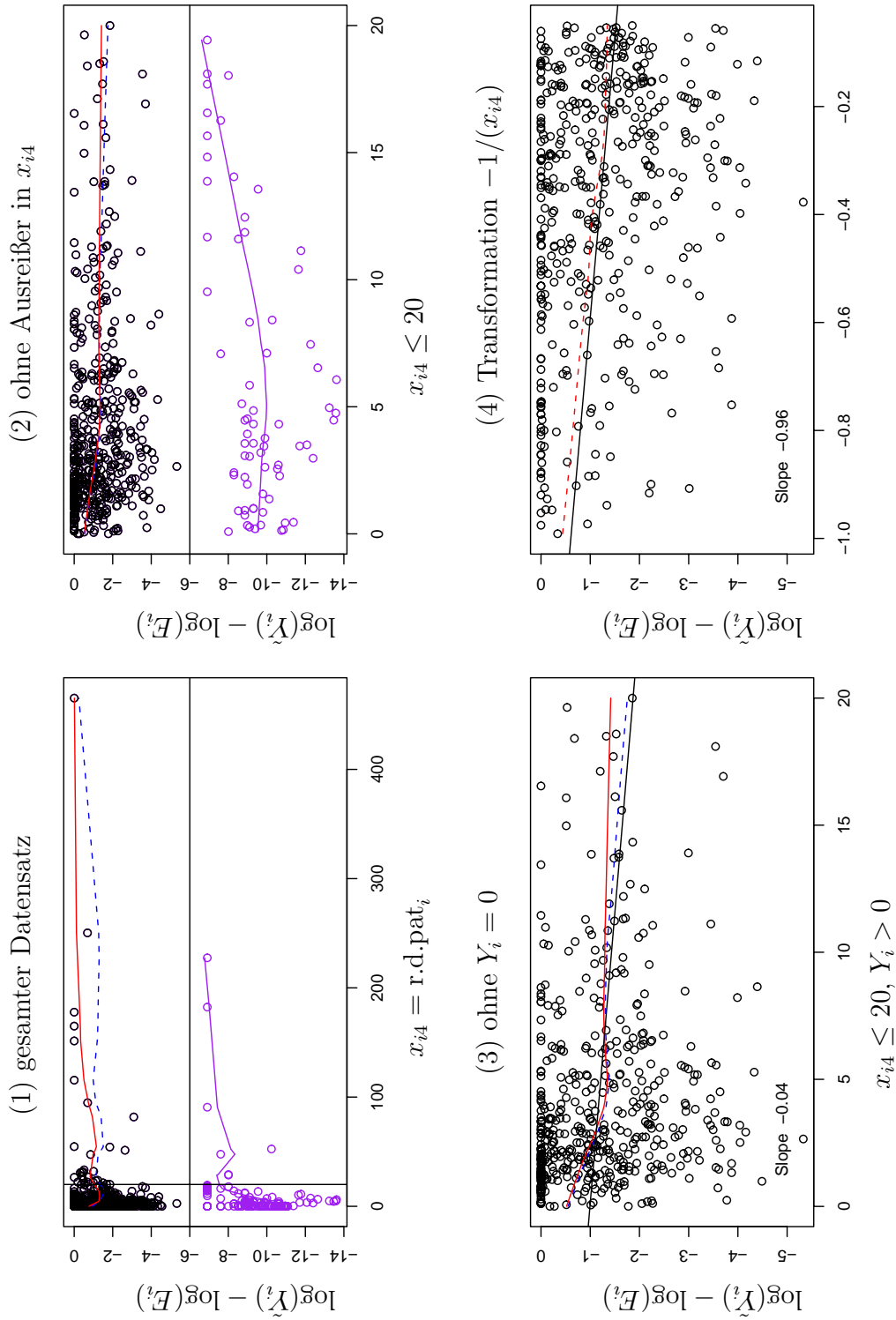


Abbildung 4.5: Plots $\ln(Y_i + 0.001) - \ln(E_i)$ gegen $x_{i4} = \text{r.d.pat}_i$; mit *lowess()* geglättete Daten für alle Daten (blau gestrichelt), Daten $Y_i = 0$ (lila) und $Y_i > 0$ (rot); Transformationen für Datenausschnitte mit $Y_i = 0$, Regressionslinien in schwarz

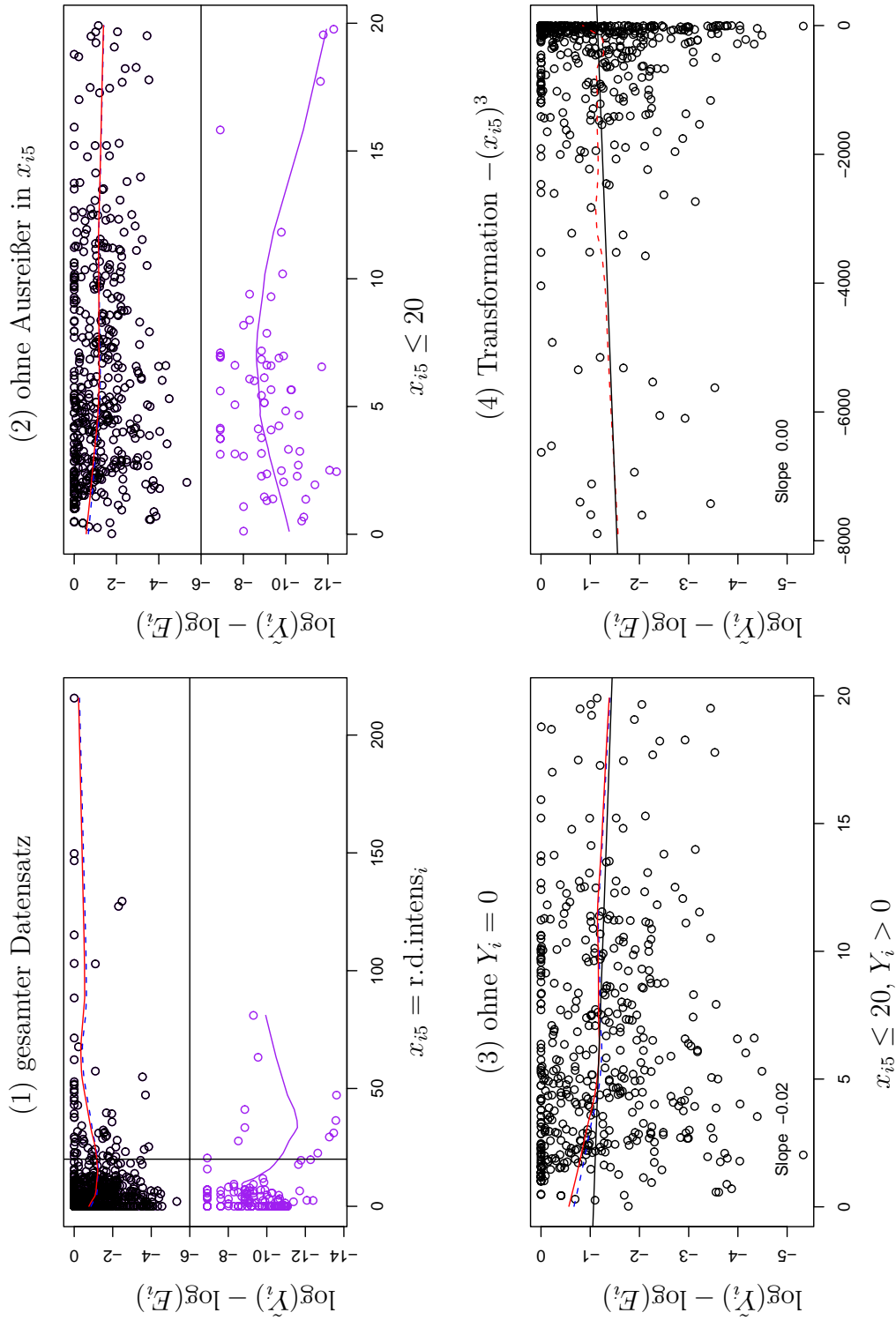


Abbildung 4.6: Plots $\ln(Y_i + 0.001) - \ln(E_i)$ gegen $x_{i5} = \text{r.d.intens}_i$; mit *lowess()* geglättete Daten für alle Daten (blau gestrichelt), Daten $Y_i = 0$ (lila) und $Y_i > 0$ (rot); Transformationen für Datenausschnitte mit $Y_i = 0$, Regressionslinien in schwarz

Zusammenfassend werden noch einmal die Transformationen für die Modellierung des Erwartungswertes aufgeführt.

```
ln.std.5.mean <- log(std.5.mean + 0.001)
sqrt.emp      <- sqrt(emp)
r.d.pat.inv   <- 1/(r.d.pat + 0.001)
r.d.intens.1  <- poly(r.d.intens,3)[,1]
r.d.intens.2  <- poly(r.d.intens,3)[,2]
r.d.intens.3  <- poly(r.d.intens,3)[,3]
```

Auf das Minus in den Transformationen von "r.d.pat" und "r.d.intens" wird verzichtet, da sich damit lediglich Regressionskoeffizienten mit umgekehrtem Vorzeichen ergeben, das Regressionsmodell sonst aber identisch bleibt. Ganz ohne Transformation geht "breadth" in die Modelle ein.

Die Transformationen werden mit Hilfe von empirischen Log Means in Abb. 4.7 verifiziert. Dazu werden für jede Kovariable \mathbf{X}_j k_j Scoring-Klassen gebildet. Diese sind $I_1 := [a_1, b_1], I_2 := (a_2, b_2], \dots, I_{k_j} := (a_{k_j}, b_{k_j}]$; $I_1 \cup \dots \cup I_{k_j} = [\min_i(x_{ij}), \max_i(x_{ij})]$. Man definiert außerdem

$$\begin{aligned} n_l^j &:= \#x_{ij} \in I_l, \quad l = 1, \dots, k_j \\ &= \text{"Anzahl der Beobachtungen von } \mathbf{X}_j \text{ in Klasse } l\text{"} \\ m_l^j &:= \max \left(\frac{1}{n_l^j} \sum_{\{i: x_{ij} \in I_l\}} \frac{Y_i}{E_i}, 0.001 \right), \quad l = 1, \dots, k_j \quad (4.3) \\ &\quad \text{"Mittelwerte von } Y_i/E_i \text{ je Klasse } I_l\text{"} \\ \log \text{mean}_l^j &:= \log(m_l^j), \quad l = 1, \dots, k_j. \end{aligned}$$

Man beachte, daß bei der Berechnung der Log Means in Formel (4.3) gegebenenfalls ein Shift von 0.001 vorgenommen wird, wenn eine Klasse den Mittelwert 0 hat. Da "r.d.miss" und "branches" die einzigen kategorialen Variablen sind, muß für alle verbleibenden metrischen Variablen ein Scoring-Ansatz durchgeführt werden. Diese werden in fünf Klassen eingeteilt; Klassengrenzen sind dabei die Quantile von 0, 20%, 40%, 60%, 80% und 100%. Dargestellt werden die Log Means jeweils an den Klassenmittelpunkten. Bei "r.d.pat" und "r.d.intens" können nur jeweils drei Klassen gebildet werden, da bei beiden etwa 30% der Werte den kleinsten Wert - nämlich 0 - haben. Klassengrenzen können daher nur für drei Klassen eindeutig bestimmt werden.

Die Graphen in Abb. 4.7 wurden mit der Funktion *poisson.main2()* erzeugt. Diese stellt eine Weiterentwicklung der Funktion *poisson.main()* vom Lehrstuhl für Mathematische Statistik (<http://www-m1.ma.tum.de/nbu/modreg/poissonmain.html>) der TU-München dar. Folgende Erweiterungen wurden gemacht: die Plots werden für metrische Daten gemäß dem Scoring-Ansatz an der Abszisse nun an den Klassenmittelpunkten und nicht mehr an äquidistanten Punkten auf der Abszisse geplottet. Dies ermöglicht eine realistischere Abbildung der Daten. Ferner wird der Range zwischen niedrigstem und höchstem Datenpunkt unter dem Graphen in Klammern angezeigt. Die Prozedur wurde für "R" lauffähig gemacht. Die Mittelwerte m_l^j werden automatisch überprüft und gegebenenfalls um 0.001 geshiftet. Wie bisher werden Konfidenzbänder der empirischen Log Means angezeigt. Die Funktion *poisson.main2()* kann auf der beigefügten CD gefunden werden.

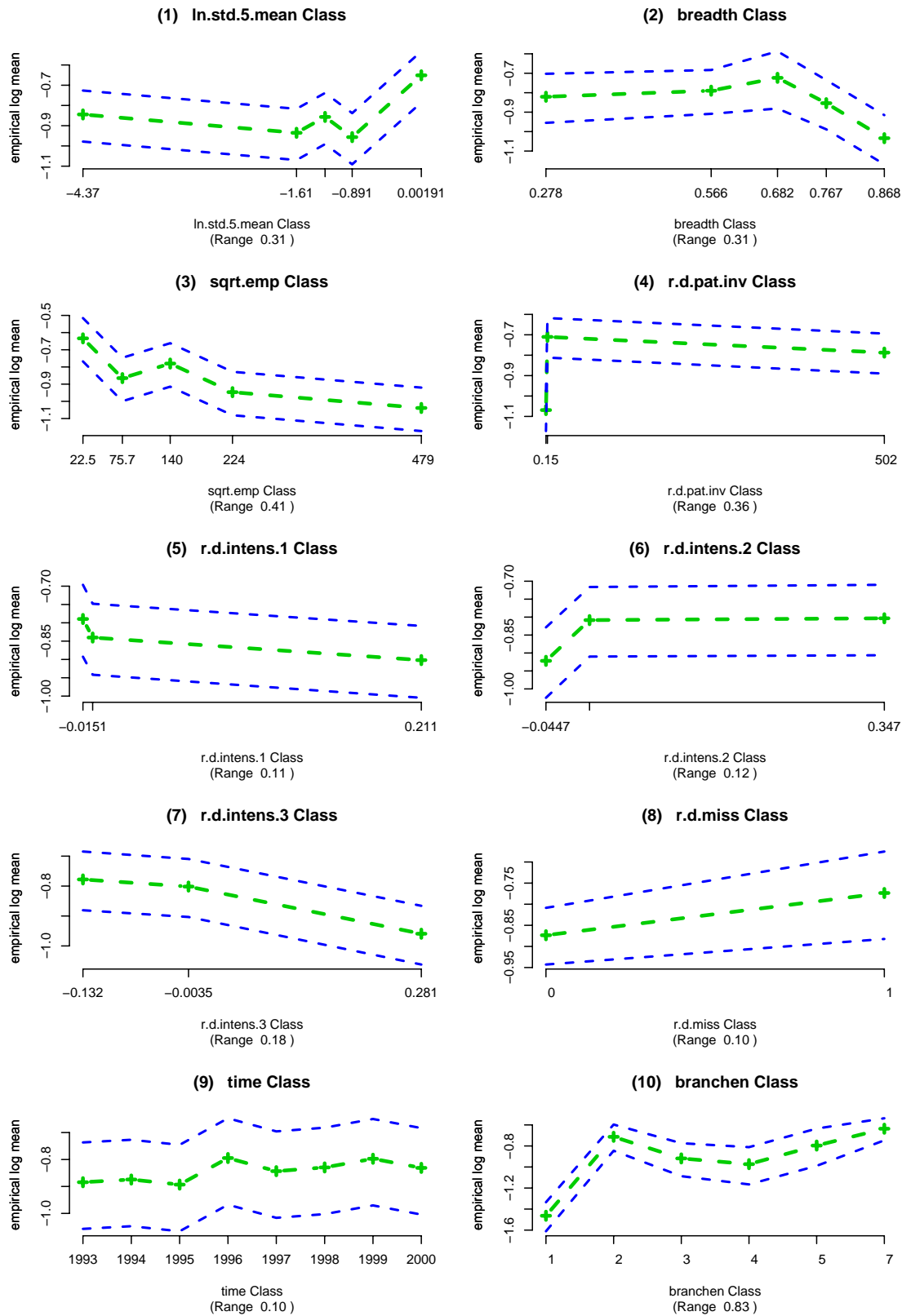


Abbildung 4.7: Empirische Log Means der Kovariablen mit "pat.cum.year" als Exposure

Die Graphen bestätigen die bisherigen Ergebnisse weitgehend, da sich meist annähernd lineare Polygonzüge ergeben. Abb. 4.7 (1) und (2) sind weitgehends linear mit Ausnahme der letzten bzw. letzten beiden Klassen. Linear fallend ist "sqrt.emp" in Abb. 4.7 (3). Je größer die Zahl der Mitarbeiter ist, desto geringer scheint die erwartete Anzahl an ausgelagerten Patentanmeldungen zu sein. Die letzte Klasse von "r.d.pat.inv" entspricht den Nullen in "r.d.pat", diese Beobachtungen scheinen geringe Erwartungswerte zu haben (s. Abb. 4.7 (4)). In Abb. 4.7 (5)–(7) sind die empirischen Log Means des linearen, quadratischen und kubischen Terms der Transformation von "r.d.intens" dargestellt. Dabei ergibt sich für den linearen und kubischen Anteil je ein linear leicht fallender, für den quadratischen Anteil ein steigender Polygonzug. Fehlende Angaben zu Forschung und Entwicklung werden durch eine 1 in "r.d.miss" angezeigt. In diesem Fall, ergibt sich ein höherer Erwartungswert als bei vollständigen Daten (s. Abb. 4.7 (8)). Die Variable "time" hat den geringsten range und verläuft fast horizontal; in welche ihrer Klassen eine Beobachtung fällt, spielt also offensichtlich kaum eine Rolle (s. Abb. 4.7 (9)). Ein reiner zeitlicher Einfluß (d.h. als Haupteffekt) ist daher auszuschließen. Zu "branchen" (s. Abb. 4.7 (10)) ist festzustellen, daß die *log means* für die Branchen 3, 4 und 5, sowie für 2 und 7 annähernd gleich sind. Deshalb sollen diese Branchen für die Modellierung des Erwartungswertes zusammengefaßt werden. Die Branchendummies lauten dann:

```
elec.tel.other      <- elec.tel + other.br
eng.car.sup.med.biot <- engineer + car.sup + med.biot
```

Branche Chemical / Pharma hat den geringsten Erwartungswert, möglicherweise lagern sie wenige Patente aus, da der Spezialisierungsgrad in dieser Branche sehr hoch ist, und es nicht ausreichend qualifizierte Patentanwälte gibt. Die Branchen 3, 4 und 5 sind durchweg Branchen mit technischem Fokus, es handelt sich dabei um Engineering, Cars + Supplier und Medtech + Biotech.

Die Branchen-Zusammenfassung läßt sich mit Hilfe von Profilen der Form $\log(\tilde{Y}_i/E_i)$ sogar noch besser begründen. Dazu wird $\log(\tilde{Y}_i/E_i)$ für alle Unternehmen nach Branchen und Jahren getrennt berechnet und in Abb. 4.8 dargestellt. Damit kann man untersuchen, wie homogen die Auslagerungsraten der Unternehmen je Branche sind, und welche zeitliche Entwicklung vorliegt. Der Branchenmedian ist jeweils als schwarzer Polygonzug mit Kringeln dargestellt. Mediane werden hier Mittelwerten vorgezogen, da sie weniger sensitiv auf Ausreißer reagieren. Die Branchen "Electr./ Telco" (s. Abb. 4.8 (2)), "Cars + Supplier" (s. Abb. 4.8 (4)) sowie "Medtech/ Biotech" (s. Abb. 4.8 (5)) sind dabei sehr homogen, die übrigen weniger.

In Abb. 4.9 (1) werden die Branchen-Mediane in einer gemeinsamen Graphik dargestellt. Die eben zusammengefaßten Branchen sind in Abb. 4.9 (2) zu sehen.

Man erkennt deutliche Branchenunterschiede, ein eindeutiger zeitlicher Trend ist jedoch nicht auszumachen. Eine Interaktion aus Branchenzugehörigkeit und Zeit wird daher ausgeschlossen. In Abb. 4.9 (2) werden zusätzlich Mediane über alle Jahre als horizontale Linien dargestellt. Insbesondere "Electr./ Telco & Other" haben einen hohen Anteil an ausgelagerten Patenten; "Chem./ Pharma" dagegen den geringsten.

Nun zur Untersuchung von Interaktionen. Mit Hilfe von *Pairs()*-Plots in Abb. 4.10 kann man sehen, daß "breadth" und "sqrt.emp" untereinander stark korreliert sind. Je höher

die Anzahl der Mitarbeiter eines Unternehmens ist, desto höher ist offenbar auch die Breite der Forschungsgebiete. Große Unternehmen scheinen auf mehr Gebieten forschen zu können als kleine.

Auch "ln.std.5.mean" und "breadth" sind korreliert: je breiter ein Unternehmen forscht, umso volatiler sind seine Anmeldungszahlen. Forschungsbreite wird offensichtlich durch die Freigabe großer Forschungsetats erreicht, was zu periodisch auftretenden vermehrten Patent-Anmeldungen führt. Für die übrigen Variablen ist kein Muster erkennbar.

Mit *pairs()*-Plots können jedoch nur Korrelationen metrischer Variablen untersucht werden. Um umfassend nach Interaktionen zu suchen, wird mit empirischen Log Means gearbeitet. Auch hier werden Scoring-Klassen für metrische Kovariablen gebildet. Für zwei Kovariablen \mathbf{X}_1 mit k_1 Scoring-Klassen und \mathbf{X}_2 mit k_2 Scoring-Klassen berechnet man:

$$\begin{aligned}
 n_{lm} &:= \#Y_i \in I_l^1 \cap I_m^2, & l = 1, \dots, k_1, m = 1, \dots, k_2 \\
 &= \text{"Anzahl der Beobachtungen von } \mathbf{Y}_i \text{ in Zelle } (l, m)\text{"} \\
 m_{lm} &:= \max \left(\frac{1}{n_{lm}} \sum_{\{i: Y_i \in I_l^1 \cap I_m^2\}} \frac{Y_i}{E_i}, 0.001 \right), \\
 & & l = 1, \dots, k_1, m = 1, \dots, k_2 \\
 & & \text{"Mittelwerte von } Y_i/E_i \text{ in Zelle } (l, m)\text{"} \\
 \log \text{mean}_{lm} &:= \log(m_{lm}), & l = 1, \dots, k_1, m = 1, \dots, k_2.
 \end{aligned} \tag{4.4}$$

Die Zuordnung $\{i : Y_i \in I_l^1 \cap I_m^2\}$ entspricht der Erstellung einer Kreuztabelle. Dabei schließt (4.4) einen Shift leerer Zellen der Kreuztabelle um 0.001 mit ein. Der Umfang der zu betrachtenden Interaktionen wird dadurch beschränkt, daß nur zweidimensionale Interaktionen betrachtet werden; höher-dimensionale Effekte werden vernachlässigt. Die empirischen Log Means werden mit der "R"-Funktion *interactions()* berechnet. Diese Funktion stellt eine Weiterentwicklung der Funktion *poisson.inter()* vom Lehrstuhl für Mathematische Statistik an der TU-München dar. Während *poisson.inter()* unter <http://www-m1.ma.tum.de/nbu/modreg/poissoninter.html> gefunden werden kann, ist *interactions()* auf der beigefügten CD zu finden. Gegenüber der ursprünglichen Version werden Plots für metrische Daten gemäß dem Scoring-Ansatz an der Abszisse an den Klassenmittelpunkten und nicht an äquidistanten Punkten auf der Abszisse geplottet. Entsteht in der Kreuztabelle eine leere Zelle, so wird diese automatisch um 0.001 geschiftet.

Interaktionen sind daran zu erkennen, daß sich die Polygonzüge in Abb. 4.11 für die verschiedenen Klassen abschnittsweise *nicht* annähernd parallel bewegen. Dann bewirkt eine Veränderung des einen Merkmals nämlich eine nicht-proportionale Veränderung des anderen Merkmals. Dies trifft auf viele Variablenkombinationen zu, herausgegriffen werden aber die vier deutlichsten Interaktionen:

```

ln.std.5.mean.breadth <- ln.std.5.mean*breadth
chem.pha.ln.std.5.mean <- chem.pha*ln.std.5.mean
chem.pha.sqrt.emp     <- chem.pha*sqrt.emp
r.d.miss.chem.pha     <- r.d.miss*chem.pha

```

Zusammenfassung

Es wurden folgende Variablen ausgewählt:

```
X <- cbind(Intercept, ln.std.5.mean, breadth, sqrt.emp, r.d.pat.inv, r.d.intens.1,  
           r.d.intens.2, r.d.intens.3, r.d.miss, chem.pha, elec.tel.other, time,  
           ln.std.5.mean.breadth, chem.pha.ln.std.5.mean, chem.pha.sqrt.emp,  
           r.d.miss.chem.pha)
```

Eine Notationstabelle mit einer Zusammenfassung aller in eingeführten Notationen kann am Anfang von Anhang D gefunden werden.

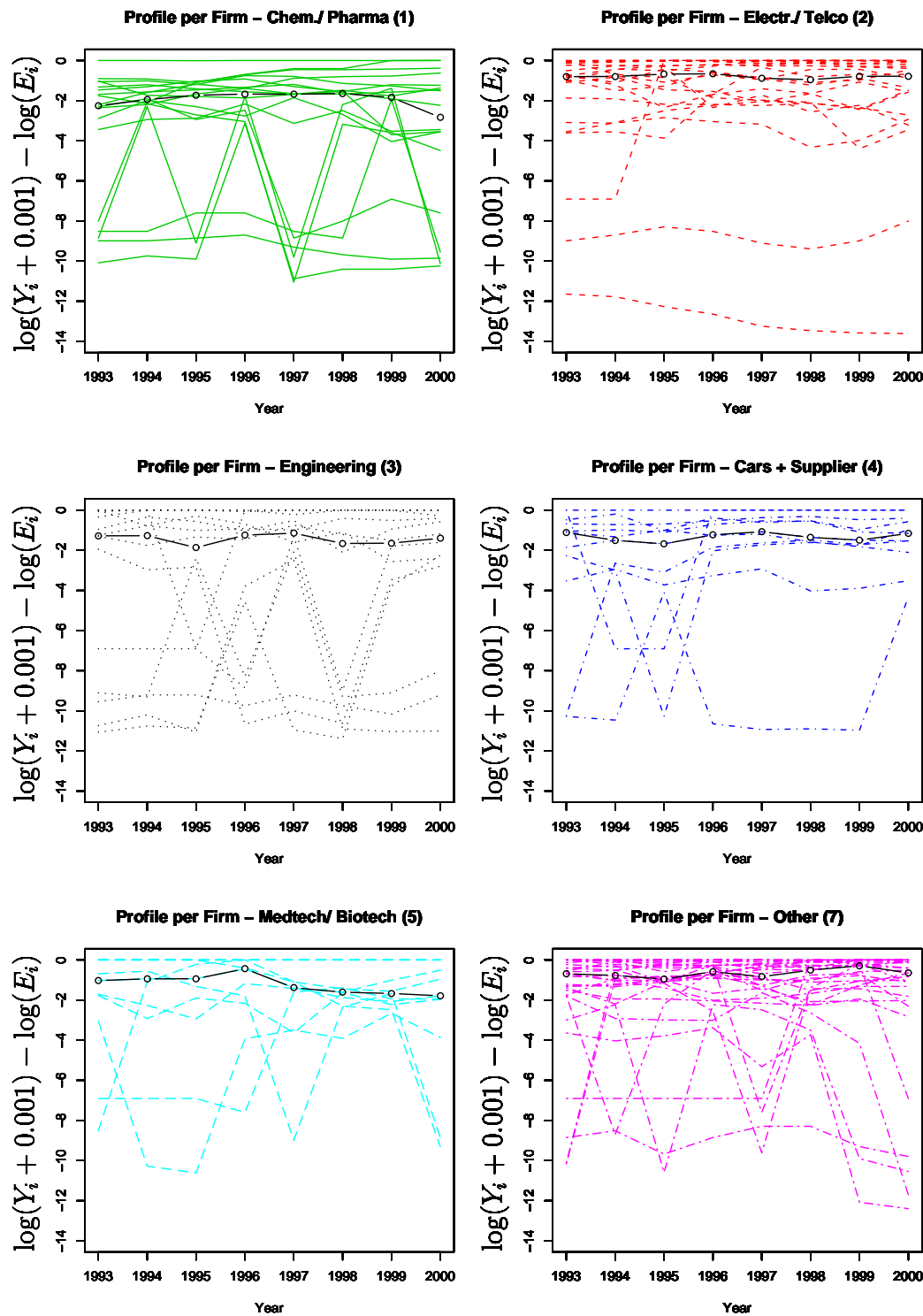


Abbildung 4.8: Profile der logarithmierten Anteile ausgelagerter Patente nach Branchen getrennt; Branchen-Mediane als durchgezogene schwarze Linien

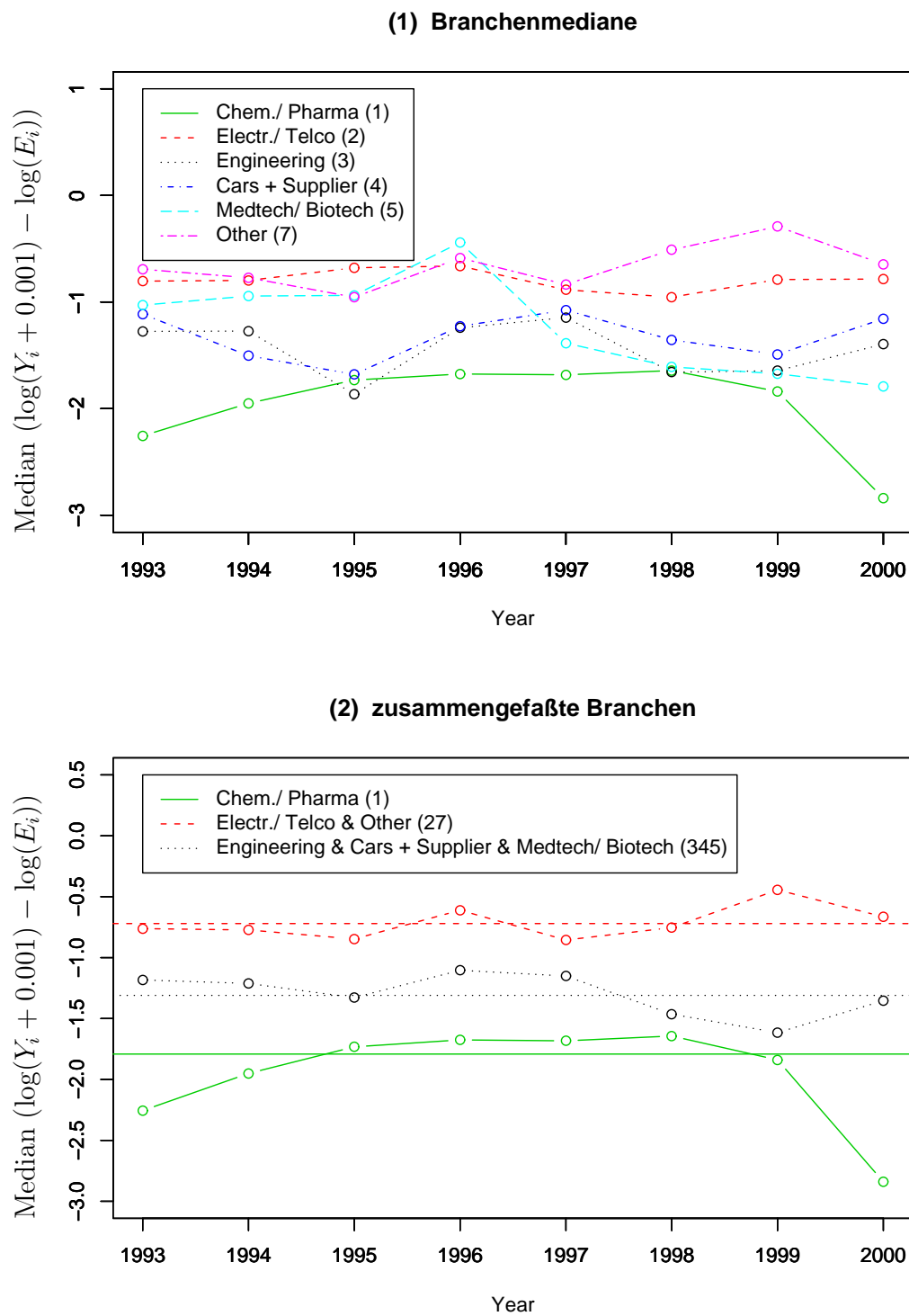
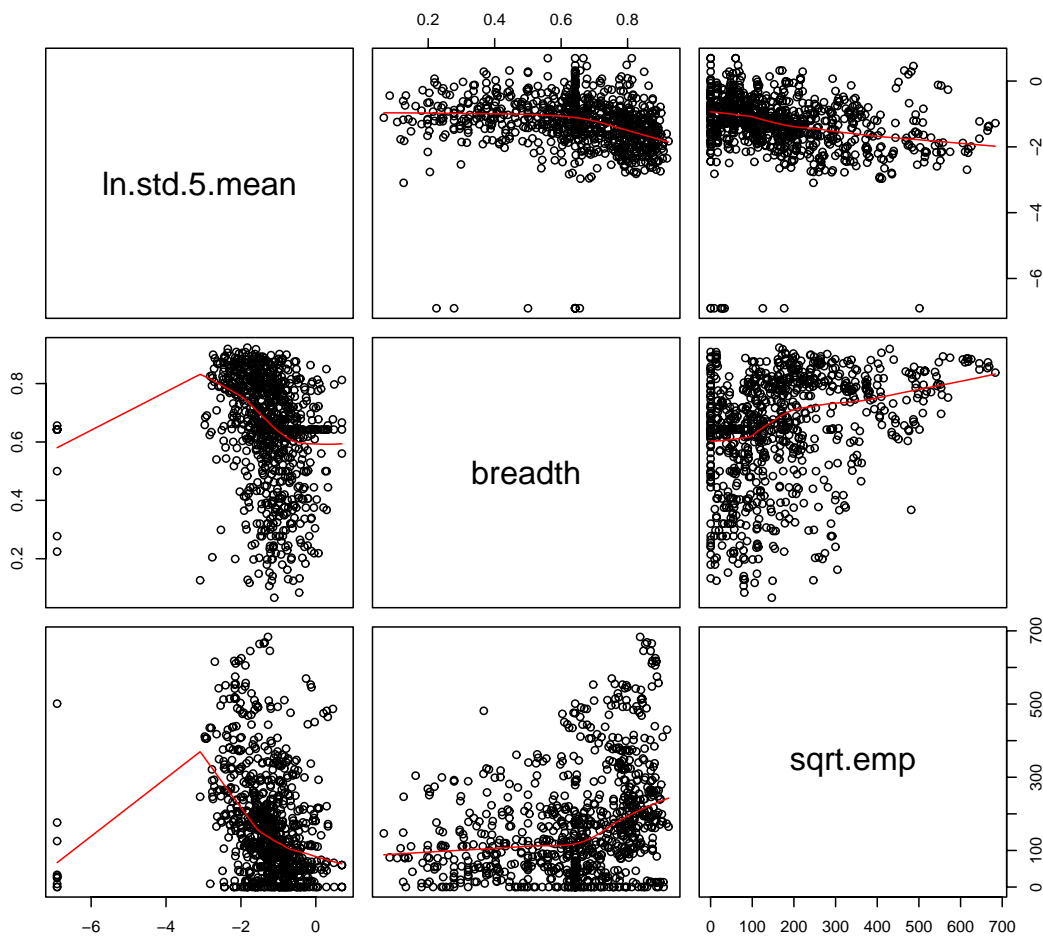


Abbildung 4.9: Profile der Branchenmediane im Zeitverlauf (1); zusammengefaßte Branchen und zeitunabhängige Mediane über alle Jahre als horizontale Linien (2)

Abbildung 4.10: Pairs-Plot der ausgewählter Haupteffekte mit *lowess()*-Glättungslinien

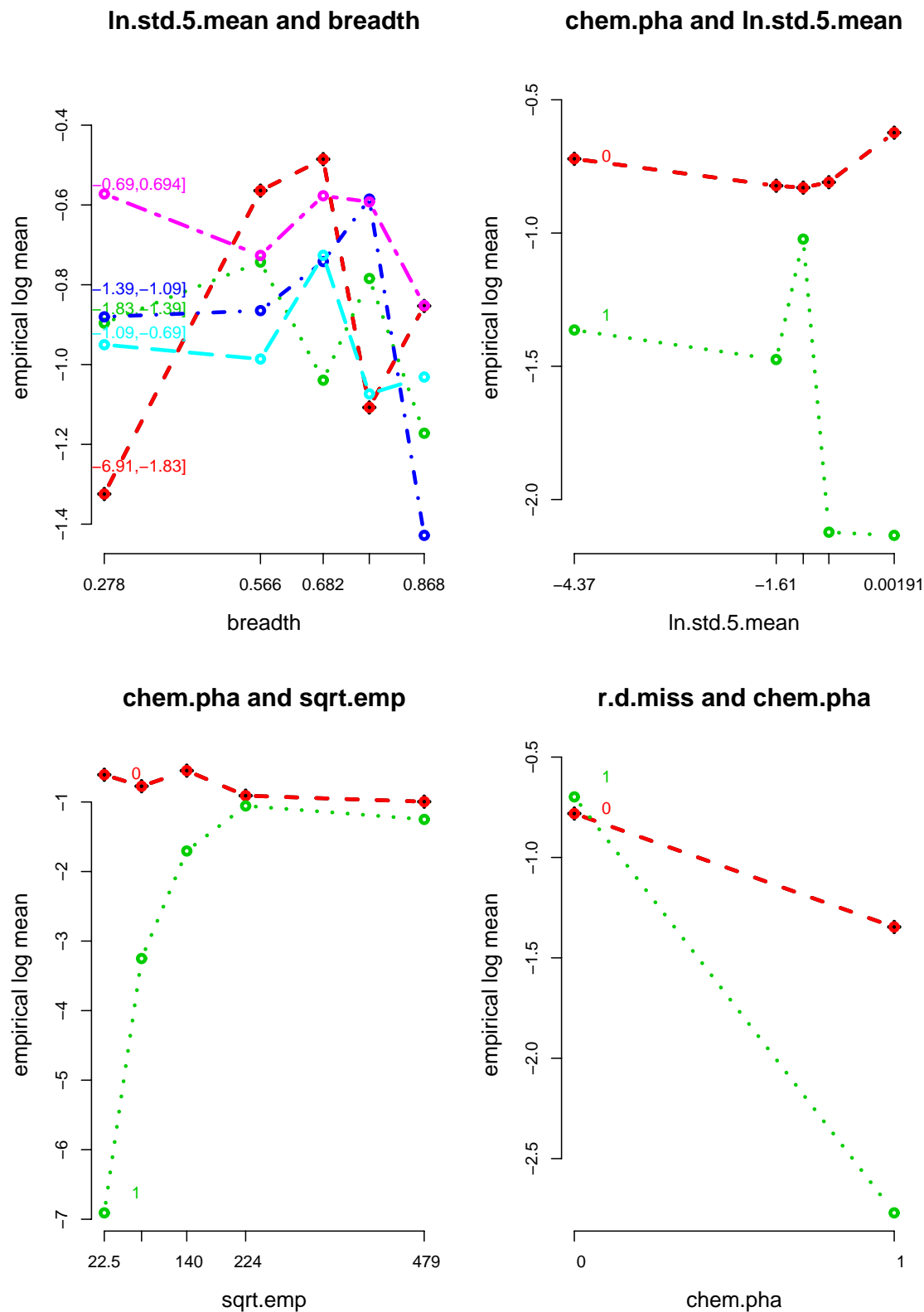


Abbildung 4.11: Empirische Log Means zur Untersuchung von zweidimensionalen Interaktionseffekten

4.2.2 Kovariablen für die Modellierung der Überdispersion

Paßt man ein gewöhnliches $Poi(\mu_i)$ -GLM an die Daten und verwendet dabei die eben für den Erwartungswert ausgewählten Kovariablen, so kann man aus dem "R"-Summary Rückschlüsse auf die Dispersion der Daten ziehen: es ist bekannt, daß die Residual Deviance approximativ χ^2_{n-p-1} -verteilt ist. Der Erwartungswert dieser Verteilung ist $n - p - 1$. Bei 16 Kovariablen bleiben demnach noch $n - p - 1 = 856 - 15 - 1 = 840$ Freiheitsgrade. Übersteigt die Residual Deviance diesen Wert, so ist dies als Hinweis auf geringe Anpassungsqualität zu werten (s. Myers et al. [29, S. 113]). Ein Summary des Poissonmodells ist auf S. 240 zu finden. Da die Residual Deviance die Anzahl der Freiheitsgrade deutlich übersteigt, kann man auf hohe Überdispersion schließen.

Residual deviance: 20627 on 840 degrees of freedom

Es gibt noch keine etablierte Methode, um in einem Regressionsmodell mit Regression auf den Dispersionsparameter Kovariablen zu wählen, die einen signifikanten Einfluß auf ihn haben. Daher wird in diesem Abschnitt versucht, ein gut begründbares Verfahren zu entwickeln, das sich auf die Verknüpfungsfunktion des Log-Link bezieht. Man betrachtet die ursprünglichen Kovariablen ohne Transformationen.

Es ist bekannt, daß für Poissonverteilte Daten gilt: $Y \sim Poi(\mu) \Rightarrow E(Y) = Var(Y) = \mu$. Ein einfacher Test auf Überdispersion kann daher so konstruiert werden, daß die empirische Stichprobenvarianz mit dem Mittelwert der Daten verglichen wird. Diese Statistiken werden getrennt nach Jahren $t = 1993, \dots, 2000$ und der Branche $D(i) \in \{"1", \dots, "5", "7"\}$ betrachtet. Die Anzahl der betrachteten Unternehmen in Branche $D(i) = j$ sei n_j . Für diese Zelldaten sind Mittelwert und empirische Varianz gegeben durch

$$\begin{aligned}\hat{\mu}_{j,t}(\mathbf{Y}) &= \frac{1}{n_j} \sum_{\{i,t: D(i)=j, T=t\}} Y_i \\ \text{und } \hat{\sigma}_{j,t}^2(\mathbf{Y}) &= \frac{1}{n_j - 1} \sum_{\{i,t: D(i)=j, T=t\}} (Y_i - \hat{\mu}_{j,t}(\mathbf{Y}))^2.\end{aligned}$$

Allgemein gilt nun aber für Verallgemeinert Poisson-verteilte Daten

$$\begin{aligned}\sigma_i^2 &= \mu_i \cdot \varphi_i^2 \\ \Leftrightarrow \varphi_i &= \sqrt{\frac{\sigma_i^2}{\mu_i}}.\end{aligned}$$

Daher erhält man mit $\varphi_i = 1 + e^{\mathbf{w}_i^t \boldsymbol{\alpha}}$, daß

$$\mathbf{w}_i^t \boldsymbol{\alpha} = \log \left(\sqrt{\frac{\hat{\sigma}_{j,t}^2(\mathbf{Y})}{\hat{\mu}_{j,t}(\mathbf{Y})}} - 1 \right) =: f_{j,t}(\mathbf{Y}).$$

Die Funktionswerte sind in Tabelle 4.2 dargestellt.

$D(i)$	$n_{D(i)}$	1993	1994	1995	1996	1997	1998	1999	2000
Chem./ Pharma "1"	18	$\hat{\mu}_{1,1} = 24.7$	30.3	35.6	44.7	52.3	50.9	50.1	56.2
		$\hat{\sigma}_{1,1}^2 = 746.4$	1610.4	2442.3	3932.0	7898.2	7190.6	8812.8	12309.8
		$f_{1,1} = 1.50$	1.84	1.99	2.13	2.42	2.39	2.51	2.62
Electr./ Telco "2"	22	54.7	55.9	67.6	81.1	88.9	112.6	126.0	146.7
		12650.9	12382.0	16741.3	21751.5	24587.1	36325.5	48137.5	65028.4
		2.65	2.63	2.69	2.73	2.75	2.83	2.92	3.00
Engineering "3"	14	7.1	7.3	7.9	7.2	10.7	11.1	15.3	15.4
		62.1	98.2	99.5	86.0	250.7	362.1	644.4	627.3
		0.67	0.98	0.93	0.90	1.34	1.55	1.70	1.68
Cars + Supplier "4"	11	17.6	14.7	17.5	26.6	35.7	36.9	31.6	36.1
		613.7	442.2	780.1	1451.1	2032.6	1997.5	1333.5	1274.1
		1.59	1.50	1.73	1.85	1.88	1.85	1.70	1.60
Medtech/ Biotech "5"	11	6.3	7.6	5.8	6.9	6.3	7.3	10.7	10.7
		29.8	45.1	31.4	36.1	41.0	62.6	131.6	226.2
		0.17	0.36	0.28	0.25	0.44	0.66	0.92	1.28
Other "7"	31	12.7	13.5	15.5	17.3	19.5	19.4	21.2	20.3
		414.0	394.9	593.2	707.9	979.6	938.8	1258.2	1028.4
		1.55	1.48	1.64	1.69	1.81	1.78	1.90	1.81

Tabelle 4.2: Geschätzte Mittelwerte und Varianzen je Branche und Jahr; Funktionswerte $f_{j,t}(\mathbf{Y})$ als Indikator für Überdispersion

Läge keine Überdispersion vor, so wären Erwartungswert und Varianz identisch, der Quotient $\frac{\hat{\sigma}_{j,t}^2}{\hat{\mu}_{j,t}}$ müßte also für alle Zellen etwa 1 sein. Innerhalb des Logarithmus müßten demnach Werte nahe der Null entstehen. Es ergeben sich jedoch sehr hohe Quotienten, die in allen Fällen für starke Überdispersion sprechen: sie schwanken zwischen 4.8 und 443. Man kann sagen: je höher die Funktionswerte $f_{j,t}(\mathbf{Y})$ sind, desto höher ist die Überdispersion. Ist der Wert für eine Branche in einem bestimmten Jahr größer als 0, so übersteigt die Varianz den Erwartungswert bereits um mehr als das vierfache. Insgesamt deuten die Funktionswerte $f_{j,t}$ in Tabelle 4.2 für alle Branchen auf starke Überdispersion hin, die teilweise im Zeitverlauf noch ansteigt. Die Werte sind in Abb. 4.12 (1) graphisch dargestellt. Man sieht, daß es sich anbietet, die Branchen "4" und "7" zur Branche "Cars + Supplier & Other" zusammenzufassen, da ihre Funktionswerte sehr nahe beisammen liegen. Die Funktionswerte der so entstehenden gemeinsamen Branche ist in Abb. 4.12 (2) zu sehen.

```
car.suppl.other <- car.suppl + other.br
```

Als Referenzkategorie soll die Branche "Electr./ Telco" dienen, da sie im betrachteten Zeitrahmen keine Veränderung erkennen läßt und gleichzeitig maximale Funktionswerte hat. Die übrigen Branchen unterscheiden sich demnach deutlich von der Referenzbranche. Ein linearer zeitlicher Einfluß zusammen mit der Branche wird angenommen für "Chem./ Pharma", "Engineering" und "Medtech/ Biotech", da deren Funktionswerte annähernd linear ansteigen. Es wird sich jedoch herausstellen, daß diese Interaktionen nicht signifikant sind. Als signifikante Branchen werden sich "Engineering", "Cars / Supplier + Other" und "Medtech / Biotech" herausstellen.

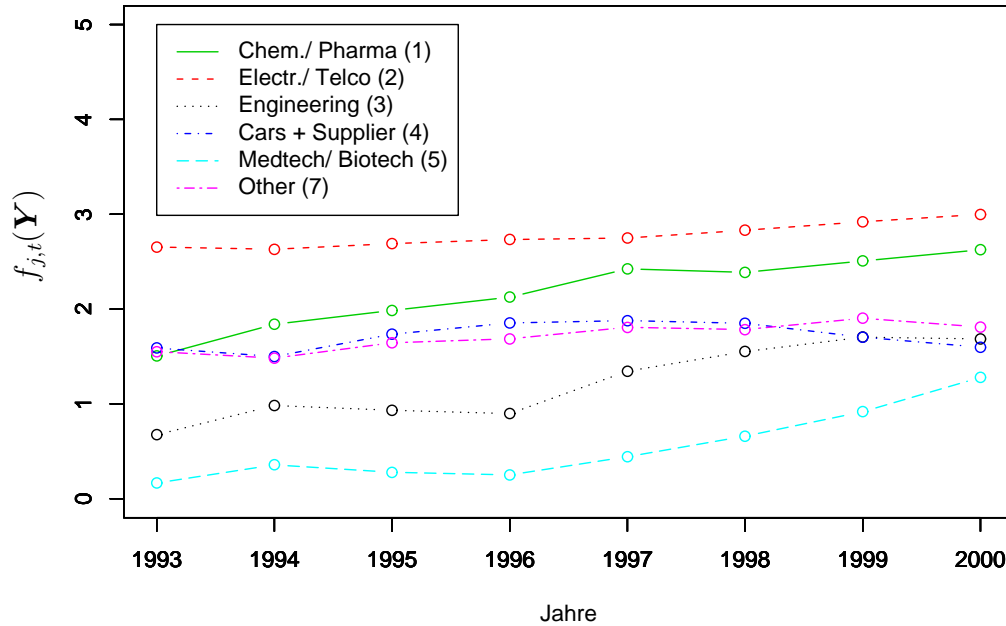
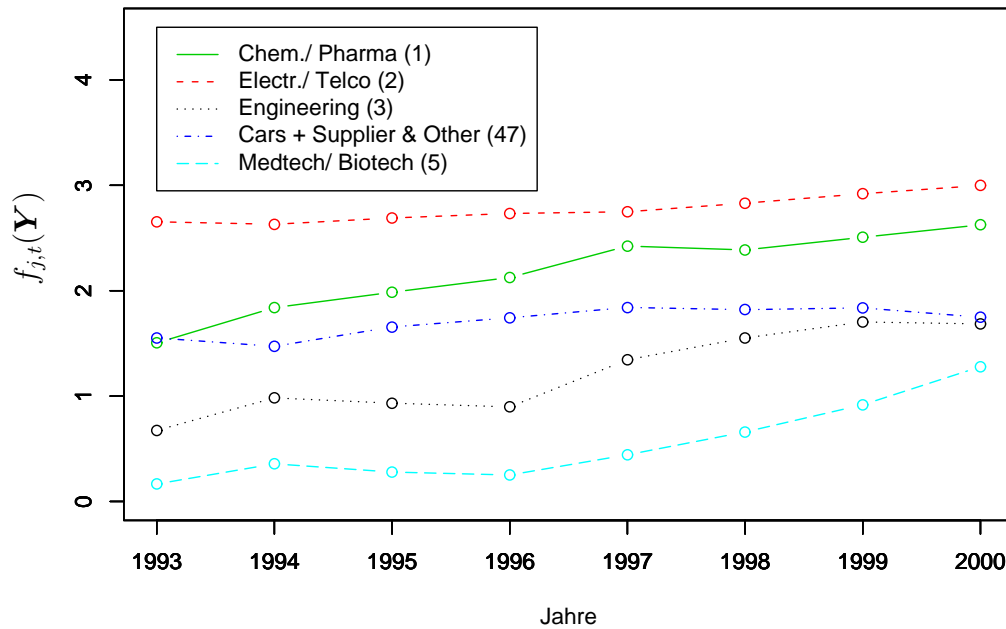
(1) Überdispersionsfunktion nach Branchen getrennt**(2) Branchen 4 und 7 zusammengefaßt**

Abbildung 4.12: Funktionswerte $f_{j,t}(\mathbf{Y})$ für $t = 1993, \dots, 2000$ (1); Branchen "4" und "7" zusammengefaßt zu "Cars + Supplier & Other" (2)

Nun wird der Einfluß der Kovariablen auf $f_{l(i)}(\mathbf{Y})$ untersucht. Erneut werden für metrische Variablen Scoring-Klassen $l(i)$ wie in Abschnitt 4.2.1, S. 98 gebildet. Die Funktionswerte sind in Tabelle 4.3 aufgelistet und werden in Abb. 4.13 graphisch dargestellt. Je höher der unter den Graphen angegebene Range ist, desto höher ist der Einfluß bestimmter Scoring-Klassen. Dazu kann man etwa Klassen mit niedrigen und Klassen mit hohen Funktionswerten gruppieren und die Zugehörigkeit der Beobachtung zur jeweiligen Gruppe durch Dummyvektoren festlegen.

Es wird so vorgegangen, daß Dummyvariablen zur Indikation signifikanter Scoring-Klassen erzeugt werden, wenn der Range der Funktionswerte über alle Klassen 1 übersteigt, also $\max_{l(i)}(f_{l(i)}(\mathbf{Y})) - \min_{l(i)}(f_{l(i)}(\mathbf{Y})) \geq 1$. Dies trifft für "breadth" (s. Abb. 4.13 (2)), "emp" (Abb. 4.13 (3)) und "r.d.intens" (Abb. 4.13 (5)) zu. Wurden die erzeugten Dummyvektoren nämlich in die Designmatrix für die Überdispersion in ein $GP(\mu_i, \varphi_i)$ -Modell aufgenommen, so zeigte sich, daß sie einen signifikanten Einfluß hatten, wenn ihr Range etwa den Wert 1 überstieg. Exemplarisch wird nun für "emp" gezeigt, wie die Dummyvektoren unter "R" erzeugt werden.

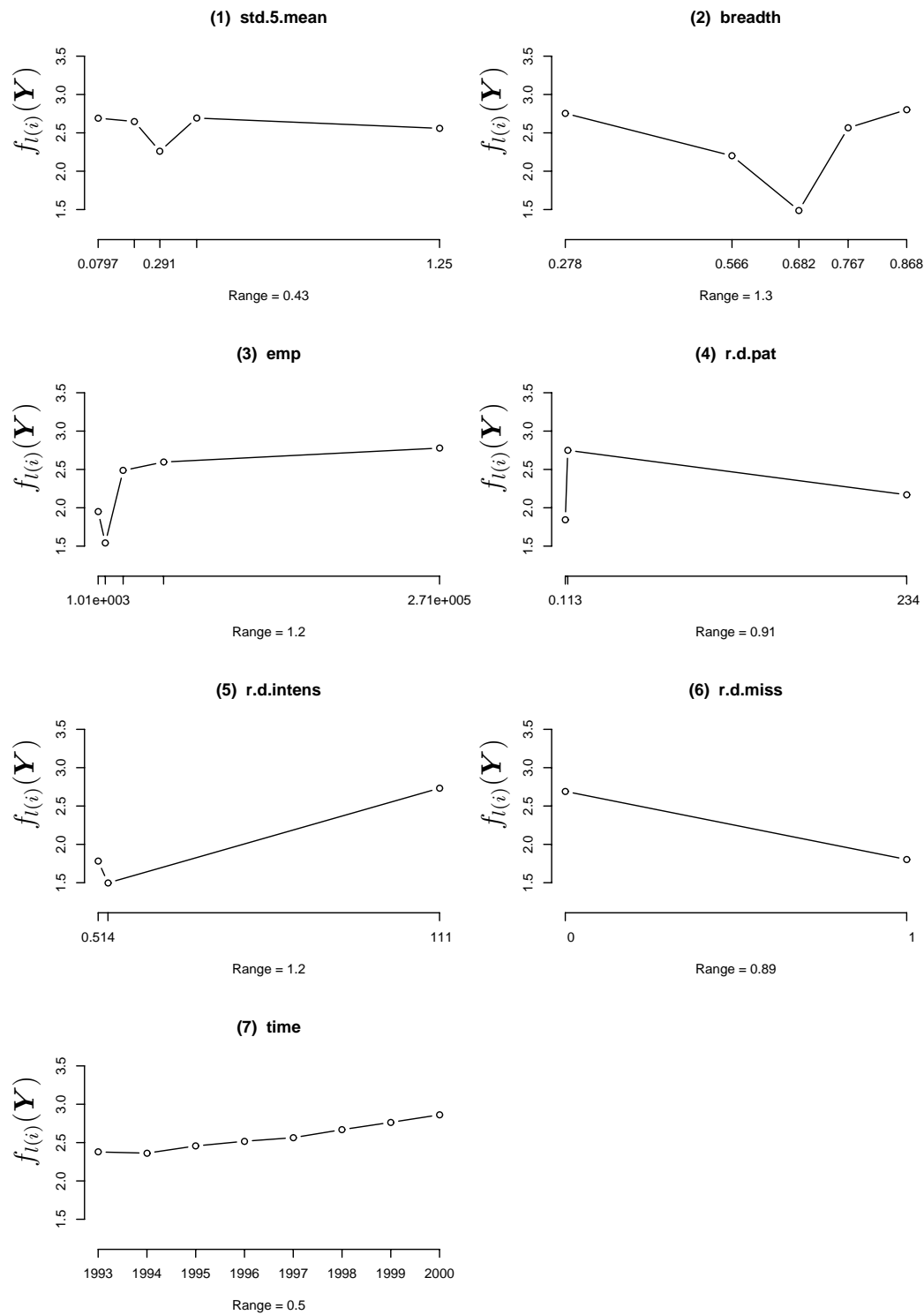
```
emp.cut <- cut(emp, breaks = quantile(emp, probs = c(0, 0.2, 0.4, 0.6, 0.8, 1)))
emp.k12 <- rep(0,856)
emp.k12[emp.cut==levels(emp.cut)[1]] <- 1
emp.k12[emp.cut==levels(emp.cut)[2]] <- 1
```

Insgesamt werden für die Modellierung der Überdispersion also ausschließlich kategoriale Variablen verwendet. Sie lauten:

```
W <- cbind(Intercept, engineer, car.suppl.other, med.biot, time, breadth.k145,
           emp.k12, r.d.intens.k12)
```

Variable	Klasse 1	Klasse 2	Klasse 3	Klasse 4	Klasse 5	Klasse 6	Klasse 7	Klasse 8
std.5.mean	Klasse = [0, 0.159]	(0.159, 0.248]	(0.248, 0.334]	(0.334, 0.501]	(0.501, 2]			
	$n_{1,1} = 172$	171	171	171	171			
	$\hat{\sigma}_{1,1}^2 = 19423.1$	7325.3	3153.0	5153.1	4081.9			
	$\hat{\mu}_{1,1} = 78.2$	32.0	28.1	20.7	21.1			
	$f_{1,1} = 2.7$	2.6	2.3	2.7	2.6			
breadth	[0.0666, 0.49]	(0.49, 0.642]	(0.642, 0.721]	(0.721, 0.812]	(0.812, 0.923]			
	172	215	127	171	171			
	6237.8	1837.5	636.0	10372.3	20215.3			
	22.4	18.3	21.6	52.9	66.2			
	2.8	2.2	1.5	2.6	2.8			
emp	[0, 2020]	(2020, 11320]	(11320, 30249]	(30249, 75322]	(75322, 466938]			
	172	171	171	171	171			
	954.9	228.3	6727.2	9461.9	21415.2			
	14.8	7.1	39.7	45.5	73.3			
	1.9	1.5	2.5	2.6	2.8			
r.d.pat	[0, 0.226]	(0.226, 3.36]	(3.36, 465]					
	283	290	283					
	815.3	19835.2	1942.7					
	15.2	71.7	20.4					
	1.8	2.7	2.2					
r.d.intens	[0, 1.03]	(1.03, 6.34]	(6.34, 216]					
	283	290	283					
	669.1	473.8	21154.7					
	13.9	15.8	79.0					
	1.8	1.5	2.7					
r.d.miss	0	1						
	602	254						
	11181.0	732.6						
	45.1	14.7						
	2.7	1.8						
time	1993	1994	1995	1996	1997	1998	1999	2000
	107	107	107	107	107	107	107	107
	3116.7	3204.8	4461.8	6028.9	7517.9	10304.8	13307.4	17904.0
	22.4	23.8	27.8	33.6	38.4	43.3	46.8	52.3
	2.4	2.4	2.5	2.5	2.6	2.7	2.8	2.9

Tabelle 4.3: Funktionswerte $f_{l(i)}(\mathbf{Y})$ zur Untersuchung der Überdispersion je nach Klasse der Variablen

Abbildung 4.13: Überdispersionsfunktion $f_{l(i)}(Y)$ je Klasse der Kovariablen

4.2.3 Kovariablen für die Modellierung des Nullenüberschuß

Um Kovariablen für die Modellierung des Nullenüberschuß auszuwählen, werden empirische Logits der Kovariablen gebildet. Diese Idee geht auf die binäre Regression zurück: das Ereignis "Beobachtung Y_i ist eine Null" ist ein binäres Zufallsexperiment. Empirische Logits werden wie folgt berechnet. Das Kroneckersymbol δ_{ij} zeige an, ob Beobachtung i in Klasse j liege:

$$\begin{aligned}\delta_{ij} &:= \begin{cases} 1 & i \in \text{Klasse } j \\ 0 & \text{sonst} \end{cases}, \\ \hat{\omega}_j &:= \frac{\#\{Y_i = 0, \delta_{ij} = 1\}}{\#\{Y_i = 0\}} \\ \text{und } \hat{\text{logit}}_j &:= \log \left(\frac{\hat{\omega}_j + \frac{1}{2}}{1 - \hat{\omega}_j + \frac{1}{2}} \right).\end{aligned}\tag{4.5}$$

Da die Klassengrenzen durch Quantile festgelegt wurden, ist die Anzahl der Beobachtungen in jeder Klasse etwa gleich. Hat eine Kovariable \mathbf{X}_j keinen Einfluß auf die Anzahl der Nullen in \mathbf{Y} , so erwartet man, daß bei k_j Klassen etwa $\frac{1}{k_j}$ der Nullen von \mathbf{Y} auf jede Klasse entfallen. Der empirische Logit ist Schätzer für die Wahrscheinlichkeit einer Null in \mathbf{Y} in Klasse k_j gegenüber allen übrigen Klassen. Man benutzt einen Shift um $\frac{1}{2}$, da es auch vorkommen kann, daß keine einzige Beobachtung $Y_i = 0$ in eine bestimmte Klasse fallen kann. Der empirische Logit von $\hat{\omega}_j = \frac{1}{k_j}$ ist in Abb. 4.14 jeweils als waagerechte Linie eingezeichnet. Noch einmal: sie stellt den empirischen Logit im Fall ohne Nullenüberschuß dar. Hohe Abweichungen von dieser Linie deuten auf hohen Einfluß der Klasse auf den Nullenüberschuß hin.

Zur Berechnung der empirischen Logits wird die "R"-Funktion *main2.plot()* verwendet, die ebenfalls auf der CD zu finden ist. Sie ist eine Weiterentwicklung der Funktion *main1.plot*, die vom Lehrstuhl Statistik der TU-München für die Logistische Regression entwickelt wurde (siehe <http://www-m1.ma.tum.de/nbu/modreg/BinomReg.shtml>). Die Änderungen bestehen in den Abszissenpunkten, an denen geplottet wird und der Berechnung der durchschnittlichen Nullen pro Klasse $\hat{\omega}_j$.

Die empirischen Logits aus Tabelle 4.4 sind mit Konfidenzbändern in Abb. 4.14 graphisch dargestellt. Es werden Dummyvektoren nach der Klassenzugehörigkeit der Kovariablen gebildet und diese in die Regression für den Nullenüberschuß aufgenommen. Im einzelnen sind dies "breadth" (s. Abb. 4.14 (2)), "emp" (Abb. 4.14 (3)) und "r.d.pat" (Abb. 4.14 (4)). Bei "breadth" zeigt der Dummyvektor an, ob sich die Beobachtung in Klasse 1 oder 2 von "breadth" befindet. Bei "emp" und "r.d.pat" wird jeweils Klasse 2 in die Regression aufgenommen. Ferner werden die Dummyvektoren der Branchen "1" und "3", also Chemical / Pharma und Engineering zusammengefaßt (siehe Abb. 4.14 (7)). Grund: alle diese Klassen zeigen empirische Logits, die stark von der gepunkteten Linie abweichen und daher starken Einfluß auf die Nullenwahrscheinlichkeit haben. Die Erstellung des Dummyvektors unter "R" wird exemplarisch für "breadth" gezeigt.

```
breadth.cut <-  
  cut(breadth, breaks = quantile(breadth, probs = c(0, 0.2, 0.4, 0.6, 0.8, 1)))  
breadth.k12 <- rep(0,856)  
breadth.k12[breadth.cut==levels(breadth.cut)[1]] <- 1  
breadth.k12[breadth.cut==levels(breadth.cut)[2]] <- 1
```

Zusammenfassung

Insgesamt werden folgende Kovariablen in die Regression für den Nullenüberschuß aufgenommen.

```
Z <- cbind(Intercept, breadth.k12, emp.k2, r.d.pat.k2, chem.pha.engin)
```

Variable	Klasse 1	Klasse 2	Klasse 3	Klasse 4	Klasse 5	Klasse 6	Klasse 7	Klasse 8
std.5.mean	Klasse = [0, 0.159] n = 172 emp. logit = -0.94	(0.159, 0.248] 171 -0.72	(0.248, 0.334] 171 -0.53	(0.334, 0.501] 171 -0.42	(0.501, 2] 171 -0.51			
breadth	[0.0666, 0.49] 172 -0.3	(0.49, 0.642] 215 -0.18	(0.642, 0.721] 127 -0.88	(0.721, 0.812] 171 -0.85	(0.812, 0.923] 171 -0.96			
emp	[0, 2020] 172 -0.48	(2020, 11320] 171 -0.21	(11320, 30249] 171 -0.72	(30249, 75322] 171 -0.83	(75322, 466938] 171 -0.9			
r.d.pat	[0, 0.226] 283 -0.15	(0.226, 3.36] 290 -0.63	(3.36, 465] 283 -0.24					
r.d.intens	[0, 1.03] 283 -0.16	(1.03, 6.34] 290 -0.37	(6.34, 216] 283 -0.48					
r.d.miss	0 602 0.24	1 254 -0.24						
branche	1 144 -0.51	2 176 -0.75	3 112 -0.48	4 88 -0.9	5 88 -0.9	7 248 -0.63		
time	1 107 -0.7	2 107 -0.75	3 107 -0.75	4 107 -0.86	5 107 -0.77	6 107 -0.83	7 107 -0.86	8 107 -0.77

Tabelle 4.4: Empirische Logits nach Formel 4.5 der Kovariablen zur Untersuchung des Nullenüberschuß

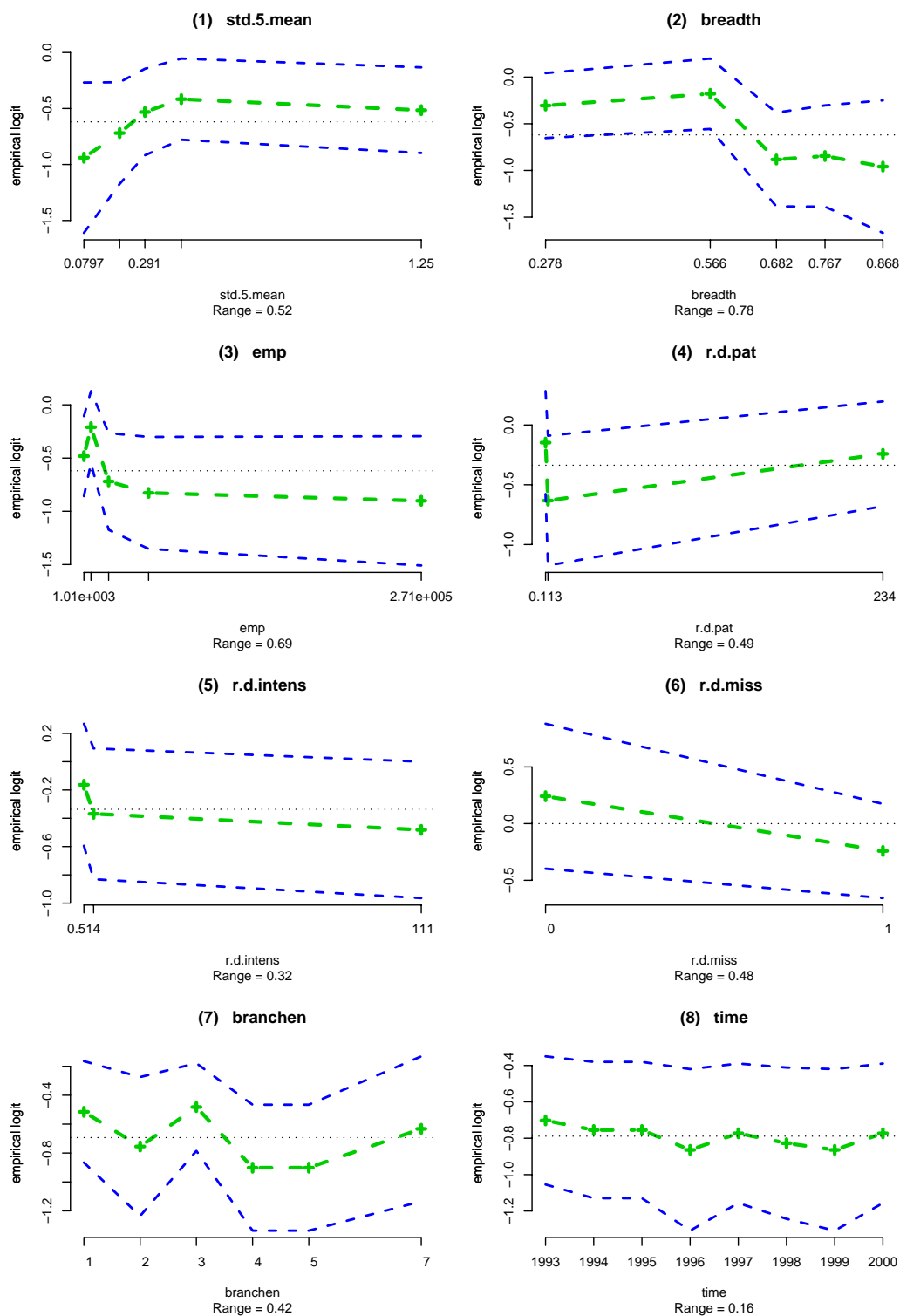


Abbildung 4.14: Geplottete empirische Logits der Kovariablen mit Konfidenzbändern (bei metrischen Kovariablen am Klassenmittelpunkt); die gestrichelte Linie zeigt jeweils den empirische Logit ohne Nullenüberschuß an

4.3 Vergleichskriterien für die Anpassungsgüte von Regressionsmodellen

Nun interessiert man sich dafür, wie man die Anpassungsgüte verschiedener Regressionsmodelle vergleichen kann. Für eingebettete Modelle (s. z.B. Lehmann und Romano [25, S. 301]) gibt es die Möglichkeit, dafür etwa den Partial Deviance – Test zu verwenden.

Definition 4.1 (Eingebettete Modelle).

Ein Regressionsmodell R (reduced) sei eingebettet (engl. "nested") in ein Modell F (full), wenn R bei geeigneter Parametrisierung von F in F enthalten ist. Dann gilt für die linearen Prädiktoren η_R von R und η_F von F für Matrizen $\mathbf{X}_R \in \mathbb{R}^{n \times p_1}$ und $\mathbf{X}_F \in \mathbb{R}^{n \times p_2}$

$$\begin{aligned}\eta_R &= \mathbf{1}\delta_0 + \mathbf{X}_R\boldsymbol{\delta}_R \\ \eta_F &= \mathbf{1}\delta_0 + \mathbf{X}_R\boldsymbol{\delta}_R + \mathbf{X}_F\boldsymbol{\delta}_F.\end{aligned}$$

Für genestete Modelle kann man den Partial Deviance – Test heranziehen. Für Modelle, die auf der verallgemeinerten Poissonverteilung oder einer ZIGP-Verteilung basieren, ergibt sich jedoch ein Problem: man kann die dafür benötigte Residual Deviance nicht geeignet berechnen. Die (unskalierte) Devianz (s. McCullagh und Nelder [26, S. 130]) berechnet man allgemein als

$$Dev := 2 \sum_{i=1}^n \left(l(y_i, y_i) - l(y_i, \mu_i) \right), \quad (4.6)$$

also mit der Differenz aus Likelihood mit maximalem Fit, der durch Einsetzen der Daten anstelle der Erwartungswerte entsteht, und dem Likelihood ausgewertet an μ_i . Speziell für die Poissonverteilung gilt

$$Dev := 2 \sum_{i=1}^n \left(y_i \log \left(\frac{y_i}{\mu_i} \right) - (y_i - \mu_i) \right).$$

Nimmt man jedoch einen Dispersionsparameter oder einen Zero Inflation - Parameter mit ins Modell auf, so stellt der erste Term in (4.6) nur noch bedingt den maximalen Fit dar: einsetzen kann man in den Loglikelihood lediglich die Schätzungen $\hat{\varphi}_i$ und $\hat{\omega}_i$. Die wahren Werte sind jedoch unbekannt.

$$\hat{Dev}_{ZIGP} := 2 \sum_{i=1}^n \left(l(y_i, y_i, \hat{\varphi}_i, \hat{\omega}_i) - l(y_i, \mu_i, \hat{\varphi}_i, \hat{\omega}_i) \right),$$

Damit stellt \hat{Dev}_{ZIGP} kein geeignetes Maß für Residual Deviance - Tests oder Partial Deviance - Tests dar.

Geht man nun vom Poisson-Modell aus und erweitert dieses um Dispersions- und Nullenüberschuß-Parameter, wobei diese wahlweise fest oder regressiert sein können, so erhält man neun verschiedene Modelle. Ein Vergleich dieser Modelle anhand der Patent-Daten soll zeigen, welche Erweiterung die Anpassungsgüte am stärksten erhöht. Im Abb. 4.15

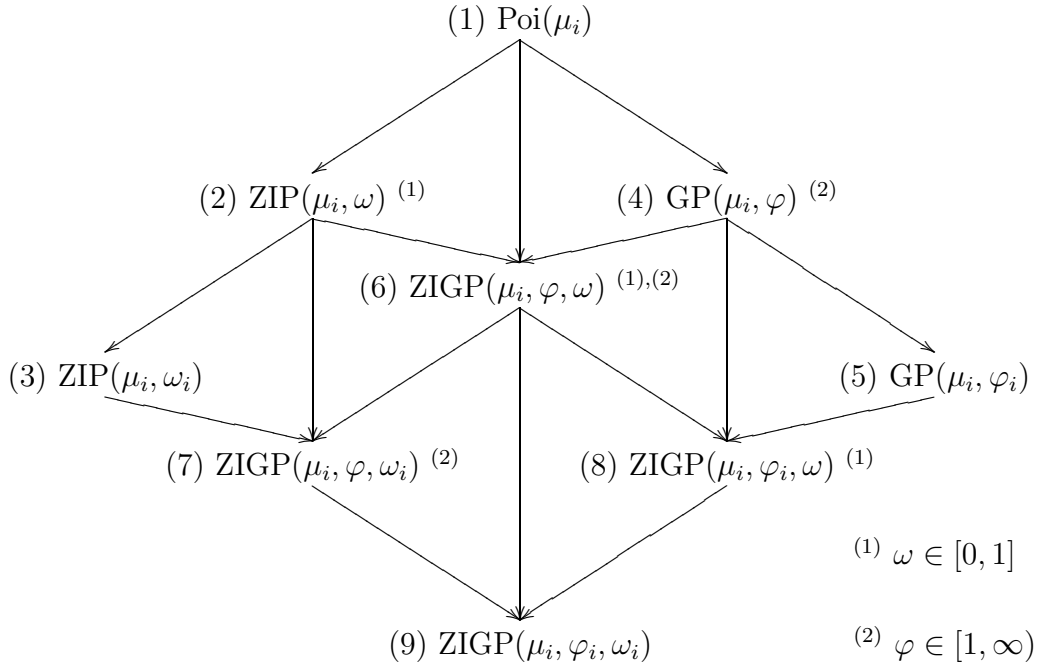


Abbildung 4.15: Struktogramm einer Evolution des Poisson-GLM's bis hin zu ZIGP-Modellen mit Regression auf allen Parametern

wird ein Struktogramm von Modellen verschiedener Entwicklungsstufen zwischen dem Poisson-GLM und dem $ZIGP(\mu_i, \varphi_i, \omega_i)$ -Regressionsmodell dargestellt.

Man stellt fest, daß viele dieser Modelle nicht *nested* sind. Dies hat zwei Gründe: zum einen enthält das reduzierte Regressionsmodell R oft einen oder mehrere Parameter des vollen Modells F nicht. Im Beispiel $R = Poi(\mu_i)$, $F = ZIP(\mu_i, \omega)$ enthält $Poi(\mu_i)$ keinen Nullenüberschuß-Parameter. Zwar gehört die Poisson-Verteilung zur Familie der ZIP-Verteilungen und kann daher als Spezialfall mit Parametrisierung $\omega = 0$ betrachtet werden. Definition 4.1 spricht aber von einer speziellen Parametrisierung der *Regressionsparameter*. Parameter ω ist bei $Poi(\mu_i)$ jedoch fest gewählt und wird bei der Modellanpassung nicht geschätzt. Ein weiteres Problem ist, daß Regressionsmodelle mit Linkfunktionen

$$\begin{aligned} \varphi_i &= 1 + e^{\mathbf{w}_i^t \boldsymbol{\alpha}} > 1 \\ \omega_i &= \frac{e^{\mathbf{z}_i^t \boldsymbol{\gamma}}}{1 + e^{\mathbf{z}_i^t \boldsymbol{\gamma}}} \quad \text{mit } \omega_i \in (0, 1) \end{aligned}$$

Werte $\varphi_i > 1$ und $\omega_i > 0$ haben. Somit ist z.B. Modell $R = GP(\mu_i, \varphi)$ mit $\varphi \in [1, \infty)$ nicht in $F = GP(\mu_i, \varphi_i)$ eingebettet. Man kann den Begriff von eingebetteten Modellen dahingehend erweitern, daß auch asymptotische Parametrisierung zugelassen wird, d.h. $R = GP(\mu_i, \varphi)$ ist in $F = GP(\mu_i, \varphi_i)$ eingebettet, wenn $\mathbf{z}_i^t \boldsymbol{\gamma} \rightarrow -\infty$. Dennoch bleibt im vorliegenden Fall das Problem bestehen, daß selbst beim Vergleich eingebetteter Modelle keine geeignete Devianzstatistik bekannt ist.

Modell (7) wurde in Kapitel 1, Modell (9) in Kapitel 2 entwickelt. Das Modell (1) ist eine "R"-Standardroutine für GLM's. Alle verbleibenden Modelle wurden durch Anpassung

der Software von Modell (9) erstellt und können auf der CD gefunden werden. Es müssen also Vergleichskriterien gefunden werden, die keine genesteten Modelle fordern. Daher wird mit Vuong-Tests und dem Akaike Informationskriterium gearbeitet.

4.3.1 Vuong - Test

Im Weiteren werden nach Vuong [37] und Clarke [3, S. 3f] Vuong - Tests zum Vergleich der Modelle vorgestellt. Ein Vuong - Test vergleicht zwei Modelle, die nicht notwendigerweise genestet sein müssen.

Das Kullback-Leibler Informationskriterium (Kullback und Leibler [23]) ist ein Maß für die Distanz zwischen zwei Statistischen Modellen. Vuong [37] definiert den *KLIC* als

$$KLIC := E_0[\log h_0(Y_i|\mathbf{x}_i)] - E_0[\log f(Y_i|\mathbf{x}_i, \hat{\boldsymbol{\delta}})],$$

wobei h_0 die wahre bedingte Wahrscheinlichkeitsfunktion von Y_i gegeben \mathbf{x}_i ist. Diese ist jedoch unbekannt. E_0 ist der Erwartungswert im wahren Modell und $\hat{\boldsymbol{\delta}}$ sind die Schätzungen von $\boldsymbol{\delta}$ im Modell mit $f(Y_i|\mathbf{x}_i, \hat{\boldsymbol{\delta}})$, das nicht das wahre Modell sein muß. Allgemein ist das beste Modell jenes, das den minimalen *KLIC* hat, da es am nächsten an der wahren (aber unbekannten) Parameterspezifikation liegt. Dieses Modell hat den maximalen Erwartungswert $E_0[\log f(Y_i|\mathbf{x}_i, \hat{\boldsymbol{\delta}})]$. Nun möchte man zwei Modelle vergleichen: seien zwei Modelle mit Wahrscheinlichkeitsfunktionen $f_1(Y_i|\mathbf{x}_i, \hat{\boldsymbol{\delta}}^1)$ und $f_2(Y_i|\mathbf{x}_i, \hat{\boldsymbol{\delta}}^2)$ gegeben, dann liegen sie beide gleich nahe an der wahren Parameterspezifikation, wenn ihre *KLIC*-Statistiken übereinstimmen, also

$$\begin{aligned} E_0[\log h_0(Y_i|\mathbf{x}_i)] - E_0[\log f_1(Y_i|\mathbf{x}_i, \hat{\boldsymbol{\delta}}^1)] &= E_0[\log h_0(Y_i|\mathbf{x}_i)] - E_0[\log f_2(Y_i|\mathbf{x}_i, \hat{\boldsymbol{\delta}}^2)] \\ \Leftrightarrow E_0 \log \frac{f_1(Y_i|\mathbf{x}_i, \hat{\boldsymbol{\delta}}^1)}{f_2(Y_i|\mathbf{x}_i, \hat{\boldsymbol{\delta}}^2)} &= 0. \end{aligned}$$

Liegt Modell 1 näher am wahren Modell als Modell 2, so ist der *KLIC* von Modell 1 geringer. Dann erhält man:

$$\begin{aligned} E_0[\log h_0(Y_i|\mathbf{x}_i)] - E_0[\log f_1(Y_i|\mathbf{x}_i, \hat{\boldsymbol{\delta}}^1)] &< E_0[\log h_0(Y_i|\mathbf{x}_i)] - E_0[\log f_2(Y_i|\mathbf{x}_i, \hat{\boldsymbol{\delta}}^2)] \\ \Leftrightarrow E_0 \log \frac{f_1(Y_i|\mathbf{x}_i, \hat{\boldsymbol{\delta}}^1)}{f_2(Y_i|\mathbf{x}_i, \hat{\boldsymbol{\delta}}^2)} &> 0 \end{aligned} \tag{4.7}$$

und im umgekehrten Fall

$$\begin{aligned} E_0[\log h_0(Y_i|\mathbf{x}_i)] - E_0[\log f_1(Y_i|\mathbf{x}_i, \hat{\boldsymbol{\delta}}^1)] &> E_0[\log h_0(Y_i|\mathbf{x}_i)] - E_0[\log f_2(Y_i|\mathbf{x}_i, \hat{\boldsymbol{\delta}}^2)] \\ \Leftrightarrow E_0 \log \frac{f_1(Y_i|\mathbf{x}_i, \hat{\boldsymbol{\delta}}^1)}{f_2(Y_i|\mathbf{x}_i, \hat{\boldsymbol{\delta}}^2)} &< 0. \end{aligned} \tag{4.8}$$

Mit anderen Worten, ein Modell sollte einem anderen vorgezogen werden, wenn seine Loglikelihood-Werte signifikant höher sind als die des anderen Modells.

Vuong definiert nun Statistiken

$$m_i := \log \left(\frac{f_1(y_i | \mathbf{x}_i, \hat{\boldsymbol{\delta}}^1)}{f_2(y_i | \mathbf{x}_i, \hat{\boldsymbol{\delta}}^2)} \right).$$

Dann ist $\mathbf{m} = (m_1, \dots, m_n)^t$ ein Vektor von Zufallsvariablen, der den Erwartungswert $\boldsymbol{\mu}_0^m = (\mu_1^m, \dots, \mu_n^m)^t := E_0(\mathbf{m})$ habe, wenn h_0 die wahre Wahrscheinlichkeitsfunktion ist. Die Nullhypothese lautet nun

$$H_0: \boldsymbol{\mu}_0^m = \mathbf{0}.$$

Anders gesagt: "Beide Modelle liegen gleich nahe an der wahren Parameterspezifikation." Der Erwartungswert $\boldsymbol{\mu}_0^m$ ist jedoch unbekannt, da die Verteilung der m_i unbekannt ist. Vuong [37, S. 312] zeigt daher, daß mit dem starken Gesetz der großen Zahlen gilt, daß

$$\frac{1}{n} LLR_n(\hat{\boldsymbol{\delta}}^1, \hat{\boldsymbol{\delta}}^2) \xrightarrow{a.s.} \boldsymbol{\mu}_0^m. \quad (4.9)$$

Dabei ist $LLR_n(\hat{\boldsymbol{\delta}}^1, \hat{\boldsymbol{\delta}}^2) := l_n^1(\hat{\boldsymbol{\delta}}^1) - l_n^2(\hat{\boldsymbol{\delta}}^2)$ die Log Likelihood-Ratio - Statistik der beiden Modelle. Man beachte, daß diese üblicherweise bei Likelihood-Ratio - Tests verwendet wird und dort in Zähler und Nenner dieselbe Wahrscheinlichkeitsfunktion enthält. Diese unterscheiden sich lediglich in der Parametrisierung. Im vorliegenden Fall unterscheiden sich jedoch die Wahrscheinlichkeitsfunktionen in Zähler und Nenner.

Formel (4.9) bedeutet, daß der Erwartungswert durch $\frac{1}{n}$ der Log Likelihood Ratio - Statistik geschätzt werden kann. Vuong zeigt ferner, daß durch geeignetes Normieren mit Hilfe des Zentralen Grenzwertsatzes gilt, daß unter H_0

$$\frac{LLR_n(\hat{\boldsymbol{\delta}}^1, \hat{\boldsymbol{\delta}}^2)}{\sqrt{n} \hat{\omega}_n} \xrightarrow{D} N(0, 1). \quad (4.10)$$

Hierbei schätzt $\sqrt{n} \hat{\omega}_n := \sqrt{\sum_{i=1}^n (m_i - \bar{m})^2}$ lediglich die Standardabweichung der Log Likelihood-Ratio - Statistik. Dazu überlegt man sich, daß man $LLR_n(\hat{\boldsymbol{\delta}}^1, \hat{\boldsymbol{\delta}}^2)$ zunächst als $\sum_{i=1}^n m_i$ schreiben kann, da

$$LLR_n(\hat{\boldsymbol{\delta}}^1, \hat{\boldsymbol{\delta}}^2) = \sum_{i=1}^n \left(\log(f_1(y_i | \mathbf{x}_i, \hat{\boldsymbol{\delta}}^1)) - \log(f_2(y_i | \mathbf{x}_i, \hat{\boldsymbol{\delta}}^2)) \right) = \sum_{i=1}^n m_i.$$

Ferner ist die empirische Varianz von $\sum_{i=1}^n m_i$ gegeben durch

$$\begin{aligned} \hat{\sigma}^2 \left(\sum_{i=1}^n m_i \right) &\stackrel{m_i \text{ iid}}{=} \sum_{i=1}^n \hat{\sigma}^2(m_i) \\ &= n \left(\frac{1}{n} \sum_{i=1}^n m_i^2 - \left[\frac{1}{n} \sum_{i=1}^n m_i \right]^2 \right) \\ &= n \left(\frac{1}{n} \sum_{i=1}^n (m_i - \bar{m})^2 \right) =: n \hat{\omega}_n^2 \end{aligned}$$

$$\Rightarrow \hat{\sigma} \left(\sum_{i=1}^n m_i \right) = \sqrt{n} \hat{\omega}_n.$$

Daher entspricht (4.10) unter H_0 schließlich

$$\nu := \frac{\sqrt{n} \left[\frac{1}{n} \sum_{i=1}^n m_i \right]}{\sqrt{\frac{1}{n} \sum_{i=1}^n (m_i - \bar{m})^2}} \xrightarrow{D} N(0, 1).$$

Zusammenfassend kann man den Vuong-Test unter Verwendung der eingeführten Notationen wie folgt definieren.

Definition 4.2 (Vuong-Test).

Sei

$$\begin{aligned} H_0: & \quad \boldsymbol{\mu}_0^m = \mathbf{0} \\ & \text{und} \\ H_1: & \quad \text{nicht } H_0. \end{aligned}$$

Verwerfe H_0 auf einem Signifikanzniveau von α , falls $|\nu| \geq z_{1-\frac{\alpha}{2}}$. Dabei ist $z_{1-\frac{\alpha}{2}}$ das $(1 - \frac{\alpha}{2})$ -Quantil der Standardnormalverteilung.

Man entscheidet sich für Modell 1, falls $\nu \geq z_{1-\frac{\alpha}{2}}$. Grund dafür ist, daß hohe Werte von ν nach Formel (4.7) für einen signifikant höheren *KLIC* von Modell 1 gegenüber Modell 2 sprechen. Schließlich entscheidet man sich für Modell 2, falls $\nu \leq -z_{1-\frac{\alpha}{2}}$. Man kann analog mit Formel (4.8) argumentieren.

4.3.2 Akaike Informationskriterium

Vor allem zur Variablenselektion wird oft das Akaike Informationskriterium (z.B. Heiberger und Holland [18, S. 572]) verwendet. Der AIC-Wert (Akaike Information Criterion) ist durch

$$AIC := -2 \, l(\hat{\boldsymbol{\delta}}) + 2k$$

gegeben. Dabei ist $\hat{\boldsymbol{\delta}}$ der Vektor der geschätzten Regressions-Parameter und k die Anzahl seiner Komponenten. Je kleiner das Akaike Informationskriterium ist, desto besser ist das Modell. Dies liegt einerseits darin, daß der Maximum-Likelihood - Schätzer den AIC-Wert minimiert, andererseits wird mit dem Strafterm $2k$ der Hinzunahme von Kovariablen eine Schranke gesetzt.

Mit jeder Kovariablen, die in ein bestehendes Modell aufgenommen wird, steigt dessen Genauigkeit. Der AIC-Wert sinkt jedoch nur dann, wenn die Steigerung der Anpassungsqualität zu einer um mindestens den Wert Eins erhöhten Log Likelihood - Funktion führt. Da die Zunahme der Log Likelihood - Funktion mit steigender Kovariablenzahl geringer wird, sinkt die Akaike-Information ausgehend vom Nullmodell im Allgemeinen zunächst, um ab einer gewissen Anzahl von Parametern wieder anzuwachsen.

4.4 Vergleich der Regressionsmodelle mit AIC–Werten und Vuong–Tests

Für die nachfolgende Tabelle 4.5 werden die neun Modelle nun verglichen. Der Vergleich von Regressionsmodellen findet auf zwei Ebenen statt: als "vollständige" Modelle werden die Modelle bezeichnet, in denen die Kovariablen im Umfang wie aus der explorativen Datenanalyse eingehen. Aus den vorangegangenen Abschnitten werden noch einmal die Designmatrizen zur Modellierung von Erwartungswert (X), Überdispersion (W) und Nullenüberschuß (Z) zusammengestellt.

```
X <- cbind(Intercept, ln.std.5.mean, breadth, sqrt.emp, r.d.pat.inv, r.d.intens.1,
           r.d.intens.2, r.d.intens.3, r.d.miss, chem.pha, elec.tel.other, time,
           ln.std.5.mean.breadth, chem.pha.ln.std.5.mean, chem.pha.sqrt.emp,
           r.d.miss.chem.pha)
W <- cbind(Intercept, engineer, car.suppl.other, med.biot, time, breadth.k145,
           emp.k12, r.d.intens.k12)
Z <- cbind(Intercept, breadth.k12, emp.k2, r.d.pat.k2, chem.pha.engin)
```

Nun müssen die Kovariablen aber nach dem Wald–Test nicht signifikant sein. Einige von ihnen sind in manchen der neun Modelle signifikant, in anderen nicht. Nicht-signifikante Variablen werden für jedes der neun Modelle sequentiell eliminiert. Variablen, die jedoch in einer signifikanten Interaktion enthalten sind, verbleiben im Modell. Siehe dazu die Modell-Outputs in Anhang D. Diese so entstehende zweite Vergleichsebene werde als "verkürzt" bezeichnet.

Oft bringt es große numerische Vorteile, metrische Variablen zu normieren. Dafür wurden die Kovariablen $\mathbf{X}_j := (x_{1j}, \dots, x_{nj})^t$ wie folgt normiert:

$$\mathbf{X}_j^* := \frac{(\mathbf{X}_j - \bar{\mathbf{X}}_j)}{\hat{\sigma}(\mathbf{X}_j)}, \quad \text{so daß} \quad (4.11)$$

$$\frac{1}{n-1} \sum_{j=1}^n \mathbf{X}_j^* = 1.$$

Dabei ergeben sich jedoch andere Koeffizientenschätzer als im unnormierten Fall. Auf die Normierung von Dummyvariablen wurde generell verzichtet.

Zu jedem Modell sind im ersten Block von Tabelle 4.5 (erste sechs Zeilen) zunächst für "vollständige", danach für "verkürzte" Modelle folgende Statistiken angegeben: Loglikelihood - Wert, dann die Anzahl der geschätzten Parameter sowie AIC-Wert.

Zudem sind die neun Modelle jeweils zeilenweise (I) und spaltenweise (II) angetragen. Die Einträge der so entstehenden unteren Dreiecksmatrix stellen die Ergebnisse des Vuong–Tests dar, wobei das Modell aus (I) mit dem aus (II) verglichen wird. Als Signifikanzniveau wurde dafür 5% gewählt. In den ersten beiden Zeilen jeder Zelle wird jeweils angegeben, wie die Vuong-Statistik lautet und welches Modell - ob (I), (II) oder keines - sie bevorzugt. Danach folgt der P-Wert dieser Teststatistik. In den letzten beiden Zeilen jeder Zelle wird analog für "verkürzte" Modelle vorgegangen.

Beispiel:

Der Block links oben in der unteren Dreiecksmatrix vergleicht Modell (I), also $ZIP(\mu_i, \omega)$ mit Modell (II), also $Poi(\mu_i)$. Werden dieselben Kovariablen für die Modellierung des Erwartungswertes verwendet, so lautet die Vuong-Statistik $\nu = 4.22$. Die Zeile "V: (I) 0.00" bedeutet, daß sich der Vuong-Test aufgrund der Test-Statistik für Modell (I), also das ZIP-Modell entscheidet. Der P-Wert beträgt 0.00. Eliminiert man jedoch nicht signifikante Werte in der Modellierung von Erwartungswert und Nullenüberschuß, so erhöht sich die Vuong-Statistik sogar leicht auf $\nu = 4.23$. Der Test entscheidet sich wieder für das ZIP-Modell und der P-Wert ist weiter 0.00.

Nun sollen die Ergebnisse ausgewertet werden. Dazu wird die Hinzunahme eines Dispersionsparameters, eines Nullenüberschußparameters und die Regression auf den Parametern untersucht.

Hinzunahme eines Nullenüberschußparameters

Die Hinzunahme eines Nullenüberschußparameters senkt den AIC-Wert. Vergleicht man etwa Modell (1) $Poi(\mu_i)$ mit Modell (2) $ZIP(\mu_i, \omega)$, so sinkt der AIC-Wert im Fall "vollständiger" Designmatrizen von 23 896 auf 19 183, das sind rund 20% weniger. Der Vuong-Test bevorzugt das ZIP-Modell mit hoch-signifikanter Vuongstatistik von $\nu = 4.2$.

Hinzunahme eines Dispersionsparameters

Die stärkste Verbesserung des AIC-Wertes ruft ein Dispersionsparameter hervor. So beträgt der AIC-Wert bei (1) $Poi(\mu_i)$ wie gehabt 23 896, bei (4) $GP(\mu_i, \varphi)$ bereits 6 861 - im "verkürzten" Fall 6 852. Das ist in beiden Fällen eine Abnahme von ca. 71%. Die Vuong-Statistiken sind in diesem Fall mit $\nu = 10.8$ sehr hoch.

Regression auf den Nullenüberschußparameter

Läßt man Regression auf den Nullenüberschußparameter zu, so senkt dies den AIC-Wert. Ein Beispiel dafür sind die Modelle (2) $ZIP(\mu_i, \omega)$ und (3) $ZIP(\mu_i, \omega_i)$. Hier sinkt im vollständigen Fall der AIC-Wert um rund 0.4% geringfügig von 19 183 auf 19 110. Ferner ergibt sich eine Vuong-Statistik von $\nu = 4.32$, was ebenfalls ein sehr hoher Wert ist (der P-Wert ist 0.00).

Regression auf den Dispersionsparameter

Dazu kann man Modell (4) $GP(\mu_i, \varphi)$ mit Modell (5) $GP(\mu_i, \varphi_i)$ vergleichen. Der AIC-Wert sinkt bei identischer Designmatrix für den Erwartungswert von 6 861 auf 6 754, rund. 1.5% weniger. Die Vuong-Statistik beträgt $\nu = 4.14$.

(II)	(1) $Poi(\mu_i)$	(2) $ZIP(\mu_i, \omega)$	(3) $ZIP(\mu_i, \omega_i)$	(4) $GP(\mu_i, \varphi)$	(5) $GP(\mu_i, \varphi_i)$	(6) $ZIGP(\mu_i, \varphi, \omega)$	(7) $ZIGP(\mu_i, \varphi, \omega_i)$	(8) $ZIGP(\mu_i, \varphi_i, \omega)$	(9) $ZIGP(\mu_i, \varphi_i, \omega_i)$
vollständig	Log Lkhd. = -11931.9 # Param. = 16 AIC = 23896	-9574.6 17 19183	-9533.8 21 19110	-3413.4 17 6861	-3353.1 24 6754	-3308.5 18 6653	-3289.5 22 6623	-3261.4 25 6573	-3239.2 29 6536
verkürzt	Log Lkhd. = -11931.9 # Param. = 16 AIC = 23896	-9574.6 17 19183	-9533.8 21 19110	-3416.1 10 6852	-3356.8 18 6750	-3308.6 14 6645	-3298.5 12 6621	-3269 15 6568	-3245.2 19 6528
(I)									
vollständig	$\nu = 4.2$ V: (I) 0.00								
(2) $ZIP(\mu_i, \omega)$ verkürzt	$\nu = 4.2$ V: (I) 0.00								
vollständig	$\nu = 4.27$ V: (I) 0.00		$\nu = 4.32$ V: (I) 0.00						
(3) $ZIP(\mu_i, \omega_i)$ verkürzt	$\nu = 4.27$ V: (I) 0.00		$\nu = 4.32$ V: (I) 0.00						
vollständig	$\nu = 10.8$ V: (I) 0.00		$\nu = 9.94$ V: (I) 0.00	$\nu = 9.88$ V: (I) 0.00					
(4) $GP(\mu_i, \varphi)$ verkürzt	$\nu = 10.8$ V: (I) 0.00		$\nu = 9.94$ V: (I) 0.00	$\nu = 9.88$ V: (I) 0.00					
vollständig	$\nu = 10.8$ V: (I) 0.00		$\nu = 10.1$ V: (I) 0.00	$\nu = 10$ V: (I) 0.00	$\nu = 4.14$ V: (I) 0.00				
(5) $GP(\mu_i, \varphi_i)$ verkürzt	$\nu = 10.8$ V: (I) 0.00		$\nu = 10.1$ V: (I) 0.00	$\nu = 9.99$ V: (I) 0.00	$\nu = 3.94$ V: (I) 0.00				
vollständig	$\nu = 10.8$ V: (I) 0.00		$\nu = 10.2$ V: (I) 0.00	$\nu = 10.2$ V: (I) 0.00	$\nu = 4.26$ V: (I) 0.00	$\nu = 2.02$ V: (I) 0.04			
(6) $ZIGP(\mu_i, \varphi, \omega)$ verkürzt	$\nu = 10.8$ V: (I) 0.00		$\nu = 10.3$ V: (I) 0.00	$\nu = 10.2$ V: (I) 0.00	$\nu = 4.21$ V: (I) 0.00	$\nu = 2.08$ V: (I) 0.04			
vollständig	$\nu = 10.8$ V: (I) 0.00		$\nu = 10.3$ V: (I) 0.00	$\nu = 10.2$ V: (I) 0.00	$\nu = 4.65$ V: (I) 0.00	$\nu = 2.66$ V: (I) 0.01	$\nu = 2.6$ V: (I) 0.01		
(7) $ZIGP(\mu_i, \varphi, \omega_i)$ verkürzt	$\nu = 10.8$ V: (I) 0.00		$\nu = 10.3$ V: (I) 0.00	$\nu = 10.2$ V: (I) 0.00	$\nu = 4.29$ V: (I) 0.00	$\nu = 2.32$ V: (I) 0.02	$\nu = 1.88$ V: none 0.06		
vollständig	$\nu = 10.8$ V: (I) 0.00		$\nu = 10.3$ V: (I) 0.00	$\nu = 10.2$ V: (I) 0.00	$\nu = 5.3$ V: (I) 0.00	$\nu = 4.28$ V: (I) 0.00	$\nu = 3.69$ V: (I) 0.00	$\nu = 1.92$ V: none 0.06	
(8) $ZIGP(\mu_i, \varphi_i, \omega)$ verkürzt	$\nu = 10.8$ V: (I) 0.00		$\nu = 10.3$ V: (I) 0.00	$\nu = 10.2$ V: (I) 0.00	$\nu = 5.15$ V: (I) 0.00	$\nu = 4.01$ V: (I) 0.00	$\nu = 3.1$ V: (I) 0.00	$\nu = 2.16$ V: (I) 0.03	
vollständig	$\nu = 10.9$ V: (I) 0.00		$\nu = 10.3$ V: (I) 0.00	$\nu = 10.3$ V: (I) 0.00	$\nu = 5.88$ V: (I) 0.00	$\nu = 5$ V: (I) 0.00	$\nu = 4.71$ V: (I) 0.00	$\nu = 3.96$ V: (I) 0.00	$\nu = 2.81$ V: (I) 0.00
(9) $ZIGP(\mu_i, \varphi_i, \omega_i)$ verkürzt	$\nu = 10.8$ V: (I) 0.00		$\nu = 10.3$ V: (I) 0.00	$\nu = 10.3$ V: (I) 0.00	$\nu = 5.69$ V: (I) 0.00	$\nu = 4.73$ V: (I) 0.00	$\nu = 4.28$ V: (I) 0.00	$\nu = 3.91$ V: (I) 0.00	$\nu = 2.9$ V: (I) 0.00

Tabelle 4.5: Vergleich der Modelle mit AIC-Statistiken und Vuong-Tests von (I) gegen (II)

Die Hinzunahme eines Dispersionsparameters bringt für den vorliegenden Datensatz deutliche Vorteile. Auch Regression auf die zusätzlichen Verteilungsparameter verbessern die Anpassungsqualität, doch auch hier überwiegt der Effekt beim Dispersionsparameter den beim Nullenüberschußparameter. "Verkürzte" Modelle haben erwartungsgemäß geringere AIC-Werte als "vollständige", da keine nicht-signifikanten Variablen mehr enthalten sind. Ein Fall, in dem der Vuong-Test unentschieden ist, die Nullhypothese also nicht verworfen werden kann, liegt beim Vergleich von Modell (7) $ZIGP(\mu_i, \varphi, \omega_i)$ mit Modell (8) $ZIGP(\mu_i, \varphi_i, \omega)$ vor. Hier ergibt sich im "vollständigen" Fall $\nu = 1.92$. Der P-Wert ist entsprechend 0.06. Dieses Ergebnis kann man so deuten, daß ein variabler Dispersionsparameter bei festem Nullenüberschußparameter keinen signifikanten Vorteil gegenüber einem Modell mit festem Dispersionsparameter und variablem Nullenüberschußparameter hat.

Insgesamt kann man jedoch feststellen, daß mit Modell (9) $ZIGP(\mu_i, \varphi_i, \omega_i)$ aus Kapitel 2 ein nach AIC-Wert und Vuong-Test "bestes" Modell vorliegt. Gegenüber dem $Poi(\mu_i)$ -Modell senkt sich der AIC-Wert durch die Erweiterung von 16 auf 29 Parametern (also 13 neue Kovariablen) von 23 896 auf 6 536. Das ist etwas mehr als ein Viertel. Zusätzlich kann man aber durch Elimination von nicht signifikanten Variablen in letzterem Modell sogar mit 19 Kovariablen auskommen und einen noch geringeren AIC-Wert von 6 528 erzielen. Damit läßt sich der AIC-Wert des Poisson-GLM's durch geeignete Modellanpassung um 73% senken.

4.5 Interpretation des Modells mit höchster Anpassungsgüte

Nun soll das nach Vuong-Test und AIC-Wert "beste" Modell (9), verkürzte Form, interpretiert werden. Eine Zusammenfassung dieses $ZIGP(\mu_i, \varphi_i, \omega_i)$ -Modells ist im Anhang D auf S. 254 angeführt. Man stellt fest, daß "r.d.miss" bei der sequentiellen Elimination nicht-signifikanter Effekte für dieses Modell eliminiert wurde. Auch bei Wagner [38, S. 27] ist diese Variable nicht signifikant. Dies ist ein sehr wünschenswertes Ergebnis, da sonst ein signifikanter systematischer Fehler im Modell enthalten wäre.

Eine Untersuchung des Einflusses einer ausgewählten Kovariablen auf Outsourcing-Quote, Überdispersion und Nullen-Wahrscheinlichkeit wird dadurch ermöglicht, daß alle übrigen Kovariablen, die in diese Größe eingehen, festgehalten werden. Bei metrischen Kovariablen werden sie durch ihren Modus oder Mittelwert ersetzt, bei kategorialen Variablen wird i.d.R. jede Kategorie einzeln untersucht. Modi werden hier benutzt, wenn sich Grafiken als links- oder rechtsschief erweisen. Auch für die Exposure "pat.cum.year" wird ihr Modus verwendet.

4.5.1 Einfluß der Kovariablen auf die Outsourcing-Quote

Da man sich weniger für eine Vorhersage absoluter Patentzahlen interessiert als vielmehr dafür, wie hoch eine prozentuale Outsourcing-Quote $\frac{E(Y_i)}{E_i}$ von Patenten ist, wird der Einfluß der Kovariablen auf den Quotienten aus Erwartungswert und Exposure untersucht. Um die Modi der metrischen Variablen zu bestimmen, wird mit Histogrammen und der

"R"-Funktion *density()* gearbeitet. Diese erzeugt mittels der Kernel-Regression (s. Myers et al. [29, S. 298f]) aus einem Histogramm eine approximative graphische Darstellung der Wahrscheinlichkeitsverteilung. Für die Regression wird eine Kernel-Funktion benötigt, für diese wird gemäß der Standardeinstellung in *density()* die Gaußfunktion verwendet. Der *density()*-Output wird in den Grafiken in Abb. 4.16 über den Histogrammen geplottet. Die Skala der Ordinatenachse zeigt lediglich die absoluten Häufigkeiten der Histogramme, auf die relativen Häufigkeiten wird verzichtet. Die senkrechte rote Linie zeigt jeweils den Modus an.

Da "ln.std.5.mean" und "breadth" im Modell interagieren, wird ein gemeinsames Histogramm (Abb. 4.17 (1)) erstellt. Eine Darstellung der Höhenlinien mit der Anzahl der Beobachtungen je Zelle ist in Abb. 4.17 (2) zu sehen. Dort wurden die beiden Zellen, die mit 36 die meisten Beobachtungen aufweisen, rot markiert. Eine 2D-Kernel-Dichte für zweidimensionale Kreuztabelle kann mit der "R"-Funktion *kde2d* aus der "library MASS" erzeugt werden (s. Abb. 4.17 (3)). Ihre Höhenlinien sowie der (approximative) Modus sind in Abb. 4.17 (4) zu sehen.

In Tabelle 4.6 werden die Modi aufgelistet. Der Modus der Exposure wird mit E^M bezeichnet, Modi von Kovariablen werden $t_k(x_j)^M$ genannt. Die Funktion $t_k(\cdot)$ steht dabei für die Transformationen, die gemäß Abschnitt 4.2.1, S. 98, auf die Variablen angewandt wurden. Beispiel: sei $x_3 = emp$, dann ist $t_3(x_3) = \sqrt{emp} = sqrt.emp$.

Variable	Modus
"pat.cum.year"	$E^M := 13.36$
"sqrt.emp"	$t_3(x_3)^M := 58.35$
"r.d.pat.inv"	$t_4(x_4)^M := 2.824$
"r.d.intens.1"	$t_5(x_5)^M := -0.0153$
"ln.std.5.mean.breadth"	$(t_1(x_1)^M, x_2^M) := (-1.000, 0.7769)$

Tabelle 4.6: Modi metrischer Variablen; bezieht sich auf transformierte, aber unnormierte Kovariablen

Der Einfluß auf die Outsourcing-Rate soll für *untransformierte* Kovariablen dargestellt werden, d.h. in der Form *vor* der Anwendung von Transformationen. Ferner muß beachtet werden, daß alle Kovariablen aus numerischen Gründen nach Formel (4.11), S. 122, normiert wurden (auch die Modi in Tabelle 4.6 stehen für unnormierte Werte). Da Transformationen wie die Anwendung des Logarithmus oder der Kehrwert positive Werte erfordern, wurden die Kovariablen bei der Modellanpassung zuerst transformiert und danach normiert. Um die Normierung a posteriori zu berücksichtigen, müssen Mittelwert und geschätzte Standardabweichung von $t_k(x_k)$ a priori, also vor der Normierung, abgespeichert werden. Sie werden als $\bar{t}_k(x_k)$ und $\hat{\sigma}(t_k(x_k))$ bezeichnet.

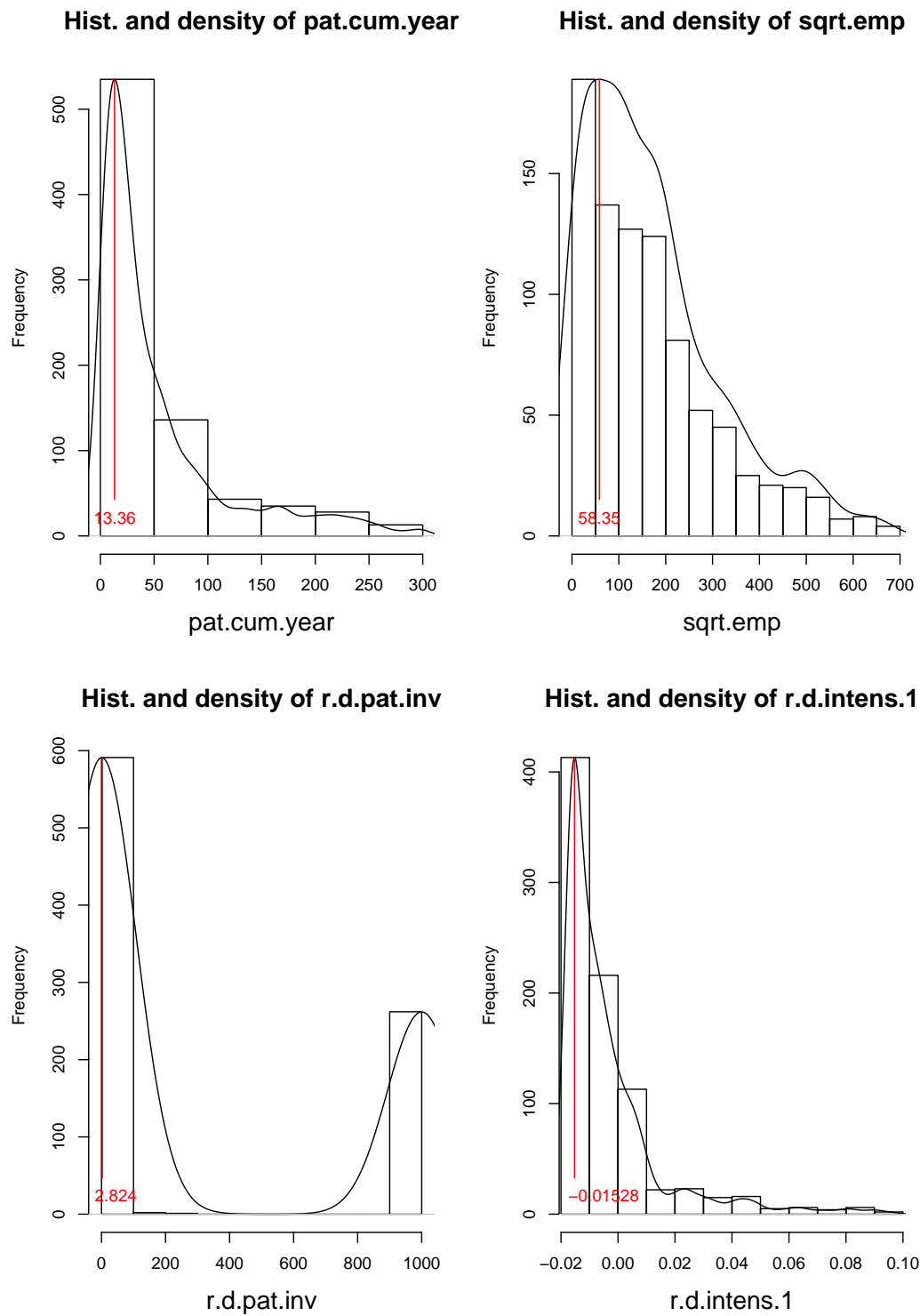
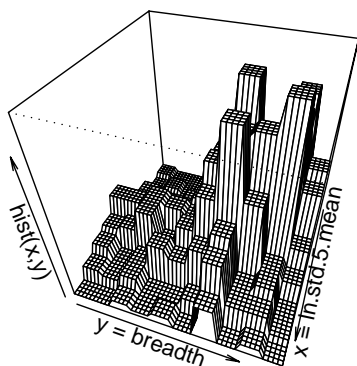
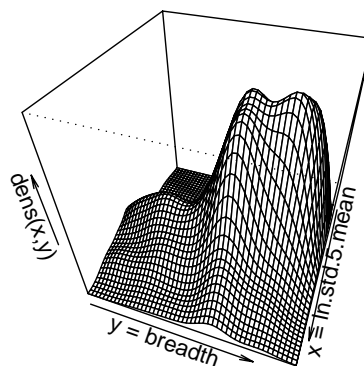


Abbildung 4.16: Modi der unnormierten metrischen Kovariablen, die in die μ -Regression eingehen und reine Haupteffekte sind.

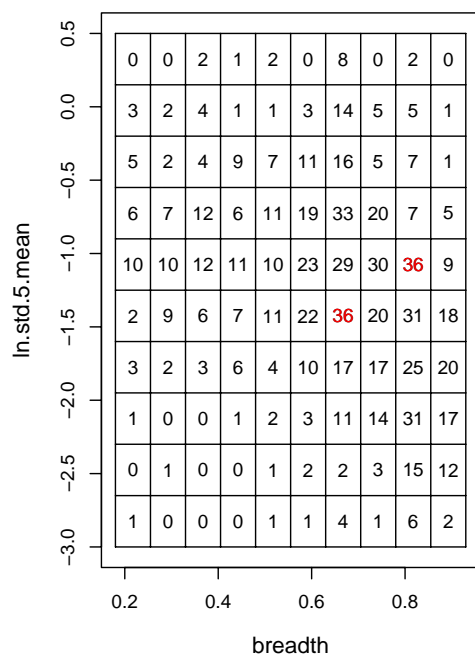
(1) Histogramm



(3) 2D – Kernel density



(2) Höhenlinien Histogramm



(4) Höhenlinien 2D-density

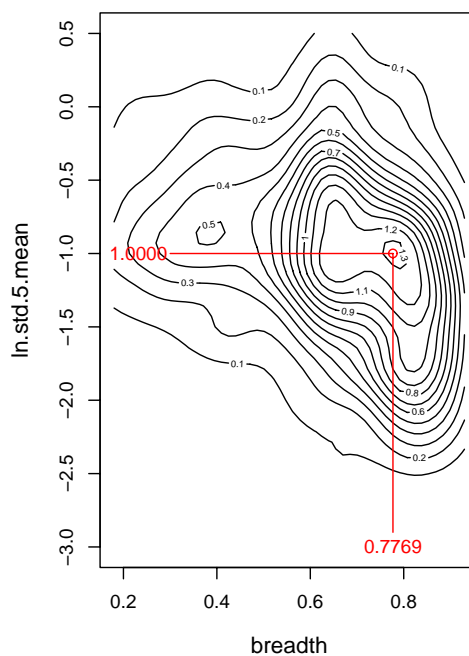


Abbildung 4.17: (1) - gemeinsames Histogramm von "ln.std.5.mean" und "breadth"; (2) - Anzahl der Beobachtungen je Zelle; (3) - 2D-Kernel-density; (4) - Höhenlinien mit Modus der Kernel-density.

Dann erhält man aufgrund des Log-Link für μ_i , daß

$$\begin{aligned} \frac{\hat{\mu}(x_k)}{E^M} &:= \frac{1}{E^M} \exp \left(\hat{\beta}_0 + \frac{t_1(x_1)^M - \bar{t}_1(x_1)}{\hat{\sigma}(t_1(x_1))} \cdot \hat{\beta}_1 + \dots + \frac{t_k(x_k) - \bar{t}_k(x_k)}{\hat{\sigma}(t_k(x_k))} \cdot \hat{\beta}_k \right. \\ &\quad \left. + \dots + \frac{t_7(x_7)^M - \bar{t}_7(x_7)}{\hat{\sigma}(t_7(x_7))} \cdot \hat{\beta}_7 + t_1(x_1)^M \cdot x_2^M \cdot \hat{\beta}_8 \right). \end{aligned}$$

Beispiel: diese Formel lautet für "emp" explizit:

$$\begin{aligned} \frac{\hat{\mu}(emp)}{E^M} &= \frac{1}{E^M} \exp \left(-0.951 + \frac{-1 - (-1.296)}{0.889} \cdot 0.031 + \frac{0.777 - 0.642}{0.191} \cdot 0.041 + \right. \\ &\quad \left. + \frac{\sqrt{emp} - 170.09}{145.22} \cdot (-0.394) + \dots \right. \\ &\quad \left. + \frac{-1 - (-1.296)}{0.889} \cdot \frac{0.777 - 0.642}{0.191} \cdot (-0.104) \right). \end{aligned}$$

In Abhängigkeit der interagierenden Variablen "std.5.mean" und "breadth" lautet sie

$$\begin{aligned} \frac{\hat{\mu}(std.5.mean, breadth)}{E^M} &= \frac{1}{E^M} \exp \left(-0.951 + \frac{\log(std.5.mean) - (-1.296)}{0.889} \cdot 0.031 \right. \\ &\quad \left. + \frac{breadth - 0.642}{0.191} \cdot 0.041 \right. \\ &\quad \left. + \frac{58.35 - 170.09}{145.22} \cdot (-0.394) + \dots \right. \\ &\quad \left. + \frac{\log(std.5.mean) - (-1.296)}{0.889} \cdot \frac{breadth - 0.642}{0.191} \right. \\ &\quad \left. \cdot (-0.104) \right). \end{aligned}$$

Da kategoriale Variablen "chem.pha" und "elec.tel.other" mit jeweils 2 Levels den Erwartungswert beeinflussen, müssen je 3 Settings untersucht werden. Die drei Fälle sind die Branchen Chemical / Pharma sowie Electro / Telecommunication / Other und alle verbleibenden Branchen. In den Abb. 4.18 (1)–(3) wird die vorhergesagte Outsourcing-Rate als Funktion der Kovariablen dargestellt.

Die Outsourcing-Raten bewegen sich etwa im Intervall [1.5%, 12%] (s. auch Abb. 4.19). Diese Prozentzahlen sind allerdings mit Vorsicht zu genießen, da sie sich auf eine fixe Exposure beziehen. Interessant ist eher ihre Relation der Quoten untereinander.

Die Branchen Electronics, Telecommunication & Other haben die höchste Outsourcing-Quote. Electro. / Telco. wurde bei Wagner [38] als Referenzkategorie verwendet, die Branche "Other" identifiziert jedoch auch er als die mit der höchsten Outsourcing-Rate. Auch für Chemical / Pharma stimmen die Ergebnisse überein, da er sie als diejenige Branche mit weitaus niedrigster Rate benennt. Gründe für die Branchenunterschiede werden später bei der Untersuchung der Branchendummies genannt.

Die Anzahl der Mitarbeiter (s. Abb. 4.18 (1)) hat hohen Einfluß: je geringer "emp", desto höher ist $\frac{\hat{\mu}(x_k)}{E^M}$. Kleinere Unternehmen leisten sich offenbar keine eigene Patentabteilung. Zu diesem Ergebnis kommt auch Wagner [38, S. 28].

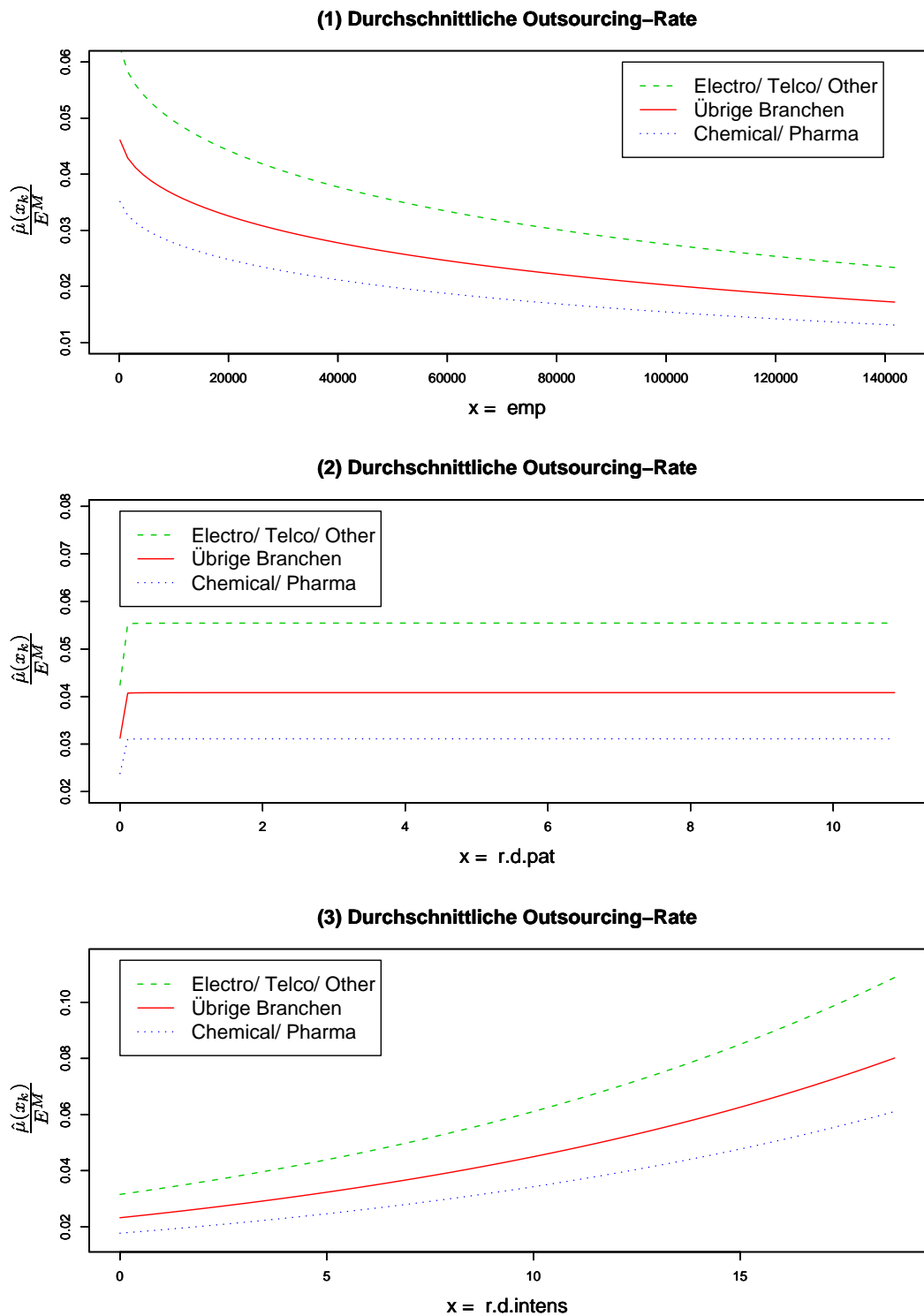


Abbildung 4.18: Einfluß von "emp" (1), "r.d.pat" (2) sowie "r.d.intens" (3) auf die Outsourcing-Quote $\frac{\hat{\mu}(x_k)}{E^M}$ unter Fixierung der übrigen Kovariablen durch ihre Modi

Die Graphik zu "r.d.pat" (s. Abb. 4.18 (2)) zeigt eine stark verzerrte Darstellung der Funktion $\hat{\beta}_4 \cdot (-\frac{1}{x_4})$. Diese Verzerrung hat ihre Ursache im betragsmäßig geringen Regressionskoeffizienten von nur $\hat{\beta}_4 = -0.123$. Offenbar hat es einen hohen Einfluß, wenn "r.d.pat" sehr klein ist. Man erinnere sich, daß 262 Beobachtungen (also etwa ein Drittel des Datenumfanges) einen Wert $r.d.pat = 0$ aufwiesen. Diese stimmt in 254 Fällen davon mit $r.d.miss = 1$ überein; es handelt sich also überwiegend um fehlende Angaben zu F+E – Ausgaben. Interessanterweise ist "r.d.pat" signifikant, "r.d.miss" aber nicht. Mit "r.d.miss" werden allerdings auch fehlende Daten bei "r.d.intens" herausgestellt. Zur Interpretation läßt sich sagen, daß diejenigen Unternehmen, die $r.d.pat = 0$ aufweisen, durchschnittlich lediglich 20000 Mitarbeiter haben, die durchschnittliche Mitarbeiterzahl aller beobachteten Unternehmen jedoch ca. 50000 beträgt. Kleinere Unternehmen haben demnach tendenziell nicht nur seltener eigene Patentabteilungen, sondern können auch weniger statistische Daten bereitstellen. Ob dem Patent teure Forschungsarbeit vorausgeht (hoher "r.d.pat"), spielt offenbar eine untergeordnete Rolle. Daher ist der Graph für größere Werte fast horizontal. Zu diesem Ergebnis kommt auch Wagner [38, S. 28].

Für "r.d.intens" (s. Abb. 4.18 (3)) stellt man fest, daß Unternehmen mit hoher Forschungsintensität ihre Patentanmeldungen bevorzugt selbst vornehmen. Wagner [38, S. 28] erhält aus seinem Modell ebenfalls einen positiven Einfluß dieser Variablen. Wagner bemerkt jedoch, daß intuitiv eher ein negativer Einfluß erklärbar ist: je höher die Forschungsausgaben pro Mitarbeiter, desto wahrscheinlicher ist es, daß dieses Unternehmen eigene Spezialisten in einer Patentabteilung beschäftigen. In diesen Unternehmen wird Forschung insgesamt größer geschrieben.

Nicht zu vergessen sind die kategorialen Variablen "chem.pha" und "elec.tel.other". Da sich für jede Kategorie nur eine punktuelle Vorhersage der Outsourcing-Rate ergibt, kann die Untersuchung anhand einer Tabelle vorgenommen werden.

Branche	Outsourcing-Quote
Electro / Telecommunication / Other	5.54%
verbleibende Branchen	4.08%
Chemical / Pharma	3.11%

Tabelle 4.7: Outsourcing-Rate je Branche unter Fixierung der übrigen Kovariablen durch ihre Modi

Dies bestätigt noch einmal die angesprochenen Branchenunterschiede. Die Branchen Chemical / Pharma haben die geringste erwartete Auslagerungsrate. Insbesondere in der Pharmaindustrie deckt ein Patent oft ein ganzes Produkt, z.B. ein Medikament, ab. Daß Pharmaunternehmen die Anmeldung selbst vornehmen, liegt daran, daß die Patente eine sehr hohe Bedeutung für sie haben, sie sind Teil ihres Geschäftsmodells. Teil jeder Anmeldung ist eine ausführliche Archiv-Recherche, ob ein ähnliches Patent bereits vergeben ist. Da Pharmaunternehmen darauf angewiesen sind, diese Recherche schnell und zuverlässig abzuwickeln, haben Unternehmen dieser Branche als erste eigene Patentdatenbanken entwickelt. Daher liegt es nahe, die gesamte Anmeldung von internen Abteilungen

abwickeln zu lassen. Eine ganz andere Rolle spielen Patente in den Branchen Electronics, Telecommunication & Other, in denen sehr viele Patente angemeldet werden. Besonders die Elektronik-Branche wird stark von Halbleiterfirmen dominiert, die sehr viele "kleine" Patente anmelden. Häufig schließen sich Unternehmen zusammen und schaffen einen neuen Produktstandard (ein einfaches Beispiel ist ein neues Stecksystem für PC's). Dies wird als *Kreuzlizensierung* bezeichnet. Sie führt dazu, daß bestehende Patente der einzelnen Unternehmen durch neue gemeinsame ersetzt werden. Diese Aufgabe übernehmen oft gemeinsame externe Patentanwälte.

$$\frac{\hat{\mu}(\text{std.5.mean}, \text{breadth})}{E^M}$$

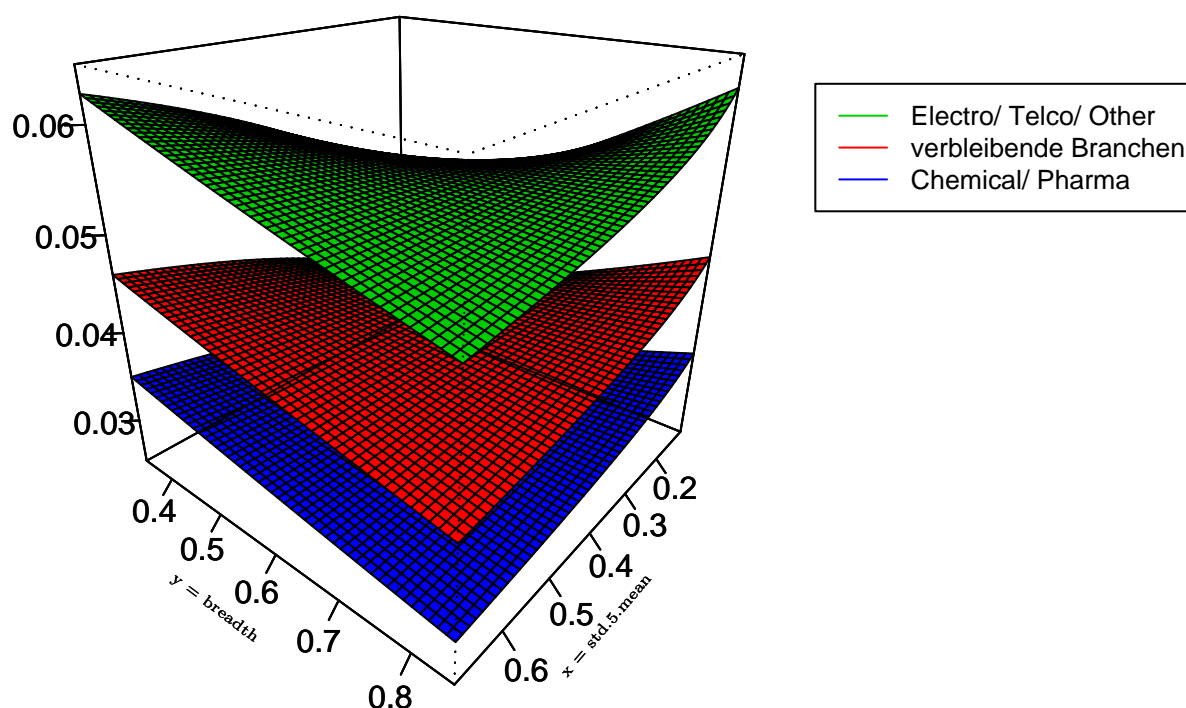


Abbildung 4.19: Einfluß von "std.5.mean" und "breadth" auf die geschätzte Outsourcing-Rate $\hat{\mu}(\text{std.5.mean}, \text{breadth})/E^M$ je Branche

Abb. 4.19 zeigt die vorhergesagte Outsourcing-Rate in Abhängigkeit von "std.5.mean" und "breadth" getrennt nach Branchenzugehörigkeit an. Die Krümmung kommt durch die Interaktion im Modell zustande. Je höher "breadth", desto mehr Patente werden von externen Anwälten angemeldet. Dasselbe gilt für "std.5.mean", wobei die Veränderung hier geringer ausfällt. Je höher die Forschungsbreite, desto eher lohnt es sich für Unternehmen, eigene Patentanwälte zu beschäftigen. In der Grafik "vorne", also für "breadth" bei hohem "std.5.mean" und für "std.5.mean" bei hohem "breadth" fällt die Änderung der Auslagerungsrate besonders gering aus. Dies paßt mit den geringen geschätzten Regressionskoeffizienten für "breadth" und "ln.std.5.mean" zusammen. Beide Kovariablen

sind nicht signifikant und verblieben nur aufgrund ihrer signifikanten Interaktion im Modell. Die angesprochenen Branchenunterschiede bestätigen sich auch hier. Die Interaktion scheint am höchsten zu sein, wenn "std.5.mean" und "breadth" gering sind. Dies korrespondiert zum negativen Vorzeichen des Regressionskoeffizienten der Interaktion. Geringe Forschungsbreite und geringe Streuung in den Anmeldungszahlen lassen besonders geringe Auslagerungsraten erwarten.

4.5.2 Einfluß der Kovariablen auf die Überdispersion

Nach Satz 1.6 ist das Verhältnis aus Varianz und Erwartungswert einer Zufallsvariablen $Y_i \sim ZIGP(\mu_i, \varphi_i, \omega_i)$ gegeben durch

$$\frac{Var(Y_i)}{E(Y_i)} = \varphi_i^2 + \mu_i \omega_i.$$

Um diesen Term als Funktion der für φ_i ausgewählten Kovariablen darzustellen, müssen zunächst μ_i und ω_i geschätzt werden. Im untersuchten Modell erweist sich ein Histogramm von $\hat{\mu}$ (also der *fitted means* aus Modell (9)) als unimodal und stark rechtsschief. Der Modus beträgt 7.55, der Mittelwert bereits 34.8. Im Folgenden soll mit dem Modus der μ_i , $\mu^M := 7.55$, gearbeitet werden.

Das Histogramm von $\hat{\omega}$ (fitted values von ω in Modell (9)) ist trimodal. Da einem Unternehmen eine A-Priori - Abschätzung seines Nullenüberschusses vor der Anpassung des Modells nicht bekannt sein dürfte, macht es in diesem Fall wenig Sinn, drei Fälle getrennt zu untersuchen. Daher wird hier der Mittelwert $\bar{\omega} := 0.070$ verwendet.

In die Regression gehen ausschließlich kategoriale Variablen ein, wobei "time" die einzige Kovariable ist, die normiert wird. Bei allen anderen handelt es sich um Dummies.

- $x_1 = engineer$
- $x_2 = car.suppl.other$
- $x_3 = time$
- $x_4 = emp.k12$
- $x_5 = r.d.intens.k12$

Die Schätzung $\hat{\varphi}(x_k)$ für das Beispiel $x_k := time$ hat aufgrund der modifizierten Log-Linkfunktion die Form

$$\begin{aligned} \hat{\varphi}(time) := & 1 + \exp\left(\hat{\alpha}_0 + x_1^M \cdot \hat{\alpha}_1 + \dots + \frac{time - \bar{time}}{\hat{\sigma}(time)} \cdot \hat{\alpha}_3 + \dots \right. \\ & \left. + x_5^M \cdot \hat{\alpha}_5\right). \end{aligned} \quad (4.12)$$

Damit kann man dann

$$\frac{Var(\hat{Y})}{E(\hat{Y})}(time) = \hat{\varphi}^2(time) + \mu^M \cdot \bar{\omega} \quad (4.13)$$

berechnen.

Sowohl "engineer" als auch "car.suppl.other" sind kategoriale Branchendummies. In diesem Fall wird jede Kategorie getrennt betrachtet. Soll das Jahr "time" fixiert werden, so wird von der Prämisse ausgegangen, daß nur der aktuellste Zeitpunkt der Daten interessant ist. Daher wird mit $time = 2000$ gearbeitet. Die Dummyvariable "emp.k12" kann man zurückführen auf "emp", dann gibt sie an, ob ein Unternehmen mehr ($= 0$) oder weniger ($= 1$) als 11291 Mitarbeiter hat. Analog kann man für "r.d.intens.k12" vorgehen: $r.d.intens.k12 = 0$ ist äquivalent zu $r.d.intens \geq 6.3$ (hohe Forschungsintensität); $r.d.intens.k12 = 1$ steht für $r.d.intens < 6.3$ (geringe Forschungsintensität).

Beispiel: man möchte den Einfluß der Branchenzugehörigkeit zu Cars, Suppliers & Other untersuchen. Die Dummyvariable "engineer" muß dann auf 0 gesetzt werden. Es werde ein kleines Unternehmen mit weniger als 11291 Mitarbeitern und geringer Forschungsintensität $r.d.intens < 6.3$ betrachtet. Dann erhält man mit Formeln (4.12) und (4.13)

$$\begin{aligned} \frac{\hat{Var}(Y)}{E(Y)}(car.suppl.other) &= 1 + \exp\left(1.968 + 0 \cdot (-0.426) \right. \\ &\quad \left. + car.suppl.other \cdot (-0.488) + \dots \right. \\ &\quad \left. + 1 \cdot (-0.305)\right) + 7.55 \cdot 0.070. \end{aligned}$$

Damit entstehen lediglich punktuelle Quotienten $\frac{\hat{Var}(Y)}{E(Y)}(x_k)$, diese werden in nachfolgender Tabelle für mögliche Kombinationen berechnet.

Branche	Mitarbeiter	Forschungsintensität	$\frac{\hat{Var}(Y)}{E(Y)}(x_k)$	# Fi.
Cars, Supplier & Other	$emp \geq 11291$	$r.d.intens \geq 6.344$	44.4	12
Cars, Supplier & Other	$emp \geq 11291$	$r.d.intens < 6.344$	27.0	14
Cars, Supplier & Other	$emp < 11291$	$r.d.intens \geq 6.344$	17.6	4
Cars, Supplier & Other	$emp < 11291$	$r.d.intens < 6.344$	11.5	12
Engineering	$emp \geq 11291$	$r.d.intens \geq 6.344$	49.3	0
Engineering	$emp \geq 11291$	$r.d.intens < 6.344$	29.8	6
Engineering	$emp < 11291$	$r.d.intens \geq 6.344$	19.2	1
Engineering	$emp < 11291$	$r.d.intens < 6.344$	12.4	7
verbleibende Branchen	$emp \geq 11291$	$r.d.intens \geq 6.344$	103.6	18
verbleibende Branchen	$emp \geq 11291$	$r.d.intens < 6.344$	60.6	17
verbleibende Branchen	$emp < 11291$	$r.d.intens \geq 6.344$	37.6	12
verbleibende Branchen	$emp < 11291$	$r.d.intens < 6.344$	23.1	4

Tabelle 4.8: $\frac{\hat{Var}(Y)}{E(Y)}(x_k)$ für das Jahr 2000

Die Quotienten bewegen sich zwischen 11 und 104, deuten also auf starke Überdispersion hin. Es läßt sich feststellen, daß eine hohe Mitarbeiterzahl den größten Einfluß auf den Quotienten hat: $\frac{\hat{Var}(Y)}{E(Y)}(x_k)$ steigt für Cars, Supplier & Other bei hoher Forschungsinten-

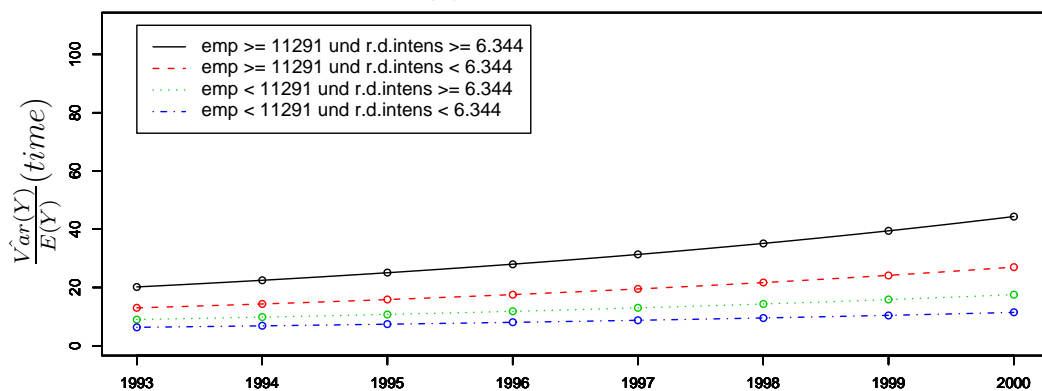
sität von 17.6 auf 44.4 Die Anzahl der Mitarbeiter ist ein Maß für die Größe des Unternehmens. Je höher das Unternehmen, desto mehr Patentanmeldungen erwartet man. Dies ließe wiederum eine Stabilisierung der Anmeldungszahlen und damit ein Absinken von $\frac{\hat{Var}(Y)}{E(Y)}(x_k)$ erwarten. Für hohe Mitarbeiterzahlen steigt der Quotient jedoch. Dies liegt daran, daß "kleine" Mitarbeiterzahlen sich in $[1; 11290]$ bewegen, "große" Zahlen jedoch in $[11291; 466900]$ liegen. Diese Tatsache läßt wiederum auf große Varianzen schließen. Auf "r.d.intens" läßt sich diese Argumentation übertragen: die Überdispersion steigt, je höher die Forschungsintensität ist. Ein Beispiel dafür ist die Branche Engineering bei hoher Mitarbeiterzahl. Dort steigt der Quotient von 29.8 auf 49.3. Hier stehen sich Intervalle für die Forschungsintensität von $[0; 6.344)$ und $[6.344; 215.17]$ Euro pro Mitarbeiter gegenüber.

Die Branchen Cars, Supplier and Other (erste vier Zeilen) haben die geringste Überdispersion, Engineering steht an zweiter Stelle. Alle verbleibenden Branchen (letzter Block) haben hohe Überdispersion. Für die Fahrzeugbranche kann man anführen, daß in der Entwicklung eines Produkts im Gegensatz etwa zur Pharmabranche sehr viele Patente stecken. Die Anzahl der angemeldeten Patente wird im Unternehmensmanagement oft als Stellschraube für die zukünftige Wettbewerbsfähigkeit der Produkte gewertet. Daher wird auch mit Zielvorgaben für die Anmeldezahlen gearbeitet. Diese zu erfüllen, fällt in der Autobranche leichter als etwa in der Pharmabranche, da Abteilungen notfalls auch kleine Neuerungen als Patent einreichen werden, um die Zielvorgabe zu erfüllen. Andererseits führt dies dazu, daß oft eine "Punktlandung" bei den Anmeldezahlen erreicht wird, und sei es nur, um die Anmeldezahlen des Vorjahres auch im laufenden Jahr zu erreichen. Damit sinkt jedoch die Varianz in den Anmeldezahlen.

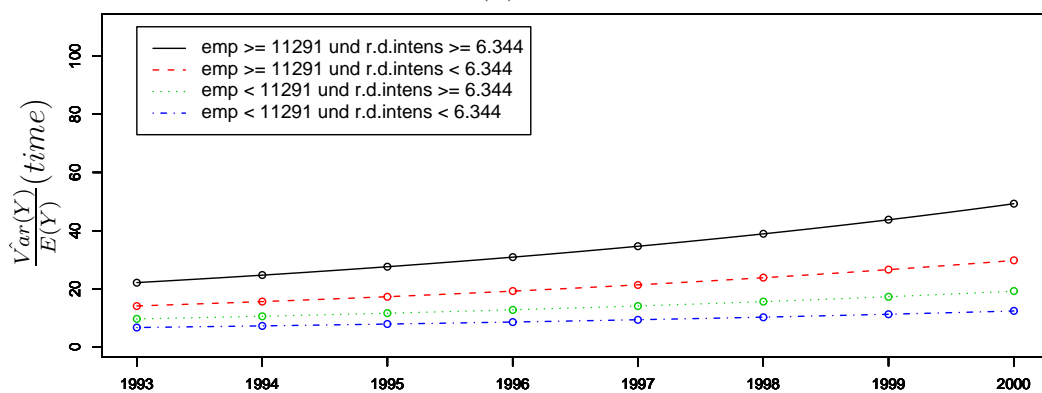
Abb. 4.20 stellt den Einfluß des Beobachtungsjahres auf den Variationskoeffizienten nach Branchen getrennt dar. Es ergibt sich stets ein positiver superlinearer Einfluß der Zeit, die die Ergebnisse aus der explorativen Datenanalyse bestätigt (s. Abb. 4.12 (2) auf S. 109). Die geringste Überdispersion läßt Cars / Suppliers / Other erwartet (Abb. 4.20 (1)).

Hier bewegt sich $\frac{\hat{Var}(Y)}{E(Y)}(time)$ etwa in $[8, 40]$, mittlere Werte etwa in $[8, 50]$ werden für Engineering (Abb. 4.20 (2)) vorhergesagt. Die höchste Überdispersion haben alle übrigen Branchen (Abb. 4.20 (3)) mit $\frac{\hat{Var}(Y)}{E(Y)}(time) \in [10, 110]$. Die möglichen Kombinationen aus geringer bzw. hoher Mitarbeiterzahl und Forschungsintensität liefern die bereits bekannten Ergebnisse: die höchste Überdispersion erhält man bei hohem "emp" und hohem "r.d.intens".

$$(1) \frac{\hat{Var}(Y)}{E(Y)}(time) \text{ (Cars / Supplier / Other)}$$



$$(2) \frac{\hat{Var}(Y)}{E(Y)}(time) \text{ (Engineering)}$$



$$(3) \frac{\hat{Var}(Y)}{E(Y)}(time) \text{ (verbleibende Branchen)}$$

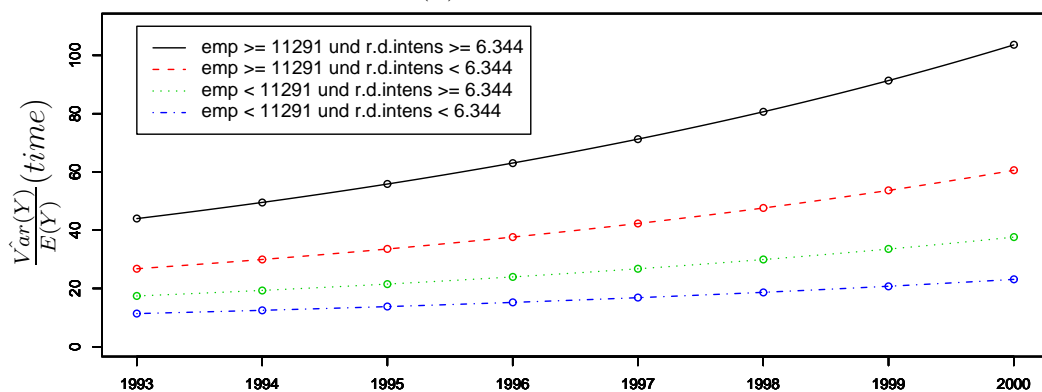


Abbildung 4.20: Einfluß von "time" auf den Variationskoeffizienten getrennt nach Branchen

4.5.3 Einfluß der Kovariablen auf die Nullen-Wahrscheinlichkeit

Für ein Unternehmen ist eine Vorhersage des Nullenüberschuß wenig aussagekräftig. Sie interessieren sich eher dafür, wie hoch denn die Nullenwahrscheinlichkeit $P(Y_i = 0)$ insgesamt ist. Die Nullenwahrscheinlichkeit ist die Wahrscheinlichkeit, daß ein Unternehmen in einem Jahr *keine einzige Patentanmeldung auslagert*.

Eine Null kann durch den Auswahlprozeß oder die GP-Variable entstehen, daher ist

$$P(Y_i = 0) = \omega_i + (1 - \omega_i) \cdot \exp\left(-\frac{\mu_i}{\varphi_i}\right).$$

Zur Schätzung von $P(Y_i = 0)$ wird μ_i wie bisher durch den Modus seiner *fitted means* $\mu^M := 7.55$ ersetzt. Das Histogramm von $\hat{\varphi}$, also der *fitted values* von φ in Modell (9), ist unimodal und rechtsschief. Daher wird mit dem Modus $\varphi^M := 3.56$ gearbeitet.

Der Nullenüberschuß ω_i wird als Funktion der Kovariablen x_k betrachtet. Als Kovariablen gehen nur die Dummyvariablen "breadth.k12", "r.d.pat.k2" und "chem.pha.engin" in den Nullenüberschuß ein. Diese wurden nicht normiert. Ganz allgemein gilt daher

$$\hat{\omega}(x_k) = \frac{\exp(\hat{\gamma}_0 + x_1^M \cdot \hat{\gamma}_1 + \dots + x_k \cdot \hat{\gamma}_k + \dots + x_q^M \cdot \hat{\gamma}_q)}{1 + \exp(\hat{\gamma}_0 + x_1^M \cdot \hat{\gamma}_1 + \dots + x_k \cdot \hat{\gamma}_k + \dots + x_q^M \cdot \hat{\gamma}_q)}.$$

Damit kann man die Nullenwahrscheinlichkeit als $\hat{P}(Y = 0 \mid x_k) := \hat{\omega}(x_k) + (1 - \hat{\omega}(x_k)) \exp\left(-\frac{\mu^M}{\varphi^M}\right)$ schätzen.

Nun ein konkretes Beispiel: man habe ein Unternehmen mit geringer technischer Forschungsbreite (*breadth* < 0.642). Dieses Unternehmen gehöre einer der Branchen Chemical/ Pharma/ Engineering an. Wie ist nun der Einfluß der Ausgaben für Forschung und Entwicklung pro Patent?

$$\hat{\omega}(r.d.pat) = \frac{\exp((-4.282) + 1 \cdot 2.271 + r.d.pat \cdot (-1.241) + 1 \cdot 1.085)}{1 + \exp((-4.282) + 1 \cdot 2.271 + r.d.pat \cdot (-1.241) + 1 \cdot 1.085)},$$

und

$$\hat{P}(Y = 0 \mid r.d.pat) = \hat{\omega}(r.d.pat) + (1 - \hat{\omega}(r.d.pat)) \exp\left(-\frac{7.55}{3.56}\right).$$

In Tabelle 4.9 sind die Nullen-Wahrscheinlichkeiten für mögliche Kovariablen-Kombinationen gegeben.

Die Wahrscheinlichkeit einer Null bewegt sich zwischen 13% und 37%. Einen großen positiven Einfluß auf $\hat{P}(Y = 0)$ hat ein kleiner "breadth". Je geringer die technische Breite der angemeldeten Patente ist, desto höher ist die Wahrscheinlichkeit, daß gar kein Patent ausgelagert wurde (d.h. alle Patente wurden selbst angemeldet!). Ein Beispiel dafür ist für "verbleibende Branchen", $r.d.pat \notin (0.233, 3.353]$. Dort erhöht sich $\hat{P}(Y = 0)$ für sinkenden "breadth" von 13.2 % auf 22.4 %. Dies macht Sinn, denn je geringer die Forschungsbreite, desto eher lassen sich Patentanmeldungen über einen eigenen Spezialisten für das Forschungsgebiet erfassen. Forscht das Unternehmen jedoch auf vielen Gebieten, so muß es viele Spezialisten beschäftigen. In diesem Fall ist eine Auslagerung oft wirtschaftlicher.

Branche	Forschungsko. / Patent	Forschungsbreite	$\hat{P}(Y = 0)$	# Fi.
Chem./ Phar. & Enging.	$r.d.pat \notin (0.233, 3.353]$	$breadth \geq 0.642$	15.5%	9
Chem./ Phar. & Enging.	$r.d.pat \notin (0.233, 3.353]$	$breadth < 0.642$	37.0%	12
Chem./ Phar. & Enging.	$r.d.pat \in (0.233, 3.353]$	$breadth \geq 0.642$	13.1%	7
Chem./ Phar. & Enging.	$r.d.pat \in (0.233, 3.353]$	$breadth < 0.642$	21.1%	4
verbleibende Branchen	$r.d.pat \notin (0.233, 3.353]$	$breadth \geq 0.642$	13.2%	22
verbleibende Branchen	$r.d.pat \notin (0.233, 3.353]$	$breadth < 0.642$	22.4%	27
verbleibende Branchen	$r.d.pat \in (0.233, 3.353]$	$breadth \geq 0.642$	12.4%	18
verbleibende Branchen	$r.d.pat \in (0.233, 3.353]$	$breadth < 0.642$	15.3%	8

Tabelle 4.9: Funktionswerte $\hat{\omega}(x_k) + (1 - \hat{\omega}(x_k)) \cdot \exp\left(-\frac{\mu^M}{\varphi^M}\right)$

Die Branchen Chemical, Pharma und Engineering lassen eine hohe Nullenwahrscheinlichkeit erwarten. So beträgt sie z.B. für $r.d.pat \in (0.233, 3.353]$ und $breadth < 0.642$ $\hat{P}(Y = 0) = 21.1\%$, für alle verbleibenden Branchen nur 15.3% . Ein Grund dafür ist, daß insbesondere Pharmaunternehmen eigene Patentdatenbanken für die Patentrecherche entwickelt haben und daher ihre Patente häufiger ausschließlich selbst anmelden. Für $r.d.pat \notin (0.233, 3.353]$ ergibt sich gegenüber $r.d.pat \in (0.233, 3.353]$ eine höhere vorhergesagte Nullenwahrscheinlichkeit. Ein Beispiel sind Unternehmen aus Chemical, Pharma und Engineering mit $breadth \geq 0.642$. Dort steigt $\hat{P}(Y = 0)$ von 13.1% auf 15.5% . Offenbar gibt es bezüglich "r.d.pat" zwei gegenläufige Effekte: zunächst ist für sehr kleine "r.d.pat" eine erhöhte Nullenwahrscheinlichkeit festzustellen. Unternehmen mit Beobachtungen $r.d.pat_i = 0$ haben jedoch wie angesprochen überwiegend gar keine Angaben zu Forschungsausgaben gemacht. Eine Interpretation dieser Werte ist demnach schwer, da keine spezifischen Merkmale dieser Unternehmen bekannt sind. Geht Patenten aber teure Forschungsarbeit voraus (hoher "r.d.pat"), so melden Unternehmen diese Patente bevorzugt alle selbst an. Offenbar vertraut man internen Abteilungen eher, diese komplexen Patente fehlerfrei anzumelden.

Kapitel 5

Zusammenfassung

In Kapitel 1 wurde gezeigt, daß gewöhnliche Poisson-Modelle bei hoher Überdispersion oder Nullenüberschuß oft unzureichend sind. Die Hinzunahme eines Dispersionsparameters in einer $GP(\mu, \varphi)$ -Verteilung führt zu einem $GP(\mu_i, \varphi)$ -Regressionsmodell. Doch auch ein Zero Inflation – Parameter kann sinnvoll sein. Dazu wurden Zero-Inflated-Verteilungen eingeführt. Auch Mixture-Ansätze mit Zero Inflated Generalized Poisson-Modellen sind bekannt. Alle verwendeten Verteilungen gehören jedoch nicht zur Familie der exponentiellen Verteilungen, damit lassen sich nur Regressionsmodelle konstruieren, die keine GLM's sind. Standardtheorie über GLM's ist damit nicht anwendbar. Neu ist jedoch, auch Regression auf den Nullenüberschuß-Parameter zuzulassen. In allen entwickelten Regressionsmodellen ist jedoch nur der häufige Fall der Überdispersion abbildbar. Nun kann man ein $ZIGP(\mu_i, \varphi, \omega_i)$ -Regressionsmodell konstruieren. Zu diesem Modell wurden die für Datenanpassung und Modell-Beurteilung notwendigen Funktionen eingeführt, die in eine Software einfließen (sie ist in Anhang A erklärt). Die Datenanpassung geschieht mittels Maximum Likelihood-Schätzern. Da aber nicht bewiesen werden kann, daß die Fisher-Information stets positiv definit ist, müssen ML-Schätzer nicht existieren und eindeutig sein. Für die im Rahmen dieser Arbeit betrachteten Daten ergab sich an der Stelle des ermittelten Schätzwertes allerdings stets negative Definitheit der beobachteten Fisher-Information, was zumindest auf die Existenz der ML-Schätzer hindeutet. Im Abschnitt über Konsistenz und Konvergenz wurde jedoch mit Annahme (A1) stets vorausgesetzt, daß die Fisher-Information positiv definit ist. In einer Simulationsstudie wurde die Konvergenzgeschwindigkeit einzelner Regressionsparameter anhand von empirischer Verzerrung, gemittelten MSE und QQ-Plots verglichen. Man kann die asymptotische Normalität und weitere Eigenschaften zur Asymptotik der Parameter unter bestimmten Voraussetzungen beweisen.

Der Beweis folgte in Kapitel 2. In diesem Kapitel wurde zunächst das Regressionsmodell um die Regression auf den Dispersionsparameter φ erweitert. Analog zu Kapitel 1 wurde eine Software (Anhang B) erstellt und eine Simulationsstudie durchgeführt. Ferner wurde bewiesen, daß das Regressionsmodell (und damit auch das aus Kapitel 1) für kompakte Regressoren konsistent und konvergent ist. Diese Eigenschaften umfassen auch die asymptotische Normalität der ML-Schätzer.

Kapitel 3 befaßte sich mit Zero Modified-Modellen. Oft möchte man testen, ob ein Nullenüberschuß in den Daten vorliegt. Dies ist aufgrund des Logit-Links bei der Regression

auf den Nullenüberschuß-Parameter nicht möglich, da $\omega = 0$ außerhalb des Parameter-raums liegt. Es wurde exemplarisch vorgeführt, wie man das $ZIGP(\mu_i, \varphi, \omega_i)$ -Modell aus Kapitel 1 zu einem Zero Modified-Modell $ZMGP(\mu_i, \varphi, \omega_i)$ -Modell erweitern kann. Dazu muß man beschränkte Erwartungswerte annehmen. Diese Modelle ermöglichen dann Tests der Form $H_0 : \omega = 0$ gegen $H_1 : \text{"nicht } H_0\text{"}$ oder bei Regression auf ω Tests auf Nullenüberschuß in verschiedenen Beobachtungsgruppen.

In Kapitel 4 wurde ein Patent-Datensatz untersucht. In einer explorativen Datenanalyse wurden Variablen für die Regression auf Erwartungswert, Dispersionsparameter und Nullenüberschuß ausgewählt und Transformationen vorgenommen. Danach wurden alle Modelle angefangen beim Poisson-GLM bis hin zum $ZIGP(\mu_i, \varphi_i, \omega_i)$ -Regressionsmodell aus Kapitel 2 verglichen. Es zeigte sich bereits eine Verbesserung der Anpassungsgüte bei der Hinzunahme eines Nullenüberschußparameters, eine deutliche Verbesserung aber durch einen Dispersionsparameter. Dies deutete darauf hin, daß zwar ein Nullenüberschuß, hauptsächlich jedoch hohe Überdispersion in den Daten vorlag. Regression auf diese Parameter verbesserte die Ergebnisse noch einmal. So konnte man feststellen, daß beim Vergleich von $Poi(\mu_i)$ - und $ZIGP(\mu_i, \varphi_i, \omega_i)$ -Modellen durch die Erweiterung von 16 auf 29 Parametern (also 13 neue Kovariablen) der AIC-Wert von 23896 auf 6536 sank. Zusätzlich erhielt man aber durch Elimination von nicht signifikanten Variablen in letzterem Modell sogar mit 19 Kovariablen einen noch geringeren AIC-Wert von 6528. Zuletzt wurde eine Interpretation des Modells mit geringstem AIC-Wert vorgenommen. Mitarbeiterzahl und Branche hatten den größten Einfluß auf Erwartungswert und Dispersionsparameter. Für die Nullenwahrscheinlichkeit spielte besonders die technische Forschungsbreite eine große Rolle.

Abbildungsverzeichnis

1.1	QQ-Plots zentrierter und normierter ML Schätzer basierend auf $N = 100$ Datensätzen im $ZIGP(\mu_i, \varphi, \omega_i)$ -Modell mit $\mu_i \in [2, 5]$ und $\omega_i \in [0.1, 0.25]$.	31
1.2	QQ-Plots zentrierter und normierter ML Schätzer basierend auf $N = 100$ Datensätzen im $ZIGP(\mu_i, \varphi, \omega_i)$ -Modell mit $\mu_i \in [2, 5]$ und $\omega_i \in [0.1, 0.5]$.	32
1.3	QQ-Plots zentrierter und normierter ML Schätzer basierend auf $N = 100$ Datensätzen im $ZIGP(\mu_i, \varphi, \omega_i)$ -Modell mit $\mu_i \in [2, 15]$ und $\omega_i \in [0.1, 0.25]$.	33
1.4	QQ-Plots zentrierter und normierter ML Schätzer basierend auf $N = 100$ Datensätzen im $ZIGP(\mu_i, \varphi, \omega_i)$ -Modell mit $\mu_i \in [2, 15]$ und $\omega_i \in [0.1, 0.5]$.	34
2.1	QQ-Plots zentrierter und normierter ML Schätzer basierend auf $N = 100$ Datensätzen im $ZIGP(\mu_i, \varphi_i, \omega_i)$ -Modell mit $\mu_i \in [2, 5]$ und $\omega_i \in [0.1, 0.25]$.	65
2.2	QQ-Plots zentrierter und normierter ML Schätzer basierend auf $N = 100$ Datensätzen im $ZIGP(\mu_i, \varphi_i, \omega_i)$ -Modell mit $\mu_i \in [2, 5]$ und $\omega_i \in [0.1, 0.5]$.	66
2.3	QQ-Plots zentrierter und normierter ML Schätzer basierend auf $N = 100$ Datensätzen im $ZIGP(\mu_i, \varphi_i, \omega_i)$ -Modell mit $\mu_i \in [2, 15]$ und $\omega_i \in [0.1, 0.25]$.	67
2.4	QQ-Plots zentrierter und normierter ML Schätzer basierend auf $N = 100$ Datensätzen im $ZIGP(\mu_i, \varphi_i, \omega_i)$ -Modell mit $\mu_i \in [2, 15]$ und $\omega_i \in [0.1, 0.5]$.	68
3.1	Linkfunktion für Zero Modified Modelle mit $\kappa_\omega^u(\kappa_\mu^o) = -0.1$	71
3.2	QQ-Plots zentrierter und normierter ML Schätzer basierend auf $N = 100$ Datensätzen im $ZMGP(\mu_i, \varphi, \omega_i)$ -Modell mit $\mu_i \in [2, 5]$ und $\omega_i \in [0.1, 0.25]$.	82
3.3	QQ-Plots zentrierter und normierter ML Schätzer basierend auf $N = 100$ Datensätzen im $ZMGP(\mu_i, \varphi, \omega_i)$ -Modell mit $\mu_i \in [2, 5]$ und $\omega_i \in [0.1, 0.5]$.	83
3.4	QQ-Plots zentrierter und normierter ML Schätzer basierend auf $N = 100$ Datensätzen im $ZMGP(\mu_i, \varphi, \omega_i)$ -Modell mit $\mu_i \in [2, 15]$ und $\omega_i \in [0.1, 0.25]$.	84
3.5	QQ-Plots zentrierter und normierter ML Schätzer basierend auf $N = 100$ Datensätzen im $ZMGP(\mu_i, \varphi, \omega_i)$ -Modell mit $\mu_i \in [2, 15]$ und $\omega_i \in [0.1, 0.5]$.	85
4.1	(1) – Histogramm des Response; (2) – Histogramm des Quotientenvektors \mathbf{Q} , $Q_i := \frac{Y_i}{E_i}$; (3)-(10) – Boxplots der metrischen Kovariablen, $i = 1, \dots, 856$.	90
4.2	Plots $\ln(Y_i + 0.001) - \ln(E_i)$ gegen $x_{i1} = \text{std.5.mean}_i$; mit <i>lowess()</i> geglättete Daten für alle Daten (blau gestrichelt), Daten $Y_i = 0$ (lila) und $Y_i > 0$ (rot); Transformationen für Datenausschnitte mit $Y_i = 0$, Regressionslinien in schwarz	93

4.3	Plots $\ln(Y_i + 0.001) - \ln(E_i)$ gegen $x_{i2} = \text{breadth}_i$; mit <i>lowess()</i> geglättete Daten für alle Daten (blau gestrichelt), Daten $Y_i = 0$ (lila) und $Y_i > 0$ (rot); Transformationen für Datenausschnitte mit $Y_i = 0$, Regressionslinien in schwarz	94
4.4	Plots $\ln(Y_i + 0.001) - \ln(E_i)$ gegen $x_{i3} = \text{emp}_i$; mit <i>lowess()</i> geglättete Daten für alle Daten (blau gestrichelt), Daten $Y_i = 0$ (lila) und $Y_i > 0$ (rot); Transformationen für Datenausschnitte mit $Y_i = 0$, Regressionslinien in schwarz	95
4.5	Plots $\ln(Y_i + 0.001) - \ln(E_i)$ gegen $x_{i4} = \text{r.d.pat}_i$; mit <i>lowess()</i> geglättete Daten für alle Daten (blau gestrichelt), Daten $Y_i = 0$ (lila) und $Y_i > 0$ (rot); Transformationen für Datenausschnitte mit $Y_i = 0$, Regressionslinien in schwarz	96
4.6	Plots $\ln(Y_i + 0.001) - \ln(E_i)$ gegen $x_{i5} = \text{r.d.intens}_i$; mit <i>lowess()</i> geglättete Daten für alle Daten (blau gestrichelt), Daten $Y_i = 0$ (lila) und $Y_i > 0$ (rot); Transformationen für Datenausschnitte mit $Y_i = 0$, Regressionslinien in schwarz	97
4.7	Empirische Log Means der Kovariablen mit "pat.cum.year" als Exposure .	99
4.8	Profile der logarithmierten Anteile ausgelagerter Patente nach Branchen getrennt; Branchen-Mediane als durchgezogene schwarze Linien	103
4.9	Profile der Branchenmediane im Zeitverlauf (1); zusammengefaßte Branchen und zeitunabhängige Mediane über alle Jahre als horizontale Linien (2)	104
4.10	Pairs-Plot der ausgewählter Haupteffekte mit <i>lowess()</i> -Glättungslinien .	105
4.11	Empirische Log Means zur Untersuchung von zweidimensionalen Interaktionseffekten	106
4.12	Funktionswerte $f_{j,t}(\mathbf{Y})$ für $t = 1993, \dots, 2000$ (1); Branchen "4" und "7" zusammengefaßt zu "Cars + Supplier & Other" (2)	109
4.13	Überdispersionsfunktion $f_{l(i)}(\mathbf{Y})$ je Klasse der Kovariablen	112
4.14	Geplottete empirische Logits der Kovariablen mit Konfidenzbändern (bei metrischen Kovariablen am Klassenmittelpunkt); die gestrichelte Linie zeigt jeweils den empirische Logit ohne Nullenüberschuß an	116
4.15	Struktogramm einer Evolution des Poisson-GLM's bis hin zu ZIGP-Modellen mit Regression auf allen Parametern	118
4.16	Modi der unnormierten metrischen Kovariablen, die in die μ -Regression eingehen und reine Haupteffekte sind.	127
4.17	(1) - gemeinsames Histogramm von "ln.std.5.mean" und "breadth"; (2) - Anzahl der Beobachtungen je Zelle; (3) - 2D-Kernel-density; (4) - Höhenlinien mit Modus der Kernel-density.	128
4.18	Einfluß von "emp" (1), "r.d.pat" (2) sowie "r.d.intens" (3) auf die Outsourcing-Quote $\frac{\hat{\mu}(x_k)}{E^M}$ unter Fixierung der übrigen Kovariablen durch ihre Modi	130
4.19	Einfluß von "std.5.mean" und "breadth" auf die geschätzte Outsourcing-Rate $\hat{\mu}(\text{std.5.mean}, \text{breadth})/E^M$ je Branche	132
4.20	Einfluß von "time" auf den Variationskoeffizienten getrennt nach Branchen	136

Tabellenverzeichnis

1.1	Range von $\boldsymbol{\mu}$ und $\boldsymbol{\omega}$ in je zwei Settings	27
1.2	Mittlere MLE-Schätzungen und MSE-Statistiken für $ZIGP(\mu_i, \varphi, \omega_i)$ in mehreren Parameter-Settings basierend auf $N = 100$ Datensätzen; geschätzte Standardabweichungen der Statistiken in Klammern.	30
2.1	Range von $\boldsymbol{\mu}$, $\boldsymbol{\varphi}$ und $\boldsymbol{\omega}$ in je zwei Settings	62
2.2	Mittlere MLE-Schätzungen und MSE-Statistiken für $ZIGP(\mu_i, \varphi_i, \omega_i)$ in mehreren Parameter-Settings basierend auf $N = 100$ Datensätzen; geschätzte Standardabweichungen der Statistiken in Klammern.	64
3.1	Range von $\boldsymbol{\mu}$ und $\boldsymbol{\omega}$ in je zwei Settings	79
3.2	Mittlere MLE-Schätzungen und MSE-Statistiken für $ZMGP(\mu_i, \varphi, \omega_i)$ in mehreren Parameter-Settings basierend auf $N = 100$ Datensätzen; geschätzte Standardabweichungen der Statistiken in Klammern.	81
4.1	Quantile und Mittelwerte zur Untersuchung metrischer Variablen	91
4.2	Geschätzte Mittelwerte und Varianzen je Branche und Jahr; Funktionswerte $f_{j,t}(\mathbf{Y})$ als Indikator für Überdispersion	108
4.3	Funktionswerte $f_{l(i)}(\mathbf{Y})$ zur Untersuchung der Überdispersion je nach Klasse der Variablen	111
4.4	Empirische Logits nach Formel 4.5 der Kovariablen zur Untersuchung des Nullenüberschuß	115
4.5	Vergleich der Modelle mit AIC-Statistiken und Vuong-Tests von (I) gegen (II)	124
4.6	Modi metrischer Variablen; bezieht sich auf transformierte, aber unnormierte Kovariablen	126
4.7	Outsourcing-Rate je Branche unter Fixierung der übrigen Kovariablen durch ihre Modi	131
4.8	$\frac{\widehat{Var}(Y)}{E(Y)}(x_k)$ für das Jahr 2000	134
4.9	Funktionswerte $\hat{\omega}(x_k) + (1 - \hat{\omega}(x_k)) \cdot \exp\left(-\frac{\mu^M}{\varphi^M}\right)$	138
A.1	Übersicht über die "R"-Funktionen und deren Zweck	148
B.1	Übersicht über die "R"-Funktionen und deren Zweck	177
C.1	Übersicht über die "R"-Funktionen und deren Zweck	211

D.1	Notationstabelle aller verwendeten Variablen	239
-----	--	-----

Literaturverzeichnis

- [1] G.Kemble Bennett and H.F. Martz. A continuous empirical Bayes smoothing technique. *Biometrika*, 59:361–368, 1972.
- [2] P. J. Bickel and D. A. Freedman. Some asymptotic theory for the bootstrap. *Annals of Statistics*, 9:1196–1217, 1981.
- [3] Kevin A. Clarke. A Simple Distribution-Free Test for Nonnested Hypotheses. Department of Political Science, University of Rochester, Harkness Hall, 2003.
- [4] P. C. Consul and Felix Famoye. Generalized Poisson regression model. *Comm. Statist. Theory Methods*, 21(1):89–109, 1992.
- [5] P.C. Consul. *Generalized Poisson Distributions*. Marcel Dekker, Inc., New York, 1989.
- [6] P.C. Consul and G.C. Jain. A generalization of the poisson distribution. *Technometrics*, 15(4):791–799, 1973.
- [7] C. Czado and A. Min. Consistency and asymptotic normality of the maximum likelihood estimator in a zero-inflated generalized Poisson regression. (<http://www.stat.uni-muenchen.de/sfb386/papers/dsp/paper423.ps>), 2005.
- [8] Ekkehart Dietz and Dankmar Böhning. On estimation of the Poisson parameter in zero-modified Poisson models. *Comput. Statist. Data Anal.*, 34(4):441–459, 2000.
- [9] Michael J. Evans and Jeffrey S. Rosenthal. *Probability and Statistics*. W.H. Freeman and Company, New York, 2nd edition, 2004.
- [10] L. Fahrmeir and H. Kaufmann. Consistency and asymptotic normality of the maximum likelihood estimator in generalized linear models. *Ann. Statist.*, 13(1):342–368, 1985.
- [11] Ludwig Fahrmeir and Gerhard Tutz. *Multivariate Statistical Modelling Based on Generalized Linear Models*. Springer-Verlag, Berlin, 2nd edition, 2001.
- [12] Felix Famoye. Generalized poisson random variate generation. *American Journal of Mathematical and Management Sciences*, 17(3-4):219–237, 1997.
- [13] Felix Famoye and K. P. Singh. Zero-inflated generalized Poisson model with an application to domestic violence data. *Journal of Data Science*, 4(1):117–130, 2006.

- [14] Julian James Faraway. *Linear models with R*. Chapman & Hall, Boca Raton, Florida, 2005.
- [15] Marek Fisz. *Wahrscheinlichkeitsrechnung und Mathematische Statistik*. VEB Deutscher Verlag der Wissenschaften, Berlin, 2 edition, 1958.
- [16] Allan Gut. *Probability: a graduate course*. Springer Science + Business Media, New York, 2nd edition, 2005.
- [17] Jerry Hausmann, Bronwyn H. Hall, and Zvi Griliches. Econometric Models for Count Data with an Application to the Patents - R&D Relationship. *Econometrica*, Vol. 52, No. 4:909–938, 1984.
- [18] Richard M. Heiberger and Burt Holland. *Statistical analysis and data display. An intermediate course with examples in S-Plus, R, and SAS*. Springer Texts in Statistics. New York, NY: Springer. xxiv, 729 p. EUR 79.95/net; sFr 135.50, 2004.
- [19] Harro Heuser. *Lehrbuch der Analysis. Teil 2*. Mathematische Leitfäden. Stuttgart: B. G. Teubner. 736 S., 100 Abb., 576 Aufg., z.T. mit Lösungen. DM 58.00 , 1981.
- [20] Norman L. Johnson, Samuel Kotz, and Adrienne W. Kemp. *Univariate discrete distributions. 2. ed*. Wiley Series in Probability and Mathematical Statistics. Probability and Mathematical Statistics. New York: John Wiley & Sons. xx, 565 p. , 1992.
- [21] Maurice G. Kendall and William R. Buckland. *A Dictionary of Statistical Terms*. Oliver and Boyd, Edinburgh, 1957.
- [22] Konrad Königsberger. *Analysis 1*. Springer Verlag, Berlin, 4 edition, 1999.
- [23] Solomon Kullback and R.A. Leibler. On Information and Sufficiency. *Annals of Mathematical Statistics*, 2:79–86, 1951.
- [24] Diane Lambert. Zero-inflated poisson regression with an application to defects in manufacturing. *Technometrics*, 34(1):1–14, 1992.
- [25] E.L. Lehmann and Joseph P. Romano. *Testing statistical hypotheses. 3rd ed*. Springer Texts in Statistics. New York, NY: Springer. xiv, 784 p. EUR 64.95/net; sFr 115.00; \$ 89.95 , 2005.
- [26] P. McCullagh and J.A. Nelder. *Generalized linear models. 2nd ed*. Monographs on Statistics and Applied Probability. 37. London etc.: Chapman and Hall. xix, 511 p., 1989.
- [27] P. Heinz Müller. *Wahrscheinlichkeitsrechnung und mathematische Statistik: Lexikon der Stochastik*. Akademie Verlag GmbH, Berlin, 5 edition, 1991.
- [28] John Mullahy. Specification and testing of some modified count data models. *Journal of Econometrics*, 33:341–365, 1986.

- [29] Raymond H. Myers, Douglas C. Montgomery, and G. Geoffrey Vining. *Generalized linear models. With applications in engineering and the sciences*. Wiley Series in Probability and Statistics. Chichester: Wiley. xiv, 342 p. \$ 89.95, 2002.
- [30] Helmut Pruscha. *Angewandte Methoden der mathematischen Statistik. Lineare, log-lineare, logistische Modelle. (Applied methods of mathematical statistics. Linear, log-linear and logistic models)*. Teubner Skripten zur Mathematischen Stochastic. Stuttgart: B. G. Teubner. 391 S., 1989.
- [31] Martin Ridout, John Hinde, and Clarice G. B. Demétrio. A Score Test for Testing a Zero-Inflated Poisson Regression Model Against Zero-Inflated Negative Binomial Alternatives. *BIOMETRICS*, 57:219–223, 2001.
- [32] M.S. Ridout, C.G.B. Demétrio, and J.P. Hinde. Models for count data with many zeros. *International Biometric Conference, Cape Town*, pages 179–190, 1998.
- [33] G.A.F. Seber. *Linear regression analysis*. Wiley Series in Probability and Mathematical Statistics. New York etc.: John Wiley & Sons. XVII, 465 p., 1977.
- [34] L. R. Shenton and P. C. Consul. On the probabilistic structure and properties of discrete lagrangian distribution. *A modern course on statistical distribution in scientific work*, Volume 1, 1974.
- [35] Daniel Stekeler. Verallgemeinerte Poissonregression und daraus abgeleitete Zero-Inflated und Zero-Hurdle Regressionsmodelle. Diplomarbeit, Centre of Mathematical Sciences, Munich University of Technology, <http://www-m4.ma.tum.de/Diplarb/ind.txt.html>, Garching bei München, 2004.
- [36] Geert Verbeke and Geert Molenberghs. The Use of Score Tests for Inference on Variance Components. *Biometrics*, 59:254–262, 2003.
- [37] Quang H. Vuong. Likelihood Ratio Tests for Model Selection and Non-Nested Hypotheses. *Econometrica*, 57(2):307–333, 1989.
- [38] Stefan Wagner. An Empirical Analysis of Make-or-Buy Decisions in Patenting. 2005.

Anhang A

Software für das $ZIGP(\mu_i, \varphi, \omega_i)$ –Regressionsmodell

Übersicht über die verwendeten Funktionen

	FUNKTIONSNAME	ZWECK	SEITE
[01]	<i>wald.test</i>	Paßt ein $ZIGP(\mu_i, \varphi, \omega_i)$ –Regressionsmodell an einen Datensatz an und gibt ein Summary mit Wald-Teststatistiken aus.	S. 149
[02]	<i>mle.zigp</i>	Berechnet die ML-Schätzer in einem $ZIGP(\mu_i, \varphi, \omega_i)$ –Regressionsmodell.	S. 153
[03]	<i>rzigp</i>	Erzeugt eine ZIGP-Zufallsvariable mit der Invertierungsmethode.	S. 157
[04]	<i>response.zigp</i>	Erzeugt einen ZIGP-Zufallsvektor aus vorgegebenen linearen Prädiktoren.	S. 159
[05]	<i>log.likelihood.zigp</i>	Berechnet die Loglikelihoodfunktion in einem $ZIGP(\mu_i, \varphi, \omega_i)$ –Regressionsmodell.	S. 161
[06]	<i>fit.zigp</i>	Berechnet die 'fitted values' des angepaßten Modells.	S. 163
[07]	<i>summary.zigp</i>	Gibt eine Auflistung aller Inputs und Outputs der Modellanpassung zurück.	S. 165
[08]	<i>gradient</i>	Berechnet den Gradienten der Loglikelihoodfunktion.	S. 168
[09]	<i>optimized.run</i>	Berechnet einen gültigen Startwert der Loglikelihoodfunktion für die 'Optim'-Methode.	S. 172
[10]	<i>FM</i>	Berechnet die erwartete Fisherinformation.	S. 174

Tabelle A.1: Übersicht über die "R"-Funktionen und deren Zweck

Hauptfunktionen (können direkt vom Benutzer ausgeführt werden)

[01] *wald.test*

Fitting ZIGP(μ_i , ϕ , ω_i) - Regression Models

Description:

'wald.test' is used to fit ZIGP(μ_i , ϕ , ω_i) - Regression Models.

Needs additional Functions:

```
'mle.zigp'
'FM'
'log.likelihood.zigp'
'fit.zigp'
'gradient'
'summary.zigp'
'optimized.run'
# for usage with 'init = T' needs external functions (in 'R' code!)
'mle.zigp.full.like'
'log.likelihood.zigp.full'
'fit.zigp1'
'summary.zigp1'
```

Usage:

```
wald.test(Yin, Xin, Zin, Offset = rep(1,length(Yin)), init = T)
```

Arguments:

Yin: response vector of length n .

Xin: design matrix of dim $(n \times p)$ for mean modelling.

Zin: design matrix of dim $(n \times q)$ for zero inflation modelling.

Offset: exposure for individual observation lengths. Defaults to a vector of 1.
The offset MUST NOT be in 'log' scale.

init: a logical value indicating whether initial optimization values for zero inflation regression parameters are set to -1 (init = F) or are estimated by a ZIGP(μ_i , ϕ , ω_i)-model (init = T). Defaults to 'T'.

Details:

In order to include an intercept in a design matrix, one has to add a vector of ones to the design matrix:
'Intercept <- rep(1,n)'.

If the output should have variable names additionally to parameter tokens (such as 'b0' or 'g0'), create the design matrix by
'Z <- cbind(Intercept, gender, height)'.

Example:

```
## Number of damages in car insurance.

damage <- c(0,1,0,0,0,4,2,0,1,0,1,1,0,2,0,0,1,0,0,1,0,0,0)
Intercept <- rep(1,length(damage))
insurance.year <- c(1,1.2,0.8,1,2,1,1.1,1,1,1.1,1.2,1.3,0.9,1.4,1,1,1,1.2,1,
1,1,1,1)
drivers.age <- c(25,19,30,48,30,18,19,29,24,54,56,20,38,18,23,58,
47,36,25,28,38,39,42)
# abroad: driver has been abroad for longer time (=1)
abroad <- c(0,0,0,1,0,0,1,0,0,0,0,0,1,0,0,0,0,1,0,1,1,1,1)
Y <- damage
X <- cbind(Intercept, drivers.age)
Z <- cbind(abroad)

wald.test(Yin=Y, Xin=X, Zin=Z, Offset = insurance.year, init = F)

[1]
[2]          MU REGRESSION
[3]  b0      Intercept      1.65557      1.15970      1.42758      0.15341
[4]  b1      drivers.age    -0.06554      0.04263     -1.53737      0.12420
[5]          OMEGA REGRESSION
[6]  g0      abroad        -1.01005      2.49831     -0.40430      0.68600
[7]          PHI           1.11699      0.29456
[8]
[9]      Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
[10]      Iterations              351
[11]      Log Likelihood          -23.6
[12]      Pearson Chi Squared           5.5
[13]      AIC                      55
[14]      Range Mu                0.12      2.25
[15]      Range Omega              0.27      0.50
```

```

wald.test <- function(Yin, Xin, Zin, Offset=rep(1,length(Yin)), init=T)
{
  Y <- Yin
  X <- Xin
  Z <- Zin
  k.beta <- dim(X)[2]
  k.gamma <- dim(Z)[2]
  n <- length(Y)
  t.i <- Offset

  # Estimate coefficients
  ausgabe <- mle.zigp(Y, X, Z, Offset = Offset, summary = FALSE, init=init)

  hat.beta <- ausgabe$Coefficients.Mu
  hat.phi <- ausgabe$Dispersion.Parameter
  hat.gamma <- ausgabe$Coefficients.Omega

  # Compute square root of diagonal elements of FM^-1
  B <- FM(hat.beta, hat.phi, hat.gamma, X, Z, Offset = Offset)
  sd.vector <- sqrt( diag(solve(B, tol = 1e-20)) )
  hat.sd.beta <- sd.vector[1:k.beta]
  hat.sd.phi <- sd.vector[k.beta+1]
  hat.sd.gamma <- sd.vector[(k.beta+2):(k.beta+1+k.gamma)]

  # Compute T-Statistics
  z.stat.beta <- hat.beta/hat.sd.beta
  z.stat.gamma <- hat.gamma/hat.sd.gamma

  # Compute P-Values
  p.value.beta <- 2*pnorm(-abs(z.stat.beta))
  p.value.gamma <- 2*pnorm(-abs(z.stat.gamma))

  # Create ***
  glimpse.beta <- rep("",length(z.stat.beta))
  glimpse.gamma <- rep("",length(z.stat.gamma))
  for (i in 1:length(z.stat.beta)) {
    if (p.value.beta[i] < 0.001) {glimpse.beta[i] <- "***"}
    if (p.value.beta[i] >= 0.001 & p.value.beta[i] < 0.01) {glimpse.beta[i] <- "**"}
    if (p.value.beta[i] >= 0.01 & p.value.beta[i] < 0.05) {glimpse.beta[i] <- "*"}
    if (p.value.beta[i] >= 0.05 & p.value.beta[i] < 0.1) {glimpse.beta[i] <- "."} }
  for (i in 1:length(z.stat.gamma)) {
    if (p.value.gamma[i] < 0.001) {glimpse.gamma[i] <- "***"}
    if (p.value.gamma[i] >= 0.001 & p.value.gamma[i] < 0.01) {glimpse.gamma[i] <- "**"}
    if (p.value.gamma[i] >= 0.01 & p.value.gamma[i] < 0.05) {glimpse.gamma[i] <- "*"}
    if (p.value.gamma[i] >= 0.05 & p.value.gamma[i] < 0.1) {glimpse.gamma[i] <- "."} }

  # Create output
  coef.names.beta <- paste("b",c(0:(k.beta-1)),sep="")
  coef.names.gamma <- paste("g",c(0:(k.gamma-1)),sep="")
  coef.desc.beta <- double(k.beta)
  coef.desc.gamma <- double(k.gamma)
  for (i in 1:k.beta) {
    coef.desc.beta[i] <- colnames(X, do.NULL=FALSE)[i]
    if (is.matrix(X)) { if (max(X[,i])==1&min(X[,i])==1)

```

```

    {coef.desc.beta[i] <- "Intercept"} }
else{ if (max(X[i])==1&min(X[i])==1) {coef.desc.beta[i] <- "Intercept"} }
}
for (i in 1:k.gamma) {
coef.desc.gamma[i] <- colnames(Z, do.NULL=FALSE)[i]
if (is.matrix(Z)) { if (max(Z[,i])==1&min(Z[,i])==1)
    {coef.desc.gamma[i] <- "Intercept"} }
else{ if (max(Z[i])==1&min(Z[i])==1) {coef.desc.gamma[i] <- "Intercept"} }
}

output <- matrix("",1+k.beta+1+k.gamma+10,7)
output[2:(1+k.beta+1+k.gamma+2),1] <- c("",coef.names.beta,"",coef.names.gamma,"")
output[2:(1+k.beta+1+k.gamma+2),2] <-
    c("MU REGRESSION",coef.desc.beta, "OMEGA REGRESSION",coef.desc.gamma,"PHI")
output[1:(1+k.beta+1+k.gamma+2),3] <- c("Estimate","",formatC(hat.beta,5,format="f"),
    "",formatC(hat.gamma,5,format="f"),formatC(hat.phi,5,format="f"))
output[1:(1+k.beta+1+k.gamma+2),4] <- c("Std. Error","",
    formatC(hat.sd.beta,5,format="f"),"",formatC(hat.sd.gamma,5,format="f"),
    formatC(hat.sd.phi,5,format="f"))
output[1:(1+k.beta+1+k.gamma+2),5] <- c("z value","",
    formatC(z.stat.beta,5,format="f"),"",formatC(z.stat.gamma,5,format="f"),"")
output[1:(1+k.beta+1+k.gamma+2),6] <- c("Pr(>|z|)", "",
    formatC(p.value.beta,5,format="f"),"",formatC(p.value.gamma,5,format="f"),"")
output[1:(1+k.beta+1+k.gamma+2),7] <- c("", "",glimpse.beta,"",glimpse.gamma,"")
output[(1+k.beta+1+k.gamma+4),2] <- "Signif. codes: 0"
output[(1+k.beta+1+k.gamma+4),3] <- "'***' 0.001"
output[(1+k.beta+1+k.gamma+4),4] <- "'**' 0.01"
output[(1+k.beta+1+k.gamma+4),5] <- "'*' 0.05"
output[(1+k.beta+1+k.gamma+4),6] <- "'.' 0.1"
output[(1+k.beta+1+k.gamma+4),7] <- "' ' 1"
output[(1+k.beta+1+k.gamma+5),2] <- "Iterations"
output[(1+k.beta+1+k.gamma+5),4] <- ausgabe$Iterations[1]
output[(1+k.beta+1+k.gamma+6),2] <- "Log Likelihood"
output[(1+k.beta+1+k.gamma+6),4] <-
    formatC(ausgabe$Log.Likelihood,digits=1,format="f")
output[(1+k.beta+1+k.gamma+7),2] <- "Pearson Chi Squared"
output[(1+k.beta+1+k.gamma+7),4] <- formatC(ausgabe$Pearson,digits=1,format="f")
output[(1+k.beta+1+k.gamma+8),2] <- "AIC"
output[(1+k.beta+1+k.gamma+8),4] <- round(ausgabe$AIC)
output[(1+k.beta+1+k.gamma+9),2] <- "Range Mu"
output[(1+k.beta+1+k.gamma+9),4] <- formatC(ausgabe$Range.Mu[1],digits=2,format="f")
output[(1+k.beta+1+k.gamma+9),5] <- formatC(ausgabe$Range.Mu[2],digits=2,format="f")
output[(1+k.beta+1+k.gamma+10),2] <- "Range Omega"
output[(1+k.beta+1+k.gamma+10),4] <-
    formatC(ausgabe$Range.Omega[1],digits=2,format="f")
output[(1+k.beta+1+k.gamma+10),5] <-
    formatC(ausgabe$Range.Omega[2],digits=2,format="f")
output2 <- data.frame(output)
return(output2)
}

```

[02] *mle.zigp*

Maximum Likelihood Estimates

Description:

'mle.zigp' is used to calculate the MLEs of the regression parameters for mean and zero inflation and the overdispersion parameter.

Needs additional Functions:

```
'log.likelihood.zigp'
'fit.zigp'
'gradient'
'summary.zigp'
'optimized.run'
# for usage with 'init = T' needs external functions (in 'R' code!)
'mle.zigp.full.like'
'log.likelihood.zigp.full'
'fit.zigp1'
'summary.zigp1'
```

Usage:

```
mle.zigp(Yin, Xin, Zin, Offset = rep(1,length(Yin)), summary = T, init = FALSE)
```

Arguments:

Yin: response vector of length n.

Xin: design matrix of dim (n x p) for mean modelling.

Zin: design matrix of dim (n x q) for zero inflation modelling.

Offset: exposure for individual observation lengths. Defaults to a vector of 1.
The offset MUST NOT be in 'log' scale.

summary: a logical value indicating whether a list summary is shown (=T) or returned to the user (=F), i.e. if values of the ML estimation should be extracted, summary has to be set to 'F'. Defaults to 'T'.

init: a logical value indicating whether initial optimization values for zero inflation regression parameters are set to -1 (init = F) or are estimated by a ZIGP(μ_i , ϕ , ω)-model (init = T). Defaults to 'T'.

Value:

If summary is set to 'F', the following values are returned:

ZI.Parameter	Zero Inflation estimate
Coefficients.Mu	Regression Parameters estimates for mean
Coefficients.Omega	Regression Parameters estimates for ZI
Dispersion.Parameter	Dispersion Parameter Estimate
Range.Mu	mean range

Range.Omega	ZI parameter range
Log.Likelihood	log likelihood value at MLE
Residuals	residuals
Pearson	Pearson Chi Squared Statistics
AIC	Akaike Information Criterion
Iterations	number of iterations needed
Time	time needed
Message	convergence message of underlying 'optim' routine
Response	response vector
Fitted.Values	fitted values
Design.Mu	design matrix for mean
Design.Omega	design matrix for ZI

See Also:

```
wald.test().
```

Example:

```
## Number of damages in car insurance.

damage <- c(0,1,0,0,0,4,2,0,1,0,1,1,0,2,0,0,1,0,0,1,0,0,0)
Intercept <- rep(1,length(damage))
insurance.year <- c(1,1.2,0.8,1,2,1,1.1,1,1,1.1,1.2,1.3,0.9,1.4,1,1,1,1.2,1,
1,1,1,1)
drivers.age <- c(25,19,30,48,30,18,19,29,24,54,56,20,38,18,23,58,
47,36,25,28,38,39,42)
# abroad: driver has been abroad for longer time (=1)
abroad <- c(0,0,0,1,0,0,1,0,0,0,0,0,1,0,0,0,0,1,0,1,1,1,1)
Y <- damage
X <- cbind(Intercept, drivers.age)
Z <- cbind(abroad)

mle.zigp(Yin=Y, Xin=X, Zin=Z, Offset = insurance.year, summary = T, init = F)

[1] Range for ZI-Parameters: 0.2669691 0.5
[2] Coefficients for mu: 1.655569 -0.06553921
[3] Coefficients for omega: -1.010055
[4] Dispersion Parameter: 1.116992
[5] Pearson Chi Squared: 5.52589
[6] Range of mu: 0.1169855 2.253163
[7] Range of omega: 0.2669691 0.5
[8] Message: "NULL"
[9] AIC: 55.24655
```

```

mle.zigp <- function(Yin, Xin, Zin, Offset = rep(1,length(Yin)), summary = T,
                    init = FALSE)
{
  X <- Xin
  Y <- Yin
  Z <- Zin
  if(is.matrix(X)) {
    n <- dim(X)[1]
    k.beta <- dim(X)[2]
  }
  else {
    n <- length(X)
    k.beta <- 1
  }
  if(is.matrix(Z)) {
    k.gamma <- dim(Z)[2]
    nz <- dim(Z)[1]
  }
  else {
    k.gamma <- 1
    nz <- length(Z)
  }

  if(nz==n){
    t.i <- Offset

  if(init){
    #optimized initial values
    start.delta <- optimized.run(Y,X,Z)
    # maximization of log likelihood
    start <- proc.time()[2]
    opt <- optim(par = start.delta, fn = log.likelihood.zigp, gr=gradient,
                method = "CG")
  }
  else
  {
    # simple initial values
    alpha <- log(0.2)
    gamma <- rep(-1,k.gamma)
    beta <- summary(glm(Y ~ offset(log(t.i)) + 1 + X, family = poisson(link=log)
                        ))$coefficients[, 1]
    start.delta <- c(beta,alpha,gamma)
    # maximization of log likelihood
    start <- proc.time()[2]
    opt <- optim(par = start.delta, fn = log.likelihood.zigp, gr=gradient,
                method = "CG")
  }

  delta <- opt$par
  it <- opt$counts
  log.likelihood <- - opt$value
  if(is.null(opt$message)) {message <- "NULL"}
  else { message <- opt$message }
  AIC <- -2 * log.likelihood + 2 * (k.beta + 1 + k.gamma)

```

```

time.needed <- proc.time()[2] - start

# results
coef <- double(k.beta + k.gamma + 1)
coef <- delta
beta <- coef[1 : k.beta]
phi <- 1 + exp(coef[k.beta + 1])
gamma <- coef[(k.beta + 2) : (k.beta + k.gamma + 1)]

fit <- fit.zigp(delta)
res <- Y - fit$fit
chsq <-
  sum((Y - fit$fit) / sqrt((1-fit$omega)*phi*fit$fit+
    fit$omega*(1-fit$omega)*fit$fit^2))

range.mu <- c(min(fit$mu), max(fit$mu))
range.omega <- c(min(fit$omega), max(fit$omega))
# compute zero inflation parameters omega
if(is.matrix(Z)){ eta.omega <- Z %*% gamma }
else { eta.omega <- Z * gamma }
omega <- exp(eta.omega)/(1+exp(eta.omega))

# summarize results
mle.data <- list(ZI.Parameter = omega,
  Coefficients.Mu = beta,
  Coefficients.Omega = gamma,
  Dispersion.Parameter = phi,
  Range.Mu = range.mu,
  Range.Omega = range.omega,
  Log.Likelihood = log.likelihood,
  Residuals = res,
  Pearson = chsq,
  AIC = AIC,
  Iterations = it,
  Time = time.needed,
  Message = message,
  Response = Y,
  Fitted.Values = fit$fit,
  Design.Mu = X,
  Design.Omega = Z)
if(summary) {
  summary.zigp(mle.data)
}
else {
  return(mle.data)
}
}
else{
return("X and Z have different numbers of rows!")
}
}

```


Hilfsfunktionen (werden i.d.R. nicht direkt vom Benutzer ausgeführt)

[03] *rzigp*

ZIGP random variable generator

Description:

‘rzigp’ generates ZIGP random variables using the inversion method.

Usage:

```
rzigp(n, mu, phi, omega)
```

Arguments:

n: length of output random vector.

mu: mean

phi: dispersion parameter

omega: zero inflation parameter

Value:

Generates a ZIGP random vector of length n.

See Also:

```
response.zigp().
```

Example:

```
# set seed for random variable generator
set.seed(2)
rzigp(3, 2, 1.5, 0.2)
[1] 3 1 0
```

```

rzigp <- function(n, mu = stop("no mu arg"), phi = stop("no phi arg"),
  omega = stop("no omega arg"))
{
  # check if parameters are valid
  if(omega < 0) {return("omega has to be in [0,1]!")}
  if(omega > 1) {return("omega has to be in [0,1]!")}

  # inversion method
  x <- double(n)
  for(i in 1:n) {
    y <- integer(1)
    c <- double(1)
    #P(y=0)
    p <- omega + (1-omega) * exp( - mu/phi)
    s <- p
    u <- runif(1, 0, 1)
    if (u > s){
      #P(y=1)
      p <- (1-omega)*mu/phi*exp(-1/phi*(mu+phi-1))
      s <- s + p
      y <- 1
      while (u > s) {
        y <- y + 1
        p <- (mu + (phi-1)*y)^(y-1) / (mu + (phi-1)*(y-1))^(y-2) / (y * phi) *
          exp((1-phi)/phi) * p
        s <- s + p
      }
    }
    x[i] <- y
  }
  return(x)
}

```

[04] *response.zigp*

ZIGP response generator

Description:

'response.zigp' generates a ZIGP response vector for a given setting of linear predictors.

Needs additional Functions:

'rzigp'

Usage:

```
response.zigp(X, Z, beta, phi, gamma)
```

Arguments:

X: design matrix of dim (n x p) for modelling of mean

Z: design matrix of dim (n x q) for modelling of zero inflation

beta: regression parameters for mean of length p

phi: dispersion parameter

gamma: regression parameters for zero inflation of length q

Value:

Generates a ZIGP response vector of length n.

See Also:

rzigp().

Example:

```
X <- matrix(c(1:3,4,3,5),3,2)
Z <- rep(1,3)
beta <- c(5,-2)
phi <- 1.5
gamma <- -10
# set seed for random variable generator
set.seed(1)
response.zigp(X,Z,beta,phi,gamma)
[1] 0 55 125
```

```

response.zigp <- function(X, Z, beta, phi, gamma)
{
# get number of observations from X and Z
  if(is.matrix(X)) {
    nx <- dim(X)[1]
  }
  else {
    nx <- length(X)
  }
  if(is.matrix(Z)) {
    nz <- dim(Z)[1]
  }
  else {
    nz <- length(Z)
  }

  if (nx==nz) {
    n <- nx
    mu <- double(n)
    omega <- double(n)
    Y <- double(n)

    if(is.matrix(X)) {
      mu <- exp(X%*%beta)
    }
    else {
      mu <- exp(X*beta)
    }
    if(is.matrix(Z)) {
      omega <- exp(Z%*%gamma)/(1+exp(Z%*%gamma))
    }
    else {
      omega <- exp(Z*gamma)/(1+exp(Z*gamma))
    }

# create ZIGP response for all i
    for(i in 1:n) {
      Y[i] <- rzigp(1, mu[i], phi, omega[i])
    }
    return(Y)
  }
  else {
    print("X and Z have different numbers of rows!")
  }
}

```

[05] *log.likelihood.zigp*

Log Likelihood of ZIGP distribution

Description:

'log.likelihood.zigp' calculates the log likelihood function of the ZIGP distribution.

Usage:

```
log.likelihood.zigp(delta)
```

Arguments:

delta: a parameter vector of length (p+q+1) with $\dim(X) = (n \times p)$, $\dim(Z) = (n \times q)$. Create delta by pasting 'delta <- c(beta, ALPHA, gamma)', where ALPHA is one-dimensional with $\phi = 1 + \exp(\alpha) \Leftrightarrow \alpha = \log(\phi-1)$. Reason: with ALPHA, the problem of finding MLE's is unrestricted - with phi, it is not. Unrestricted problems are numerically more stable. beta is the vector of regression parameters for the mean modelling, phi is the dispersion parameter and gamma is the vector of regression parameters for the ZI modelling.

Details:

The design matrices have to be defined as X (for mean) and Z (for ZI). n has to be defined as the number of observations $\dim(X)[1]$. k.beta has to be defined as the length of beta. k.gamma has to be defined as the length of gamma. Y has to be defined as the response vector. t.i has to be defined as the exposure.

Example:

```
Y <- c(3,0,2)
X <- matrix(c(1:3,4,3,5),3,2)
Z <- rep(1,3)
n <- dim(X)[1]
t.i <- rep(1,n)
beta <- c(5,-2)
phi <- 1.5
alpha <- log(phi-1)
gamma <- -10
k.beta <- length(beta)
k.gamma <- length(gamma)
delta <- c(beta,alpha,gamma)
log.likelihood.zigp(delta)
[1] 107.2714
```

```

log.likelihood.zigp <- function(delta)
{
  eta.mu <- double(n)
  eta.omega <- double(n)
  s1 <- double(1)
  s2 <- double(1)

  if(k.beta == 1) {
    eta.mu <- X * delta[1]
  }
  else {
    beta <- delta[1:k.beta]
    eta.mu <- X %*% beta
  }
  if(k.gamma == 1) {
    eta.omega <- Z * delta[k.beta+2]
  }
  else {
    gamma <- delta[(k.beta + 2) : (k.beta + k.gamma + 1)]
    eta.omega <- Z %*% gamma
  }

  mu.i <- t.i*exp(eta.mu)
  phi <- 1+exp(delta[k.beta+1])
  k.i <- exp(eta.omega)
  temp3 <- (mu.i + (phi-1)*Y)

  # terms of log likelihood with 1_{Y_i = 0}
  s1 <- sum(ifelse(Y == 0,1,0)* (log(k.i+exp(-1/phi*mu.i)) -
    log(1+k.i)))
  # terms of log likelihood with 1_{Y_i > 0} (excluding factorial)
  s2 <- sum(ifelse(Y>0,1,0)*(-log(1+k.i)+log(t.i)+eta.mu+(Y-1)*log(temp3)-
    Y*log(phi)-1/phi*temp3) - lgamma(Y+1))

  l <- s1 + s2
  return( - l)
}

```

[06] *fit.zigp*

Fitted Values

Description:

'fit.zigp' computes fitted values.

Usage:

```
fit.zigp(delta)
```

Arguments:

delta: a parameter vector of length (p+q+1) with $\dim(X) = (n \times p)$,
 $\dim(Z) = (n \times q)$. Create delta by pasting
 'delta <- c(beta, phi, gamma)'.
 beta is the vector of regression parameters for the mean modelling,
 phi is the dispersion parameter and gamma is the vector of regression
 parameters for the ZI modelling.

Details:

The design matrices have to be defined as X (for mean) and Z (for ZI).
 n has to be defined as the number of observations $\dim(X)[1]$.
 k.beta has to be defined as the length of beta.
 k.gamma has to be defined as the length of gamma.
 t.i has to be defined as the exposure.

Example:

```
X <- matrix(c(1:3,4,3,5),3,2)
Z <- rep(1,3)
n <- dim(X)[1]
beta <- c(5,-2)
phi <- 1.5
gamma <- -10
k.beta <- length(beta)
k.gamma <- length(gamma)
t.i <- rep(1,n)
delta <- c(beta,phi,gamma)
fit.zigp(delta)
[1] $fit
[2]      [,1]
[3] [1,]  0.04978481
[4] [2,]  54.59567139
[5] [3,] 148.40642146
[6] $mu
[7]      [,1]
[8] [1,]  0.04978707
[9] [2,]  54.59815003
[10] [3,] 148.41315910
[11] $omega
[12] [1] 4.539787e-05 4.539787e-05 4.539787e-05
```

```

fit.zigp <- function(delta)
{
  eta.mu <- double(n)
  eta.omega <- double(n)
  mu <- double(n)
  omega <- double(n)
  fit <- double(n)

  if(k.beta == 1) {
    eta.mu <- X * delta[1]
  }
  else {
    beta <- delta[1:k.beta]
    eta.mu <- X %*% beta
  }
  if(k.gamma == 1) {
    eta.omega <- Z * delta[k.beta+2]
  }
  else {
    gamma <- delta[(k.beta + 2) : (k.beta + k.gamma + 1)]
    eta.omega <- Z %*% gamma
  }

  # computation of mu, omega and mean
  mu <- t.i*exp(eta.mu)
  omega <- exp(eta.omega)/(1+exp(eta.omega))
  fit <- (1 - omega) * mu

  return(list(fit = fit, mu = mu, omega = omega))
}

```


[07] *summary.zigp*

Summary

Description:

‘summary.zigp’ summarizes the results of the maximum likelihood estimation.

Usage:

```
summary.zigp(mle.data.in)
```

Arguments:

mle.data.in: a list of result parameters given by function ‘mle.zigp’

```

summary.zigp <- function(mle.data.in)
{
  X <- mle.data.in$Design.Mu
  if(is.matrix(X)) {
    nx <- dim(X)[1]
    k.beta <- dim(X)[2]
  }
  else {
    nx <- length(X)
    k.beta <- 1
  }
  Z <- mle.data.in$Design.Omega
  if(is.matrix(Z)) {
    nz <- dim(Z)[1]
    k.gamma <- dim(Z)[2]
  }
  else {
    nz <- length(Z)
    k.gamma <- 1
  }

  # range of ZI parameters
  out0 <- matrix(double(2), 1, 2)
  out0[1, 1] <- min(mle.data.in$ZI.Parameter)
  out0[1, 2] <- max(mle.data.in$ZI.Parameter)
  colnames(out0) <- c("", "")
  rownames(out0) <- c("Range for ZI-Parameters: ")
  print(out0)
  # beta
  out1 <- matrix(double(k.beta), 1, k.beta)
  kopf <- rep("", k.beta)
  for(i in 1:k.beta) {
    out1[1, i] <- mle.data.in$Coefficients.Mu[i]
  }
  colnames(out1) <- kopf
  rownames(out1) <- c("Coefficients for mu: ")
  print(out1)
  # gamma
  out2 <- matrix(double(k.gamma), 1, k.gamma)
  kopf <- rep("", k.gamma)
  for(i in 1:k.gamma) {
    out2[1, i] <- mle.data.in$Coefficients.Omega[i]
  }
  colnames(out2) <- kopf
  rownames(out2) <- c("Coefficients for omega: ")
  print(out2)
  # dispersion parameter
  out3 <- matrix(double(1), 1, 1)
  out3[1, 1] <- mle.data.in$Dispersion.Parameter
  colnames(out3) <- c("")
  rownames(out3) <- c("Dispersion Parameter: ")
  print(out3)
  # Pearson Chi Squared
  out4 <- matrix(double(1), 1, 1)

```

```

out4[1, 1] <- mle.data.in$Pearson
colnames(out4) <- c("")
rownames(out4) <- c("Pearson Chi Squared:      ")
print(out4)
# range of mu
out5 <- matrix(double(2), 1, 2)
out5[1, 1] <- mle.data.in$Range.Mu[1]
out5[1, 2] <- mle.data.in$Range.Mu[2]
colnames(out5) <- c("", "")
rownames(out5) <- c("Range of mu:              ")
print(out5)
# range of omega
out6 <- matrix(double(2), 1, 2)
out6[1, 1] <- mle.data.in$Range.Omega[1]
out6[1, 2] <- mle.data.in$Range.Omega[2]
colnames(out6) <- c("", "")
rownames(out6) <- c("Range of omega:          ")
print(out6)
# optimization message
out7 <- matrix(double(1), 1, 1)
out7[1, 1] <- mle.data.in$Message
colnames(out7) <- c("")
rownames(out7) <- c("Message:                ")
print(out7)
# AIC
out8 <- matrix(double(1), 1, 1)
out8[1, 1] <- mle.data.in$AIC
colnames(out8) <- c("")
rownames(out8) <- c("AIC:                      ")
print(out8)
return()
}

```

[08] *gradient*

Die explizite Berechnung eines Gradienten ist zwar nicht notwendig, da für die 'optim'-Methode "CG" Gradienten gegebenenfalls numerisch geschätzt werden. Allerdings erhöhen explizite Gradienten die Konvergenzgeschwindigkeit enorm.

Gradient of log likelihood

Description:

'gradient' calculates the gradient of the log likelihood function.

Usage:

```
gradient(delta)
```

Arguments:

delta: a parameter vector of length (p+q+1) with $\dim(X) = (n \times p)$, $\dim(Z) = (n \times q)$. Create delta by pasting 'delta <- c(beta, ALPHA, gamma)', where ALPHA is one-dimensional with $\phi = 1 + \exp(\alpha) \Leftrightarrow \alpha = \log(\phi-1)$. Reason: with ALPHA, the problem of finding MLE's is unrestricted - with phi, it is not. Unrestricted problems are numerically more stable. beta is the vector of regression parameters for the mean modelling, phi is the dispersion parameter and gamma is the vector of regression parameters for the ZI modelling.

mle.data.in: a list of result parameters given by function 'mle.zigp'

Details:

This function used for the 'optim' method. Some 'optim' methods need a gradient and have to approximate it in every iteration step. This slows down the process a lot. Therefore, an explicit gradient function increases convergence speed and ensures correct convergence even if the log likelihood function is not continuous due to the indicator function of the ZI parameter.

The design matrices have to be defined as X (for mean) and Z (for ZI). n has to be defined as the number of observations $\dim(X)[1]$. k.beta has to be defined as the length of beta. k.gamma has to be defined as the length of gamma. t.i has to be defined as the exposure.

Example:

```
Y <- c(3,0,2)
X <- matrix(c(1:3,4,3,5),3,2)
Z <- rep(1,3)
n <- dim(X)[1]
beta <- c(5,-2)
phi <- 1.5
alpha <- log(phi-1)
gamma <- -10
```

```
k.beta <- length(beta)
k.gamma <- length(gamma)
t.i <- rep(1,n)
delta <- c(beta,alpha,gamma)
gradient(delta)
[1] 289.8153379 480.6197593 -64.3128614 -0.9998638
```

```

gradient <- function(delta)
{
  grad <- double(k.beta + 1 + k.gamma)
  eta.mu <- double(n)
  eta.omega <- double(n)
  s1 <- double(1)
  s2 <- double(1)

  if(k.beta == 1) {
    eta.mu <- X * delta[1]
    X <- cbind(X, rep(0, length(X)))
  }
  else {
    beta <- delta[1:k.beta]
    eta.mu <- X %*% beta
  }
  if(k.gamma == 1) {
    eta.omega <- Z * delta[k.beta+2]
    Z <- cbind(Z, rep(0, length(Z)))
  }
  else {
    gamma <- delta[(k.beta + 2) : (k.beta + k.gamma + 1)]
    eta.omega <- Z %*% gamma
  }

  mu.i <- t.i*exp(eta.mu)
  k.i <- exp(eta.omega)
  phi <- 1+exp(delta[k.beta+1])
  P0 <- exp(-1/phi*mu.i)

  # Derivative for beta
  for (j in 1:k.beta)
  {
    # terms of gradient with 1_{Y_i = 0}
    s1 <- sum(ifelse(Y == 0, 1, 0) * X[,j] * (P0 * (-1/phi) * mu.i / (k.i + P0)))
    # terms of gradient with 1_{Y_i > 0}
    s2 <- sum(ifelse(Y > 0, 1, 0) * X[,j] * (1 + (Y-1) * mu.i / (mu.i + (phi-1) * Y) - 1/phi * mu.i))
    grad[j] <- s1 + s2
  }

  # Derivative for phi
  # terms of gradient with 1_{Y_i = 0}
  s1 <- sum(ifelse(Y == 0, 1, 0) * (1/phi^2 * P0 * mu.i / (k.i + P0)))
  # terms of gradient with 1_{Y_i > 0}
  s2 <- sum(ifelse(Y > 0, 1, 0) * (Y * (Y-1) / (mu.i + (phi-1) * Y) - Y/phi + (mu.i - Y) / phi^2))
  grad[k.beta+1] <- s1 + s2

  # Derivative for gamma
  for (j in 1:k.gamma)
  {
    # terms of gradient with 1_{Y_i = 0}
    s1 <- sum(ifelse(Y == 0, 1, 0) * Z[,j] * k.i * (1 / (k.i + P0)))

```

```
# terms of gradient independent of Y_i
  s2 <- -sum(Z[,j]*k.i/(1+k.i))
  grad[k.beta+1+j] <- s1 + s2
}

return(-grad)
}
```

[09] *optimized.run*

Calculation of valid initial values

Description:

'optimized.run' calculates valid initial values for the 'optim' method.

Needs additional Functions:

```
# needs external functions (in 'R' code!)
'mle.zigp.full.like'
'log.likelihood.zigp.full'
'fit.zigp1'
'summary.zigp1'
```

Usage:

```
optimized.run(Y,X,Z)
```

Arguments:

Yin: response vector of length n.

Xin: design matrix of dim (n x p) for mean modelling.

Zin: design matrix of dim (n x q) for zero inflation modelling.

Details:

This function uses ''R'' routines for ZIGP(μ_i, ϕ, ω) regression.
Therefore, compile
'mle.zigp.full.like'
'log.likelihood.zigp.full'
'fit.zigp1'
'summary.zigp1'.

t.i has to be defined as the exposure.

n has to be defined as the number of observations `dim(X)[1]`.

Example:

```
Y <- c(3,0,2)
X <- matrix(c(1:3,4,3,5),3,2)
Z <- rep(1,3)
n <- dim(X)[1]
t.i <- rep(1,n)
optimized.run(Y,X,Z)
[1]          X1          X2          Z
[2]  -0.6783368  0.4559966 -1.6094379 -1.3862944
```



```

optimized.run <- function(Y,X,Z)
{

# improved initial values
# get initial beta, alpha & omega
out <- mle.zigp.full.like(Y, X, Offset = t.i, summary=FALSE)
beta.start <- out$Coefficients
alpha.start <- log(out$Dispersion.Parameter-1)
omega.first <- out$ZI.Parameter

# LM for gamma
rechte.seite <- rep(log(omega.first) - log(1-omega.first),n)
out<-lm(rechte.seite ~ Z-1)
gamma.start <- out$coefficients
rm(rechte.seite,out)

start.delta <- c(beta.start,alpha.start,gamma.start)
rm(beta.start,alpha.start,gamma.start)

return(start.delta)
}

```

[10] *FM*

Fisher Information

Description:

'FM' calculates the (Expected) Fisher Information matrix.

Usage:

```
FM(beta, phi, gamma, X, Z, Offset = NULL)
```

Arguments:

beta: regression parameters for mean of length p

phi: dispersion parameter

gamma: regression parameters for zero inflation of length q

X: design matrix of dim (n x p) for mean modelling.

Z: design matrix of dim (n x q) for zero inflation modelling.

Offset: exposure for individual observation lengths. Defaults to a vector of 1.
The offset MUST NOT be in 'log' scale.

Example:

```
X <- matrix(c(1:3,4,3,5),3,2)
Z <- rep(1,3)
beta <- c(5,-2)
phi <- 1.5
gamma <- -10
FM(beta, phi, gamma, X, Z)
      [,1]      [,2]      [,3]      [,4]
[1,] 6.935767e+02 1.139715e+03 -2.218288e+00 -1.506747e-06
[2,] 1.139715e+03 1.875348e+03 -3.600850e+00 -6.026989e-06
[3,] -2.218288e+00 -3.600850e+00 1.797916e+00 1.004498e-06
[4,] -1.506747e-06 -6.026989e-06 1.004498e-06 9.079169e-05
```

```

FM <- function(beta, phi, gamma, X, Z, Offset = NULL)
{

k.beta <- length(beta)
k.gamma <- length(gamma)

FM <- array(0,c(1+k.beta+k.gamma,1+k.beta+k.gamma))

if(is.matrix(X)){ eta.mu <- X%*%beta }
else{eta.mu <- X*beta}
if(is.matrix(Z)){ eta.gamma <- Z%*%gamma }
else{eta.gamma <- Z*gamma}

if (is.null(Offset)) {mu <- exp(eta.mu)}
else {t.i <- Offset
      mu <- t.i*exp(eta.mu)}
k <- exp(eta.gamma)
P0 <- exp(-1/phi*mu)
# a_i = E[I_{Y_i = 0}]
a <- (k+P0)/(1+k)

temp <- double(1)

if(k.beta==1){ X <- cbind(X,rep(0,length(X))) }
if(k.gamma==1){ Z <- cbind(Z,rep(0,length(Z))) }

# the second derivative w.r.t. "beta"
for (i in 1:k.beta)
{
  for (j in 1:k.beta)
  {
    temp <- sum( X[,i]*X[,j]*mu* ( a*(-1/phi*P0^2+(mu-phi)/(phi^2)*P0*k)/
      ((k+P0)^2) + ((phi-1)*mu)/(phi^2*(mu-2+2*phi)*(1+k)) - (1-a)/phi ) )
    FM[i,j] <- -temp
  }
}

# the second derivative w.r.t. "phi"
temp <- sum( a*(1/(phi^4)*P0*mu^2*k-2/(phi^3)*P0*mu*(k+P0))/((k+P0)^2) -
  (mu*(mu+2*phi))/(phi^2*(mu-2+2*phi)*(1+k)) +
  mu/(1+k)*(1/phi^2+2/phi^3) - 2*(1-a)*mu/(phi^3) )
FM[k.beta+1,k.beta+1] <- -temp

# the second derivative w.r.t. "gamma"
for (i in 1:k.gamma)
{
  for (j in 1:k.gamma)
  {
    temp <- sum( Z[,i]*Z[,j]*k* ( a*P0/((k+P0)^2)-1/((1+k)^2) ) )
    FM[k.beta+1+i,k.beta+1+j] <- -temp
  }
}

# the mixed derivative w.r.t. "beta" and "phi"

```

```

for (i in 1:k.beta)
{
temp <- sum( X[,i]*mu* ( a*(-1/phi^3*P0*mu*k+1/(phi^2)*P0*(k+P0))/((k+P0)^2)-
mu/(phi^2*(mu-2+2*phi)*(1+k))+(1-a)/(phi^2) ) )
FM[k.beta+1,i] <- -temp
FM[i,k.beta+1] <- -temp
}

# the mixed derivative w.r.t. "beta" and "gamma"
for (i in 1:k.beta)
{
for (j in 1:k.gamma)
{
temp <- sum( X[,i]*Z[,j]*a*P0*mu*k/(phi*(k+P0)^2) )
FM[k.beta+1+j,i] <- -temp
FM[i,k.beta+1+j] <- -temp
}
}

# the mixed derivative w.r.t. "phi" and "gamma"
for (i in 1:k.gamma)
{
temp <- sum( Z[,i]* a * (-1/phi^2 * P0 * mu * k)/((k+P0)^2) )
FM[k.beta+1+i,k.beta+1] <- -temp
FM[k.beta+1,k.beta+1+i] <- -temp
}

return(FM)
}

```

Anhang B

Software für das $ZIGP(\mu_i, \varphi_i, \omega_i)$ –Regressionsmodell

Übersicht über die verwendeten Funktionen

	FUNKTIONSNAME	ZWECK	SEITE
[01]	<i>wald.test</i>	Paßt ein $ZIGP(\mu_i, \varphi_i, \omega_i)$ –Regressionsmodell an einen Datensatz an und gibt ein Summary mit Wald-Teststatistiken aus.	S. 178
[02]	<i>mle.zigp</i>	Berechnet die ML-Schätzer in einem $ZIGP(\mu_i, \varphi_i, \omega_i)$ –Regressionsmodell.	S. 183
[03]	<i>rzigp</i>	Erzeugt eine ZIGP-Zufallsvariable mit der Invertierungsmethode.	S. 189
[04]	<i>response.zigp</i>	Erzeugt einen ZIGP-Zufallsvektor aus vorgegebenen linearen Prädiktoren.	S. 191
[05]	<i>log.likelihood.zigp</i>	Berechnet die Loglikelihoodfunktion in einem $ZIGP(\mu_i, \varphi_i, \omega_i)$ –Regressionsmodell.	S. 194
[06]	<i>fit.zigp</i>	Berechnet die 'fitted values' des angepaßten Modells.	S. 196
[07]	<i>summary.zigp</i>	Gibt eine Auflistung aller Inputs und Outputs der Modellanpassung zurück.	S. 199
[08]	<i>gradient</i>	Berechnet den Gradienten der Loglikelihoodfunktion.	S. 202
[09]	<i>optimized.run</i>	Berechnet einen gültigen Startwert der Loglikelihoodfunktion für die 'Optim'-Methode.	S. 206
[10]	<i>FM</i>	Berechnet die erwartete Fisherinformation.	S. 208

Tabelle B.1: Übersicht über die "R"-Funktionen und deren Zweck

Hauptfunktionen (können direkt vom Benutzer ausgeführt werden)[01] *wald.test*

Fitting ZIGP(mu_i, phi_i, omega_i) - Regression Models

Description:

'wald.test' is used to fit ZIGP(mu_i, phi_i, omega_i) - Regression Models.

Needs additional Functions:

```
'mle.zigp'
'FM'
'log.likelihood.zigp'
'fit.zigp'
'gradient'
'summary.zigp'
'optimized.run'
# for usage with 'init = T' needs external functions (in 'R' code!)
'mle.zigp.full.like'
'log.likelihood.zigp.full'
'fit.zigp1'
'summary.zigp1'
```

Usage:

```
wald.test(Yin, Xin, Win, Zin, Offset=rep(1,length(Yin)), init=T)
```

Arguments:

Yin: response vector of length n.

Xin: design matrix of dim (n x p) for mean modelling.

Win: design matrix of dim (n x r) for overdispersion modelling.

Zin: design matrix of dim (n x q) for zero inflation modelling.

Offset: exposure for individual observation lengths. Defaults to a vector of 1.
The offset MUST NOT be in 'log' scale.

init: a logical value indicating whether initial optimization values for dispersion are set to -2.5 and values for zero inflation regression parameters are set to -1 (init = F) or are estimated by a ZIGP(mu_i, phi, omega)-model (init = T). Defaults to 'T'.

Details:

In order to include an intercept in a design matrix, one has to add a vector of ones to the design matrix:

```
'Intercept <- rep(1,n)'
```

If the output should have variable names additionally to parameter tokens (such as 'b0', 'a0' or 'g0'), create the design matrix by 'W <- cbind(Intercept, gender, height)'.

Example:

```
## Number of damages in car insurance.

damage <- c(0,1,0,0,0,4,2,0,1,0,1,1,0,2,0,0,1,0,0,1,0,0,0)
Intercept <- rep(1,length(damage))
insurance.year <- c(1,1.2,0.8,1,2,1,1.1,1,1,1.1,1.2,1.3,0.9,1.4,1,1,1,1.2,1,
1,1,1,1)
drivers.age <- c(25,19,30,48,30,18,19,29,24,54,56,20,38,18,23,58,
47,36,25,28,38,39,42)
# for overdispersion: car brand dummy in {1,2,3}, brand = 1 is reference
brand <- c(1,2,1,3,3,2,2,1,1,3,2,2,1,3,1,3,2,2,1,1,3,3,2)
brand2 <- ifelse(brand==2,1,0)
brand3 <- ifelse(brand==3,1,0)
W <- t(rbind(brand2,brand3))
# abroad: driver has been abroad for longer time (=1)
abroad <- c(0,0,0,1,0,0,1,0,0,0,0,0,1,0,0,0,0,1,0,1,1,1,1)
Y <- damage
X <- cbind(Intercept, drivers.age)
Z <- cbind(abroad)

wald.test(Yin=Y, Xin=X, Win=W, Zin=Z, Offset = insurance.year, init = F)
```

		Estimate	Std. Error	z value	Pr(> z)
1					
2	MU REGRESSION				
3	b0 Intercept	1.47148	1.07377	1.37038	0.17057
4	b1 drivers.age	-0.05075	0.03907	-1.29897	0.19395
5	PHI REGRESSION				
6	a0 brand2	-8.64637	2132.15915	-0.00406	0.99676
7	a1 brand3	0.17339	1.50296	0.11536	0.90816
8	OMEGA REGRESSION				
9	g0 abroad	-1.10339	2.46771	-0.44713	0.65478
10					

```

11      Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
12      Iterations      43
13      Log Likelihood    -23.4
14      Pearson Chi Squared      2.5
15      AIC      57
16      Range Mu      0.23      2.45
17      Range Phi      1.00      2.19
18      Range Omega      0.25      0.50
```

```

wald.test <- function(Yin, Xin, Win, Zin, Offset=rep(1,length(Yin)), init=T)
{
  Y <- Yin
  X <- Xin
  W <- Win
  Z <- Zin
  k.beta <- dim(X)[2]
  k.alpha <- dim(W)[2]
  k.gamma <- dim(Z)[2]
  n <- length(Y)

  # Estimate coefficients
  ausgabe <- mle.zigp(Y, X, W, Z, Offset = Offset, summary = FALSE, init=init)

  hat.beta <- ausgabe$Coefficients.Mu
  hat.alpha <- ausgabe$Coefficients.Phi
  hat.gamma <- ausgabe$Coefficients.Omega

  # Compute square root of diagonal elements of FM-1
  B <- FM(hat.beta, hat.alpha, hat.gamma, X, W, Z, Offset = Offset)
  sd.vector <- sqrt( diag(solve(B, tol = 1e-20)) )
  hat.sd.beta <- sd.vector[1:k.beta]
  hat.sd.alpha <- sd.vector[(k.beta+1):(k.beta+k.alpha)]
  hat.sd.gamma <- sd.vector[(k.beta+k.alpha+1):(k.beta+k.alpha+k.gamma)]

  # Compute T-Statistics
  z.stat.beta <- hat.beta/hat.sd.beta
  z.stat.alpha <- hat.alpha/hat.sd.alpha
  z.stat.gamma <- hat.gamma/hat.sd.gamma

  # Compute P-Values
  p.value.beta <- 2*pnorm(-abs(z.stat.beta))
  p.value.alpha <- 2*pnorm(-abs(z.stat.alpha))
  p.value.gamma <- 2*pnorm(-abs(z.stat.gamma))

  # Create ***
  glimpse.beta <- rep("",length(z.stat.beta))
  glimpse.alpha <- rep("",length(z.stat.alpha))
  glimpse.gamma <- rep("",length(z.stat.gamma))
  for (i in 1:length(z.stat.beta)) {
    if (p.value.beta[i] < 0.001) {glimpse.beta[i] <- "***"}
    if (p.value.beta[i] >= 0.001 & p.value.beta[i] < 0.01) {glimpse.beta[i] <- "**"}
    if (p.value.beta[i] >= 0.01 & p.value.beta[i] < 0.05) {glimpse.beta[i] <- "*"}
    if (p.value.beta[i] >= 0.05 & p.value.beta[i] < 0.1) {glimpse.beta[i] <- "."} }
  for (i in 1:length(z.stat.alpha)) {
    if (p.value.alpha[i] < 0.001) {glimpse.alpha[i] <- "***"}
    if (p.value.alpha[i] >= 0.001 & p.value.alpha[i] < 0.01) {glimpse.alpha[i] <- "**"}
    if (p.value.alpha[i] >= 0.01 & p.value.alpha[i] < 0.05) {glimpse.alpha[i] <- "*"}
    if (p.value.alpha[i] >= 0.05 & p.value.alpha[i] < 0.1) {glimpse.alpha[i] <- "."} }
  for (i in 1:length(z.stat.gamma)) {
    if (p.value.gamma[i] < 0.001) {glimpse.gamma[i] <- "***"}
    if (p.value.gamma[i] >= 0.001 & p.value.gamma[i] < 0.01) {glimpse.gamma[i] <- "**"}
    if (p.value.gamma[i] >= 0.01 & p.value.gamma[i] < 0.05) {glimpse.gamma[i] <- "*"}
    if (p.value.gamma[i] >= 0.05 & p.value.gamma[i] < 0.1) {glimpse.gamma[i] <- "."} }

```



```

# Create output
coef.names.beta <- paste("b",c(0:(k.beta-1)),sep="")
coef.names.alpha <- paste("a",c(0:(k.alpha-1)),sep="")
coef.names.gamma <- paste("g",c(0:(k.gamma-1)),sep="")
coef.desc.beta <- double(k.beta)
coef.desc.alpha <- double(k.alpha)
coef.desc.gamma <- double(k.gamma)
for (i in 1:k.beta) {
  coef.desc.beta[i] <- colnames(X, do.NULL=FALSE)[i]
  if (is.matrix(X)) { if (max(X[,i])==1&min(X[,i])==1)
    {coef.desc.beta[i] <- "Intercept"} }
  else{ if (max(X[i])==1&min(X[i])==1) {coef.desc.beta[i] <- "Intercept"} }
}
for (i in 1:k.alpha) {
  coef.desc.alpha[i] <- colnames(W, do.NULL=FALSE)[i]
  if (is.matrix(W)) { if (max(W[,i])==1&min(W[,i])==1)
    {coef.desc.alpha[i] <- "Intercept"} }
  else{ if (max(W[i])==1&min(W[i])==1) {coef.desc.alpha[i] <- "Intercept"} }
}
for (i in 1:k.gamma) {
  coef.desc.gamma[i] <- colnames(Z, do.NULL=FALSE)[i]
  if (is.matrix(Z)) { if (max(Z[,i])==1&min(Z[,i])==1)
    {coef.desc.gamma[i] <- "Intercept"} }
  else{ if (max(Z[i])==1&min(Z[i])==1) {coef.desc.gamma[i] <- "Intercept"} }
}

output <- matrix("",1+k.beta+k.alpha+k.gamma+12,7)
output[2:(1+k.beta+k.alpha+k.gamma+3),1] <-
  c("",coef.names.beta,"", coef.names.alpha,"", coef.names.gamma)
output[2:(1+k.beta+k.alpha+k.gamma+3),2] <-
  c("MU REGRESSION",coef.desc.beta,"PHI REGRESSION", coef.desc.alpha,
    "OMEGA REGRESSION", coef.desc.gamma)
output[1:(1+k.beta+k.alpha+k.gamma+3),3] <-
  c("Estimate","",formatC(hat.beta,5,format="f"),"",formatC(hat.alpha,5,format="f"),
    "",formatC(hat.gamma,5,format="f"))
output[1:(1+k.beta+k.alpha+k.gamma+3),4] <-
  c("Std. Error","",formatC(hat.sd.beta,5,format="f"),"",
    formatC(hat.sd.alpha,5,format="f"),"",formatC(hat.sd.gamma,5,format="f"))
output[1:(1+k.beta+k.alpha+k.gamma+3),5] <-
  c("z value","",formatC(z.stat.beta,5,format="f"),"",
    formatC(z.stat.alpha,5,format="f"),"",formatC(z.stat.gamma,5,format="f"))
output[1:(1+k.beta+k.alpha+k.gamma+3),6] <-
  c("Pr(>|z|)", "",formatC(p.value.beta,5,format="f"), "",
    formatC(p.value.alpha,5,format="f"), "",formatC(p.value.gamma,5,format="f"))
output[1:(1+k.beta+k.alpha+k.gamma+3),7] <-
  c("", "",glimpse.beta,"",glimpse.alpha,"",glimpse.gamma)
output[(1+k.beta+k.alpha+k.gamma+5),2] <- "Signif. codes: 0"
output[(1+k.beta+k.alpha+k.gamma+5),3] <- "'***' 0.001"
output[(1+k.beta+k.alpha+k.gamma+5),4] <- "'**' 0.01"
output[(1+k.beta+k.alpha+k.gamma+5),5] <- "'*' 0.05"
output[(1+k.beta+k.alpha+k.gamma+5),6] <- "'.' 0.1"
output[(1+k.beta+k.alpha+k.gamma+5),7] <- "' ' 1"
output[(1+k.beta+k.alpha+k.gamma+6),2] <- "Iterations"

```

```

output[(1+k.beta+k.alpha+k.gamma+6),4] <- ausgabe$Iterations[1]
output[(1+k.beta+k.alpha+k.gamma+7),2] <- "Log Likelihood"
output[(1+k.beta+k.alpha+k.gamma+7),4] <-
  formatC(ausgabe$Log.Likelihood,digits=1,format="f")
output[(1+k.beta+k.alpha+k.gamma+8),2] <- "Pearson Chi Squared"
output[(1+k.beta+k.alpha+k.gamma+8),4] <- formatC(ausgabe$Pearson,digits=1,format="f")
output[(1+k.beta+k.alpha+k.gamma+9),2] <- "AIC"
output[(1+k.beta+k.alpha+k.gamma+9),4] <- round(ausgabe$AIC)
output[(1+k.beta+k.alpha+k.gamma+10),2] <- "Range Mu"
output[(1+k.beta+k.alpha+k.gamma+10),4] <-
  formatC(ausgabe$Range.Mu[1],digits=2,format="f")
output[(1+k.beta+k.alpha+k.gamma+10),5] <-
  formatC(ausgabe$Range.Mu[2],digits=2,format="f")
output[(1+k.beta+k.alpha+k.gamma+11),2] <- "Range Phi"
output[(1+k.beta+k.alpha+k.gamma+11),4] <-
  formatC(ausgabe$Range.Phi[1],digits=2,format="f")
output[(1+k.beta+k.alpha+k.gamma+11),5] <-
  formatC(ausgabe$Range.Phi[2],digits=2,format="f")
output[(1+k.beta+k.alpha+k.gamma+12),2] <- "Range Omega"
output[(1+k.beta+k.alpha+k.gamma+12),4] <-
  formatC(ausgabe$Range.Omega[1],digits=2,format="f")
output[(1+k.beta+k.alpha+k.gamma+12),5] <-
  formatC(ausgabe$Range.Omega[2],digits=2,format="f")
output2 <- data.frame(output)
return(output2)
}

```

[02] *mle.zigp*

Maximum Likelihood Estimates

Description:

'mle.zigp' is used to calculate the MLEs of the regression parameters for mean, overdispersion and zero inflation.

Needs additional Functions:

```
'log.likelihood.zigp'
'fit.zigp'
'gradient'
'summary.zigp'
'optimized.run'
# for usage with 'init = T' needs external functions (in 'R' code!)
'mle.zigp.full.like'
'log.likelihood.zigp.full'
'fit.zigp1'
'summary.zigp1'
```

Usage:

```
mle.zigp(Yin, Xin, Win, Zin, Offset = rep(1,length(Yin)),
summary = TRUE, init = FALSE)
```

Arguments:

Yin: response vector of length n.

Xin: design matrix of dim (n x p) for mean modelling.

Win: design matrix of dim (n x r) for overdispersion modelling.

Zin: design matrix of dim (n x q) for zero inflation modelling.

Offset: exposure for individual observation lengths. Defaults to a vector of 1.
The offset MUST NOT be in 'log' scale.

summary: a logical value indicating whether a list summary is shown (=T) or returned to the user (=F), i.e. if values of the ML estimation should be extracted, summary has to be set to 'F'. Defaults to 'T'.

init: a logical value indicating whether initial optimization values for dispersion are set to -2.5 and values for zero inflation regression parameters are set to -1 (init = F) or are estimated by a ZIGP(μ_i , ϕ , ω)-model (init = T). Defaults to 'T'.

Value:

If summary is set to 'F', the following values are returned:

ZI.Parameter	Zero Inflation estimate
--------------	-------------------------

Coefficients.Mu	Regression Parameters estimates for mean
Coefficients.Phi	Regression Parameters estimates for overdispersion
Coefficients.Omega	Regression Parameters estimates for ZI
Dispersion.Parameter	Dispersion Parameter Estimate
Range.Mu	mean range
Range.Phi	dispersion parameter range
Range.Omega	ZI parameter range
Log.Likelihood	log likelihood value at MLE
Residuals	residuals
Pearson	Pearson Chi Squared Statistics
AIC	Akaike Information Criterion
Iterations	number of iterations needed
Time	time needed
Message	convergence message of underlying 'optim' routine
Response	response vector
Fitted.Values	fitted values
Design.Mu	design matrix for mean
Design.Phi	design matrix for overdispersion
Design.Omega	design matrix for ZI

See Also:

```
wald.test().
```

Example:

```
## Number of damages in car insurance.

damage <- c(0,1,0,0,0,4,2,0,1,0,1,1,0,2,0,0,1,0,0,1,0,0,0)
Intercept <- rep(1,length(damage))
insurance.year <- c(1,1.2,0.8,1,2,1,1.1,1,1,1.1,1.2,1.3,0.9,1.4,1,1,1,1.2,1,
1,1,1,1)
drivers.age <- c(25,19,30,48,30,18,19,29,24,54,56,20,38,18,23,58,
47,36,25,28,38,39,42)
# for overdispersion: car brand dummy in {1,2,3}, brand = 1 is reference
brand <- c(1,2,1,3,3,2,2,1,1,3,2,2,1,3,1,3,2,2,1,1,3,3,2)
brand2 <- ifelse(brand==2,1,0)
brand3 <- ifelse(brand==3,1,0)
W <- t(rbind(brand2,brand3))
# abroad: driver has been abroad for longer time (=1)
abroad <- c(0,0,0,1,0,0,1,0,0,0,0,0,1,0,0,0,0,1,0,1,1,1,1)
Y <- damage
X <- cbind(Intercept, drivers.age)
Z <- cbind(abroad)

mle.zigp(Yin=Y, Xin=X, Win=W, Zin=Z, Offset = insurance.year,
summary = T, init = F)

[1] Range for ZI-Parameters: 0.2491062 0.5
[2] Range of Dispersion Pars: 1.000176 2.189325
[3] Coefficients for mu: 1.471478 -0.05075418
[4] Coefficients for phi: -8.646371 0.1733860
[5] Coefficients for omega: -1.103385
[6] Pearson Chi Squared: 2.501088
```

[7]	Range of μ :	0.2294054 2.445806
[8]	Message:	"NULL"
[9]	AIC:	56.88305

```

mle.zigp <- function(Yin, Xin, Win, Zin, Offset = rep(1,length(Yin)), summary = TRUE,
                     init = FALSE)
{
  Y <- Yin
  X <- Xin
  W <- Win
  Z <- Zin
  if(is.matrix(X)) {
    n <- dim(X)[1]
    k.beta <- dim(X)[2]
  }
  else {
    n <- length(X)
    k.beta <- 1
  }
  if(is.matrix(W)) {
    k.alpha <- dim(W)[2]
    nw <- dim(W)[1]
  }
  else {
    k.alpha <- 1
    nw <- length(W)
  }
  if(is.matrix(Z)) {
    k.gamma <- dim(Z)[2]
    nz <- dim(Z)[1]
  }
  else {
    k.gamma <- 1
    nz <- length(Z)
  }

  if(nz==n & nw==n){
    t.i <- Offset

    if(init){ #optimized initial values
      start.delta <- optimized.run(Y,X,W,Z)

      # maximization of log likelihood
      start <- proc.time()[2]
      opt <- optim(par = start.delta, fn = log.likelihood.zigp, gr=gradient,
                  method = "BFGS")
      # method = "CG")
    }
    else
    {
      # simple initial values
      alpha <- rep(-2.5,k.alpha)
      gamma <- rep(-2.5,k.gamma)
      beta <- summary(glm(Y ~ offset(log(t.i)) + 1 + X, family = poisson(link=log)
                          ))$coefficients[, 1]
      start.delta <- c(beta,alpha,gamma)
      # maximization of log likelihood w.r.t. delta s.t. above limits hold
      start <- proc.time()[2]
    }
  }
}

```

```

    opt <- optim(par = start.delta, fn = log.likelihood.zigp, gr=gradient,
    method = "BFGS")
    # method = "CG")
  }

  delta <- opt$par
  it <- opt$counts
  log.likelihood <- - opt$value
  if(is.null(opt$message)) {message <- "NULL"}
  else { message <- opt$message }
  AIC <- -2 * log.likelihood + 2 * (k.beta + k.alpha + k.gamma)
  time.needed <- proc.time()[2] - start

# results
coef <- double(k.beta + k.alpha + k.gamma)
coef <- delta
beta <- coef[1 : k.beta]
alpha <- coef[(k.beta + 1) : (k.beta + k.alpha)]
gamma <- coef[(k.beta + k.alpha + 1) : (k.beta + k.alpha + k.gamma)]

fit <- fit.zigp(delta)
res <- Y - fit$fit
chsqr <- sum((Y - fit$fit) / sqrt((1-fit$omega)*fit$phi*fit$fit+
    fit$omega*(1-fit$omega)*fit$fit^2))

range.mu <- c(min(fit$mu), max(fit$mu))
# compute zero inflation parameters omega
if(is.matrix(Z)){ eta.omega <- Z %*% gamma }
else { eta.omega <- Z * gamma }
omega <- exp(eta.omega)/(1+exp(eta.omega))
range.omega <- c(min(omega), max(omega))
# compute dispersion parameters phi
if(is.matrix(W)){ eta.phi <- W %*% alpha }
else { eta.phi <- W * alpha }
phi <- 1+ exp(eta.phi)
range.phi <- c(min(phi), max(phi))

# summarize results
mle.data <- list(ZI.Parameter = omega,
  Coefficients.Mu = beta,
  Coefficients.Phi = alpha,
  Coefficients.Omega = gamma,
  Dispersion.Parameter = phi,
  Range.Mu = range.mu,
  Range.Phi = range.phi,
  Range.Omega = range.omega,
  Log.Likelihood = log.likelihood,
  Residuals = res,
  Pearson = chsq,
  AIC = AIC,
  Iterations = it,
  Time = time.needed,
  Message = message,
  Response = Y,

```

```
Fitted.Values = fit$fit,
Design.Mu = X,
Design.Phi = W,
Design.Omega = Z)
if(summary) {
  summary.zigp(mle.data)
}
else {
  return(mle.data)
}
}
else{
return("X, W and Z have different numbers of rows!")
}
}
```


Hilfsfunktionen (werden i.d.R. nicht direkt vom Benutzer ausgeführt)

[03] *rzigp*

ZIGP random variable generator

Description:

‘rzigp’ generates ZIGP random variables using the inversion method.

Usage:

```
rzigp(n, mu, phi, omega)
```

Arguments:

n: length of output random vector.

mu: mean

phi: dispersion parameter

omega: zero inflation parameter

Value:

Generates a ZIGP random vector of length n.

See Also:

```
response.zigp().
```

Example:

```
# set seed for random variable generator
set.seed(2)
rzigp(3, 2, 1.5, 0.2)
[1] 3 1 0
```

```

rzigp <- function(n, mu = stop("no mu arg"), phi = stop("no phi arg"),
  omega = stop("no omega arg"))
{
  # check if parameters are valid
  if(omega < 0) {return("omega has to be in [0,1]!")}
  if(omega > 1) {return("omega has to be in [0,1]!")}

  # inversion method
  x <- double(n)
  for(i in 1:n) {
    y <- integer(1)
    c <- double(1)
    #P(y=0)
    p <- omega + (1-omega) * exp( - mu/phi)
    s <- p
    u <- runif(1, 0, 1)
    if (u > s){
      #P(y=1)
      p <- (1-omega)*mu/phi*exp(-1/phi*(mu+phi-1))
      s <- s + p
      y <- 1
      while (u > s) {
        y <- y + 1
        p <- (mu + (phi-1)*y)^(y-1) / (mu + (phi-1)*(y-1))^(y-2) / (y * phi) *
          exp((1-phi)/phi) * p
        s <- s + p
      }
    }
    x[i] <- y
  }
  return(x)
}

```

[04] *response.zigp*

ZIGP response generator

Description:

'response.zigp' generates a ZIGP response vector for a given setting of linear predictors.

Needs additional Functions:

'rzigp'

Usage:

```
response.zigp(X, W, Z, beta, alpha, gamma)
```

Arguments:

X: design matrix of dim (n x p) for modelling of mean

W: design matrix of dim (n x r) for modelling of overdispersion

Z: design matrix of dim (n x q) for modelling of zero inflation

beta: regression parameters for mean of length p

alpha: regression parameters for overdispersion of length r

gamma: regression parameters for zero inflation of length q

Value:

Generates a ZIGP response vector of length n.

See Also:

rzigp().

Example:

```
X <- matrix(c(1:3,4,3,5),3,2)
W <- c(3,-4,-1)
Z <- rep(1,3)
beta <- c(5,-2)
alpha <- 3.4
gamma <- -10
# set seed for random variable generator
set.seed(1)
response.zigp(X,W,Z,beta,alpha,gamma)
[1] 0 55 132
```

```

"response.zigp" <- function(X, W, Z, beta, alpha, gamma)
{
# get number of observations from X, W and Z
  if(is.matrix(X)) {
    nx <- dim(X)[1]
  }
  else {
    nx <- length(X)
  }
  if(is.matrix(W)) {
    nw <- dim(W)[1]
  }
  else {
    nw <- length(W)
  }
  if(is.matrix(Z)) {
    nz <- dim(Z)[1]
  }
  else {
    nz <- length(Z)
  }

  if (nx==nz&nx==nw) {
    n <- nx
    mu <- double(n)
    phi <- double(n)
    omega <- double(n)
    Y <- double(n)

    if(is.matrix(X)) {
      mu <- exp(X%*%beta)
    }
    else {
      mu <- exp(X*beta)
    }
    if(is.matrix(W)) {
      phi <- 1 + exp(W%*%alpha)
    }
    else {
      phi <- 1 + exp(W*alpha)
    }
    if(is.matrix(Z)) {
      omega <- exp(Z%*%gamma)/(1+exp(Z%*%gamma))
    }
    else {
      omega <- exp(Z*gamma)/(1+exp(Z*gamma))
    }
# create ZIGP response for all i
    for(i in 1:n) {
      Y[i] <- rzigp(1, mu[i], phi[i], omega[i])
    }
    return(Y)
  }
  else {

```

```
    print("X, W and Z do not have the same numbers of rows!")  
  }  
}
```

[05] *log.likelihood.zigp*

Log Likelihood of ZIGP distribution

Description:

'log.likelihood.zigp' calculates the log likelihood function of the ZIGP distribution.

Usage:

```
log.likelihood.zigp(delta)
```

Arguments:

delta: a parameter vector of length (p+r+q) with $\dim(X) = (n \times p)$, $\dim(W) = (n \times r)$, $\dim(Z) = (n \times q)$. Create delta by pasting 'delta <- c(beta, alpha, gamma)'.
 beta is the vector of regression parameters for the mean modelling.
 alpha is the vector of regression parameters for overdispersion modelling.
 gamma is the vector of regression parameters for the ZI modelling.

Details:

The design matrices have to be defined as X (for mean), W (for overdispersion) and Z (for ZI).
 n has to be defined as the number of observations $\dim(X)[1]$.
 k.beta has to be defined as the length of beta.
 k.alpha has to be defined as the length of alpha.
 k.gamma has to be defined as the length of gamma.
 Y has to be defined as the response vector.
 t.i has to be defined as the exposure.

Example:

```
Y <- c(3,0,2)
X <- matrix(c(1:3,4,3,5),3,2)
W <- c(3,-4,-1)
Z <- rep(1,3)
n <- dim(X)[1]
t.i <- rep(1,n)
beta <- c(5,-2)
alpha <- 3.4
gamma <- -10
k.beta <- length(beta)
k.gamma <- length(gamma)
delta <- c(beta,alpha,gamma)
log.likelihood.zigp(delta)
[1] 160.2377
```

```

log.likelihood.zigp <- function(delta)
{
  eta.mu <- double(n)
  eta.phi <- double(n)
  eta.omega <- double(n)
  s1 <- double(1)
  s2 <- double(1)

  if(k.beta == 1) {
    eta.mu <- X * delta[1]
  }
  else {
    beta <- delta[1:k.beta]
    eta.mu <- X %*% beta
  }

  if(k.alpha == 1) {
    eta.phi <- W * delta[k.beta + 1]
  }
  else {
    alpha <- delta[(k.beta + 1) : (k.beta + k.alpha)]
    eta.phi <- W %*% alpha
  }

  if(k.gamma == 1) {
    eta.omega <- Z * delta[k.beta + k.alpha + 1]
  }
  else {
    gamma <- delta[(k.beta + k.alpha + 1) : (k.beta + k.alpha + k.gamma)]
    eta.omega <- Z %*% gamma
  }

  mu.i <- t.i*exp(eta.mu)
  b.i <- exp(eta.phi)
  phi.i <- 1 + b.i
  k.i <- exp(eta.omega)
  temp1 <- (mu.i + b.i*Y)

  # terms of log likelihood with 1_{Y_i = 0}
  s1 <- sum(ifelse(Y == 0,1,0)* (log(k.i+exp(-1/phi.i*mu.i)) -
    log(1+k.i)))
  # terms of log likelihood with 1_{Y_i > 0} (excluding factorial)
  s2 <- sum(ifelse(Y>0,1,0)*(-log(1+k.i)+log(t.i)+eta.mu+(Y-1)*log(temp1)-
    Y*log(phi.i)-1/phi.i*temp1) - lgamma(Y+1))

  l <- s1 + s2
  return( - l)
}

```

[06] *fit.zigp*

Fitted Values

Description:

'fit.zigp' computes fitted values.

Usage:

```
fit.zigp(delta)
```

Arguments:

delta: a parameter vector of length (p+r+q) with $\dim(X) = (n \times p)$, $\dim(W) = (n \times r)$, $\dim(Z) = (n \times q)$. Create delta by pasting 'delta <- c(beta, alpha, gamma)'.
 beta is the vector of regression parameters for the mean modelling.
 alpha is the vector of regression parameters for overdispersion modelling.
 gamma is the vector of regression parameters for the ZI modelling.

Details:

The design matrices have to be defined as X (for mean), W (for overdispersion) and Z (for ZI).
 n has to be defined as the number of observations $\dim(X)[1]$.
 k.beta has to be defined as the length of beta.
 k.alpha has to be defined as the length of alpha.
 k.gamma has to be defined as the length of gamma.
 t.i has to be defined as the exposure.

Example:

```
X <- matrix(c(1:3,4,3,5),3,2)
W <- c(3,-4,-1)
Z <- rep(1,3)
n <- dim(X)[1]
beta <- c(5,-2)
alpha <- 3.4
gamma <- -10
k.beta <- length(beta)
k.gamma <- length(gamma)
t.i <- rep(1,n)
delta <- c(beta,alpha,gamma)
fit.zigp(delta)
[1] $fit
[2]      [,1]
[3] [1,]  0.04978481
[4] [2,]  54.59567139
[5] [3,] 148.40642146
[6] $mu
[7]      [,1]
[8] [1,]  0.04978707
[9] [2,]  54.59815003
```



```
[10] [3,] 148.41315910
[11] $phi
[12] [1] 26904.186074      1.000001      1.033373
[13] $omega
[14] [1] 4.539787e-05 4.539787e-05 4.539787e-05
```

```

fit.zigp <- function(delta)
{
  eta.mu <- double(n)
  eta.phi <- double(n)
  eta.omega <- double(n)
  mu <- double(n)
  phi <- double(n)
  omega <- double(n)
  fit <- double(n)

  if(k.beta == 1) {
    eta.mu <- X * delta[1]
  }
  else {
    beta <- delta[1:k.beta]
    eta.mu <- X %*% beta
  }
  if(k.alpha == 1) {
    eta.phi <- W * delta[k.beta + 1]
  }
  else {
    alpha <- delta[(k.beta + 1) : (k.beta + k.alpha)]
    eta.phi <- W %*% alpha
  }
  if(k.gamma == 1) {
    eta.omega <- Z * delta[k.beta + k.alpha + 1]
  }
  else {
    gamma <- delta[(k.beta + k.alpha + 1) : (k.beta + k.alpha + k.gamma)]
    eta.omega <- Z %*% gamma
  }

  # computation of mu, phi, omega and mean
  mu <- t.i*exp(eta.mu)
  phi <- 1 + exp(eta.phi)
  omega <- exp(eta.omega)/(1+exp(eta.omega))
  fit <- (1 - omega) * mu

  return(list(fit = fit, mu = mu, phi = phi, omega = omega))
}

```

[07] *summary.zigp*

Summary

Description:

‘summary.zigp’ summarizes the results of the maximum likelihood estimation.

Usage:

```
summary.zigp(mle.data.in)
```

Arguments:

mle.data.in: a list of result parameters given by function ‘mle.zigp’

```

summary.zigp <- function(mle.data.in)
{
  X <- mle.data.in$Design.Mu
  if(is.matrix(X)) {
    nx <- dim(X)[1]
    k.beta <- dim(X)[2]
  }
  else {
    nx <- length(X)
    k.beta <- 1
  }
  W <- mle.data.in$Design.Phi
  if(is.matrix(W)) {
    nw <- dim(W)[1]
    k.alpha <- dim(W)[2]
  }
  else {
    nw <- length(W)
    k.alpha <- 1
  }
  Z <- mle.data.in$Design.Omega
  if(is.matrix(Z)) {
    nz <- dim(Z)[1]
    k.gamma <- dim(Z)[2]
  }
  else {
    nz <- length(Z)
    k.gamma <- 1
  }

  # range of ZI parameters
  out0 <- matrix(double(2), 1, 2)
  out0[1, 1] <- min(mle.data.in$ZI.Parameter)
  out0[1, 2] <- max(mle.data.in$ZI.Parameter)
  colnames(out0) <- c("", "")
  rownames(out0) <- c("Range for ZI-Parameters: ")
  print(out0)
  # range of Dispersion Parameters
  out1 <- matrix(double(2), 1, 2)
  out1[1, 1] <- min(mle.data.in$Dispersion.Parameter)
  out1[1, 2] <- max(mle.data.in$Dispersion.Parameter)
  colnames(out1) <- c("", "")
  rownames(out1) <- c("Range of Dispersion Pars: ")
  print(out1)
  # beta
  out2 <- matrix(double(k.beta), 1, k.beta)
  kopf <- rep("", k.beta)
  for(i in 1:k.beta) {
    out2[1, i] <- mle.data.in$Coefficients.Mu[i]
  }
  colnames(out2) <- kopf
  rownames(out2) <- c("Coefficients for mu: ")
  print(out2)
  # alpha

```

```

out3 <- matrix(double(k.alpha), 1, k.alpha)
kopf <- rep("", k.alpha)
for(i in 1:k.alpha) {
  out3[1, i] <- mle.data.in$Coefficients.Phi[i]
}
colnames(out3) <- kopf
rownames(out3) <- c("Coefficients for phi:  ")
print(out3)
# gamma
out4 <- matrix(double(k.gamma), 1, k.gamma)
kopf <- rep("", k.gamma)
for(i in 1:k.gamma) {
  out4[1, i] <- mle.data.in$Coefficients.Omega[i]
}
colnames(out4) <- kopf
rownames(out4) <- c("Coefficients for omega:  ")
print(out4)
# Pearson Chi Squared
out5 <- matrix(double(1), 1, 1)
out5[1, 1] <- mle.data.in$Pearson
colnames(out5) <- c("")
rownames(out5) <- c("Pearson Chi Squared:  ")
print(out5)
# range of mu
out6 <- matrix(double(2), 1, 2)
out6[1, 1] <- mle.data.in$Range.Mu[1]
out6[1, 2] <- mle.data.in$Range.Mu[2]
colnames(out6) <- c("", "")
rownames(out6) <- c("Range of mu:  ")
print(out6)
# optimization message
out7 <- matrix(double(1), 1, 1)
out7[1, 1] <- mle.data.in$Message
colnames(out7) <- c("")
rownames(out7) <- c("Message:  ")
print(out7)
# AIC
out8 <- matrix(double(1), 1, 1)
out8[1, 1] <- mle.data.in$AIC
colnames(out8) <- c("")
rownames(out8) <- c("AIC:  ")
print(out8)
return()
}

```

[08] *gradient*

Die explizite Berechnung eines Gradienten ist zwar nicht notwendig, da für die 'optim'-Methode "CG" Gradienten gegebenenfalls numerisch geschätzt werden. Allerdings erhöhen explizite Gradienten die Konvergenzgeschwindigkeit enorm.

Gradient of log likelihood

Description:

'gradient' calculates the gradient of the log likelihood function.

Usage:

```
gradient(delta)
```

Arguments:

delta: a parameter vector of length (p+r+q) with $\dim(X) = (n \times p)$, $\dim(W) = (n \times r)$, $\dim(Z) = (n \times q)$. Create delta by pasting 'delta <- c(beta, alpha, gamma)'.
 beta is the vector of regression parameters for the mean modelling.
 alpha is the vector of regression parameters for overdispersion modelling.
 gamma is the vector of regression parameters for the ZI modelling.

Details:

This function used for the 'optim' method. Some 'optim' methods need a gradient and have to approximate it in every iteration step. This slows down the process a lot. Therefore, an explicit gradient function increases convergence speed and ensures correct convergence even if the log likelihood function is not continuous due to the indicator function of the ZI parameter.

The design matrices have to be defined as X (for mean), W (for overdispersion) and Z (for ZI).

n has to be defined as the number of observations $\dim(X)[1]$.

k.beta has to be defined as the length of beta.

k.alpha has to be defined as the length of alpha.

k.gamma has to be defined as the length of gamma.

t.i has to be defined as the exposure.

Example:

```
Y <- c(3,0,2)
X <- matrix(c(1:3,4,3,5),3,2)
W <- c(3,-4,-1)
Z <- rep(1,3)
n <- dim(X)[1]
beta <- c(5,-2)
alpha <- 3.4
gamma <- -10
k.beta <- length(beta)
k.gamma <- length(gamma)
t.i <- rep(1,n)
```

```
delta <- c(beta,alpha,gamma)
gradient(delta)
[1] 423.8616107 704.1026859 7.5116293 -0.9998638
```

```

gradient <- function(delta)
{
  grad <- double(k.beta + k.alpha + k.gamma)
  eta.mu <- double(n)
  eta.phi <- double(n)
  eta.omega <- double(n)
  s1 <- double(1)
  s2 <- double(1)

  if(k.beta == 1) {
    eta.mu <- X * delta[1]
    X <- cbind(X, rep(0, length(X)))
  }
  else {
    beta <- delta[1:k.beta]
    eta.mu <- X %*% beta
  }
  if(k.alpha == 1) {
    eta.phi <- W * delta[k.beta + 1]
    W <- cbind(W, rep(0, length(W)))
  }
  else {
    alpha <- delta[(k.beta + 1) : (k.beta + k.alpha)]
    eta.phi <- W %*% alpha
  }
  if(k.gamma == 1) {
    eta.omega <- Z * delta[k.beta + k.alpha + 1]
    Z <- cbind(Z, rep(0, length(Z)))
  }
  else {
    gamma <- delta[(k.beta + k.alpha + 1) : (k.beta + k.alpha + k.gamma)]
    eta.omega <- Z %*% gamma
  }

  mu.i <- t.i*exp(eta.mu)
  k.i <- exp(eta.omega)
  b.i <- exp(eta.phi)
  phi.i <- 1 + b.i
  P0 <- exp(-1/phi.i*mu.i)

  # Derivative for beta
  for (j in 1:k.beta)
  {
    # terms of gradient with 1_{Y_i = 0}
    s1 <- sum(ifelse(Y == 0, 1, 0) * X[,j] * (P0 * (-1/phi.i) * mu.i / (k.i + P0)))
    # terms of gradient with 1_{Y_i > 0}
    s2 <- sum(ifelse(Y > 0, 1, 0) * X[,j] * (1 + (Y-1) * mu.i / (mu.i + b.i * Y) - 1/phi.i * mu.i))
    grad[j] <- s1 + s2
  }

  # Derivative for alpha
  for (j in 1:k.alpha)
  {

```



```

# terms of gradient with 1_{Y_i = 0}
  s1 <- sum(ifelse(Y == 0,1,0)*W[,j]*b.i*(1/phi.i^2*P0*mu.i/(k.i+P0)))
# terms of gradient with 1_{Y_i > 0}
  s2 <- sum(ifelse(Y>0,1,0)*W[,j]*b.i*(Y*(Y-1)/(mu.i + b.i*Y)-Y/phi.i+
    (mu.i-Y)/phi.i^2))
  grad[k.beta+j] <- s1 + s2
}

# Derivative for gamma
for (j in 1:k.gamma)
{
# terms of gradient with 1_{Y_i = 0}
  s1 <- sum(ifelse(Y == 0,1,0)* Z[,j]* k.i * (1/(k.i + P0)))
# terms of gradient independent of Y_i
  s2 <- -sum(Z[,j]*k.i/(1+k.i))
  grad[k.beta+k.alpha+j] <- s1 + s2
}

return(-grad)
}

```

[09] *optimized.run*

Calculation of valid initial values

Description:

'optimized.run' calculates valid initial values for the 'optim' method.

Needs additional Functions:

```
# needs external functions (in 'R' code!)
'mle.zigp.full.like'
'log.likelihood.zigp.full'
'fit.zigp1'
'summary.zigp1'
```

Usage:

```
optimized.run(Y,X,W,Z)
```

Arguments:

Yin: response vector of length n.

Xin: design matrix of dim (n x p) for mean modelling.

Win: design matrix of dim (n x r) for overdispersion modelling.

Zin: design matrix of dim (n x q) for zero inflation modelling.

Details:

This function uses ''R'' routines for ZIGP(μ_i, ϕ_i, ω_i) regression. Therefore, compile
'mle.zigp.full.like'
'log.likelihood.zigp.full'
'fit.zigp1'
'summary.zigp1'.

t.i has to be defined as the exposure.

n has to be defined as the number of observations dim(X)[1].

Example:

```
Y <- c(3,0,2)
X <- matrix(c(1:3,4,3,5),3,2)
W <- c(3,-4,-1)
Z <- rep(1,3)
n <- dim(X)[1]
t.i <- rep(1,n)
optimized.run(Y,X,W,Z)
[1]          X1          X2          W          Z
[2]  -0.6783368  0.4559966  0.1238029 -1.3862944
```

```

optimized.run <- function(Y,X,W,Z)
{

# improved initial values
# get initial beta, phi & omega
out <- mle.zigp.full.like(Y, X, Offset = t.i, summary=FALSE)
beta.start <- out$Coefficients
phi.first <- out$Dispersion.Parameter
omega.first <- out$ZI.Parameter

# LM for alpha
rechte.seite <- rep(log(phi.first - 1),n)
out<-lm(rechte.seite ~ W-1)
alpha.start <- out$coefficients

# LM for gamma
rechte.seite <- rep(log(omega.first)-log(1-omega.first),n)
out<-lm(rechte.seite ~ Z-1)
gamma.start <- out$coefficients
rm(rechte.seite,out)

start.delta <- c(beta.start,alpha.start,gamma.start)
rm(beta.start,alpha.start,gamma.start)

return(start.delta)
}

```

[10] *FM*

Fisher Information

Description:

‘FM’ calculates the (Expected) Fisher Information matrix.

Usage:

```
FM(beta, alpha, gamma, X, W, Z, Offset = NULL)
```

Arguments:

beta: regression parameters for mean of length p

alpha: regression parameters for overdispersion of length r

gamma: regression parameters for zero inflation of length q

X: design matrix of dim (n x p) for mean modelling.

W: design matrix of dim (n x r) for overdispersion modelling.

Z: design matrix of dim (n x q) for zero inflation modelling.

Offset: exposure for individual observation lengths. Defaults to a vector of 1.
The offset MUST NOT be in ‘log’ scale.

Example:

```
X <- matrix(c(1:3,4,3,5),3,2)
W <- c(3,-4,-1)
Z <- rep(1,3)
beta <- c(5,-2)
alpha <- 3.4
gamma <- -10
FM(beta, alpha, gamma, X, W, Z)
      [,1]      [,2]      [,3]      [,4]
[1,] 1.469180e+03 2.412237e+03 6.249334e-03 -8.400641e-11
[2,] 2.412237e+03 3.965799e+03 1.040260e-02 -3.360257e-10
[3,] 6.249334e-03 1.040260e-02 2.101615e-03 2.520099e-10
[4,] -8.400641e-11 -3.360257e-10 2.520099e-10 9.079162e-05
```

```

FM <- function(beta, alpha, gamma, X, W, Z, Offset = NULL)
{

k.beta <- length(beta)
k.alpha <- length(alpha)
k.gamma <- length(gamma)

FM <- array(0,c(k.beta+k.alpha+k.gamma,k.beta+k.alpha+k.gamma))

if(is.matrix(X)){ eta.mu <- X%*%beta }
else{eta.mu <- X*beta}
if(is.matrix(W)){ eta.phi <- W%*%alpha }
else{eta.phi <- W*alpha}
if(is.matrix(Z)){ eta.gamma <- Z%*%gamma }
else{eta.gamma <- Z*gamma}

if (is.null(Offset)) {mu <- exp(eta.mu)}
else {t.i <-< Offset
      mu <- t.i*exp(eta.mu)}
b <- exp(eta.phi)
phi <- 1+b
k <- exp(eta.gamma)
P0 <- exp(-1/phi*mu)
# a_i = E[I_{Y_i = 0}]
a <- (k+P0)/(1+k)

temp <- double(1)

if(k.beta==1){ X <- cbind(X,rep(0,length(X))) }
if(k.alpha==1){ W <- cbind(W,rep(0,length(W))) }
if(k.gamma==1){ Z <- cbind(Z,rep(0,length(Z))) }

# the second derivative w.r.t. "beta"
for (i in 1:k.beta)
{
  for (j in 1:k.beta)
  {
    temp <- sum( X[,i]*X[,j]*mu* ( a*(-1/phi*P0^2+(mu-phi)/(phi^2)*P0*k)/
      ((k+P0)^2) + (b*mu)/(phi^2*(mu-2+2*phi)*(1+k)) - (1-a)/phi ) )
    FM[i,j] <- -temp
  }
}

# the second derivative w.r.t. "alpha"
for (i in 1:k.alpha)
{
  for (j in 1:k.alpha)
  {
    temp <- sum( W[,i]*W[,j]*b*( a*P0*mu/((k+P0)^2)*(mu*b*k/phi^4+(1/phi^2-2*b/phi^3)*
      (k+P0))+mu^2/(phi^2*(mu-2+2*phi)*(1+k)) + 2*mu/(1+k)*(-1/phi^2+b/phi^3) +
      (1-a)/(phi^3)*(mu*phi-2*mu*b) ) )
    FM[k.beta+i,k.beta+j] <- -temp
  }
}
}

```

```

# the second derivative w.r.t. "gamma"
for (i in 1:k.gamma)
{
  for (j in 1:k.gamma)
  {
    temp <- sum( Z[,i]*Z[,j]*k* (a*P0/((k+P0)^2)-1/((1+k)^2) ) )
    FM[k.beta+k.alpha+i,k.beta+k.alpha+j] <- -temp
  }
}

# the mixed derivative w.r.t. "beta" and "alpha"
for (i in 1:k.beta)
{
  for (j in 1:k.alpha)
  {
    temp <- sum( X[,i]*W[,j]*mu*b* ( a*(-1/phi^3*P0*mu*k+1/(phi^2)*P0*(k+P0))/((k+P0)^2)-
      mu/(phi^2*(mu-2+2*phi)*(1+k))+(1-a)/(phi^2) ) )
    FM[k.beta+j,i] <- -temp
    FM[i,k.beta+j] <- -temp
  }
}

# the mixed derivative w.r.t. "beta" and "gamma"
for (i in 1:k.beta)
{
  for (j in 1:k.gamma)
  {
    temp <- sum( X[,i]*Z[,j]*a*P0*mu*k/(phi*(k+P0)^2) )
    FM[k.beta+k.alpha+j,i] <- -temp
    FM[i,k.beta+k.alpha+j] <- -temp
  }
}

# the mixed derivative w.r.t. "alpha" and "gamma"
for (i in 1:k.gamma)
{
  for (j in 1:k.alpha)
  {
    temp <- sum( W[,j]*Z[,i]*b*a* (-1/phi^2 * P0 * mu * k)/((k+P0)^2) )
    FM[k.beta+k.alpha+i,k.beta+j] <- -temp
    FM[k.beta+j,k.beta+k.alpha+i] <- -temp
  }
}

return(FM)
}

```

Anhang C

Software für das $ZMGP(\mu_i, \varphi, \omega_i)$ –Regressionsmodell

Übersicht über die verwendeten Funktionen

	FUNKTIONSNAME	ZWECK	SEITE
[01]	<i>wald.test</i>	Paßt ein $ZMGP(\mu_i, \varphi, \omega_i)$ –Regressionsmodell an einen Datensatz an und gibt ein Summary mit Wald-Teststatistiken aus.	S. 212
[02]	<i>mle.zmgp</i>	Berechnet die ML-Schätzer in einem $ZMGP(\mu_i, \varphi, \omega_i)$ –Regressionsmodell.	S. 216
[03]	<i>rzmgp</i>	Erzeugt eine ZMGP-Zufallsvariable mit der Invertierungsmethode.	S. 220
[04]	<i>response.zmgp</i>	Erzeugt einen ZMGP-Zufallsvektor aus vorgegebenen linearen Prädiktoren.	S. 222
[05]	<i>log.likelihood.zmgp</i>	Berechnet die Loglikelihoodfunktion in einem $ZMGP(\mu_i, \varphi, \omega_i)$ –Regressionsmodell.	S. 224
[06]	<i>fit.zmgp</i>	Berechnet die 'fitted values' des angepaßten Modells.	S. 226
[07]	<i>summary.zmgp</i>	Gibt eine Auflistung aller Inputs und Outputs der Modellanpassung zurück.	S. 228
[08]	<i>gradient</i>	Berechnet den Gradienten der Loglikelihoodfunktion.	S. 230
[09]	<i>optimized.run</i>	Berechnet einen gültigen Startwert der Loglikelihoodfunktion für die 'Optim'-Methode.	S. 233
[10]	<i>FM</i>	Berechnet die erwartete Fisherinformation.	S. 235

Tabelle C.1: Übersicht über die "R"-Funktionen und deren Zweck

Hauptfunktionen (können direkt vom Benutzer ausgeführt werden)[01] *wald.test*

Fitting ZMGP(mu_i, phi, omega_i) - Regression Models

Description:

`'wald.test'` is used to fit ZMGP(mu_i, phi, omega_i) - Regression Models.

Needs additional Functions:

```

'mle.zmgp'
'FM'
'log.likelihood.zmgp'
'fit.zmgp'
'gradient'
'summary.zmgp'
'optimized.run'
# for usage with 'init = T' needs external functions (in 'R' code!)
'mle.zigp.full.like'
'log.likelihood.zigp.full'
'fit.zigp1'
'summary.zigp1'

```

Usage:

```
wald.test(Yin, Xin, Zin, Offset=rep(1,length(Yin)), init=T, K.mu = NULL)
```

Arguments:

Yin: response vector of length n.

Xin: design matrix of dim (n x p) for mean modelling.

Zin: design matrix of dim (n x q) for zero inflation modelling.

Offset: exposure for individual observation lengths. Defaults to a vector of 1.
The offset MUST NOT be in 'log' scale.

init: a logical value indicating whether initial optimization values for zero
inflation regression parameters are set to -1 (init = F) or are estimated
by a ZIGP(mu_i, phi, omega)-model (init = T). Defaults to 'T'.

K.mu: upper boundary for mu. If unknown, an K.mu is set to 2 * max(Yin).
Defaults to NULL, i.e. 'unknown'.

Details:

In order to include an intercept in a design matrix, one has to
add a vector of ones to the design matrix:
`'Intercept <- rep(1,n)'`.

If the output should have variable names additionally to parameter tokens (such as 'b0' or 'g0'), create the design matrix by
'Z <- cbind(Intercept, gender, height)'.

Zero Modified Models allow for negative ZI-Parameters ω_i in [Kappa,1].
Kappa is given in the output.

Example:

```
## Number of damages in car insurance.

damage <- c(0,1,0,0,0,4,2,0,1,0,1,1,0,2,0,0,1,0,0,1,0,0,0)
Intercept <- rep(1,length(damage))
insurance.year <- c(1,1.2,0.8,1,2,1,1.1,1,1,1.1,1.2,1.3,0.9,1.4,1,1,1,1.2,1,
1,1,1,1)
drivers.age <- c(25,19,30,48,30,18,19,29,24,54,56,20,38,18,23,58,
47,36,25,28,38,39,42)
# abroad: driver has been abroad for longer time (=1)
abroad <- c(0,0,0,1,0,0,1,0,0,0,0,0,1,0,0,0,0,1,0,1,1,1,1)
Y <- damage
X <- cbind(Intercept, drivers.age)
Z <- cbind(abroad)

wald.test(Yin=Y, Xin=X, Zin=Z, Offset = insurance.year, init = F, K.mu = 10)
```

		Estimate	Std. Error	z value	Pr(> z)
[1]					
[2]	MU REGRESSION				
[3]	b0 Intercept	1.68299	1.12116	1.50111	0.13333
[4]	b1 drivers.age	-0.06833	0.04053	-1.68582	0.09183
[5]	OMEGA REGRESSION				
[6]	g0 abroad	-2.49917	7.52011	-0.33233	0.73964
[7]	PHI	1.12300	0.28641		
[8]					
[9]	Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1				
[10]	Iterations		351		
[11]	Log Likelihood		-23.6		
[12]	Pearson Chi Squared		6.0		
[13]	AIC		55		
[14]	Range Mu		0.10	2.20	
[15]	Range Omega		0.08	0.50	
[16]	Kappa		-4.540199e-05		

```

wald.test <- function(Yin, Xin, Zin, Offset=rep(1,length(Yin)), init=T, K.mu = NULL)
{
  Y <- Yin
  X <- Xin
  Z <- Zin
  k.beta <- dim(X)[2]
  k.gamma <- dim(Z)[2]
  n <- length(Y)

  # Estimate coefficients
  ausgabe <- mle.zmgp(Y, X, Z, Offset = Offset, summary = FALSE, init=init, K.mu = K.mu)

  hat.beta <- ausgabe$Coefficients.Mu
  hat.phi <- ausgabe$Dispersion.Parameter
  hat.gamma <- ausgabe$Coefficients.Omega

  # Compute square root of diagonal elements of FM-1
  B <- FM(hat.beta, hat.phi, hat.gamma, X, Z, K=K, Offset = Offset)
  sd.vector <- sqrt( diag(solve(B, tol = 1e-20)) )
  hat.sd.beta <- sd.vector[1:k.beta]
  hat.sd.phi <- sd.vector[k.beta+1]
  hat.sd.gamma <- sd.vector[(k.beta+2):(k.beta+1+k.gamma)]

  # Compute T-Statistics
  z.stat.beta <- hat.beta/hat.sd.beta
  z.stat.gamma <- hat.gamma/hat.sd.gamma

  # Compute P-Values
  p.value.beta <- 2*pnorm(-abs(z.stat.beta))
  p.value.gamma <- 2*pnorm(-abs(z.stat.gamma))

  # Create ***
  glimpse.beta <- rep("",length(z.stat.beta))
  glimpse.gamma <- rep("",length(z.stat.gamma))
  for (i in 1:length(z.stat.beta)) {
    if (p.value.beta[i] < 0.001) {glimpse.beta[i] <- "***"}
    if (p.value.beta[i] >= 0.001 & p.value.beta[i] < 0.01) {glimpse.beta[i] <- "**"}
    if (p.value.beta[i] >= 0.01 & p.value.beta[i] < 0.05) {glimpse.beta[i] <- "*"}
    if (p.value.beta[i] >= 0.05 & p.value.beta[i] < 0.1) {glimpse.beta[i] <- "."} }
  for (i in 1:length(z.stat.gamma)) {
    if (p.value.gamma[i] < 0.001) {glimpse.gamma[i] <- "***"}
    if (p.value.gamma[i] >= 0.001 & p.value.gamma[i] < 0.01) {glimpse.gamma[i] <- "**"}
    if (p.value.gamma[i] >= 0.01 & p.value.gamma[i] < 0.05) {glimpse.gamma[i] <- "*"}
    if (p.value.gamma[i] >= 0.05 & p.value.gamma[i] < 0.1) {glimpse.gamma[i] <- "."} }

  #create output
  coef.names.beta <- paste("b",c(0:(k.beta-1)),sep="")
  coef.names.gamma <- paste("g",c(0:(k.gamma-1)),sep="")
  coef.desc.beta <- double(k.beta)
  coef.desc.gamma <- double(k.gamma)
  for (i in 1:k.beta) {
    coef.desc.beta[i] <- colnames(X, do.NULL=FALSE)[i]
    if (is.matrix(X)) { if (max(X[,i])==1&min(X[,i])==1)
      {coef.desc.beta[i] <- "Intercept"} }
  }

```

```

else{ if (max(X[i])==1&min(X[i])==1) {coef.desc.beta[i] <- "Intercept"} }
}
for (i in 1:k.gamma) {
coef.desc.gamma[i] <- colnames(Z, do.NULL=FALSE)[i]
if (is.matrix(Z)) { if (max(Z[,i])==1&min(Z[,i])==1)
{coef.desc.gamma[i] <- "Intercept"} }
else{ if (max(Z[i])==1&min(Z[i])==1) {coef.desc.gamma[i] <- "Intercept"} }
}

output <- matrix("",1+k.beta+1+k.gamma+11,7)
output[2:(1+k.beta+1+k.gamma+2),1] <- c("",coef.names.beta,"",coef.names.gamma,"")
output[2:(1+k.beta+1+k.gamma+2),2] <-
c("MU REGRESSION",coef.desc.beta, "OMEGA REGRESSION",coef.desc.gamma,"PHI")
output[1:(1+k.beta+1+k.gamma+2),3] <-
c("Estimate","",formatC(hat.beta,5,format="f"),"",formatC(hat.gamma,5,format="f"),
formatC(hat.phi,5,format="f"))
output[1:(1+k.beta+1+k.gamma+2),4] <-
c("Std. Error","",formatC(hat.sd.beta,5,format="f"),"",
formatC(hat.sd.gamma,5,format="f"),formatC(hat.sd.phi,5,format="f"))
output[1:(1+k.beta+1+k.gamma+2),5] <- c("z value","",
formatC(z.stat.beta,5,format="f"),"",formatC(z.stat.gamma,5,format="f"),"")
output[1:(1+k.beta+1+k.gamma+2),6] <- c("Pr(>|z|)", "",
formatC(p.value.beta,5,format="f"),"",formatC(p.value.gamma,5,format="f"),
"")
output[1:(1+k.beta+1+k.gamma+2),7] <- c("", "",glimpse.beta,"",glimpse.gamma,"")
output[(1+k.beta+1+k.gamma+4),2] <- "Signif. codes: 0"
output[(1+k.beta+1+k.gamma+4),3] <- "'***' 0.001"
output[(1+k.beta+1+k.gamma+4),4] <- "'**' 0.01"
output[(1+k.beta+1+k.gamma+4),5] <- "'*' 0.05"
output[(1+k.beta+1+k.gamma+4),6] <- "'.' 0.1"
output[(1+k.beta+1+k.gamma+4),7] <- "' ' 1"
output[(1+k.beta+1+k.gamma+5),2] <- "Iterations"
output[(1+k.beta+1+k.gamma+5),4] <- ausgabe$Iterations[1]
output[(1+k.beta+1+k.gamma+6),2] <- "Log Likelihood"
output[(1+k.beta+1+k.gamma+6),4] <-
formatC(ausgabe$Log.Likelihood,digits=1,format="f")
output[(1+k.beta+1+k.gamma+7),2] <- "Pearson Chi Squared"
output[(1+k.beta+1+k.gamma+7),4] <- formatC(ausgabe$Pearson,digits=1,format="f")
output[(1+k.beta+1+k.gamma+8),2] <- "AIC"
output[(1+k.beta+1+k.gamma+8),4] <- round(ausgabe$AIC)
output[(1+k.beta+1+k.gamma+9),2] <- "Range Mu"
output[(1+k.beta+1+k.gamma+9),4] <- formatC(ausgabe$Range.Mu[1],digits=2,format="f")
output[(1+k.beta+1+k.gamma+9),5] <- formatC(ausgabe$Range.Mu[2],digits=2,format="f")
output[(1+k.beta+1+k.gamma+10),2] <- "Range Omega"
output[(1+k.beta+1+k.gamma+10),4] <-
formatC(ausgabe$Range.Omega[1],digits=2,format="f")
output[(1+k.beta+1+k.gamma+10),5] <-
formatC(ausgabe$Range.Omega[2],digits=2,format="f")
output[(1+k.beta+1+k.gamma+11),2] <- "Kappa"
output[(1+k.beta+1+k.gamma+11),4] <- ausgabe$Omega.Bound
output2 <- data.frame(output)
return(output2)
}

```

[02] *mle.zmgp*

Maximum Likelihood Estimates

Description:

'mle.zmgp' is used to calculate the MLEs of the regression parameters for mean and zero inflation and the overdispersion parameter.

Needs additional Functions:

```
'log.likelihood.zmgp'
'fit.zmgp'
'gradient'
'summary.zmgp'
'optimized.run'
# for usage with 'init = T' needs external functions (in 'R' code!)
'mle.zigp.full.like'
'log.likelihood.zigp.full'
'fit.zigp1'
'summary.zigp1'
```

Usage:

```
mle.zmgp(Yin, Xin, Zin, Offset = rep(1,length(Yin)), summary = TRUE,
         init = FALSE, K.mu = NULL)
```

Arguments:

Yin: response vector of length n.

Xin: design matrix of dim (n x p) for mean modelling.

Zin: design matrix of dim (n x q) for zero inflation modelling.

Offset: exposure for individual observation lengths. Defaults to a vector of 1.
The offset MUST NOT be in 'log' scale.

summary: a logical value indicating whether a list summary is shown (=T) or returned to the user (=F), i.e. if values of the ML estimation should be extracted, summary has to be set to 'F'. Defaults to 'T'.

init: a logical value indicating whether initial optimization values for zero inflation regression parameters are set to -1 (init = F) or are estimated by a ZIGP(μ_i , ϕ , ω)-model (init = T). Defaults to 'T'.

K.mu: upper boundary for μ . If unknown, an K.mu is set to $2 * \max(Yin)$. Defaults to NULL, i.e. 'unknown'.

Value:

If summary is set to 'F', the following values are returned:

ZI.Parameter	Zero Inflation estimate
--------------	-------------------------

Coefficients.Mu	Regression Parameters estimates for mean
Coefficients.Omega	Regression Parameters estimates for ZI
Dispersion.Parameter	Dispersion Parameter Estimate
Range.Mu	mean range
Range.Omega	ZI parameter range
Log.Likelihood	log likelihood value at MLE
Residuals	residuals
Pearson	Pearson Chi Squared Statistics
AIC	Akaike Information Criterion
Iterations	number of iterations needed
Time	time needed
Message	convergence message of underlying 'optim' routine
Response	response vector
Fitted.Values	fitted values
Design.Mu	design matrix for mean
Design.Omega	design matrix for ZI
Omega.Bound	lower boundary for omega. omega in [K,1], $K < 0$.

See Also:

```
wald.test().
```

Example:

```
## Number of damages in car insurance.

damage <- c(0,1,0,0,0,4,2,0,1,0,1,1,0,2,0,0,1,0,0,1,0,0,0)
Intercept <- rep(1,length(damage))
insurance.year <- c(1,1.2,0.8,1,2,1,1.1,1,1,1.1,1.2,1.3,0.9,1.4,1,1,1,1.2,1,
1,1,1,1)
drivers.age <- c(25,19,30,48,30,18,19,29,24,54,56,20,38,18,23,58,
47,36,25,28,38,39,42)
# abroad: driver has been abroad for longer time (=1)
abroad <- c(0,0,0,1,0,0,1,0,0,0,0,0,0,1,0,0,0,0,1,0,1,1,1,1)
Y <- damage
X <- cbind(Intercept, drivers.age)
Z <- cbind(abroad)

mle.zmgrp(Yin=Y, Xin=X, Zin=Z, Offset = insurance.year, summary = T, init = F,
          K.mu = 10)

[1] Range for ZI-Parameters: 0.07587448 0.4999773
[2] Coefficients for mu: 1.682989 -0.06833466
[3] Coefficients for omega: -2.499169
[4] Dispersion Parameter: 1.123003
[5] Pearson Chi Squared: 6.013368
[6] Range of mu: 0.1022411 2.202157
[7] Range of omega: 0.07587448 0.4999773
[8] Message: "NULL"
[9] Lower bound of omega_i: -4.540199e-05
[10] AIC: 55.28697
```

```

mle.zmgp <- function(Yin, Xin, Zin, Offset = rep(1,length(Yin)), summary = TRUE,
                    init = FALSE, K.mu = NULL)
{
  X <- Xin
  Y <- Yin
  Z <- Zin
  if(is.matrix(X)) {
    n <- dim(X)[1]
    k.beta <- dim(X)[2]
  }
  else {
    n <- length(X)
    k.beta <- 1
  }
  if(is.matrix(Z)) {
    k.gamma <- dim(Z)[2]
    nz <- dim(Z)[1]
  }
  else {
    k.gamma <- 1
    nz <- length(Z)
  }

  if(nz==n){
    t.i <- Offset
  #calculate K
  if (!is.null(K.mu)){ K.mu.upper <- K.mu }
  else { K.mu.upper <- 2*max(Y) }
  K <- exp(-K.mu.upper)/(exp(-K.mu.upper)-1)

  if(init){
    #optimized initial values
    start.delta <- optimized.run(Y,X,Z)
    # maximization of log likelihood w.r.t. delta s.t. above limits hold
    start <- proc.time()[2]
    opt <- optim(par = start.delta, fn = log.likelihood.zmgp, gr=gradient,
               method = "CG")
  }
  else
  {
    # simple initial values
    alpha <- log(0.2)
    gamma <- rep(-2.5,k.gamma)
    beta <- summary(glm(Y ~ offset(log(t.i)) + 1 + X, family = poisson(link=log)
                      ))$coefficients[, 1]
    start.delta <- c(beta,alpha,gamma)
    # maximization of log likelihood w.r.t. delta s.t. above limits hold
    start <- proc.time()[2]
    opt <- optim(par = start.delta, fn = log.likelihood.zmgp, gr=gradient,
               method = "CG")
  }

  delta <- opt$par
  it <- opt$counts

```

```

log.likelihood <- - opt$value
if(is.null(opt$message)) {message <- "NULL"}
else { message <- opt$message }
AIC <- -2 * log.likelihood + 2 * (k.beta + 1 + k.gamma)
time.needed <- proc.time()[2] - start

# results
coef <- double(k.beta + k.gamma + 1)
coef <- delta
beta <- coef[1 : k.beta]
phi <- 1 + exp(coef[k.beta + 1])
gamma <- coef[(k.beta + 2) : (k.beta + k.gamma + 1)]

fit <- fit.zmgp(delta)
res <- Y - fit$fit
chsq <- sum((Y - fit$fit) /
            sqrt((1-fit$omega)*phi*fit$fit+fit$omega*(1-fit$omega)*fit$fit^2))
range.mu <- c(min(fit$mu), max(fit$mu))
range.omega <- c(min(fit$omega), max(fit$omega))
# compute zero inflation parameters omega
if(is.matrix(Z)){ eta.omega <- Z %*% gamma }
else { eta.omega <- Z * gamma }
omega <- (K+exp(eta.omega))/(1+exp(eta.omega))

# summarize results
mle.data <- list(ZI.Parameter = omega,
  Coefficients.Mu = beta,
  Coefficients.Omega = gamma,
  Dispersion.Parameter = phi,
  Range.Mu = range.mu,
  Range.Omega = range.omega,
  Log.Likelihood = log.likelihood,
  Residuals = res,
  Pearson = chsq,
  AIC = AIC,
  Iterations = it,
  Time = time.needed,
  Message = message,
  Response = Y,
  Fitted.Values = fit$fit,
  Design.Mu = X,
  Design.Omega = Z,
  Omega.Bound = K)
if(summary) {
  summary.zmgp(mle.data)
}
else {
  return(mle.data)
}
}
else{
return("X and Z have different numbers of rows!")
}
}

```

Hilfsfunktionen (werden i.d.R. nicht direkt vom Benutzer ausgeführt)

[03] *rzmgp*

ZMGP random variable generator

Description:

‘rzmgp’ generates ZMGP random variables using the inversion method.

Usage:

```
rzmgp(n, mu, phi, omega, K)
```

Arguments:

n: length of output random vector.

mu: mean

phi: dispersion parameter

omega: zero inflation parameter

K: lower boundary for omega. omega in $[K, 1]$, $K < 0$.

Value:

Generates a ZMGP random vector of length n.

See Also:

```
response.zmgp().
```

Example:

```
# set seed for random variable generator
set.seed(2)
rzmgp(3, 2, 1.5, -0.000004, K=-0.000005)
[1] 4 2 0
```



```

rzmgp <- function(n, mu = stop("no mu arg"), phi = stop("no phi arg"),
                  omega = stop("no omega arg"), K)
{
  # check if parameters are valid
  if(omega < K) {return("omega has to be in [K,1]!")}
  if(omega > 1) {return("omega has to be in [K,1]!")}

  # inversion method
  x <- double(n)
  for(i in 1:n) {
    y <- integer(1)
    c <- double(1)
    #P(y=0)
    p <- omega + (1-omega) * exp( - mu/phi)
    s <- p
    u <- runif(1, 0, 1)
    if (u > s){
      #P(y=1)
      p <- (1-omega)*mu/phi*exp(-1/phi*(mu+phi-1))
      s <- s + p
      y <- 1
      while (u > s) {
        y <- y + 1
        p <- (mu + (phi-1)*y)^(y-1) / (mu + (phi-1)*(y-1))^(y-2) / (y * phi) *
          exp((1-phi)/phi) * p
        s <- s + p
      }
    }
    x[i] <- y
  }
  return(x)
}

```

[04] *response.zmgp*

ZMGP response generator

Description:

'response.zmgp' generates a ZIGP response vector for a given setting of linear predictors.

Needs additional Functions:

'rzmgp'

Usage:

```
response.zmgp(X, Z, beta, phi, gamma, K.mu = NULL)
```

Arguments:

X: design matrix of dim (n x p) for modelling of mean

Z: design matrix of dim (n x q) for modelling of zero inflation

beta: regression parameters for mean of length p

phi: dispersion parameter

gamma: regression parameters for zero inflation of length q

K.mu: upper boundary for mu. If unknown, an K.mu is set to $2 * \max(Y_{in})$. Defaults to NULL, i.e. 'unknown'.

Value:

Generates a ZMGP response vector of length n.

See Also:

rzmgp().

Example:

```
X <- matrix(c(1:3,4,3,5),3,2)
Z <- rep(1,3)
beta <- c(5,-2)
phi <- 1.5
gamma <- -10
# set seed for random variable generator
set.seed(1)
response.zmgp(X,Z,beta,phi,gamma,K.mu=10)
[1] 0 55 125
```

```

response.zmgrp <- function(X, Z, beta, phi, gamma, K.mu = NULL)
{
  # get number of observations from X and Z
  if(is.matrix(X)) {
    nx <- dim(X)[1]
  }
  else {
    nx <- length(X)
  }
  if(is.matrix(Z)) {
    nz <- dim(Z)[1]
  }
  else {
    nz <- length(Z)
  }

  if (nx==nz) {
    n <- nx
    mu <- double(n)
    omega <- double(n)
    Y <- double(n)

    if(is.matrix(X)) {
      mu <- exp(X%*%beta)
    }
    else {
      mu <- exp(X*beta)
    }
  }
  #calculate K
  if (!is.null(K.mu)){ K.mu.upper <- K.mu }
  else { K.mu.upper <- 2*max(mu) }
  K <- exp(-K.mu.upper)/(exp(-K.mu.upper)-1)
  if(is.matrix(Z)) {
    omega <- (K+exp(Z%*%gamma))/(1+exp(Z%*%gamma))
  }
  else {
    omega <- (K+exp(Z*gamma))/(1+exp(Z*gamma))
  }

  # create ZMGP response for all i
  for(i in 1:n) {
    Y[i] <- rzmgrp(1, mu[i], phi, omega[i], K)
  }
  return(Y)
}
else {
  print("X and Z have different numbers of rows!")
}
}

```

[05] *log.likelihood.zmgp*

Log Likelihood of ZMGP distribution

Description:

'log.likelihood.zmgp' calculates the log likelihood function of the ZMGP distribution.

Usage:

```
log.likelihood.zmgp(delta)
```

Arguments:

delta: a parameter vector of length (p+q+1) with $\dim(X) = (n \times p)$, $\dim(Z) = (n \times q)$. Create delta by pasting 'delta <- c(beta, ALPHA, gamma)', where ALPHA is one-dimensional with $\phi = 1 + \exp(\alpha) \Leftrightarrow \alpha = \log(\phi-1)$. Reason: with ALPHA, the problem of finding MLE's is unrestricted - with phi, it is not. Unrestricted problems are numerically more stable. beta is the vector of regression parameters for the mean modelling, phi is the dispersion parameter and gamma is the vector of regression parameters for the ZI modelling.

Details:

The design matrices have to be defined as X (for mean) and Z (for ZI).
n has to be defined as the number of observations $\dim(X)[1]$.
k.beta has to be defined as the length of beta.
k.gamma has to be defined as the length of gamma.
Y has to be defined as the response vector.
t.i has to be defined as the exposure.
K has to be defined as the lower boundary for omega. omega in $[K, 1]$, $K < 0$.

Example:

```
Y <- c(3,0,2)
X <- matrix(c(1:3,4,3,5),3,2)
Z <- rep(1,3)
n <- dim(X)[1]
t.i <- rep(1,n)
beta <- c(5,-2)
phi <- 1.5
alpha <- log(phi-1)
gamma <- -10
k.beta <- length(beta)
k.gamma <- length(gamma)
delta <- c(beta,alpha,gamma)
K <- -0.000005
log.likelihood.zmgp(delta)
[1] 107.3881
```

```

log.likelihood.zmgrp <- function(delta)
{
  eta.mu <- double(n)
  eta.omega <- double(n)
  s1 <- double(1)
  s2 <- double(1)

  if(k.beta == 1) {
    eta.mu <- X * delta[1]
  }
  else {
    beta <- delta[1:k.beta]
    eta.mu <- X %*% beta
  }
  if(k.gamma == 1) {
    eta.omega <- Z * delta[k.beta+2]
  }
  else {
    gamma <- delta[(k.beta + 2) : (k.beta + k.gamma + 1)]
    eta.omega <- Z %*% gamma
  }

  mu.i <- t.i*exp(eta.mu)
  phi <- 1+exp(delta[k.beta+1])
  k.i <- exp(eta.omega)
  P0 <- exp(-1/phi*mu.i)

  # terms of log likelihood with 1_{Y_i = 0}
  s1 <- sum(ifelse(Y == 0,1,0)* (log(k.i+K+P0*(1-K)) -
    log(1+k.i) ) )
  # terms of log likelihood with 1_{Y_i > 0} (excluding factorial)
  s2 <- sum(ifelse(Y>0,1,0)*(log(1-K)-log(1+k.i)+log(t.i)+eta.mu+(Y-1)*
    log(mu.i+(phi-1)*Y)-Y*log(phi)-1/phi*(mu.i+(phi-1)*Y)) - lgamma(Y+1))

  l <- s1 + s2
  return( - l)
}

```

[06] *fit.zmgp*

Fitted Values

Description:

'fit.zmgp' computes fitted values.

Usage:

```
fit.zmgp(delta)
```

Arguments:

delta: a parameter vector of length (p+q+1) with $\dim(X) = (n \times p)$, $\dim(Z) = (n \times q)$. Create delta by pasting 'delta <- c(beta, phi, gamma)'.
beta is the vector of regression parameters for the mean modelling,
phi is the dispersion parameter and gamma is the vector of regression parameters for the ZI modelling.

Details:

The design matrices have to be defined as X (for mean) and Z (for ZI).
n has to be defined as the number of observations $\dim(X)[1]$.
k.beta has to be defined as the length of beta.
k.gamma has to be defined as the length of gamma.
t.i has to be defined as the exposure.
K has to be defined as the lower boundary for omega. omega in $[K, 1]$, $K < 0$.

Example:

```
X <- matrix(c(1:3,4,3,5),3,2)
Z <- rep(1,3)
n <- dim(X)[1]
beta <- c(5,-2)
phi <- 1.5
gamma <- -10
k.beta <- length(beta)
k.gamma <- length(gamma)
t.i <- rep(1,n)
delta <- c(beta,phi,gamma)
K <- -0.000005
fit.zmgp(delta)
[1] $fit
[2]      [,1]
[3] [1,]  0.04978506
[4] [2,]  54.59594437
[5] [3,] 148.40716349
[6] $mu
[7]      [,1]
[8] [1,]  0.04978707
[9] [2,]  54.59815003
[10] [3,] 148.41315910
```

```

[11] $omega
[12] [1] 4.039810e-05 4.039810e-05 4.039810e-05

fit.zmgp <- function(delta)
{
  eta.mu <- double(n)
  eta.omega <- double(n)
  mu <- double(n)
  omega <- double(n)
  fit <- double(n)

  if(k.beta == 1) {
    eta.mu <- X * delta[1]
  }
  else {
    beta <- delta[1:k.beta]
    eta.mu <- X %*% beta
  }
  if(k.gamma == 1) {
    eta.omega <- Z * delta[k.beta+2]
  }
  else {
    gamma <- delta[(k.beta + 2) : (k.beta + k.gamma + 1)]
    eta.omega <- Z %*% gamma
  }

  # computation of mu, omega and mean
  mu <- t.i*exp(eta.mu)
  omega <- (K+exp(eta.omega))/(1+exp(eta.omega))
  fit <- (1 - omega) * mu

  return(list(fit = fit, mu = mu, omega = omega))
}

```

[07] *summary.zmgp*

Summary

Description:

'summary.zmgp' summarizes the results of the maximum likelihood estimation.

Usage:

```
summary.zmgp(mle.data.in)
```

Arguments:

mle.data.in: a list of result parameters given by function 'mle.zmgp'

```
summary.zmgp <- function(mle.data.in)
{
  X <- mle.data.in$Design.Mu
  if(is.matrix(X)) {
    nx <- dim(X)[1]
    k.beta <- dim(X)[2]
  }
  else {
    nx <- length(X)
    k.beta <- 1
  }
  Z <- mle.data.in$Design.Omega
  if(is.matrix(Z)) {
    nz <- dim(Z)[1]
    k.gamma <- dim(Z)[2]
  }
  else {
    nz <- length(Z)
    k.gamma <- 1
  }

  # range of ZI parameters
  out0 <- matrix(double(2), 1, 2)
  out0[1, 1] <- min(mle.data.in$ZI.Parameter)
  out0[1, 2] <- max(mle.data.in$ZI.Parameter)
  colnames(out0) <- c("", "")
  rownames(out0) <- c("Range for ZI-Parameters: ")
  print(out0)
  # beta
  out1 <- matrix(double(k.beta), 1, k.beta)
  kopf <- rep("", k.beta)
  for(i in 1:k.beta) {
    out1[1, i] <- mle.data.in$Coefficients.Mu[i]
  }
  colnames(out1) <- kopf
  rownames(out1) <- c("Coefficients for mu:      ")
  print(out1)
  # gamma
```



```

out2 <- matrix(double(k.gamma), 1, k.gamma)
kopf <- rep("", k.gamma)
for(i in 1:k.gamma) {
  out2[1, i] <- mle.data.in$Coefficients.Omega[i]
}
colnames(out2) <- Kopf
rownames(out2) <- c("Coefficients for omega: ")
print(out2)
# dispersion parameter
out3 <- matrix(double(1), 1, 1)
out3[1, 1] <- mle.data.in$Dispersion.Parameter
colnames(out3) <- c("")
rownames(out3) <- c("Dispersion Parameter: ")
print(out3)
# Pearson Chi Squared
out4 <- matrix(double(1), 1, 1)
out4[1, 1] <- mle.data.in$Pearson
colnames(out4) <- c("")
rownames(out4) <- c("Pearson Chi Squared: ")
print(out4)
# range of mu
out5 <- matrix(double(2), 1, 2)
out5[1, 1] <- mle.data.in$Range.Mu[1]
out5[1, 2] <- mle.data.in$Range.Mu[2]
colnames(out5) <- c("", "")
rownames(out5) <- c("Range of mu: ")
print(out5)
# range of omega
out6 <- matrix(double(2), 1, 2)
out6[1, 1] <- mle.data.in$Range.Omega[1]
out6[1, 2] <- mle.data.in$Range.Omega[2]
colnames(out6) <- c("", "")
rownames(out6) <- c("Range of omega: ")
print(out6)
# optimization message
out7 <- matrix(double(1), 1, 1)
out7[1, 1] <- mle.data.in$Message
colnames(out7) <- c("")
rownames(out7) <- c("Message: ")
print(out7)
# lower bound for omega
out8 <- matrix(double(1), 1, 1)
out8[1, 1] <- mle.data.in$Omega.Bound
colnames(out8) <- c("")
rownames(out8) <- c("Lower bound of omega_i: ")
print(out8)
# AIC
out9 <- matrix(double(1), 1, 1)
out9[1, 1] <- mle.data.in$AIC
colnames(out9) <- c("")
rownames(out9) <- c("AIC: ")
print(out9)
return()
}

```

[08] gradient

Die explizite Berechnung eines Gradienten ist zwar nicht notwendig, da für die 'optim'-Methode "CG" Gradienten gegebenenfalls numerisch geschätzt werden. Allerdings erhöhen explizite Gradienten die Konvergenzgeschwindigkeit enorm.

Gradient of log likelihood

Description:

'gradient' calculates the gradient of the log likelihood function.

Usage:

```
gradient(delta)
```

Arguments:

delta: a parameter vector of length (p+q+1) with $\dim(X) = (n \times p)$, $\dim(Z) = (n \times q)$. Create delta by pasting 'delta <- c(beta, ALPHA, gamma)', where ALPHA is one-dimensional with $\phi = 1 + \exp(\alpha) \Leftrightarrow \alpha = \log(\phi-1)$. Reason: with ALPHA, the problem of finding MLE's is unrestricted - with phi, it is not. Unrestricted problems are numerically more stable. beta is the vector of regression parameters for the mean modelling, phi is the dispersion parameter and gamma is the vector of regression parameters for the ZI modelling.

mle.data.in: a list of result parameters given by function 'mle.zigp'

Details:

This function used for the 'optim' method. Some 'optim' methods need a gradient and have to approximate it in every iteration step. This slows down the process a lot. Therefore, an explicit gradient function increases convergence speed and ensures correct convergence even if the log likelihood function is not continuous due to the indicator function of the ZI parameter.

The design matrices have to be defined as X (for mean) and Z (for ZI). n has to be defined as the number of observations $\dim(X)[1]$. k.beta has to be defined as the length of beta. k.gamma has to be defined as the length of gamma. t.i has to be defined as the exposure. K has to be defined as the lower boundary for omega. omega in $[K, 1]$, $K < 0$.

Example:

```
Y <- c(3,0,2)
X <- matrix(c(1:3,4,3,5),3,2)
Z <- rep(1,3)
n <- dim(X)[1]
beta <- c(5,-2)
phi <- 1.5
alpha <- log(phi-1)
```

```

gamma <- -10
k.beta <- length(beta)
k.gamma <- length(gamma)
t.i <- rep(1,n)
delta <- c(beta,alpha,gamma)
K <- -0.000005
gradient(delta)
[1] 289.815338 480.619759 -64.312861 -1.123626

gradient <- function(delta)
{
  grad <- double(k.beta + 1 + k.gamma)
  eta.mu <- double(n)
  eta.omega <- double(n)
  s1 <- double(1)
  s2 <- double(1)

  if(k.beta == 1) {
    eta.mu <- X * delta[1]
    X <- cbind(X,rep(0,length(X)))
  }
  else {
    beta <- delta[1:k.beta]
    eta.mu <- X %*% beta
  }
  if(k.gamma == 1) {
    eta.omega <- Z * delta[k.beta+2]
    Z <- cbind(Z,rep(0,length(Z)))
  }
  else {
    gamma <- delta[(k.beta + 2) : (k.beta + k.gamma + 1)]
    eta.omega <- Z %*% gamma
  }

  mu.i <- t.i*exp(eta.mu)
  k.i <- exp(eta.omega)
  phi <- 1+exp(delta[k.beta+1])
  P0 <- exp(-1/phi*mu.i)

  # Derivative for beta
  for (j in 1:k.beta)
  {
    # terms of gradient with 1_{Y_i = 0}
    s1 <- sum(ifelse(Y == 0,1,0)* X[,j]*(P0*(-1/phi)*mu.i*(1-K)/(k.i+K+P0*(1-K))))
    # terms of gradient with 1_{Y_i > 0}
    s2 <- sum(ifelse(Y>0,1,0)* X[,j]*(1+(Y-1)*mu.i/(mu.i+(phi-1)*Y)-1/phi*mu.i))
    grad[j] <- s1 + s2
  }

  # Derivative for phi
  # terms of gradient with 1_{Y_i = 0}
  s1 <- sum(ifelse(Y == 0,1,0)* (1/phi^2*P0*mu.i*(1-K)/(k.i+K+P0*(1-K))))
  # terms of gradient with 1_{Y_i > 0}

```

```

    s2 <- sum(ifelse(Y>0,1,0)* (Y*(Y-1)/(mu.i+(phi-1)*Y)-Y/phi+(mu.i-Y)/phi^2))
    grad[k.beta+1] <- s1 + s2

# Derivative for gamma
for (j in 1:k.gamma)
{
# terms of gradient with 1_{Y_i = 0}
  s1 <- sum(ifelse(Y == 0,1,0)* Z[,j]*k.i/(k.i+K+P0*(1-K)))
# terms of gradient independent of Y_i
  s2 <- -sum(Z[,j]*k.i/(1+k.i))
  grad[k.beta+1+j] <- s1 + s2
}

return(-grad)
}

```

[09] *optimized.run*

Calculation of valid initial values

Description:

'optimized.run' calculates valid initial values for the 'optim' method.

Needs additional Functions:

```
# needs external functions (in 'R' code!)
'mle.zigp.full.like'
'log.likelihood.zigp.full'
'fit.zigp1'
'summary.zigp1'
```

Usage:

```
optimized.run(Y,X,Z)
```

Arguments:

Yin: response vector of length n.

Xin: design matrix of dim (n x p) for mean modelling.

Zin: design matrix of dim (n x q) for zero inflation modelling.

Details:

This function uses ''R'' routines for ZIGP(μ_i , ϕ , ω) regression. Therefore, compile
'mle.zigp.full.like'
'log.likelihood.zigp.full'
'fit.zigp1'
'summary.zigp1'.

t.i has to be defined as the exposure.

n has to be defined as the number of observations `dim(X)[1]`.

Example:

```
Y <- c(3,0,2)
X <- matrix(c(1:3,4,3,5),3,2)
Z <- rep(1,3)
n <- dim(X)[1]
t.i <- rep(1,n)
optimized.run(Y,X,Z)
[1]          X1          X2          Z
[2]  -0.6783368  0.4559966 -1.6094379 -1.3862944
```

```

optimized.run <- function(Y,X,Z)
{

# improved initial values
# get initial beta, phi & omega
out <- mle.zigp.full.like(Y, X, Offset = t.i, summary=FALSE)
beta.start <- out$Coefficients
phi.start <- log(out$Dispersion.Parameter-1)
omega.first <- out$ZI.Parameter

# LM for gamma
rechte.seite <- rep(log(omega.first)-log(1-omega.first),n)
out<-lm(rechte.seite ~ Z-1)
gamma.start <- out$coefficients
rm(rechte.seite,out)

start.delta <- c(beta.start,phi.start,gamma.start)
rm(beta.start,phi.start,gamma.start)

return(start.delta)
}

```

[10] *FM*

Fisher Information

Description:

'FM' calculates the (Expected) Fisher Information matrix.

Usage:

```
FM(beta, phi, gamma, X, Z, K, Offset = NULL)
```

Arguments:

beta: regression parameters for mean of length p

phi: dispersion parameter

gamma: regression parameters for zero inflation of length q

X: design matrix of dim (n x p) for mean modelling.

Z: design matrix of dim (n x q) for zero inflation modelling.

K: lower boundary for omega. omega in [K,1], K < 0.

Offset: exposure for individual observation lengths. Defaults to a vector of 1.
The offset MUST NOT be in 'log' scale.

Example:

```
X <- matrix(c(1:3,4,3,5),3,2)
Z <- rep(1,3)
beta <- c(5,-2)
phi <- 1.5
gamma <- -10
K <- -0.000005
FM(beta, phi, gamma, X, Z, K)
      [,1]      [,2]      [,3]      [,4]
[1,] 6.935802e+02 1.139721e+03 -2.218299e+00 -1.506755e-06
[2,] 1.139721e+03 1.875357e+03 -3.600868e+00 -6.027020e-06
[3,] -2.218299e+00 -3.600868e+00 1.797925e+00 1.004503e-06
[4,] -1.506755e-06 -6.027020e-06 1.004503e-06 1.020288e-04
```

```

FM <- function(beta, phi, gamma, X, Z, K, Offset = NULL)
{

k.beta <- length(beta)
k.gamma <- length(gamma)

FM <- array(0,c(1+k.beta+k.gamma,1+k.beta+k.gamma))

if(is.matrix(X)){ eta.mu <- X%*%beta }
else{eta.mu <- X*beta}
if(is.matrix(Z)){ eta.gamma <- Z%*%gamma }
else{eta.gamma <- Z*gamma}

if (is.null(Offset)) {mu <- exp(eta.mu)}
else {t.i <- Offset
      mu <- t.i*exp(eta.mu)}
k <- exp(eta.gamma)
P0 <- exp(-1/phi*mu)
# a_i = E[I_{Y_i = 0}]
a <- (k+K+P0*(1-K))/(1+k)

temp <- double(1)

if(k.beta==1){ X <- cbind(X,rep(0,length(X))) }
if(k.gamma==1){ Z <- cbind(Z,rep(0,length(Z))) }

# the second derivative w.r.t. "beta"
for (i in 1:k.beta)
{
  for (j in 1:k.beta)
  {
    temp <- sum( X[,i]*X[,j]*mu* ( a*(1-K)*(-1/phi*P0^2*(1-K)+(mu-phi)/
      (phi^2)*P0*(k+K))/((k+K+P0*(1-K))^2) + ((phi-1)*mu*(1-K))/
      (phi^2*(mu-2+2*phi)*(1+k)) - (1-a)/phi ) )
    FM[i,j] <- -temp
  }
}

# the second derivative w.r.t. "phi"
temp <- sum( a*(1-K)*((-2*mu*P0/phi^3)*(k+K+P0*(1-K))+P0/phi^4*mu^2*(k+K))/
  ((k+K+P0*(1-K))^2) - (mu*(mu+2*phi)*(1-K))/(phi^2*(mu-2+2*phi)*(1+k)) +
  mu*(1-K)/(1+k)*(1/phi^2+2/phi^3) - 2*(1-a)*mu/(phi^3) )
FM[k.beta+1,k.beta+1] <- -temp

# the second derivative w.r.t. "gamma"
for (i in 1:k.gamma)
{
  for (j in 1:k.gamma)
  {
    temp <- sum( Z[,i]*Z[,j]*k* ( a*(K+P0*(1-K))/((k+K+P0*(1-K))^2)-1/((1+k)^2) ) )
    FM[k.beta+1+i,k.beta+1+j] <- -temp
  }
}
}

```



```

# the mixed derivative w.r.t. "beta" and "phi"
for (i in 1:k.beta)
{
temp <- sum( X[,i]*mu* ( a*(1-K)*(-1/phi^3*P0*mu*(k+K)+1/(phi^2)*P0*(k+K+P0*(1-K)))/
((k+K+P0*(1-K))^2)-mu*(1-K)/(phi^2*(mu-2+2*phi)*(1+k))+(1-a)/(phi^2) ) )
FM[k.beta+1,i] <- -temp
FM[i,k.beta+1] <- -temp
}

# the mixed derivative w.r.t. "beta" and "gamma"
for (i in 1:k.beta)
{
for (j in 1:k.gamma)
{
temp <- sum( X[,i]*Z[,j]*a*(1-K)*P0*mu*k/(phi*(k+K+P0*(1-K))^2) )
FM[k.beta+1+j,i] <- -temp
FM[i,k.beta+1+j] <- -temp
}
}

# the mixed derivative w.r.t. "phi" and "gamma"
for (i in 1:k.gamma)
{
temp <- sum( Z[,i]* a * (-1/phi^2 * P0 * (1-K) * mu * k)/((k+K+P0*(1-K))^2) )
FM[k.beta+1+i,k.beta+1] <- -temp
FM[k.beta+1,k.beta+1+i] <- -temp
}

return(FM)
}

```

Anhang D

Summaries der Modelle für den Patent–Datensatz

Zunächst eine Zusammenfassung der verwendeten Notationen.

$Y = \text{out.int}$	<i>Response:</i> Anzahl der Patente, die von externen Patentanwälten eingereicht wurden
$E = \text{pat.cum.year}$	<i>Exposure:</i> Anzahl insgesamt eingereichter Patente
<hr/>	
<i>Branchendummies</i>	
chem.pha	Dummyvektor für die Branche "Chemical / Pharma"
elec.tel	Dummyvektor für die Branche "Electro / Telecommunication"
engineer	Dummyvektor für die Branche "Engineering"
car.supp	Dummyvektor für die Branche "Cars and Suppliers"
med.biot	Dummyvektor für die Branche "Medtech / Biotech"
other.br	Dummyvektor für alle übrigen Branchen
elec.tel.other	Dummyvektor für die Branchen "Electro / Telecommunication / Other"
eng.car.supp.med.biot	Dummyvektor für die Branchen "Engineering / Cars and Suppliers / Medtech and Biotech"
car.supp.other	Dummyvektor für die Branchen "Cars and Suppliers / Other"
chem.pha.engin	Dummyvektor für die Branchen "Chemical / Pharma / Engineering"
<hr/>	
<i>Variablen und Transformationen</i>	
std.5.mean	geschätzte Standardabweichung der Bewerbungen e. Unternehmens
ln.std.5.mean	$\log(\text{std.5.mean} + 0.001)$
breadth	Maß für die Streuung der Bewerbungen in technologischen Gebieten
emp	Anzahl der Mitarbeiter
sqrte.emp	$\sqrt{\text{emp}}$
r.d.pat	Verhältnis aus Ausgaben für Forschung und Entwicklung und der Anzahl angemeldeter Patente
r.d.pat.inv	$1/(\text{r.d.pat} + 0.001)$
r.d.intens.1	$\text{poly}(\text{r.d.intens}, 3)[1]$: Ausgaben für Fo. u. Entw. in Relation zur Anzahl der Mitarbeiter
r.d.intens.2	$\text{poly}(\text{r.d.intens}, 3)[2]$: quadratischer Term
r.d.intens.3	$\text{poly}(\text{r.d.intens}, 3)[3]$: kubischer Term
r.d.miss	Dummyvar.: Angaben zu Fo. u. Entw. fehlen (= 1) / feh. nicht (= 0)
<hr/>	
<i>Interaktionen</i>	
ln.std.5.mean.breadth	Interaktion zw. ln.std.5.mean u. breadth: $\text{ln.std.5.mean} * \text{breadth}$
chem.pha.ln.std.5.mean	$\text{chem.pha} * \text{ln.std.5.mean}$
chem.pha.sqrte.emp	$\text{chem.pha} * \text{sqrte.emp}$
r.d.miss.chem.pha	$\text{r.d.miss} * \text{chem.pha}$
<hr/>	
<i>diskretisierte Variablen</i>	
breadth.k145	Dummyvariable für Klassen 1, 4, 5 in diskretisiertem "breadth"
emp.k12	Dummyvariable für Klassen 1 und 2 in diskretisiertem "emp"
r.d.intens.k12	Dummyvariable für Klassen 1 und 2 in diskretisiertem "r.d.intens"
breadth.k12	Dummyvariable für Klassen 1 und 2 in diskretisiertem "breadth"
emp.k2	Dummyvariable für die Klasse 2 in diskretisiertem "emp"
r.d.pat.k2	Dummyvariable für die Klasse 2 in diskretisiertem "r.d.pat"

Tabelle D.1: Notationstabelle aller verwendeten Variablen

1.) $Poi(\mu_i)$ vollständig

```
fmla <- Y ~ offset(ln.pat.cum.year) + 1 + ln.std.5.mean + breadth + sqrt.emp +
  r.d.pat.inv + r.d.intens.1 + r.d.intens.2 + r.d.intens.3 + r.d.miss +
  chem.pha + elec.tel.other + time + ln.std.5.mean.breadth +
  chem.pha.ln.std.5.mean + chem.pha.sqrt.emp + r.d.miss.chem.pha
patent.glm <- glm(formula=fmla,family=poisson, x=TRUE, y=TRUE)
summary(patent.glm)
Call:
glm(formula = fmla, family = poisson, x = TRUE, y = TRUE)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-23.2520	-3.1602	-0.8611	1.7222	19.0188

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.447216	0.056352	-25.682	< 2e-16 ***
ln.std.5.mean	-0.128834	0.008948	-14.398	< 2e-16 ***
breadth	0.140444	0.007507	18.708	< 2e-16 ***
sqrt.emp	-0.378785	0.005771	-65.634	< 2e-16 ***
r.d.pat.inv	-0.717685	0.083016	-8.645	< 2e-16 ***
r.d.intens.1	-0.024902	0.012508	-1.991	0.0465 *
r.d.intens.2	0.004089	0.007417	0.551	0.5815
r.d.intens.3	0.077209	0.012212	6.322	2.58e-10 ***
r.d.miss	1.199283	0.177474	6.758	1.40e-11 ***
chem.pha	-0.003154	0.025868	-0.122	0.9030
elec.tel.other	0.416499	0.017611	23.650	< 2e-16 ***
time	0.061070	0.006447	9.472	< 2e-16 ***
ln.std.5.mean.breadth	-0.066455	0.008313	-7.994	1.30e-15 ***
chem.pha.ln.std.5.mean	0.241026	0.027844	8.656	< 2e-16 ***
chem.pha.sqrt.emp	-0.175610	0.026625	-6.596	4.23e-11 ***
r.d.miss.chem.pha	-0.875958	0.104302	-8.398	< 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 26941 on 855 degrees of freedom
 Residual deviance: 20627 on 840 degrees of freedom
 AIC: 23896

Number of Fisher Scoring iterations: 5

1.) $Poi(\mu_i)$ verkürzt ist identisch

Für alle nachfolgenden nicht verkürzten Modelle wird mit

```
X <- cbind(Intercept, ln.std.5.mean, breadth, sqrt.emp, r.d.pat.inv, r.d.intens.1,
  r.d.intens.2, r.d.intens.3, r.d.miss, chem.pha, elec.tel.other, time,
  ln.std.5.mean.breadth, chem.pha.ln.std.5.mean, chem.pha.sqrt.emp,
  r.d.miss.chem.pha)
W <- cbind(Intercept, engineer, car.suppl.other, med.biot, time, breadth.k145,
  emp.k12, r.d.intens.k12)
Z <- cbind(Intercept, breadth.k12, emp.k2, r.d.pat.k2, chem.pha.engin)
```

gearbeitet.

2.) $ZIP(\mu_i, \omega)$ vollständig

```
wald.test(Y, X, Offset=pat.cum.year)
1
2
3      MU REGRESSION
4 b0      Intercept    -1.26533    0.03922   -32.26347    0.00000    ***
5 b1      ln.std.5.mean  -0.04634    0.00697    -6.65062    0.00000    ***
6 b2      breadth       0.00645    0.00517     1.24878    0.21175
7 b3      sqrt.emp      -0.34037    0.00422   -80.61601    0.00000    ***
8 b4      r.d.pat.inv   -0.55486    0.05754    -9.64383    0.00000    ***
9 b5      r.d.intens.1   0.03783    0.00747     5.06810    0.00000    ***
10 b6     r.d.intens.2   -0.03408    0.00552    -6.17760    0.00000    ***
11 b7     r.d.intens.3   0.05093    0.00732     6.96030    0.00000    ***
12 b8     r.d.miss       0.96210    0.12324     7.80658    0.00000    ***
13 b9     chem.pha      -0.03278    0.01876    -1.74766    0.08052     .
14 b10    elec.tel.other  0.40691    0.01265    32.17264    0.00000    ***
15 b11    time          0.05315    0.00473    11.24592    0.00000    ***
16 b12    ln.std.5.mean.breadth -0.17396    0.00670   -25.95014    0.00000    ***
17 b13    chem.pha.ln.std.5.mean  0.21373    0.02052    10.41435    0.00000    ***
18 b14    chem.pha.sqrt.emp -0.21984    0.01997   -11.00575    0.00000    ***
19 b15    r.d.miss.chem.pha -0.54631    0.06362    -8.58708    0.00000    ***
20      OMEGA          0.11863    0.01157
21
22      Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
23      Iterations      111
24      Log Likelihood   -9574.6
25      AIC              19183
26      Range Mu        0.34    578.29
```

2.) $ZIP(\mu_i, \omega)$ verkürzt ist identisch

3.) $ZIP(\mu_i, \omega_i)$ vollständig

```
X <- cbind(Intercept, ln.std.5.mean, breadth, sqrt.emp, r.d.pat.inv, r.d.intens.1,
           r.d.intens.2, r.d.intens.3, r.d.miss, chem.pha, elec.tel.other, time,
           ln.std.5.mean.breadth, chem.pha.ln.std.5.mean, chem.pha.sqrt.emp,
           r.d.miss.chem.pha)
```

```
Z <- cbind(Intercept, breadth.k12, emp.k2, r.d.pat.k2, chem.pha.engin)
wald.test(Y, X, Z, Offset=pat.cum.year, init=F)
```

		Estimate	Std. Error	z value	Pr(> z)	
1						
2	MU REGRESSION					
3	b0 Intercept	-1.28162	0.05452	-23.50574	0.00000	***
4	b1 ln.std.5.mean	-0.04643	0.00974	-4.76636	0.00000	***
5	b2 breadth	0.00630	0.00732	0.86116	0.38915	
6	b3 sqrt.emp	-0.34081	0.00576	-59.17261	0.00000	***
7	b4 r.d.pat.inv	-0.58150	0.07990	-7.27824	0.00000	***
8	b5 r.d.intens.1	0.03735	0.01059	3.52795	0.00042	***
9	b6 r.d.intens.2	-0.03377	0.00776	-4.35057	0.00001	***
10	b7 r.d.intens.3	0.05076	0.01028	4.93759	0.00000	***
11	b8 r.d.miss	1.01877	0.17103	5.95670	0.00000	***
12	b9 chem.pha	-0.03161	0.02679	-1.18019	0.23792	
13	b10 elec.tel.other	0.40634	0.01764	23.03809	0.00000	***
14	b11 time	0.05335	0.00654	8.15677	0.00000	***
15	b12 ln.std.5.mean.breadth	-0.17428	0.00955	-18.25192	0.00000	***
16	b13 chem.pha.ln.std.5.mean	0.21456	0.02939	7.30086	0.00000	***
17	b14 chem.pha.sqrt.emp	-0.22150	0.02840	-7.79849	0.00000	***
18	b15 r.d.miss.chem.pha	-0.52407	0.09490	-5.52241	0.00000	***
19	OMEGA REGRESSION					
20	g0 Intercept	-3.21585	0.25437	-12.64253	0.00000	***
21	g1 breadth.k12	1.25539	0.25427	4.93724	0.00000	***
22	g2 emp.k2	0.80855	0.25576	3.16137	0.00157	**
23	g3 r.d.pat.k2	-0.53423	0.26292	-2.03192	0.04216	*
24	g4 chem.pha.engin	1.22163	0.23409	5.21858	0.00000	***
25						
26						
27	Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					
28	Iterations		137			
29	Log Likelihood		-9533.8			
30	Pearson Chi Squared		26.9			
31	AIC		19110			
32	Range Mu		0.34	578.22		
33	Phi		1			
34	Range Omega		0.02	0.52		

3.) $ZIP(\mu_i, \omega_i)$ verkürzt ist identisch

4.) $GP(\mu_i, \varphi)$ vollständig

```
X <- cbind(Intercept, ln.std.5.mean, breadth, sqrt.emp, r.d.pat.inv, r.d.intens.1,
  r.d.intens.2, r.d.intens.3, r.d.miss, chem.pha, elec.tel.other, time,
  ln.std.5.mean.breadth, chem.pha.ln.std.5.mean, chem.pha.sqrt.emp,
  r.d.miss.chem.pha)
```

```
wald.test(Y, X, Offset=pat.cum.year)
```

		Estimate	Std. Error	z value	Pr(> z)	
1						
2	MU REGRESSION					
3	b0 Intercept	-0.73886	0.14666	-5.03777	0.00000	***
4	b1 ln.std.5.mean	0.00926	0.04025	0.22995	0.81813	
5	b2 breadth	0.32049	0.03655	8.76970	0.00000	***
6	b3 sqrt.emp	-0.55637	0.03225	-17.25316	0.00000	***
7	b4 r.d.pat.inv	-0.35384	0.19191	-1.84377	0.06522	.
8	b5 r.d.intens.1	-0.04964	0.03599	-1.37903	0.16788	
9	b6 r.d.intens.2	0.19893	0.03140	6.33488	0.00000	***
10	b7 r.d.intens.3	-0.11391	0.03280	-3.47308	0.00051	***
11	b8 r.d.miss	0.18336	0.41412	0.44277	0.65793	
12	b9 chem.pha	-0.40857	0.12111	-3.37363	0.00074	***
13	b10 elec.tel.other	0.19061	0.07338	2.59749	0.00939	**
14	b11 time	-0.03839	0.03107	-1.23559	0.21661	
15	b12 ln.std.5.mean.breadth	0.01698	0.03591	0.47278	0.63637	
16	b13 chem.pha.ln.std.5.mean	-0.00025	0.14365	-0.00171	0.99864	
17	b14 chem.pha.sqrt.emp	0.07049	0.13830	0.50969	0.61027	
18	b15 r.d.miss.chem.pha	-0.44416	0.31606	-1.40529	0.15993	
19	PHI	8.24540	0.44748			
20						
21	Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					
22	Iterations		109			
23	Log Likelihood		-3413.4			
24	AIC		6861			
25	Range Mu		0.23	341.12		

4.) $GP(\mu_i, \varphi)$ verkürzt

```

X <- cbind(Intercept, breadth, sqrt.emp, r.d.pat.inv, r.d.intens.1,
           r.d.intens.2, r.d.intens.3, chem.pha, elec.tel.other)
wald.test(Y, X, Offset=pat.cum.year)

```

		Estimate	Std. Error	z value	Pr(> z)	
1						
2	MU REGRESSION					
3	b0 Intercept	-0.68634	0.07273	-9.43654	0.00000	***
4	b1 breadth	0.32751	0.03594	9.11369	0.00000	***
5	b2 sqrt.emp	-0.55700	0.03028	-18.39580	0.00000	***
6	b3 r.d.pat.inv	-0.28062	0.04183	-6.70833	0.00000	***
7	b4 r.d.intens.1	-0.05428	0.03536	-1.53527	0.12472	
8	b5 r.d.intens.2	0.19850	0.03071	6.46426	0.00000	***
9	b6 r.d.intens.3	-0.11062	0.03266	-3.38698	0.00071	***
10	b7 chem.pha	-0.42817	0.09578	-4.47039	0.00001	***
11	b8 elec.tel.other	0.17646	0.07289	2.42094	0.01548	*
12	PHI	8.20312	0.44220			
13						
14	Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					
15	Iterations		45			
16	Log Likelihood		-3416.1			
17	AIC		6852			
18	Range Mu		0.23	362.87		

5.) $GP(\mu_i, \varphi_i)$ vollständig

```

X <- cbind(Intercept, ln.std.5.mean, breadth, sqrt.emp, r.d.pat.inv, r.d.intens.1,
  r.d.intens.2, r.d.intens.3, r.d.miss, chem.pha, elec.tel.other, time,
  ln.std.5.mean.breadth, chem.pha.ln.std.5.mean, chem.pha.sqrt.emp,
  r.d.miss.chem.pha)
W <- cbind(Intercept, engineer, car.supp.other, med.biot, time, breadth.k145,
  emp.k12, r.d.intens.k12)
wald.test(Y, X, W, Offset=pat.cum.year, init=F)

```

		Estimate	Std. Error	z value	Pr(> z)	
1						
2	MU REGRESSION					
3	b0 Intercept	-1.21121	0.13690	-8.84769	0.00000	***
4	b1 ln.std.5.mean	0.01286	0.03774	0.34064	0.73338	
5	b2 breadth	0.27546	0.03719	7.40652	0.00000	***
6	b3 sqrt.emp	-0.45732	0.03324	-13.75904	0.00000	***
7	b4 r.d.pat.inv	-0.35211	0.17716	-1.98746	0.04687	*
8	b5 r.d.intens.1	0.00955	0.03992	0.23920	0.81095	
9	b6 r.d.intens.2	0.13959	0.03453	4.04218	0.00005	***
10	b7 r.d.intens.3	-0.08192	0.03295	-2.48660	0.01290	*
11	b8 r.d.miss	0.19933	0.37935	0.52545	0.59927	
12	b9 chem.pha	0.09403	0.13501	0.69651	0.48611	
13	b10 elec.tel.other	0.36579	0.08108	4.51127	0.00001	***
14	b11 time	0.06878	0.04168	1.64993	0.09896	.
15	b12 ln.std.5.mean.breadth	0.02397	0.03499	0.68509	0.49329	
16	b13 chem.pha.ln.std.5.mean	-0.01024	0.14267	-0.07176	0.94279	
17	b14 chem.pha.sqrt.emp	0.07944	0.14096	0.56356	0.57305	
18	b15 r.d.miss.chem.pha	-0.55093	0.31509	-1.74851	0.08038	.
19	PHI REGRESSION					
20	a0 Intercept	1.93974	0.11640	16.66451	0.00000	***
21	a1 engineer	-0.33690	0.17472	-1.92820	0.05383	.
22	a2 car.supp.other	-0.59727	0.10236	-5.83528	0.00000	***
23	a3 med.biot	-0.63028	0.19334	-3.25991	0.00111	**
24	a4 time	0.23578	0.05708	4.13033	0.00004	***
25	a5 breadth.k145	0.40127	0.08701	4.61152	0.00000	***
26	a6 emp.k12	-0.44662	0.10705	-4.17213	0.00003	***
27	a7 r.d.intens.k12	-0.26057	0.10730	-2.42828	0.01517	*
28						
29						
30	Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					
31	Iterations		276			
32	Log Likelihood		-3353.1			
33	AIC		6754			
34	Range Mu		0.17	370.71		
35	Range Phi		2.27	15.89		

5.) $GP(\mu_i, \varphi_i)$ verkürzt

```

X <- cbind(Intercept, breadth, sqrt.emp, r.d.pat.inv, r.d.intens.1,
           r.d.intens.2, r.d.intens.3, r.d.miss, chem.pha, elec.tel.other,
           r.d.miss.chem.pha)
W <- cbind(Intercept, car.suppl.other, med.biot, time, breadth.k145,
           emp.k12, r.d.intens.k12)
wald.test(Y, X, W, Offset=pat.cum.year, init=F)

```

		Estimate	Std. Error	z value	Pr(> z)	
1						
2	MU REGRESSION					
3	b0 Intercept	-1.15649	0.13426	-8.61398	0.00000	***
4	b1 breadth	0.26438	0.03618	7.30800	0.00000	***
5	b2 sqrt.emp	-0.45866	0.03149	-14.56675	0.00000	***
6	b3 r.d.pat.inv	-0.36366	0.17925	-2.02876	0.04248	*
7	b4 r.d.intens.1	0.01630	0.03941	0.41359	0.67917	
8	b5 r.d.intens.2	0.13043	0.03377	3.86218	0.00011	***
9	b6 r.d.intens.3	-0.07399	0.03283	-2.25380	0.02421	*
10	b7 r.d.miss	0.20257	0.38325	0.52856	0.59711	
11	b8 chem.pha	0.01045	0.10531	0.09920	0.92098	
12	b9 elec.tel.other	0.28746	0.07134	4.02960	0.00006	***
13	b10 r.d.miss.chem.pha	-0.60816	0.30210	-2.01311	0.04410	*
14	PHI REGRESSION					
15	a0 Intercept	1.91400	0.11405	16.78227	0.00000	***
16	a1 car.suppl.other	-0.51435	0.09169	-5.60970	0.00000	***
17	a2 med.biot	-0.47988	0.17512	-2.74032	0.00614	**
18	a3 time	0.17580	0.04040	4.35102	0.00001	***
19	a4 breadth.k145	0.39511	0.08611	4.58815	0.00000	***
20	a5 emp.k12	-0.47022	0.10518	-4.47069	0.00001	***
21	a6 r.d.intens.k12	-0.32711	0.10227	-3.19854	0.00138	**
22						
23						
24	Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					
25	Iterations		265			
26	Log Likelihood		-3356.8			
27	AIC		6750			
28	Range Mu		0.20	335.88		
29	Range Phi		2.40	14.16		

6.) $ZIGP(\mu_i, \varphi, \omega)$ vollständig

```
X <- cbind(Intercept, ln.std.5.mean, breadth, sqrt.emp, r.d.pat.inv, r.d.intens.1,
  r.d.intens.2, r.d.intens.3, r.d.miss, chem.pha, elec.tel.other, time,
  ln.std.5.mean.breadth, chem.pha.ln.std.5.mean, chem.pha.sqrt.emp,
  r.d.miss.chem.pha)
```

```
wald.test(Y, X, Offset=pat.cum.year)
```

		Estimate	Std. Error	z value	Pr(> z)	
1						
2	MU REGRESSION					
3	b0 Intercept	-0.93923	0.08684	-10.81604	0.00000	***
4	b1 ln.std.5.mean	0.07816	0.02074	3.76849	0.00016	***
5	b2 breadth	0.11841	0.01555	7.61292	0.00000	***
6	b3 sqrt.emp	-0.47753	0.01309	-36.48889	0.00000	***
7	b4 r.d.pat.inv	-0.35687	0.12387	-2.88112	0.00396	**
8	b5 r.d.intens.1	0.09154	0.01680	5.44889	0.00000	***
9	b6 r.d.intens.2	0.00901	0.01378	0.65397	0.51313	
10	b7 r.d.intens.3	-0.00985	0.01608	-0.61271	0.54007	
11	b8 r.d.miss	0.51878	0.26524	1.95587	0.05048	.
12	b9 chem.pha	-0.42212	0.05802	-7.27604	0.00000	***
13	b10 elec.tel.other	0.35299	0.03533	9.99200	0.00000	***
14	b11 time	-0.03744	0.01408	-2.66013	0.00781	**
15	b12 ln.std.5.mean.breadth	-0.13365	0.02015	-6.63413	0.00000	***
16	b13 chem.pha.ln.std.5.mean	-0.02109	0.06606	-0.31932	0.74948	
17	b14 chem.pha.sqrt.emp	-0.09054	0.06412	-1.41216	0.15790	
18	b15 r.d.miss.chem.pha	-0.60322	0.19243	-3.13471	0.00172	**
19	PHI	6.41803	0.23443			
20	OMEGA	0.03988	0.01002			
21						
22	Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					
23	Iterations		261			
24	Log Likelihood		-3308.5			
25	AIC		6653			
26	Range Mu		0.15	474.94		

6.) $ZIGP(\mu_i, \varphi, \omega)$ verkürzt

```

X <- cbind(Intercept, ln.std.5.mean, breadth, sqrt.emp, r.d.pat.inv,
           r.d.intens.1, r.d.miss, chem.pha, elec.tel.other, time,
           ln.std.5.mean.breadth, r.d.miss.chem.pha)
wald.test(Y, X, Offset=pat.cum.year)

```

		Estimate	Std. Error	z value	Pr(> z)	
1						
2	MU REGRESSION					
3	b0 Intercept	-0.86274	0.07949	-10.85343	0.00000	***
4	b1 ln.std.5.mean	0.08225	0.01988	4.13818	0.00004	***
5	b2 breadth	0.11916	0.01548	7.69827	0.00000	***
6	b3 sqrt.emp	-0.48019	0.01274	-37.67953	0.00000	***
7	b4 r.d.pat.inv	-0.24134	0.11127	-2.16885	0.03009	*
8	b5 r.d.intens.1	0.09469	0.01640	5.77270	0.00000	***
9	b6 r.d.miss	0.27088	0.23966	1.13028	0.25836	
10	b7 chem.pha	-0.46682	0.04606	-10.13542	0.00000	***
11	b8 elec.tel.other	0.34920	0.03433	10.17296	0.00000	***
12	b9 time	-0.03799	0.01362	-2.78897	0.00529	**
13	b10 ln.std.5.mean.breadth	-0.13229	0.01994	-6.63302	0.00000	***
14	b11 r.d.miss.chem.pha	-0.52071	0.18690	-2.78604	0.00534	**
15	PHI	6.41208	0.23368			
16	OMEGA	0.04046	0.01009			
17						
18	Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					
19	Iterations		257			
20	Log Likelihood		-3308.6			
21	AIC		6645			
22	Range Mu		0.14	491.51		

7.) $ZIGP(\mu_i, \varphi, \omega_i)$ vollständig

```
X <- cbind(Intercept, ln.std.5.mean, breadth, sqrt.emp, r.d.pat.inv, r.d.intens.1,
           r.d.intens.2, r.d.intens.3, r.d.miss, chem.pha, elec.tel.other, time,
           ln.std.5.mean.breadth, chem.pha.ln.std.5.mean, chem.pha.sqrt.emp,
           r.d.miss.chem.pha)
```

```
Z <- cbind(Intercept, breadth.k12, emp.k2, r.d.pat.k2, chem.pha.engin)
wald.test(Y, X, Z, Offset=pat.cum.year, init=T)
```

		Estimate	Std. Error	z value	Pr(> z)	
1						
2	MU REGRESSION					
3	b0 Intercept	-0.76235	0.13585	-5.61178	0.00000	***
4	b1 ln.std.5.mean	0.08064	0.04016	2.00781	0.04466	*
5	b2 breadth	0.07327	0.03274	2.23760	0.02525	*
6	b3 sqrt.emp	-0.46676	0.02840	-16.43554	0.00000	***
7	b4 r.d.pat.inv	-0.15342	0.17848	-0.85958	0.39002	
8	b5 r.d.intens.1	0.08937	0.03334	2.68029	0.00736	**
9	b6 r.d.intens.2	-0.00076	0.02881	-0.02650	0.97886	
10	b7 r.d.intens.3	-0.00212	0.03185	-0.06672	0.94680	
11	b8 r.d.miss	0.14683	0.38508	0.38129	0.70299	
12	b9 chem.pha	-0.42685	0.11485	-3.71669	0.00020	***
13	b10 elec.tel.other	0.29454	0.06964	4.22955	0.00002	***
14	b11 time	-0.04825	0.02898	-1.66466	0.09598	.
15	b12 ln.std.5.mean.breadth	-0.13717	0.03901	-3.51613	0.00044	***
16	b13 chem.pha.ln.std.5.mean	-0.00682	0.13644	-0.04999	0.96013	
17	b14 chem.pha.sqrt.emp	-0.10089	0.13222	-0.76303	0.44545	
18	b15 r.d.miss.chem.pha	-0.38340	0.33019	-1.16113	0.24559	
19	OMEGA REGRESSION					
20	g0 Intercept	-4.64256	0.74003	-6.27349	0.00000	***
21	g1 breadth.k12	2.33776	0.74369	3.14345	0.00167	**
22	g2 emp.k2	0.45814	0.53910	0.84982	0.39543	
23	g3 r.d.pat.k2	-1.45740	0.68742	-2.12012	0.03400	*
24	g4 chem.pha.engin	1.08472	0.49197	2.20485	0.02746	*
25	PHI	6.43562	0.30217			
26						
27	Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					
28	Iterations		257			
29	Log Likelihood		-3289.5			
30	Pearson Chi Squared		-173.1			
31	AIC		6623			
32	Range Mu		0.15	455.83		
33	Range Omega		0.00	0.32		

7.) $ZIGP(\mu_i, \varphi, \omega_i)$ verkürzt

```

X <- cbind(Intercept, ln.std.5.mean, breadth, sqrt.emp, r.d.pat.inv,
           r.d.intens.1, chem.pha, elec.tel.other, ln.std.5.mean.breadth)
Z <- cbind(Intercept, breadth.k12)
wald.test(Y, X, Z, Offset=pat.cum.year, init=T)

```

		Estimate	Std. Error	z value	Pr(> z)	
1						
2	MU REGRESSION					
3	b0 Intercept	-0.75737	0.06690	-11.32044	0.00000	***
4	b1 ln.std.5.mean	0.08148	0.03914	2.08174	0.03737	*
5	b2 breadth	0.09372	0.03289	2.84898	0.00439	**
6	b3 sqrt.emp	-0.48064	0.02748	-17.49292	0.00000	***
7	b4 r.d.pat.inv	-0.11268	0.03683	-3.05939	0.00222	**
8	b5 r.d.intens.1	0.08132	0.03316	2.45200	0.01421	*
9	b6 chem.pha	-0.49848	0.09068	-5.49698	0.00000	***
10	b7 elec.tel.other	0.32474	0.06851	4.74010	0.00000	***
11	b8 ln.std.5.mean.breadth	-0.14776	0.03934	-3.75638	0.00017	***
12	OMEGA REGRESSION					
13	g0 Intercept	-4.37553	0.60573	-7.22360	0.00000	***
14	g1 breadth.k12	2.19412	0.65750	3.33709	0.00085	***
15	PHI	6.37712	0.29568			
16						
17	Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					
18	Iterations		274			
19	Log Likelihood		-3298.5			
20	Pearson Chi Squared		-161.8			
21	AIC		6621			
22	Range Mu		0.14	492.50		
23	Range Omega		0.01	0.10		

8.) $ZIGP(\mu_i, \varphi_i, \omega)$ vollständig

```

X <- cbind(Intercept, ln.std.5.mean, breadth, sqrt.emp, r.d.pat.inv, r.d.intens.1,
  r.d.intens.2, r.d.intens.3, r.d.miss, chem.pha, elec.tel.other, time,
  ln.std.5.mean.breadth, chem.pha.ln.std.5.mean, chem.pha.sqrt.emp,
  r.d.miss.chem.pha)
W <- cbind(Intercept, engineer, car.supp.other, med.biot, time, breadth.k145,
  emp.k12, r.d.intens.k12)
Z <- Intercept
wald.test(Y, X, W, Z, Offset=pat.cum.year, init=T)

```

		Estimate	Std. Error	z value	Pr(> z)	
1						
2	MU REGRESSION					
3	b0 Intercept	-1.03835	0.13052	-7.95574	0.00000	***
4	b1 ln.std.5.mean	0.03006	0.03721	0.80777	0.41922	
5	b2 breadth	0.07737	0.03204	2.41452	0.01576	*
6	b3 sqrt.emp	-0.40392	0.02887	-13.98894	0.00000	***
7	b4 r.d.pat.inv	-0.20105	0.17318	-1.16091	0.24568	
8	b5 r.d.intens.1	0.14988	0.03772	3.97349	0.00007	***
9	b6 r.d.intens.2	-0.04615	0.03127	-1.47585	0.13998	
10	b7 r.d.intens.3	0.02657	0.03373	0.78772	0.43086	
11	b8 r.d.miss	0.23000	0.36805	0.62491	0.53203	
12	b9 chem.pha	-0.27417	0.13043	-2.10200	0.03555	*
13	b10 elec.tel.other	0.30553	0.07412	4.12192	0.00004	***
14	b11 time	0.04145	0.03452	1.20082	0.22982	
15	b12 ln.std.5.mean.breadth	-0.10982	0.03512	-3.12656	0.00177	**
16	b13 chem.pha.ln.std.5.mean	-0.00448	0.14344	-0.03121	0.97510	
17	b14 chem.pha.sqrt.emp	-0.03315	0.14068	-0.23566	0.81370	
18	b15 r.d.miss.chem.pha	-0.55360	0.32145	-1.72217	0.08504	.
19	PHI REGRESSION					
20	a0 Intercept	1.89484	0.11303	16.76465	0.00000	***
21	a1 engineer	-0.23787	0.17554	-1.35507	0.17540	
22	a2 car.supp.other	-0.53016	0.10670	-4.96887	0.00000	***
23	a3 med.biot	-0.02791	0.19003	-0.14686	0.88325	
24	a4 time	0.20268	0.05354	3.78565	0.00015	***
25	a5 breadth.k145	0.13312	0.09154	1.45422	0.14588	
26	a6 emp.k12	-0.48173	0.10954	-4.39766	0.00001	***
27	a7 r.d.intens.k12	-0.39286	0.10647	-3.69001	0.00022	***
28	OMEGA REGRESSION					
29	g0 Intercept	-2.86226	0.21127	-13.54778	0.00000	***
30						
31	Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					
32	Iterations		146			
33	Log Likelihood		-3261.4			
34	Pearson Chi Squared		47.7			
35	AIC		6573			
36	Range Mu		0.16	525.81		
37	Range Phi		2.20	11.35		
38	Range Omega		0.05	0.05		

8.) $ZIGP(\mu_i, \varphi_i, \omega)$ verkürzt

```

X <- cbind(Intercept, ln.std.5.mean, breadth, sqrt.emp, r.d.intens.1,
           r.d.miss, chem.pha, elec.tel.other, ln.std.5.mean.breadth)
W <- cbind(Intercept, car.supp.other, time, emp.k12, r.d.intens.k12)
Z <- Intercept
wald.test(Y, X, W, Z, Offset=pat.cum.year, init=T)
1      Estimate Std. Error   z value Pr(>|z|)
2      MU REGRESSION
3 b0      Intercept    -0.85080    0.06605 -12.88147  0.00000 ***
4 b1      ln.std.5.mean    0.02902    0.03563  0.81445  0.41539
5 b2      breadth        0.07134    0.03170  2.25018  0.02444 *
6 b3      sqrt.emp       -0.39871    0.02707 -14.73099  0.00000 ***
7 b4      r.d.intens.1     0.13082    0.03245  4.03183  0.00006 ***
8 b5      r.d.miss       -0.29344    0.07073 -4.14891  0.00003 ***
9 b6      chem.pha       -0.34522    0.09528 -3.62327  0.00029 ***
10 b7     elec.tel.other    0.28138    0.06530  4.30912  0.00002 ***
11 b8 ln.std.5.mean.breadth -0.09701    0.03389 -2.86286  0.00420 **
12      PHI REGRESSION
13 a0      Intercept     1.95487    0.08560 22.83592  0.00000 ***
14 a1      car.supp.other  -0.46021    0.09240 -4.98070  0.00000 ***
15 a2      time          0.15049    0.04249  3.54156  0.00040 ***
16 a3      emp.k12       -0.53749    0.10201 -5.26908  0.00000 ***
17 a4      r.d.intens.k12 -0.36408    0.09417 -3.86620  0.00011 ***
18      OMEGA REGRESSION
19 g0      Intercept     -2.86670    0.21200 -13.52195  0.00000 ***
20
21      Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
22      Iterations              158
23      Log Likelihood          -3269.0
24      Pearson Chi Squared        34.9
25      AIC                     6568
26      Range Mu                 0.18    432.92
27      Range Phi                2.44     9.89
28      Range Omega              0.05     0.05

```


9.) $ZIGP(\mu_i, \varphi_i, \omega_i)$ vollständig

```

X <- cbind(Intercept, ln.std.5.mean, breadth, sqrt.emp, r.d.pat.inv, r.d.intens.1,
  r.d.intens.2, r.d.intens.3, r.d.miss, chem.pha, elec.tel.other, time,
  ln.std.5.mean.breadth, chem.pha.ln.std.5.mean, chem.pha.sqrt.emp,
  r.d.miss.chem.pha)
W <- cbind(Intercept, engineer, car.supp.other, med.biot, time, breadth.k145,
  emp.k12, r.d.intens.k12)
Z <- cbind(Intercept, breadth.k12, emp.k2, r.d.pat.k2, chem.pha.engin)
wald.test(Y, X, W, Z, Offset=pat.cum.year, init=T)
1
2          MU REGRESSION
3  b0      Intercept      -1.02296    0.12577   -8.13374   0.00000   ***
4  b1      ln.std.5.mean     0.02546    0.03688    0.69016   0.49009
5  b2      breadth          0.04673    0.03226    1.44854   0.14747
6  b3      sqrt.emp        -0.39590    0.02832   -13.97844   0.00000   ***
7  b4      r.d.pat.inv      -0.19481    0.16643   -1.17049   0.24180
8  b5      r.d.intens.1      0.14672    0.03757    3.90519   0.00009   ***
9  b6      r.d.intens.2     -0.05021    0.03134   -1.60199   0.10916
10 b7      r.d.intens.3      0.02980    0.03341    0.89175   0.37252
11 b8      r.d.miss          0.24727    0.35356    0.69937   0.48432
12 b9      chem.pha         -0.24500    0.12953   -1.89137   0.05857   .
13 b10     elec.tel.other     0.28921    0.07211    4.01089   0.00006   ***
14 b11     time              0.03890    0.03385    1.14937   0.25040
15 b12     ln.std.5.mean.breadth -0.11597    0.03517   -3.29728   0.00098   ***
16 b13     chem.pha.ln.std.5.mean 0.01900    0.14340    0.13253   0.89457
17 b14     chem.pha.sqrt.emp -0.05774    0.14051   -0.41092   0.68113
18 b15     r.d.miss.chem.pha -0.32906    0.32625   -1.00862   0.31316
19          PHI REGRESSION
20 a0      Intercept         1.90610    0.11263   16.92428   0.00000   ***
21 a1      engineer         -0.41945    0.18317   -2.28996   0.02202   *
22 a2      car.supp.other    -0.51855    0.10458   -4.95825   0.00000   ***
23 a3      med.biot          0.02779    0.18901    0.14705   0.88309
24 a4      time              0.21288    0.05299    4.01708   0.00006   ***
25 a5      breadth.k145      0.11166    0.09180    1.21626   0.22389
26 a6      emp.k12          -0.55070    0.11157   -4.93587   0.00000   ***
27 a7      r.d.intens.k12    -0.35475    0.10496   -3.37997   0.00072   ***
28          OMEGA REGRESSION
29 g0      Intercept        -4.20856    0.54872   -7.66982   0.00000   ***
30 g1      breadth.k12        1.90743    0.55581    3.43182   0.00060   ***
31 g2      emp.k2             0.70129    0.44246    1.58497   0.11297
32 g3      r.d.pat.k2        -1.20366    0.53665   -2.24293   0.02490   *
33 g4      chem.pha.engin     1.10465    0.42128    2.62213   0.00874   **
34
35      Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
36      Iterations
37      Log Likelihood
38      Pearson Chi Squared
39      AIC
40      Range Mu
41      Range Phi
42      Range Omega

```

9.) $ZIGP(\mu_i, \varphi_i, \omega_i)$ verkürzt

```

X <- cbind(Intercept, ln.std.5.mean, breadth, sqrt.emp, r.d.pat.inv,
           r.d.intens.1, chem.pha, elec.tel.other, ln.std.5.mean.breadth)
W <- cbind(Intercept, engineer, car.supp.other, time, emp.k12, r.d.intens.k12)
Z <- cbind(Intercept, breadth.k12, r.d.pat.k2, chem.pha.engin)
wald.test(Y, X, W, Z, Offset=pat.cum.year, init=T)
1      Estimate Std. Error   z value Pr(>|z|)
2      MU REGRESSION
3 b0      Intercept    -0.95107    0.05921 -16.06177 0.00000 ***
4 b1      ln.std.5.mean    0.03114    0.03579  0.87025 0.38417
5 b2      breadth        0.04107    0.03168  1.29643 0.19483
6 b3      sqrt.emp       -0.39399    0.02713 -14.52455 0.00000 ***
7 b4      r.d.pat.inv    -0.12341    0.03306 -3.73272 0.00019 ***
8 b5      r.d.intens.1     0.12421    0.03311  3.75166 0.00018 ***
9 b6      chem.pha       -0.27176    0.09893 -2.74711 0.00601 **
10 b7     elec.tel.other   0.30626    0.06634  4.61627 0.00000 ***
11 b8 ln.std.5.mean.breadth -0.10360    0.03441 -3.01095 0.00260 **
12      PHI REGRESSION
13 a0      Intercept      1.96765    0.08550 23.01242 0.00000 ***
14 a1      engineer       -0.42551    0.17050 -2.49570 0.01257 *
15 a2      car.supp.other  -0.48767    0.09624 -5.06729 0.00000 ***
16 a3      time           0.16136    0.04224  3.82013 0.00013 ***
17 a4      emp.k12        -0.58670    0.10254 -5.72145 0.00000 ***
18 a5      r.d.intens.k12  -0.30482    0.09649 -3.15906 0.00158 **
19      OMEGA REGRESSION
20 g0      Intercept      -4.28182    0.58621 -7.30418 0.00000 ***
21 g1      breadth.k12     2.27093    0.57769  3.93103 0.00008 ***
22 g2      r.d.pat.k2     -1.24085    0.52165 -2.37870 0.01737 *
23 g3      chem.pha.engin  1.08537    0.41303  2.62782 0.00859 **
24
25      Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
26      Iterations              129
27      Log Likelihood          -3245.2
28      Pearson Chi Squared       42.5
29      AIC                     6528
30      Range Mu                 0.18   446.30
31      Range Phi                2.41   10.15
32      Range Omega              0.00    0.28

```