Christoph STEINKELLNER

Ein Monte Carlo EM-Algorithmus für statistische Modelle mit zensierten Daten

DIPLOMARBEIT

zur Erlangung des akademischen Grades eines
Diplom-Ingenieurs

Diplomstudium Technische Mathematik

Graz University of Technology

Technische Universität Graz

Betreuer:
Ao. Univ.-Prof. Dipl.-Ing. Dr.techn. Herwig FRIEDL

Institut für Statistik

Graz, im März 2012

Zusammenfassung

Die Maximum Likelihood-Schätzung für das Generalisierte Lineare Modell (GLM) basiert auf der Annahme, dass Response- und auch Prädiktor-Werte vollständig beobachtet werden. Eine Situation, in der das nicht der Fall ist, entsteht beim Auftreten zensierter Beobachtungen, die eine iterative Maximierung der Likelihood Funktion mittels EM-Algorithmus motivieren.
Diese Arbeit beschäftigt sich mit diesem, von Dempster, Laird und Rubin (1977) entwickelten Verfahren und diskutiert die ML-Schätzung für Modelle mit links-zensierten Beobachtungen. Ein Beispiel dafür sind Schadstoffkonzentrationen die nicht exakt gemessen werden können, wenn sie unter einer bestimmten Nachweisbarkeitsgrenze liegen. Oft ist man zusätzlich daran interessiert, die Population in Gruppen zu unterteilen, was die Motivation für ein Mischmodell liefert, in dem die Gruppenzugehörigkeit einer einzelnen Response ebenfalls als nicht-beobachtbare Zufallsvariable modelliert wird.
Ziel dieser Arbeit ist die Herleitung des Schemas eines EM-Algorithmus, wenn beide Probleme gleichzeitig vorliegen. Je nach Verteilung der Response-Variablen kann die Berechnung des Erwartungswerts im E-Schritt unangenehm sein und soll deshalb durch eine Monte Carlo-Simulation approximiert werden. Eine Implementierung des dadurch entstehenden Monte Carlo EM-Algorithmus in der `R`-Funktion `glmm.lc` bildet den Abschluss dieser Arbeit, wobei für normalverteilte Responses ein Vergleich mit einem deterministischen EM-Algorithmus möglich ist.

Abstract

The Maximum Likelihood estimation to a Generalized Linear Model (GLM) in its standard form is based on the assumption that all data are completely observed. A situation which does not satisfy this assumption arises when censored observations occur and gives motivation to use the EM algorithm to compute the Maximum Likelihood estimates iteratively.
This thesis discusses that procedure developed by Dempster et al. (1977) and focuses on the ML estimation for problems with left censored responses. Concentrations that can only be measured if they exceed a certain threshold value are mentioned to be an example. If one is interested in dividing the population into groups a mixture of different components will be an appropriate model where the group indicator is also modeled as an unobserved variable.
The aim of this thesis is to derive the scheme of an EM algorithm when both of the described problems occur at the same time. The integral that has to be computed in the E-step is not always analytically tractable and is in general replaced by a Monte Carlo simulated mean. An implementation of the `R` function `glmm.lc` which contains the resulting MCEM algorithm is the final part of this thesis. Additionally a deterministic EM algorithm is provided for gaussian responses as a comparison to the MCEM algorithm.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am ....
(Unterschrift)

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

....
date (signature)

Danksagung

Diese Arbeit steht am Ende eines Studiums, bei dem mich viele Menschen unterstützt haben. Mein besonderer Dank gilt meiner Familie, auf deren Unterstützung ich während des gesamten Studiums zählen könnte. Außerdem danke ich meinem Betreuer, Herrn Friedl, für seine Geduld und seinen Einsatz während der Erstellung dieser Arbeit.

Inhaltsverzeichnis

1 Einleitung **8**

2 Das Generalisierte Lineare Modell **10**
2.1 Die Erweiterung des einfachen Linearen Modells 10
2.1.1 Die Exponentialfamilie 11
2.1.2 Mitglieder der Exponentialfamilie 15
2.1.3 Linkfunktionen 18
2.2 Die Maximum Likelihood Schätzung 20
2.2.1 Newton-Raphson Methode und Iterative Weighted Least Squares 22
2.2.2 Fisher Scoring 24
2.2.3 Eigenschaften das Schätzers 25
2.2.4 Pearson Residuen 26
2.3 Goodness of Fit 26
2.3.1 Deviance 27
2.3.2 Analysis of Deviance 28

3 Modelle für unvollständige Daten **30**
3.1 Der EM-Algorithmus 30
3.1.1 Die Theorie 30
3.1.2 Standardfehler 34
3.1.3 Varianten des EM-Algorithmus 35
3.2 Endliche diskrete Mischungen 36
3.2.1 Direkter Zugang zur ML-Schätzung 37
3.2.2 Diskrete Mischungen als Problem unvollständiger Daten 39
3.2.3 Testen auf die Anzahl der Komponenten 42
3.2.4 Diskrete Mischungen Generalisierter Linearer Modelle . 45
3.2.5 Likelihood Spikes 50
3.3 Modelle mit zufälligen Effekten 51
3.3.1 Überdispersion 51
3.3.2 Normalverteilte zufällige Effekte 53
3.3.3 Beliebige zufällige Effekte 56
3.3.4 Shared Random Effects 58
3.4 Zensierte Daten 59
3.4.1 Überblick 59
3.4.2 Die Likelihood Funktion 61
3.4.3 Links-zensierte normalverteilte Daten 61
3.4.4 Das Tobit-Modell 64
3.4.5 Gammaverteilte zensierte Daten 68

4 Endliche Mischungen mit links-zensierten Daten 72
4.1 Motivation . . . 72
4.2 Das Modell . . . 72
4.3 Der Monte Carlo EM-Algorithmus . . . 73
4.4 Links-zensierte Daten in Mischungen von GLMs . . . 78
4.4.1 Schätzung der Dispersionsparameter . . . 79
4.4.2 Der deterministische EM-Algorithmus für normalverteilte Responses . . . 80

5 Die R-Funktion `glmm.lc` 83
5.1 Überblick . . . 83
5.2 Der Algorithmus . . . 84
5.2.1 Startwerte . . . 84
5.2.2 Monte Carlo-Simulation . . . 85
5.2.3 Weitere Optionen zur Kontrolle des Algorithmus . . . 86
5.3 Simulationen . . . 87
5.3.1 Mischung von Normalverteilungen . . . 87
5.3.2 Normalverteilte Responses in einer Mischung von Regressionsmodellen . . . 93
5.3.3 Mischung von Gammaverteilungen . . . 94
5.3.4 Gammaverteilte Responses im Regressionsmodell . . . 100
5.4 Verwendung der Funktion `glmm.lc` . . . 102
5.4.1 Generieren von Datensätzen . . . 102
5.4.2 Output . . . 103

6 Resumee 106

A Eigenschaften der Score-Funktion 107

B Programmcode 108

Literatur 140

1 Einleitung

Diese Arbeit widmet sich zuerst der Theorie des Generalisierten Linearen Modells (GLM), das dazu dient, den Erwartungswert einer ausgezeichneten Zufallsvariable, die in der Folge *Response* genannt wird, unter Verwendung von erklärenden Variablen zu modellieren. Der Begriff *Generalisiert* ist dadurch motiviert, dass das GLM eine Verallgemeinerung des einfachen Linearen Modells darstellt. Während dort der Erwartungswert einer normalverteilten Zufallsvariable durch Linearkombinationen der erklärenden Variablen direkt beschrieben wird, ist die Verwendung des GLM auch für Responses, die einer anderen, zur *Exponentialfamilie* gehörenden Verteilung folgen, möglich. Außerdem können die Linearkombinationen der erklärenden Variablen im GLM anstatt des Erwartungswerts auch eine geeignete Funktion desselben modellieren.
Diese Verallgemeinerungen werden zu Beginn von Kapitel 2 genauer behandelt und bilden die Grundlage für die Parameterschätzung basierend auf der Maximum Likelihood Theorie (Abschnitt 2.2).
Kapitel 3 beschäftigt sich mit Problemen unvollständiger Daten und dem EM-Algorithmus als Instrument der Parameterschätzung dafür. Die Log-Likelihood Funktion wird dabei iterativ maximiert, wobei jede Iteration aus der Berechnung eines Erwartungswerts (E-Schritt) und dessen Maximierung (M-Schritt) besteht. Das Hauptaugenmerk liegt dabei auf drei Arten von Problemen, bei denen der EM-Algorithmus Anwendung findet:

- Mischmodelle: Eine Population kann in Gruppen unterteilt werden, wobei die Gruppenzugehörigkeit der einzelnen Beobachtungen aber unbekannt ist.

- Modelle mit zufälligen Effekten: Die Variabilität der Daten wird durch die Anpassung eines GLM nicht ausreichend erklärt, weshalb angenommen wird, dass zusätzliche, nicht-beobachtete erklärende Variablen existieren.

- Zensierte Daten: Aufgrund von Unzulänglichkeiten in der Datenerhebung wird ein Teil der Responses nicht exakt, sondern nur entweder links-zensiert, rechts-zensiert oder intervall-zensiert beobachtet.

Wie in den Abschnitten 3.2 und 3.3 deutlich wird, stehen Mischungen von GLMs in engem Zusammenhang mit Modellen mit zufälligen Effekten. In beiden Fällen benötigt man zur Durchführung des M-Schritts die Theorie der Maximum Likelihood-Schätzung eines GLM.
Abschnitt 3.4 wird sich mit zensierten Daten beschäftigen und mit ihren

Auswirkungen auf die Parameterschätzung, wie etwa am Beispiel des Tobit-Modells (Tobin, 1958). Dabei handelt es sich um ein einfaches Regressionsmodell, in dem normalverteilte Responses nur dann exakt beobachtet werden, wenn ihre Realisierungen positive Werte annehmen. Das Schema eines EM-Algorithmus für das Tobit-Modell lässt sich einfach herleiten (Abschnitt 3.4.4).
Für andere Verteilungsannahmen können bei Modellen mit zensierten Responses allerdings Schwierigkeiten bei der Durchführung des E-Schritts auftreten, da eine geschlossene Darstellung für den zu berechnenden Erwartungswert nicht immer gefunden werden kann. Dies motiviert den Einsatz der Monte Carlo-Simulation zur Approximation dieses Erwartungswerts, was den entstehenden Algorithmus zu einem Monte Carlo EM-Algorithmus macht.
Anwendung findet dieser MCEM-Algorithmus in Kapitel 4, das sich Mischmodellen mit links-zensierten Daten widmet. Die Messung von Schadstoffkonzentrationen, die unter eine Nachweisbarkeitsgrenze fallen können und deren Verteilung außerdem von einer nicht-beobachtbaren Größe beeinflusst wird, ist das Beispiel für eine Datensituation, die die Betrachtung dieser Modellklasse motiviert.
Die Implementierung dieses MCEM-Algorithmus in der `R`-Funktion `glmm.lc` ist Inhalt von Kapitel 5. Für den Fall, dass normalverteilte Responses angenommen werden, steht dabei ein EM-Algorithmus zur Verfügung, der ohne MC-Simulation auskommt und dessen Schema in Abschnitt 4.4.2 dargestellt ist. Der Vergleich des MCEM-Algorithmus mit dem deterministischen EM-Algorithmus ist Teil der danach folgenden Simulationsstudien. Zusätzlich wird das Verhalten des MCEM-Algorithmus am Beispiel von gammaverteilten Responses untersucht.
Eine Zusammenfassung der Resultate sowie einen Ausblick auf mögliche Erweiterungen dieser Arbeit bietet schließlich Kapitel 6. Der Anhang enthält den Programmcode der implementierten `R`-Funktionen.

2 Das Generalisierte Lineare Modell

2.1 Die Erweiterung des einfachen Linearen Modells

Das einfache Lineare Modell (LM) enthält die Annahme, dass unabhängige, normalverteilte Response-Variablen $Y_1, \ldots, Y_n$ beobachtet werden mit $\mathbb{E}(Y_i) = \mu_i$ und $\text{var}(Y_i) = \sigma^2$ für $i = 1, \ldots, n$. Die Erwartungswerte $\mathbb{E}(Y_i)$ sollen dabei durch Linearkombinationen von gegebenen Prädiktor-Variablen $x_i \in \mathbb{R}^p$ modelliert werden. Der zentrale Ansatz lautet:

$$\mu_i = \sum_{j=1}^{p} x_{ij}\beta_j. \tag{1}$$

Dabei ist $\beta = (\beta_1, \ldots, \beta_p)^t$ der zum Modell gehörende Parametervektor von der Dimension p, für dessen Maximum Likelihood Schätzer (MLE) $\hat{\beta}$ es eine geschlossene Darstellung gibt. Die speziellen Annahmen des LM ermöglichen eine Regressionsanalyse mit Resultaten, die zum einen in geschlossener Darstellung verfügbar sind und zum anderen eine Vielfalt an Interpretationen, etwa in geometrischer Hinsicht, zulassen. Auf die Theorie des einfachen LM soll in dieser Arbeit nicht näher eingegangen, sondern auf Rawlings, Pantula und Dickey (1998) sowie Davison (2003) verwiesen werden. Sowohl die Festlegung auf Normalverteilung als auch die Annahme der konstanten Varianz σ^2 für alle Responses stellen Einschränkungen dar, die für viele in der Praxis vorliegende Datensituationen zu stark sind.

Eine Möglichkeit, mit Inhomogenität der Varianz umzugehen ist, die Annahme zu treffen, dass eine geeignete Transformation gefunden werden kann, welche angewandt auf die Responses Y_i modifizierte Daten liefert, die der Annahme der Varianzhomogenität entsprechen. Box und Cox (1964) führen dazu eine Familie von Transformationen $Y(\lambda)$ ein, gegeben durch

$$Y(\lambda) = \begin{cases} \frac{Y^\lambda - 1}{\lambda} & \text{für } \lambda \neq 0 \\ \log Y & \text{für } \lambda = 0 \end{cases} \tag{2}$$

und zeigen, wie der Schätzer für jenen Wert von λ gefunden werden kann, bei dem am ehesten konstante Varianz vorliegt.

Im Gegensatz dazu ist die zentrale Idee beim Generalisierten Linearen Modell (GLM), die Daten in untransformierter Form beizubehalten und stattdessen die restriktiven Annahmen im LM durch Verallgemeinerungen zu ersetzen. Die Response muss dabei nicht mehr ausschließlich normalverteilt angenommen werden, sondern darf einer Verteilung aus der *Exponentialfamilie* unter-

liegen, die die Normalverteilung als Spezialfall enthält. Damit erreicht man auch, dass die Varianz $\text{var}(Y_i)$ nicht konstant sein muss, sondern proportional zu einer Funktion des Erwartungswerts, $V(\mu_i)$, modelliert werden kann. Außerdem muss beim GLM durch den linearen Prädiktor nicht direkt der Erwartungswert μ_i modelliert werden, sondern kann in (1) durch $g(\mu_i)$ ersetzt werden, wobei $g(\cdot)$ eine geeignete Funktion bezeichnet, die in der Folge *Linkfunktion* genannt wird.

Im Folgenden sollen Details dieser Verallgemeinerungen diskutiert werden, wobei McCullagh und Nelder (1989) als Grundlage dienen.

2.1.1 Die Exponentialfamilie

Im Gegensatz zum LM, wo die Responses y_i ausschließlich normalverteilt angenommen werden, soll im GLM eine ganze Familie von Verteilungen zugänglich sein. Dazu wird nun der Begriff der Exponentialfamilie eingeführt, für den es in der Literatur verschiedene Definitionen gibt. Die folgende findet sich bei Lehmann und Romano (2005):

Definition 2.1 (k-parametrische Exponentialfamilie): Lässt sich für eine Zufallsvariable Y die Dichte- oder Wahrscheinlichkeitsfunktion $f_Y = f(y;\theta)$ in der Form

$$f(y;\theta) = C(\theta)\exp\left\{\sum_{j=1}^{k} Q_j(\theta)T_j(y)\right\}h(y) \tag{3}$$

schreiben, wobei θ den Vektor aus dem k-dimensionalen Parameterraum bezeichnet und $C(\cdot)$, $Q_j(\cdot)$, $T_j(\cdot)$ $(j = 1,\ldots,k)$ und $h(\cdot)$ bekannte Funktionen sind, dann ist $f(y;\theta)$ Dichte- oder Wahrscheinlichkeitsfunktion einer Verteilung aus der k-parametrischen Exponentialfamilie.

Bemerkung: Eine solche Funktion kann ebenso in der Form

$$f(y;\theta) = \exp\left\{\tilde{C}(\theta) + \sum_{j=1}^{k} Q_j(\theta)T_j(y) + \tilde{h}(y)\right\} \tag{4}$$

geschrieben werden. Gilt $Q_j(\theta) = \theta_j$ für $j = 1,\ldots,k$, dann liegt eine k-parametrische Exponentialfamilie in *kanonischer* Form vor.

Die von McCullagh und Nelder (1989) getroffene Verteilungsannahme der Response-Variablen im GLM stellt einen Spezialfall von (4) dar. Sie fordern,

dass f_Y von der Form

$$f(y;\theta,\phi) = \exp\left(\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right) \tag{5}$$

ist, wobei $a(\cdot)$, $b(\cdot)$ und $c(\cdot)$ bekannte Funktionen sind. Ist ϕ ebenfalls bekannt, handelt es sich bei (5) um eine einparametrige lineare Exponentialfamilie mit kanonischem Parameter θ. Andernfalls spricht man von einer *exponentiellen Dispersionsfamilie*, die als zweiparametrige Exponentialfamilie bezeichnet werden kann. Wir wollen uns von nun an immer, wenn wir von Exponentialfamilie sprechen, auf die in (5) angegebene Form beziehen.

Beispiel (Normalverteilung): Für $Y \sim N(\mu,\sigma^2)$ ist die Dichtefunktion f_Y gegeben durch

$$\begin{aligned} f(y;\mu,\sigma^2) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right) \\ &= \exp\left(\frac{y\mu - \frac{\mu^2}{2}}{\sigma^2} - \frac{1}{2}\left(\frac{y^2}{\sigma^2} + \log\left(2\pi\sigma^2\right)\right)\right). \end{aligned}$$

Für $\theta = \mu$ und $\phi = \sigma^2$ ist die Normalverteilung somit Mitglied der Exponentialfamilie mit

$$a(\phi) = \phi,\ b(\theta) = \frac{\theta^2}{2} \text{ und } c(y,\phi) = -\frac{1}{2}\left(\frac{y^2}{\sigma^2} + \log\left(2\pi\sigma^2\right)\right).$$

Wesentliche Merkmale einer Verteilung sind die Momente der Zufallsvariable. Für die Exponentialfamilie lassen sich Erwartungswert und Varianz unmittelbar aus folgenden, allgemein gültigen Eigenschaften herleiten:

Satz 2.1 (Eigenschaften der Score-Funktion) Sei Y eine Zufallsvariable mit differenzierbarer Dichte- oder Wahrscheinlichkeitsfunktion $f(y;\theta)$. Dann gilt für die Score-Funktion $\frac{\partial}{\partial\theta}\log f(y;\theta)$:

$$\mathbb{E}\left(\frac{\partial \log f(Y;\theta)}{\partial\theta}\right) = 0, \tag{6}$$

$$\mathbb{E}\left(\frac{\partial \log f(Y;\theta)}{\partial\theta}\right)^2 + \mathbb{E}\left(\frac{\partial^2 \log f(Y;\theta)}{\partial\theta^2}\right) = 0. \tag{7}$$

Beweis: siehe Anhang A

Korollar 2.2: Für eine Zufallsvariable Y mit Dichte- oder Wahrscheinlichkeitsfunktion $f(y;\theta)$ aus der Exponentialfamilie gilt

$$\begin{aligned}\mathbb{E}(Y) &= b'(\theta)\\ \operatorname{var}(Y) &= a(\phi)b''(\theta).\end{aligned}$$

Beweis: Für die Exponentialfamilie liefert (6)

$$0 = \mathbb{E}\left(\frac{\partial}{\partial\theta}\left(\frac{Y\theta - b(\theta)}{a(\phi)} + c(Y,\phi)\right)\right) = \frac{1}{a(\phi)}\mathbb{E}\left(Y - b'(\theta)\right),$$

woraus

$$\mathbb{E}(Y) = b'(\theta)$$

folgt.

Und mit (7) erhält man

$$0 = \mathbb{E}\left(\frac{\partial \log f(Y;\theta)}{\partial\theta}\right)^2 + \mathbb{E}\left(\frac{\partial^2 \log f(Y;\theta)}{\partial\theta^2}\right) = -\frac{b''(\theta)}{a(\phi)} + \frac{1}{a(\phi)^2}\underbrace{\mathbb{E}(Y - b'(\theta))^2}_{=\operatorname{var}(Y)}$$

und somit

$$\operatorname{var}(Y) = a(\phi)b''(\theta).$$

□

Die Varianz lässt sich also als Produkt zweier Funktionen schreiben. Die Funktion $b''(\theta)$ beschreibt dabei den Einfluss des Parameters θ, mit dem wegen $\mathbb{E}(Y) = b'(\theta)$ auch eine Abhängigkeit der Varianz vom Erwartungswert erklärt wird. Dieser soll von nun an mit $\mu := \mathbb{E}(Y) = b'(\theta)$ bezeichnet werden. Damit lässt sich die Varianz durch

$$\operatorname{var}(Y) = a(\phi)b''(\theta) = a(\phi)\frac{\partial b'(\theta)}{\partial\theta} = a(\phi)\frac{\partial\mu}{\partial\theta}$$

schreiben. Um den Einfluss von μ auf $\operatorname{var}(Y)$ hervorzuheben, wird ab hier der Begriff der *Varianzfunktion* verwendet. Diese ist durch $V(\mu) := \frac{\partial\mu}{\partial\theta} = b''(\theta)$ definiert und ermöglicht die Schreibweise

$$\operatorname{var}(Y) = a(\phi)V(\mu).$$

Damit lässt sich $a(\phi)$ als jener Teil der Varianz interpretieren, der von μ unabhängig ist. Bei ϕ handelt es sich um den *Dispersionsparameter*. Allgemein wird beim GLM angenommen, dass es für Beobachtungen $y_1, \ldots, y_n$

einen gemeinsamen Dispersionsparameter gibt und nur die Funktionen $a_i(\cdot)$ voneinander unterscheidbar sind, jedoch in der Form

$$a_i(\phi) = a_i \cdot \phi$$

vorliegen, wobei a_i als bekannte beobachtungsspezifische Gewichte betrachtet werden können (vgl. McCullagh und Nelder, 1989, S. 29).

Beispiel (Varianzfunktion der Normalverteilung): Für die Normalverteilung ergibt sich aus $a(\phi) = \phi = \sigma^2$ und $b(\theta) = \frac{\theta^2}{2}$

$$\text{var}(Y) = a(\phi)V(\mu) = \sigma^2 b''(\theta) = \sigma^2 \cdot 1 = \sigma^2.$$

Bei der Normalverteilung liegt also die spezielle Varianzfunktion $V(\mu) = 1$ vor, womit keine Einflussnahme der Größe des Erwartungswerts auf die Varianz zustande kommt. Wir werden in Abschnitt 2.1.2 andere Mitglieder der Exponentialfamilie betrachten, die sich in dieser Beziehung von der Normalverteilung unterscheiden. Davor soll noch die Herleitung höherer Momente diskutiert werden. Wir betrachten dazu die *Kumulantenerzeugende* Funktion $K_Y(t)$, deren Zusammenhang mit der *Momentenerzeugenden* Funktion $M_Y(t)$ durch $K_Y(t) = \log M_Y(t)$ gegeben ist. Die l-te Kumulante $\kappa_l(Y)$ wird dabei bestimmt durch

$$\kappa_l(Y) = \left.\frac{\partial^l K_Y(t)}{\partial t^l}\right|_{t=0}. \tag{8}$$

Bemerkung: Kumulanten werden üblicherweise einer Zufallsvariable zugeordnet. Da sie aber von Parametern abhängen ist auch die Notation $\kappa_l(Y, \theta, \phi)$ durchaus gebräuchlich.

Momente und Kumulanten einer Zufallsvariable stehen in enger Beziehung zu einander. Insbesondere gilt

$$\begin{aligned}\kappa_1(Y) &= \mathbb{E}(Y)\\ \kappa_2(Y) &= \mathbb{E}(Y-\mu)^2\\ \kappa_3(Y) &= \mathbb{E}(Y-\mu)^3.\end{aligned}$$

In der Exponentialfamilie gilt allgemein

$$\begin{aligned}1 &= \int_{\mathbb{R}} \exp\left(\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right) dy\\ &= \exp\left(-\frac{b(\theta)}{a(\phi)}\right) \int_{\mathbb{R}} \exp\left(\frac{y}{a(\phi)}\theta + c(y,\phi)\right) dy\end{aligned}$$

und damit

$$\int_{\mathbb{R}} \exp\left(\frac{y}{a(\phi)}\theta + c(y,\phi)\right) dy = \exp\left(\frac{b(\theta)}{a(\phi)}\right). \tag{9}$$

Die Momentenerzeugende Funktion ist definiert durch

$$M_Y(t) = \mathbb{E}(e^{tY}) = \int_{\mathbb{R}} \exp\left(\frac{y\theta - b(\theta)}{a(\phi)} + ty + c(y,\phi)\right) dy$$

$$= \exp\left(-\frac{b(\theta)}{a(\phi)}\right) \int_{\mathbb{R}} \exp\left(\frac{y}{a(\phi)}(\theta + a(\phi)t) + c(y,\phi)\right) dy.$$

Mit (9) folgt, dass

$$M_Y(t) = \exp\left(-\frac{b(\theta)}{a(\phi)}\right) \exp\left(\frac{b(\theta + a(\phi)t)}{a(\phi)}\right) = \exp\left(\frac{b(\theta + a(\phi)t) - b(\theta)}{a(\phi)}\right)$$

gilt und somit

$$K_Y(t) = \log M(t) = \frac{b\big(\theta + a(\phi)t\big) - b(\theta)}{a(\phi)}.$$

Für die l-te Kumulante $\kappa_l(y)$ resultiert damit bei der Exponentialfamilie

$$\kappa_l(Y) = K_Y^{(l)}(t)\Big|_{t=0} = a(\phi)^{l-1} b^{(l)}\big(\theta + a(\phi)t\big)\big|_{t=0} = a(\phi)^{l-1} b^{(l)}(\theta). \tag{10}$$

Die Funktion $b(\theta)$, die hierbei eine wesentliche Rolle spielt, wird als *Kumulantenfunktion* bezeichnet.

2.1.2 Mitglieder der Exponentialfamilie

In Abschnitt 2.1.1 wurde die Normalverteilung mit $Y \sim N(\mu, \sigma^2)$ als Mitglied unserer Exponentialfamilie identifiziert. Dabei konnte aus der in (5) geforderten Darstellung der Dichte die Kumulantenfunktion $b(\theta)$ sowie die Varianzfunktion $V(\mu)$ einfach abgeleitet werden. Um die Verteilungen diesbezüglich gut miteinander vergleichen zu können, soll in allen Beispielen eine Parametrisierung verwendet werden, in der der Erwartungswert der Zufallsvariable Y durch μ bezeichnet wird. Zur Bestimmung der Kumulanten wird (10) dienen.

Poissonverteilung: $Y \sim P(\mu)$

Die Wahrscheinlichkeitsfunktion

$$f(y;\mu) = \frac{\mu^y}{y!}e^{-\mu} = \exp(y\log\mu - \mu - \log y!), \quad y = 0,1,2,\ldots$$

gehört für $\theta = \log\mu$ und $\phi = 1$ zur Exponentialfamilie mit

$$a(\phi) = \phi, \qquad b(\theta) = \exp(\theta), \qquad c(y,\phi) = -\log y!.$$

Man erhält die Varianzfunktion $V(\mu) = \exp(\theta) = \mu$ und damit die Kumulanten

$$\begin{aligned} \mathbb{E}(Y) &= \exp(\theta) = \mu \\ \operatorname{var}(Y) &= a(\phi)V(\mu) = \mu \\ \kappa_l(Y) &= \exp(\theta) = \mu \quad \text{für } l > 2. \end{aligned}$$

Charakteristisch für die Poissonverteilung ist die Tatsache, dass durch μ alle Kumulanten bestimmt sind. Für den Fall, dass die Annahme $\operatorname{var}(Y) = \mathbb{E}(Y) = \mu$ für beobachtete Daten nicht plausibel erscheint und durch $\operatorname{var}(Y) = \phi\mathbb{E}(Y)$ mit $\phi \neq 1$ ersetzt werden muss, liegt keine Poissonverteilung mehr vor.

Gammaverteilung: $Y \sim \Gamma(\mu,\nu)$

Mit der durch $\mathbb{E}(Y) = \mu$ und $\operatorname{var}(Y) = \mu^2/\nu$ bestimmten Parametrisierung lässt sich die Dichte der Gammaverteilung in die Form

$$\begin{aligned} f(y;\mu,\nu) &= \exp\left(-\frac{\nu}{\mu}y + \nu\log\nu - \nu\log\mu + (\nu-1)\log y - \log\Gamma(\nu)\right) \\ &= \exp\left(\frac{y\left(-\frac{1}{\mu}\right) + \log\frac{1}{\mu}}{\frac{1}{\nu}} + \nu\log\nu + (\nu-1)\log y - \log\Gamma(\nu)\right), \\ & y > 0 \end{aligned} \tag{11}$$

bringen. Für $\theta = -1/\mu$ und $\phi = 1/\nu$ liefert dies die Exponentialfamilie mit

$$a(\phi) = \phi, \quad b(\theta) = -\log(-\theta), \quad c(y,\phi) = \frac{1}{\phi}\log\frac{1}{\phi} + \left(\frac{1}{\phi} - 1\right)\log y - \log\Gamma(\frac{1}{\phi}).$$

Man erhält die Varianzfunktion $V(\mu) = 1/\theta^2 = \mu^2$ und damit die Kumulanten

$$\begin{aligned}\mathbb{E}(Y) &= -\frac{1}{\theta} = \mu \\ \operatorname{var}(Y) &= a(\phi)V(\mu) = \frac{1}{\nu}\mu^2 \\ \kappa_l(Y) &= (l-1)!\,\nu\left(\frac{\mu}{\nu}\right)^l \quad \text{für } l > 2.\end{aligned}$$

Wegen $V(\mu) = \mu^2$ ist die Varianz bei der Gammaverteilung proportional zum Quadrat des Erwartungswerts. Im Gegensatz zur Poissonverteilung ermöglicht es hier ein Dispersionsparameter $\phi = 1/\nu$, die Varianz zu korrigieren.

Standardisierte Binomialverteilung: $\tilde{Y} = mY \sim B(m, \mu)$

Für eine binomialverteilte Zufallsvariable $\tilde{Y} \sim B(m, \mu)$ ist die Wahrscheinlichkeitsfunktion gegeben durch

$$f(\tilde{y}; m, \mu) = \binom{m}{\tilde{y}}\mu^{\tilde{y}}(1-\mu)^{m-\tilde{y}}, \quad \tilde{y} = 0, 1, 2, \ldots, m.$$

Dabei kann $\tilde{y}$ als Anzahl der Erfolge bei m Versuchen interpretiert werden und die Werte $0, \ldots, m$ annehmen. Für den Erwartungswert gilt $\mathbb{E}(\tilde{Y}) = m\mu$. Da der Parameter m aber Größe des Wertebereichs und Erwartungswert der Zufallsvariable nicht beeinflussen soll, ist es sinnvoll anstatt der absoluten die relativen Häufigkeiten der Erfolge zu betrachten. Mit der entsprechenden Standardisierung $Y := \tilde{Y}/m$ gilt $\mathbb{E}(Y) = \mu$ und

$$\begin{aligned}f(y; m, \mu) &= \binom{m}{my}\mu^{my}(1-\mu)^{m-my} \\ &= \exp\left(\log\binom{m}{my} + my\log\mu + m(1-my)\log(1-\mu)\right) \\ &= \exp\left(\frac{y\log\left(\frac{\mu}{1-\mu}\right) - \log\frac{1}{1-\mu}}{\frac{1}{m}} + \log\binom{m}{my}\right), \quad y = 0, \frac{1}{m}, \ldots, 1.\end{aligned}$$

Dabei nimmt y die Werte $0, \frac{1}{m}, \frac{2}{m}, \ldots, 1$ an und für $\theta = \log\frac{\mu}{1-\mu}$ und $\phi = \frac{1}{m}$ liegt eine Exponentialfamilie vor mit

$$a(\phi) = \phi, \qquad b(\theta) = \log\frac{1}{1-\mu} = \log(1+\exp(\theta)), \qquad c(y, \phi) = \log\binom{\frac{1}{\phi}}{\frac{y}{\phi}}.$$

Weiters gilt $\mu = \frac{\exp(\theta)}{1+\exp(\theta)}, V(\mu) = b''(\theta) = \frac{\exp(\theta)}{(1+\exp(\theta))^2} = \mu(1-\mu)$ sowie

$$\mathbb{E}(Y) = \mu$$
$$\text{var}(Y) = a(\phi)V(\mu) = \frac{1}{m}\mu(1-\mu).$$

Bemerkung 1: Die Funktion $V(\mu) = \mu(1-\mu)$ bringt die Eigenschaft mit sich, dass die Varianz der Zufallsvariable y dann am größten ist, wenn Erfolg und Misserfolg des Versuchs gleich wahrscheinlich sind, also $\mu = 1/2$ gilt. Für $\mu \to 0$ und $\mu \to 1$ strebt die Varianz gegen 0.

Bemerkung 2: In nicht-standardisierter Form gehört die Binomialverteilung keiner einparametrigen Exponentialfamilie an.

2.1.3 Linkfunktionen

Neben der Erweiterung der Verteilungsannahme ist die zweite wesentliche Verallgemeinerung gegenüber dem LM die Verwendung einer *Linkfunktion*. Im LM modellieren die erklärenden Variablen x_i $(i = 1, \ldots, n)$ die Erwartungswerte μ_i direkt durch

$$\mu_i = x_i^t \beta.$$

Für den linearen Prädiktor im GLM führen McCullagh und Nelder (1989) die Bezeichnung $\eta_i := x_i^t\beta$ ein und definieren die Linkfunktion $g(\cdot)$ als die Funktion, die den Zusammenhang zwischen linearem Prädiktor und Erwartungswert durch

$$g(\mu_i) = \eta_i$$

erklärt. Die Linkfunktion muss also einem Wert von μ eindeutig einen Wert η zuordnen, umgekehrt muss zu einem linearen Prädiktor auch der dazugehörende Erwartungswert bestimmt werden können. Für die Linkfunktion kommen also nur streng monotone, differenzierbare Funktionen in Frage, so dass die Inverse $g^{-1}(\eta) = \mu$ existiert. Das LM kann mit dieser Notation als Spezialfall, der nur den sogenannten *Identitäts-Link* $(g(\mu) = \mu)$ erlaubt, angesehen werden. Dagegen kann diese Linkfunktion etwa bei der Poissonverteilung unpassend sein, da η beliebig aus $\mathbb{R}$ sein kann, aber $\mu > 0$ gelten muss. Die Funktion $g(\cdot)$ muss also so beschaffen sein, dass $g^{-1}(\cdot)$ Prädiktorwerte auf für die Verteilung zulässige Werte von μ abbildet.
Jede Verteilung besitzt eine spezielle Linkfunktion, den *kanonischen Link*. Mit ihm wird durch den linearen Prädiktor η gleichzeitig der kanonische Parameter der Verteilung θ modelliert. Aus $\eta = \theta$ folgt

$$b'(\theta) = \mu = g^{-1}(\eta) = g^{-1}(\theta),$$

womit die kanonische Linkfunktion einer Verteilung durch $g(\mu) = b'^{-1}(\mu)$, also der Inversen der abgeleiteten Kumulantenfunktion $b(\cdot)$ bestimmt ist. Bei Verwendung des kanonischen Links vereinfacht sich die ML-Schätzung (siehe Abschnitt 2.2) und es existiert eine suffiziente Statistik für den Parametervektor β, der die Regressionskoeffizienten enthält. Der kanonische Link führt also zu angenehmen Eigenschaften im Modell und reduziert den Rechenaufwand, die beste Anpassung an vorliegende Daten ist durch ihn aber nicht notwendigerweise garantiert.

Verteilung	kanonischer Link
Normal	$\eta = \mu$
Poisson	$\eta = \log \mu$
Binomial	$\eta = \log \frac{\mu}{1-\mu}$
Gamma	$\eta = \frac{1}{\mu}$

Tabelle 1: Kanonische Linkfunktionen einiger Mitglieder der Exponentialfamilie

Der *Log-Link*, der kanonische Link der Poissonverteilung, kann als Grenzwert einer Familie von Linkfunktionen aufgefasst werden, nämlich der Potenzfamilie, die für $\mu > 0$ definiert werden kann durch

$$\eta = g(\mu) = \begin{cases} \frac{\mu^\lambda - 1}{\lambda} & \text{für } \lambda \neq 0 \\ \log \mu & \text{für } \lambda = 0 \end{cases}. \tag{12}$$

Bei der Wahl der Linkfunktion für ein Modell ist es sinnvoll, eine ganze Familie von Funktionen zu betrachten um Vergleiche anstellen zu können.

Eine große Klasse von Linkfunktionen bietet sich für die standardisierte Binomialverteilung an. Da hier $\mu \in (0, 1)$ gilt, muss eine Linkfunktion $g(\cdot)$ das Intervall $(0, 1)$ auf die Menge aller reellen Zahlen abbilden. Somit wird jede Funktion, deren Inverse eine stetige Verteilungsfunktion einer Zufallsvariable ist, zu einer zulässigen Linkfunktion für die standardisierte Binomialverteilung. Auch der kanonische Link $g(\mu) = \log \frac{\mu}{1-\mu}$ steht in dieser Beziehung zu einer Verteilungsfunktion, nämlich jener der *logistischen* Verteilung, weshalb er auch unter der Bezeichnung *Logit-Link* bekannt ist. Häufige Verwendung findet auch der sogenannte *Probit-Link*, definiert durch $g(\mu) = \Phi^{-1}(\mu)$, wobei $\Phi(\cdot)$ die Verteilungsfunktion einer $N(0, 1)$ verteilten Zufallsvariable ist. Johnson und Albert (1999, S. 79) beschreiben es als Standard, bei der Binomialverteilung in der Praxis mehrere Linkfunktionen zu testen und sich

dann für die zu entscheiden, mit der der zufriedenstellendste Fit erzielt wurde. Detailliertere Informationen zur Wahl der Linkfunktion allgemein gibt Pregibon (1980).

Welche Rolle die Linkfunktion bei der ML-Schätzung im Modell spielt, werden wir im folgenden Abschnitt sehen.

2.2 Die Maximum Likelihood Schätzung

Die bisher formulierten Verallgemeinerungen des LM lassen sich zu drei Komponenten zusammenfassen, die die Basis für die Spezifikation des GLMs liefern:

- zufällige Komponente: $Y_i \overset{ind.}{\sim} F(y_i; \theta_i, \phi_i)$
- systematische Komponente: $\eta_i = x_i^t \beta$
- Beziehung der beiden Komponenten: $g(\mu_i) = \eta_i$

Dabei ist $Y = (Y_1, \ldots, Y_n)^t$ ein Zufallsvektor mit unabhängigen Variablen Y_i, die unter der Voraussetzung der Existenz von $\mathbb{E}(Y_i)$ und $\text{var}(Y_i)$ der gleichen Verteilung F der einparametrigen Exponentialfamilie folgen, wobei aber beobachtungsspezifische kanonische Parameter θ_i und Momente $\mathbb{E}(Y_i) = \mu_i$ sowie $\text{var}(Y_i) = a(\phi_i)V(\mu_i)$ vorliegen. Weiters sei $y = (y_1, \ldots, y_n)^t$ die Realisierung von Y.
Die p-dimensionalen Vektoren $x_i = (x_{i1}, \ldots, x_{ip})^t$ enthalten die zu den beobachteten y_i gehörenden, bekannten Prädiktor-Variablen, die zur *Designmatrix* $X = (x_1, \ldots, x_n)^t \in \mathbb{R}^{n \times p}$ zusammengefasst werden. Das Hauptinteresse besteht nun darin, den ML-Schätzer für den Parametervektor $\beta = (\beta_1, \ldots, \beta_p)^t$ zu bestimmen.

Für den Vektor der kanonischen Parameter $\theta = (\theta_1, \ldots, \theta_n)^t$ ist die Log-Likelihood Funktion der Stichprobe wegen der Unabhängigkeit der y_i gegeben durch

$$l(\theta, y) = \log f(y; \theta, \phi) = \sum_{i=1}^{n} \log f(y_i; \theta_i, \phi_i)$$

und da die Response-Variablen Y_i einer Verteilung aus der Exponentialfamilie folgen, gilt wegen (5)

$$l(\theta, y) = \sum_{i=1}^{n} \left(\frac{y_i \theta_i - b(\theta_i)}{a(\phi_i)} + c(y_i, \phi_i) \right) . \tag{13}$$

Die Dispersionsparameter $\phi = (\phi_1, \ldots, \phi_n)^t$ werden an dieser Stelle als sogenannte *Nuisance Parameter*, also als bekannte Größen betrachtet.[1] Die Linkfunktion $g(\cdot)$ beschreibt den Zusammenhang zwischen dem Vektor der Erwartungswerte μ und den linearen Prädiktoren $\eta = X^t\beta$ und ermöglicht die Herleitung von Score-Gleichungen der Form

$$\frac{\partial l(\theta, y)}{\partial \beta_j} \stackrel{!}{=} 0, \qquad j = 1, \ldots, p. \tag{14}$$

Die Anwendung der Kettenregel liefert

$$\frac{\partial l(\theta, y)}{\partial \beta_j} = \frac{\partial l(\theta, y)}{\partial \theta_i} \frac{\partial \theta_i}{\partial \mu_i} \frac{\partial \mu_i}{\partial \eta_i} \frac{\partial \eta_i}{\partial \beta_j}$$

und wegen

$$\frac{\partial \mu_i}{\partial \theta_i} = \frac{\partial b'(\theta_i)}{\partial \theta_i} = b''(\theta_i) = V(\mu_i)\,,$$
$$\frac{\partial \mu_i}{\partial \eta_i} \frac{\partial \eta_i}{\partial \beta_j} = \frac{\partial \mu_i}{\partial g(\mu_i)} \frac{\partial \eta_i}{\partial \beta_j} = \frac{x_{ij}}{g'(\mu_i)}$$

folgt für die Score-Gleichungen (14)

$$\begin{aligned} \frac{\partial l(\theta, y)}{\partial \beta_j} &= \sum_{i=1}^{n} \frac{\partial}{\partial \theta_i} \left(\frac{y_i\theta_i - b(\theta_i)}{a(\phi_i)} + c(y_i, \phi_i) \right) \frac{1}{V(\mu)} \frac{x_{ij}}{g'(\mu_i)} \\ &= \sum_{i=1}^{n} \frac{y_i - \mu_i}{a(\phi_i) V(\mu_i)} \frac{x_{ij}}{g'(\mu_i)} \stackrel{!}{=} 0, \qquad j = 1, \ldots, p. \end{aligned} \tag{15}$$

Der ML-Schätzer $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_p)^t$ ist damit als die Lösung des Gleichungssystems (15) definiert. Wird für $g(\cdot)$ die kanonische Linkfunktion verwendet, gilt

$$g'(\mu_i) = \frac{\partial g(\mu_i)}{\partial \mu_i} = \frac{\partial \theta_i}{\partial \mu_i} = \frac{1}{V(\mu_i)}\,, \tag{16}$$

und das Gleichungssystem (15) vereinfacht sich zu

$$\frac{\partial l(\theta, y)}{\partial \beta_j} = \sum_{i=1}^{n} \frac{y_i - \mu_i}{a(\phi_i)} x_{ij} \stackrel{!}{=} 0, \qquad j = 1, \ldots, p. \tag{17}$$

Beim einfachen LM ist eine direkte Lösung des Gleichungssystems möglich, im Allgemeinen muss der MLE $\hat{\beta}$ iterativ bestimmt werden.

[1] Dabei kann ohne Verlust der Allgemeinheit $a(\phi_i)$ durch $a_i(\phi)$ ersetzt werden, wodurch sich ϕ als globaler Dispersionsparameter interpretieren lässt und die Funktionen $a_i(\cdot)$ als beobachtungsspezifische Gewichtungsfunktionen betrachtet werden können.

2.2.1 Newton-Raphson Methode und Iterative Weighted Least Squares

Um den ML-Schätzer $\hat{\beta}$ zu bestimmen, wendet man zunächst die Newton-Raphson Methode an. Diese liefert für die Lösung von 15 die Iterationsvorschrift

$$\beta^{(k+1)} = \beta^{(k)} - H_k^{-1} \nabla l_k, \tag{18}$$

wobei ∇l_k und H_k^{-1} den Gradient zu $l(\theta, y)$ und die Inverse der Hesse-Matrix H bezeichnen, ausgewertet jeweils in $\beta^{(k)}$. Die Einträge von ∇l sind von der Form

$$(\nabla l)_j = \frac{\partial l(\theta, y)}{\partial \beta_j} = \sum_{i=1}^{n} \frac{y_i - \mu_i}{a(\phi_i) V(\mu_i)} \frac{x_{ij}}{g'(\mu_i)}, \quad j = 1, \ldots, p,$$

und lassen sich mit

$$\begin{aligned} d_i &:= g'(\mu_i), \\ w_i &:= \frac{1}{a(\phi_i) V(\mu_i) (g'(\mu_i))^2} \end{aligned}$$

umschreiben zu

$$(\nabla l)_j = \sum_{i=1}^{n} (y_i - \mu_i) w_i d_i x_{ij}, \quad j = 1, \ldots, p. \tag{19}$$

Mit den Diagonalmatrizen $D = diag(d_i)$ und $W = diag(w_i)$ gilt damit für den Gradienten in Matrixnotation

$$\nabla l = \frac{\partial l(\theta, y)}{\partial \beta} = X^t D W (y - \mu). \tag{20}$$

Die Einträge der Hesse-Matrix sind gegeben durch

$$\begin{aligned} (H)_{jk} = \frac{\partial^2 l(\theta, y)}{\partial \beta_j \partial \beta_k} &= \sum_{i=1}^{n} \frac{\partial}{\partial \beta_k} \left((y_i - \mu_i) d_i w_i x_{ij} \right) \\ &= \sum_{i=1}^{n} \left(\frac{\partial}{\partial \beta_k} (y_i - \mu_i) d_i w_i x_{ij} + (y_i - \mu_i) \frac{\partial}{\partial \beta_k} (d_i w_i x_{ij}) \right) \\ &= \sum_{i=1}^{n} x_{ik} \left(\frac{\partial}{\partial \eta_i} (y_i - \mu_i) d_i w_i x_{ij} + (y_i - \mu_i) \frac{\partial}{\partial \eta_i} (d_i w_i x_{ij}) \right) \\ &= \sum_{i=1}^{n} -x_{ik} \left(w_i - (y_i - \mu_i) \frac{\partial d_i w_i}{\partial \eta_i} \right) x_{ij} \,, \end{aligned}$$

wodurch man die Matrixnotation

$$H = -X^t \tilde{W} X \tag{21}$$

gewinnt mit $\tilde{W} = diag(\tilde{w}_i)$ und

$$\begin{aligned}
\tilde{w}_i &= w_i - (y_i - \mu_i)\frac{\partial d_i w_i}{\partial \eta_i} \\
&= w_i - (y_i - \mu_i)\frac{\partial}{\partial \mu_i}\left(\frac{1}{a(\phi_i)V(\mu_i)g'(\mu_i)}\right)\frac{1}{g'(\mu_i)} \\
&= w_i - (y_i - \mu_i)\left(-\frac{V'(\mu_i)g'(\mu_i) + V(\mu_i)g''(\mu_i)}{a(\phi_i)(V(\mu_i)g'(\mu_i))^2 g'(\mu_i)}\right) \\
&= w_i + (y_i - \mu_i)\frac{V'(\mu_i)g'(\mu_i) + V(\mu_i)g''(\mu_i)}{a(\phi_i)(V(\mu_i))^2(g'(\mu_i))^3}.
\end{aligned} \tag{22}$$

Mit (20) und (21) folgt für die Iterationsvorschrift (18)

$$\beta^{(k+1)} = \beta^{(k)} + (X^t \tilde{W} X)^{-1} X^t D W (y - \mu). \tag{23}$$

Die Matrizen $\tilde{W}, D$ und W werden dabei im Vektor der Erwartungswerte μ ausgewertet, für den der aktuelle Schätzer $\mu^{(k)}$ von $\beta^{(k)}$ abhängt, was eine Notation der Form

$$\beta^{(k+1)} = \beta^{(k)} + (X^t \tilde{W}_k X)^{-1} X^t D_k W_k (y - \mu^{(k)}), \quad k \geq 0 \tag{24}$$

sinnvoll macht. McCullagh und Nelder (1989, S. 40 ff) zeigen wie diese Vorschrift zu einem *Iterative Weighted Least Squares* (IWLS) Problem umformuliert werden kann. Dazu werden sogenannte Pseudobeobachtungen (adjusted dependent variables) $z = (z_1, \ldots, z_n)^t$ eingeführt, die zum Zeitpunkt der k-ten Iteration definiert sind durch

$$z^{(k)} = X^t \beta^{(k)} + \tilde{W}_k^{-1} D_k W_k (y - \mu^{(k)}), \quad k \geq 0. \tag{25}$$

Damit lässt sich (24) zu

$$\beta^{(k+1)} = (X^t \tilde{W}_k X)^{-1} X^t \tilde{W}_k z^{(k)}, \quad k \geq 0 \tag{26}$$

umschreiben und entspricht dem Gleichungssystem eines *Weighted Least Squares* (WLS) Problems. Die Tatsache, dass die Pseudobeobachtungen $z^{(k)}$ sowie $\tilde{W}_k$ in $\mu^{(k)}$ ausgewertet werden und damit vom aktuellen Schätzer $\beta^{(k)}$ abhängen, macht es zu einem IWLS Problem.[2]

[2]Die Ausnahme bildet das LM, in dem $\tilde{W} = I$ und $z = y$ gilt.

2.2.2 Fisher Scoring

Mit dem Newton-Raphson Verfahren lässt sich ein Algorithmus zur Bestimmung von $\hat{\beta} = \lim_{k\to\infty} \beta^{(k)}$ herleiten. Um diesen zu vereinfachen, wird die in (21) verwendete Hessematrix üblicherweise durch ihren Erwartungswert, also die Informationsmatrix, ersetzt. Wegen $\mathbb{E}(-X^t\tilde{W}X) = -X^tWX$ erhält man für (24)

$$\beta^{(k+1)} = \beta^{(k)} + (X^tW_kX)^{-1}X^tD_kW_k(y-\mu^{(k)}), \quad k \geq 0 \tag{27}$$

was zu einem IWLS-Problem mit

$$\beta^{(k+1)} = (X^tW_kX)^{-1}X^tW_kz^{(k)}, \quad k \geq 0 \tag{28}$$

und den Pseudobeobachtungen

$$\begin{aligned} z^{(k)} &= X^t\beta^{(k)} + W_k^{-1}D_kW_k(y-\mu^{(k)}) \\ &= X^t\beta^{(k)} + D_k(y-\mu^{(k)}), \quad k \geq 0 \end{aligned} \tag{29}$$

führt. Diese Technik, die als *Fisher Scoring* (FS) bezeichnet wird, muss für den Fall des kanonischen Links nicht angewendet werden. In diesem Fall gilt wegen (16) $g''(\mu_i) = -V'(\mu_i)/(V(\mu_i))^2$ und damit folgt für (22)

$$\begin{aligned} \tilde{w}_i &= w_i + (y_i-\mu_i)\frac{V'(\mu_i)g'(\mu_i) + V(\mu_i)g''(\mu_i)}{a(\phi_i)(V(\mu_i))^2(g'(\mu_i))^3} \\ &= w_i + (y_i-\mu_i)\frac{V'(\mu_i)/V(\mu_i) + V(\mu_i)(-V'(\mu_i)/(V(\mu_i))^2}{a(\phi_i)(V(\mu_i))^2(g'(\mu_i))^3} \\ &= w_i \end{aligned}$$

und somit automatisch $\tilde{W}_k = W_k, \quad k \geq 0$.

Bemerkung 1: Für die Pseudobeobachtungen $z^{(k)}$ gilt beim FS

$$\begin{aligned} z^{(k)} &= X^t\beta^{(k)} + D_k(y-\mu^{(k)}) \\ &= \eta^{(k)} + g'(\mu^{(k)})(y-\mu^{(k)}) \\ &= g(\mu^{(k)}) + g'(\mu^{(k)})(y-\mu^{(k)}), \end{aligned}$$

womit sie als linearisierte Responses $g(y)$ interpretiert werden können, angenähert durch die Taylor-Entwicklung der Ordnung 1 um $\mu^{(k)}$.

Bemerkung 2: Es gilt $\mathbb{E}(z^{(k)}) = \eta^{(k)}$ und $\text{var}(z^{(k)}) = D_k\text{var}(Y)D_k = W_k^{-1}$, womit der Varianz-Kovarianzmatrix der Pseudobeobachtungen $z^{(k)}$ gerade

die Inverse der Gewichtsmatrix W_k entspricht.

FS liefert also einen Algorithmus zur Lösung des IWLS-Problems und somit zur Berechnung des ML-Schätzers $\hat{\beta}$. Die Iteration gemäß den Vorschriften (28) und (29) wird ausführt, bis eine hinreichend kleine relative Änderung

$$\frac{||\beta^{(k+1)} - \beta^{(k)}||}{||\beta^{(k)}||}$$

vorliegt. Dabei sind im Gegensatz zum Newton-Raphson Verfahren keine Startwerte für den Parametervektor β sondern nur Startwerte für η oder μ notwendig, was eine Erleichterung der Implementierung darstellt (vgl. Hardin und Hilbe, 2007, Kap. 3.2, 3.4).

2.2.3 Eigenschaften das Schätzers

Um Eigenschaften des MLE $\hat{\beta}$ zu untersuchen, betrachten wir die Entwicklung der Score-Funktion um den wahren Parameter β. Analog zur Herleitung der Iterationsvorschrift (23) folgt aus

$$0 = \left.\frac{\partial l(\theta, y)}{\partial \beta}\right|_{\hat{\beta}} \approx \left.\frac{\partial l(\theta, y)}{\partial \beta}\right|_{\beta} + \left.\frac{\partial^2 l(\theta, y)}{\partial \beta \partial \beta^t}\right|_{\beta} (\hat{\beta} - \beta)$$

beim FS

$$\hat{\beta} - \beta \approx (X^t W X)^{-1} X^t D W (y - \mu),$$

also

$$\begin{aligned}
\mathbb{E}(\hat{\beta}) &\approx \beta, \\
\operatorname{var}(\hat{\beta}) &\approx (X^t W X)^{-1} X^t D W \operatorname{var}(Y) W D X (X^t W X)^{-1} \\
&= (X^t W X)^{-1} X^t D W (D W D)^{-1} W D X (X^t W X)^{-1} \\
&= (X^t W X)^{-1}.
\end{aligned}$$

Dabei ist $(X^t W X)^{-1}$ als $(X^t W(\beta, \phi) X)^{-1}$ zu verstehen. Um diese Matrix zu schätzen und damit Information über die Standardfehler der $\hat{\beta}_j$ zu gewinnen, ist es üblich, die Auswertung in $\hat{\beta}$ vorzunehmen, womit

$$\widehat{\operatorname{var}}(\hat{\beta}) = (X^t W(\hat{\beta}, \phi) X)^{-1}$$

resultiert.

2.2.4 Pearson Residuen

Die Dispersionsparameter ϕ_i wurden bisher als bekannt vorausgesetzt. Wie in Abschnitt 2.1.1 erwähnt, beinhaltet ein Modell jedoch im Allgemeinen einen festen aber nicht bekannten Parameter ϕ und es liegen $a_i(\phi) = a_i\,\phi$ mit bekannten Gewichten a_i vor.[3] Es gilt dann $\phi = \mathrm{var}(Y_i)/(a_i V(\mu_i))$ und

$$\frac{1}{n}\sum_{i=1}^{n}\frac{(y_i-\mu_i)^2}{a_i V(\mu_i)}$$

ist wegen

$$\mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n}\frac{(Y_i-\mu_i)^2}{a_i V(\mu_i)}\right) = \frac{1}{n}\sum_{i=1}^{n}\frac{\mathbb{E}(Y_i-\mu_i)^2}{a_i V(\mu_i)} = \frac{1}{n}\sum_{i=1}^{n}\phi\frac{\mathrm{var}(Y_i)}{\mathrm{var}(Y_i)} = \phi$$

erwartungstreuer Schätzer für ϕ, jedoch nur dann unverzerrt, wenn μ bekannt ist. Da μ aber bei einer Modellanpassung selbst durch $\hat{\mu}$ geschätzt werden muss, verwendet man als Dispersionsschätzer die Statistik

$$\hat{\phi} = \frac{1}{n-p}\sum_{i=1}^{n}\frac{(y_i-\hat{\mu}_i)^2}{a_i V(\hat{\mu}_i)} = \frac{1}{n-p}X^2, \tag{30}$$

welche als mittlere *Pearson-Statistik* bezeichnet wird. Ihre Summanden sind die Quadrate der *Pearson Residuen*, die durch

$$r_i = \frac{y_i-\hat{\mu}_i}{\sqrt{a_i V(\hat{\mu}_i)}} \tag{31}$$

definiert sind. Sie messen den Unterschied zwischen Beobachtungen und gefitteten Werten, skaliert mit der geschätzten Standardabweichung und stellen somit eine weitere Verallgemeinerung gegenüber dem LM dar. Dort folgt die Pearson-Statistik einer χ^2-Verteilung, allgemein trifft dies in der Exponentialfamilie nur asymptotisch zu.

2.3 Goodness of Fit

Eine Modellanpassung lässt sich als Ersetzen der Datenpunkte y_i durch *gefittete* Werte $\hat{\mu}_i$ auffassen. Der exakte Fit $y_i = \hat{\mu}_i$ wird dabei im Allgemeinen nicht erreicht und es stellt sich dann die Frage nach einem Maß der Güte der Modellanpassung, welches Auskunft darüber gibt, ob die Abweichung

[3] Die Bestimmung von $\hat{\beta}$ beeinflusst dies nicht, da Unabhängigkeit zwischen ϕ und θ besteht.

von $\hat{\mu}$ gegenüber y toleriert werden kann oder nicht. Die in (30) definierte Pearson-Statistik ist eine Möglichkeit, die Qualität der Anpassung zu messen. McCullagh und Nelder (1989, S. 34 ff) weisen darauf hin, dass diese eine einfache Interpretation erlaubt, verwenden sie aber vorwiegend zur Analyse von Residuen.

2.3.1 Deviance

Einen anderen Ansatz zur Beurteilung der Modellanpassung bietet die *Likelihood Ratio Test* (LRT)-Statistik . Diese vergleicht für verschiedene Schätzer von μ die entsprechenden Werte der Log-Likelihood Funktion, die dabei als Funktion in μ betrachtet wird, dessen Schätzer im Modell durch $\hat{\mu}$ ermittelt wird. Das einfachste Modell ist das *Nullmodell*, welches nur einen Parameter enthält und somit $\mathbb{E}(Y_i) = \mu_0$ modelliert. Dabei wird die gesamte Variabilität der Responses der zufälligen Komponente zugeschrieben, während andere eventuell vorliegende Variablen unberücksichtigt bleiben. Im Gegensatz dazu enthält das volle oder *saturierte* Modell bei n Beobachtungen ebenso viele Parameter und erlaubt die exakte Anpassung durch $\hat{\mu}_i = y_i$ für $i = 1, \ldots, n$. Die Funktion $l(\mu, y) = \log f(y; \mu)$ wird dabei an der Stelle $\mu = y$ zu $\log f(y; y)$ maximiert. Dieser Ausdruck, der nur von den Beobachtungen und nicht vom Parameterschätzer abhängt, dient generell als Referenz zum Vergleich mit dem ML-Schätzer aus dem zu bewertenden Modell.

Definition 2.1 (skalierte Deviance): Seien $Y_1, \ldots, Y_n$ unabhängige Zufallsvariablen und $l(\mu, y) = \log f(y; \mu)$ die Log-Likelihood Funktion für die Realisierung $y = (y_1, \ldots, y_n)^t$. Sei außerdem ϕ der gemeinsame Dispersionsparameter. Dann wird die Statistik

$$\frac{1}{\phi} D(y, \mu) = -2\big(l(\mu, y) - l(y, y)\big) \tag{32}$$

als *skalierte Deviance* bezeichnet. Sie liefert ein Maß für den *Lack of Fit*, dessen Minimierung äquivalent zur Maximierung von $l(\mu, y)$ ist.

Bemerkung 1: Bei $D(y, \mu)$ handelt es sich um die unskalierte Deviance, die für unbekannte Dispersionsparameter berechnet werden kann.[4]

Am Beispiel der Normalverteilung zeigt sich, dass die Deviance im LM der

[4] Die Funktion `glm` in `R` gibt standardmäßig die unskalierte Deviance aus, da ϕ im Vorhinein nicht spezifierbar ist.

Fehlerquadratsumme entspricht. Für $Y_i \overset{ind}{\sim} N(\mu_i, \sigma^2)$ mit festem σ^2 gilt

$$\begin{aligned} l(\mu, y) &= \sum_{i=1}^{n} \left(-\frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2}\frac{(y_i - \mu_i)^2}{\sigma^2} \right) \\ &= -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mu_i)^2, \\ l(y, y) &= -\frac{n}{2}\log(2\pi\sigma^2) \end{aligned}$$

und somit gilt mit $\phi = \sigma^2$ für die durch den ML-Schätzer $\hat{\mu}$ skalierte Deviance

$$\frac{1}{\phi}D(y, \hat{\mu}) = \frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i - \hat{\mu}_i)^2 \sim \chi^2_{n-p}.$$

Für andere Verteilungen aus der Exponentialfamilie ist die Annahme $\frac{1}{\phi}D(y, \hat{\mu}) \sim \chi^2_{n-p}$ nur asymptotisch erfüllt, also für $n \to \infty$, was die Relevanz für die Bewertung des Fits deutlich einschränkt. Tabelle 2 enthält die Terme der unskalierten Deviance zu den Mitgliedern der Exponentialfamilie aus Abschnitt 2.1.2.

Verteilung	Deviance $D(y, \hat{\mu})$
Normal	$\sum_{i=1}^{n}(y_i - \hat{\mu}_i)^2$
Poisson	$2\sum_{i=1}^{n}\{y_i \log(y_i/\hat{\mu}_i) - (y_i - \hat{\mu}_i)\}$
Binomial	$2\sum_{i=1}^{n} m_i\{y_i \log[y_i/\hat{\mu}_i] + (1 - y_i)\log[(1 - y_i)/(1 - \hat{\mu}_i)]\}$
Gamma	$2\sum_{i=1}^{n}\{-\log(y_i/\hat{\mu}_i) + (y_i - \hat{\mu}_i)/\hat{\mu}_i\}$

Tabelle 2: Deviance-Terme für einige Mitglieder der Exponentialfamilie

2.3.2 Analysis of Deviance

Die Deviance bewertet also eine Modellanpassung durch Betrachtung der Log-Likelihood Funktion an der Stelle des ML-Schätzers $\hat{\mu}$, der im GLM eine Funktion des Vektors $\hat{\beta}$ ist. Der durch die Deviance bestimmte Wert des Lack of Fit lässt sich verkleinern, indem man die Dimension der Parameterraums vergrößert, was einem Hinzufügen von Variablen im Modell entspricht und umgekehrt wird die Deviance größer, wenn der Parameterraum eingeschränkt wird. Ziel ist es, eine Modellanpassung zu finden, die Variablen, welche nicht entscheidend zur Reduktion der Deviance beitragen, eliminiert. Dazu betrachtet man für $1 \leq q < p$ in einander geschachtelte Modelle *(nested*

models) der Form

$$\begin{aligned} &\text{Modell 1: } \eta = \beta_{q+1}x_{q+1} + \ldots + \beta_p x_p \\ &\text{Modell 2: } \eta = \beta_1 x_1 + \ldots + \beta_p x_p \end{aligned}$$

mit der äquivalenten Formulierung der Hypothesen

$$\begin{aligned} H_0 &: \beta_1 = \ldots = \beta_q = 0 \\ H_A &: \beta_1, \ldots, \beta_q \text{ beliebig.} \end{aligned}$$

Modell 2 enthält Modell 1 dabei als ein Untermodell, in welchem die ersten q Parameter Null gesetzt sind, was mit einer Eliminierung der Variablen $x_1, \ldots, x_q$ gleichbedeutend ist. Für die entsprechenden ML-Schätzer $\hat{\mu}_1$ und $\hat{\mu}_2$ liefert der LRT beim Vergleich der Modelle die Statistik

$$\begin{aligned} -2\big(l(\hat{\mu}_1, y) - l(\hat{\mu}_2, y)\big) &= -2\Big(\big[l(\hat{\mu}_1, y) - l(y, y)\big] - \big[l(\hat{\mu}_2, y) - l(y, y)\big]\Big) \\ &= \frac{1}{\phi}\Big(D(y, \hat{\mu}_1) - D(y, \hat{\mu}_2)\Big), \end{aligned} \tag{33}$$

die damit genau der Reduktion der Deviance entspricht und wenn ϕ bekannt ist, asymptotisch χ^2_q-verteilt ist. Muss ϕ geschätzt werden, dann betrachtet man für den Modellvergleich die Statistik

$$\frac{\Big(D(y, \hat{\mu}_1) - D(y, \hat{\mu}_2)\Big)\Big/q}{\hat{\phi}_2} \sim F_{q,n-q}\,,$$

wobei für $\hat{\phi}_2$ der Dispersionsschätzer des komplexeren Modells verwendet wird.

3 Modelle für unvollständige Daten

Die bisher diskutierte Modellanpassung basierte auf der Annahme, dass sowohl zu den Responses y_i als auch zu den Prädiktoren x_i uneingeschränkter Zugang besteht. In der Praxis können aber Situationen auftreten, in denen Daten, die von Interesse sind, zum Teil nicht beobachtbar sind. Dieses Problem kann in vielerlei Gestalt erscheinen, so z.B. in einem GLM, wenn relevante erklärende Variablen fehlen, weil sie aus irgend einem Grund nicht gemessen werden konnten. Eine andere Möglichkeit ist das Vorliegen von zensierten Response-Variablen, für die anstelle von $Y_i = y_i$ nur $Y_i \in [a_i, b_i]$, $Y_i \leq \tau_i$ oder $Y_i \geq \tau_i$ beobachtet wird. In beiden Situationen spielen nicht-beobachtbare Realisierungen von Zufallsvariablen eine Rolle, deren Information als wesentlich anzusehen ist und deren Einfluss in der Modellschätzung berücksichtigt werden soll.

Bevor wir näher auf diese und andere Fälle eingehen, soll ein Verfahren vorgestellt werden, das sich im Umgang mit unvollständigen Daten bewährt hat.

3.1 Der EM-Algorithmus

3.1.1 Die Theorie

Die folgenden Ausführungen halten sich an Dempster et al. (1977), die einen Algorithmus zur Bestimmung des ML-Schätzers bei unvollständig vorliegenden Daten entwickelten. Es wird dabei allgemein angenommen, dass die Gesamtheit der Daten aus zwei Teilen besteht, dem beobachtbaren Teil Y und den nicht-beobachteten und somit zufälligen Einflüssen Z. Für einen festen Wert $Z = z$ könnte man mittels ML-Schätzung die Funktion

$$l(\theta, y|z) = \log f(y; \theta|z)$$

maximieren, bei der es sich aber um eine bedingte Log-Likelihood Funktion handelt. Natürlich ist man vielmehr an der Maximierung der marginalen Log-Likelihood $l(\theta, y)$ interessiert, die die Zufälligkeit von z nicht einschränkt. Dazu betrachtet man die Dichte der vollständigen Beobachtung (y, z), die durch $f(y, z; \theta)$ ausgedrückt werden soll, wobei in θ alle unbekannten Parameter enthalten sind. Der Zusammenhang dieser gemeinsamen Dichte mit der marginalen Log-Likelihood ist dann gegeben durch

$$l(\theta, y) = \log f(y; \theta) = \log \int f(y, z; \theta)\, dz. \tag{34}$$

Dabei erlaubt eine Spezifizierung von $f(y;\theta)$ im Allgemeinen viele Arten der Wahl von $f(y,z;\theta)$. Es müssen also im Vorhinein Annahmen getroffen werden, welcher Verteilung der zufällige Einfluss Z unterliegt, um eine gemeinsame Dichte $f(y,z;\theta)$ darstellen zu können. Die Berechnung des Integrals in (34) kann sich aber je nach getroffener Annahme oder Datensituation als zu aufwendig erweisen und Schwierigkeiten bei der direkten Lösung der resultierenden Score-Gleichungen bewirken, was einen anderen Zugang motiviert. Dabei wird die bedingte Dichte von Z, gegeben Y und θ als

$$f(z|y;\theta) = \frac{f(y,z;\theta)}{f(y;\theta)} \tag{35}$$

dargestellt, womit man (34) als

$$l(\theta,y) = \log f(y;\theta) = \log f(y,z;\theta) - \log f(z|y;\theta) \tag{36}$$

schreiben kann. Die marginale Log-Likelihood Funktion, die durch $\hat{\theta}$ maximiert werden soll, ist somit die Differenz zweier Teile, die beide vom nicht-beobachteten Teil der Daten z abhängen. Wir definieren an dieser Stelle für alle Paare von Parametervektoren $(\theta,\theta^{(0)})$ die Funktion

$$Q(\theta|\theta^{(0)}) := \mathbb{E}\big(\log f(Y,Z;\theta)\big|Y=y;\theta^{(0)}\big), \tag{37}$$

der im EM-Algorithmus eine entscheidende Rolle zukommt:

Definition 3.1 (Iterationsschritt des EM-Algorithmus): Sei durch (Y,Z) eine Trennung der Daten in den beobachteten Teil Y und den nicht-beobachteten Teil Z gegeben und die Funktion $Q(\theta|\theta^{(0)})$ wie in (37) definiert. Dann ist der Iterationsschritt des EM-Algorithmus zur Berechnung von $\theta^{(k+1)}$, gegeben $\theta^{(k)}$, definiert durch:

- Berechne $Q(\theta|\theta^{(k)})$ (E-Schritt).
- Bestimme $\theta^{(k+1)}$ aus dem Parameterraum so, dass $Q(\theta|\theta^{(k)})$ maximal wird für $\theta = \theta^{(k+1)}$ (M-Schritt).

Da z nicht beobachtet wurde, besteht die zugrundeliegende Idee darin, anstatt die gemeinsame Log-Likelihood Funktion, $\log f(y,z;\theta)$, zu maximieren, jenen Wert von θ zu bestimmen, der ihre konditionale Erwartung bezüglich eines aktuellen Parameterschätzers $\theta^{(k)}$, gegeben den beobachteten Teil der Daten y, maximiert.

Der Name *EM-Algorithmus (Expectation Maximization Algorithm)* kommt

daher, dass jede Iteration aus der Berechnung eines **E**rwartungswerts und der **M**aximierung desselben besteht. Es bleibt zu überprüfen, ob dieses Verfahren zu einem Schätzer $\hat{\theta}$ führt, der die marginale Log-Likelihood $l(\theta, y)$ maximiert. Mit der zu (37) analogen Notation

$$H(\theta|\theta^{(0)}) := \mathbb{E}\big(\log f(Z|Y;\theta)\big|Y = y;\theta^{(0)}\big) \tag{38}$$

folgt aus (36)

$$\begin{aligned} Q(\theta|\theta^{(0)}) - H(\theta|\theta^{(0)}) &= \mathbb{E}\big(\log f(Y,Z;\theta)\big|Y = y;\theta^{(0)}\big) \\ &\quad - \mathbb{E}\big(\log f(Z|Y;\theta)\big|Y = y;\theta^{(0)}\big) \\ &= \mathbb{E}\big(l(\theta,Y)|Y = y;\theta^{(0)}\big) \\ &= l(\theta, y). \end{aligned} \tag{39}$$

Das heißt, dass $\hat{\theta}$, der ML-Schätzer zu $l(\theta, y)$, auch $Q(\theta|\theta^{(0)}) - H(\theta|\theta^{(0)})$ maximiert. Sei θ' jener Wert von θ, der im M-Schritt ermittelt wird, um die Funktion $Q(\theta|\theta^{(k)})$ zu maximieren. Dann gilt in jedem Fall

$$Q(\theta'|\theta^{(k)}) - Q(\theta^{(k)}|\theta^{(k)}) \geq 0 \tag{40}$$

und somit für die Differenz in der marginalen Log-Likelihood Funktion

$$\begin{aligned} l(\theta', y) - l(\theta^{(k)}, y) &= \Big(Q(\theta'|\theta^{(k)}) - Q(\theta^{(k)}|\theta^{(k)})\Big) - \Big(H(\theta'|\theta^{(k)}) - H(\theta^{(k)}|\theta^{(k)})\Big) \\ &\geq H(\theta^{(k)}|\theta^{(k)}) - H(\theta'|\theta^{(k)}). \end{aligned} \tag{41}$$

Lemma 3.2: Für alle Paare $(\theta, \theta^{(0)})$ aus dem Parameterraum gilt

$$H(\theta|\theta^{(0)}) \leq H(\theta^{(0)}|\theta^{(0)})$$

Beweis: Eine Folgerung aus der Jensen-Ungleichung liefert für jede konkave Funktion $h(\cdot)$:

$$\mathbb{E}(h(X)) \leq h(\mathbb{E}(X)). \tag{42}$$

Gemäß der Definition von $H(\cdot|\cdot)$ in (38) folgt für beliebige $\theta', \theta^{(k)}$

$$\begin{aligned} H(\theta'|\theta^{(k)}) - H(\theta^{(k)}|\theta^{(k)}) &= \mathbb{E}\big(\log f(Z|Y;\theta^{(k)})\big|Y = y;\theta^{(k)}\big) \\ &\quad - \mathbb{E}\big(\log f(Z|Y;\theta')\big|Y = y;\theta^{(k)}\big) \\ &= \mathbb{E}\left(\log \frac{f(Z|Y;\theta')}{f(Z|Y;\theta^{(k)})}\middle| Y = y;\theta^{(k)}\right) \\ &\overset{(42)}{\leq} \log \mathbb{E}\left(\frac{f(Z|Y;\theta')}{f(Z|Y;\theta^{(k)})}\middle| Y = y;\theta^{(k)}\right) \\ &= \log \int \frac{f(z|y;\theta')}{f(z|y;\theta^{(k)})} f(z|y;\theta^{(k)})\, dz \\ &= \log \int f(z|y;\theta')\, dz = \log 1 = 0. \end{aligned}$$

□

Damit folgt für (41)

$$l(\theta', y) - l(\theta^{(k)}, y) \geq 0.$$

Dies bedeutet, dass die Folge der im EM-Algorithmus berechneten Parametervektoren $\theta^{(1)}, \theta^{(2)}, \theta^{(3)}, \ldots$ ausgehend von einem Startwert $\theta^{(0)}$ eine monoton wachsende Folge von Werten $l(\theta^{(0)}, y) \leq l(\theta^{(1)}, y) \leq l(\theta^{(2)}, y), \ldots$ erzeugt. Wenn $l(\theta, y)$ nach oben beschränkt ist, führt das zu Konvergenz gegen einen Wert θ^*. Ist die Funktion $Q(\theta|\theta^{(k)})$ stetig in θ und $\theta^{(k)}$, dann ist θ^* ein stationärer Punkt der marginalen Likelihood Funktion, doch ob es sich dabei um den ML-Schätzer $\hat{\theta}$ handelt, kann von der Wahl der Startwerte abhängen.
Die Konvergenzrate des EM-Algorithmus ist linear und wird davon beeinflusst, wie groß der Anteil der unbeobachteten Datenmenge ist. Der Iterationsschritt wird so lange wiederholt, bis eine zuvor festgelegte Bedingung erfüllt ist. Naheliegend ist dabei das *Lack of Progress* Kriterium, welches erfüllt ist, wenn

$$l(\theta^{(k+1)}, y) - l(\theta^{(k)}, y) < \varepsilon$$

gilt für ε hinreichend klein. Ändert sich der Wert der Log-Likelihood Funktion nur mehr geringfügig, muss dies aber nicht bedeuten, dass der Algorithmus nahe am Ziel ist (Lindsay, 1995).
Ein alternativer Zugang zur Herleitung einer Abbruchbedingung für den EM-Algorithmus, basiert auf der *Aitken Acceleration*, einer auf alle linear konvergenten Prozesse anwendbaren Methode. Dabei macht man sich die lineare Konvergenzrate zunutze, um ausgehend von den Werten der Log-Likelihood Funktion zu den Zeitpunkten k, $k-1$ und $k-2$ den Wert von $l^{(\infty)} = l(\theta^*, y)$ vorherzusagen. Für $l^{(k)} := l(\theta^{(k)}, y)$ gilt unter Annahme einer linearen Konvergenzrate

$$l^{(k+1)} - l^{(k)} \approx c(l^{(k)} - l^{(k-1)}) \text{ für } k \geq 1$$

für eine Konstante c mit $0 < c < 1$ und damit

$$l^{(k+1)} - l^{(k)} \approx c^k(l^{(1)} - l^{(0)}) \text{ für } k \geq 1.$$

Für den Grenzwert folgt damit

$$l^{(\infty)} = \lim_{k\to\infty} l^{(k)} \approx l^{(0)} + \sum_{i=0}^{\infty} c^i(l^{(1)} - l^{(0)}) = l^{(0)} + \frac{1}{1-c}(l^{(1)} - l^{(0)}).$$

Für die Schätzung von $l^{(\infty)}$ nach $k \geq 2$ Iterationen wird der Wert c durch den Quotient der letzten beiden Differenzen geschätzt, also

$$c^{(k)} = \frac{l^{(k)} - l^{(k-1)}}{l^{(k-1)} - l^{(k-2)}},$$

was mit dem Prinzip der Aitken Acceleration die Vorhersage von

$$l_k^{(\infty)} = l^{(k-1)} + \frac{l^{(k)} - l^{(k-1)}}{1 - c^{(k)}}$$

ermöglicht. Wir sehen, dass $l_k^{(\infty)} \geq l^{(k)}$ wegen $0 < c < 1$ hält, wobei in Wirklichkeit $l_k^{(\infty)}$ viel größer als $l^{(k)}$ sein kann, nämlich dann, wenn c nahe bei 1 ist, was im Zusammenhang mit sehr langsamer Konvergenz steht. Die Bedingung, die erfüllt sein soll, damit der Algorithmus gestoppt wird, kann durch

$$|l_{k+1}^{(\infty)} - l_k^{(\infty)}| < \varepsilon$$

definiert werden (Böhning, Dietz, Schaub, Schlattmann und Lindsay, 1994). Mit

$$|l_k^{(\infty)} - l^{(k)}| < \varepsilon$$

ist ein ähnliches Kriterium gegeben, das den Algorithmus erst terminieren lässt, wenn dieser hinreichend nahe am Ziel ist. Vorausgesetzt, $l(\theta^*, y)$ wird durch $l_k^{(\infty)}$ gut geschätzt, ist dabei ε aussagekräftiger bezgl. numerischer Adäquatheit des Schätzers als im Lack of Progress Kriterium (vgl. Lindsay, 1995, Kap. 3.4).

Das Prinzip der Aitken Acceleration lässt sich auch auf den Parametervektor θ anwenden, um die Konvergenz des Algorithmus zu beschleunigen (siehe Louis, 1982), wird aber mit zunehmender Anzahl der Parameter schwieriger zu implementieren. Die eventuelle Verringerung des Rechenaufwands, die man damit erwirkt, lohnt sich außerdem nur in hinreichend naher Umgebung des MLE $\hat{\theta}$.

Eine detaillierte Betrachtung der Eigenschaften des EM-Algorithmus hinsichtlich der Konvergenz liefert Wu. Er weist außerdem bereits (1983) darauf hin, dass der EM-Algorithmus trotz vergleichsweise langsamer Konvergenz zu einem populären Instrument in der Statistik wurde. Im Gegensatz zu vielen anderen Optimierungsproblemen ist die Implementierung für zahlreiche Anwendungen einfach zu realisieren und der M-Schritt durch bestehende Pakete erfolgreich durchzuführen.

3.1.2 Standardfehler

Ein wesentliches Merkmal des EM-Algorithmus ist, dass er ML-Schätzer liefert, nicht aber Information über ihre Standardfehler. Die Art und Weise, wie der Algorithmus arbeitet, bringt nämlich keine automatische Schätzung

dafür mit sich. Die direkteste Methode, Standardfehler zu gewinnen, ist mittels Berechnung der beobachteten Informationsmatrix $I(\hat{\theta}, y)$. Louis (1982) leitet dazu die Formel

$$\begin{aligned} I(\hat{\theta}, y) = \; & \mathbb{E}\big(B_c(Y, Z; \theta) \,\big|\, Y = y; \theta = \hat{\theta}\big) \\ & - \mathbb{E}\big(S_c(Y, Z; \theta) S_c^t(Y, Z; \theta) \,\big|\, Y = y; \theta = \hat{\theta}\big) \end{aligned} \tag{43}$$

her. Dabei ist $S_c(y, z; \theta)$ der Gradient der Log-Likelihood Funktion von (y, z) und $B_c(y, z; \theta)$ ist die dazugehörende Hessematrix der zweiten Ableitungen. Nachdem $\hat{\theta}$ vom EM-Algorithmus gefunden wird, lässt sich die beobachtete Informationsmatrix durch Auswertung von (43) berechnen (Watanabe und Yamaguchi, 2003).
Bei Problemen unvollständiger Daten wird der M-Schritt in vielen Software-Paketen durch Funktionen ausgeführt, deren Arbeitsweise auf die komplette Verfügbarkeit der Daten gestützt ist. Werden dann die Standardfehler der Parameter zur Informationsmatrix der vollständigen Daten ausgegeben, sind diese immer unterschätzt, da ihre Berechnung auf der Annahme basiert, dass es sich bei der konditionalen Erwartung von z um beobachtete Daten handelt (vgl. Aitkin, Francis, Hinde und Darnell, 2009).
Diese Arbeit wird sich darauf beschränken, auf diese Tatsache bei einigen Beispielen hinzuweisen. Je nach Problem lässt sich die Schätzung eines Standardfehlers auch anders bewerkstelligen. Allgemein kann man mit der von Louis (1982) vorgestellten Technik durch den EM-Algorithmus unterschätzte Standardfehler korrigieren.

3.1.3 Varianten des EM-Algorithmus

Der EM-Algorithmus kann als Spezialfall des *GEM-Algorithmus* (Generalized Expectation Maximization Algorithm) gesehen werden. Dieser hat die Eigenschaft, dass in jeder Iteration ein neuer Schätzer $\theta^{(k+1)}$ bestimmt wird, der $Q(\theta|\theta^{(k)})$ nicht notwendigerweise maximiert, sondern nur $Q(\theta^{(k+1)}|\theta^{(k)}) \geq Q(\theta^{(k)}|\theta^{(k)})$ sicher stellt (Dempster et al., 1977). Dies kann eine Vereinfachung in der Implementierung bewirken und numerische Vorteile bringen, und die Monotonität von $(l(\theta^{(k)}, y))_{k\geq 0}$ ist immer noch gewährleistet.
Bei der Lösung spezieller Probleme wurde immer wieder der Versuch unternommen, den EM-Algorithmus hinsichtlich Effizienz zu verbessern oder in anderer Form an die Besonderheiten des Problems anzupassen. Es entstand so eine Vielzahl an Modifikationen des EM-Algorithmus, die zum Teil unter eigenen Bezeichnungen bekannt sind. Einen Überblick findet man bei Roche (2003), der bei den so gewonnenen Algorithmen zwischen zwei Arten unterscheidet:

- deterministische EM Varianten,
- stochastische EM Varianten.

Die deterministischen Varianten zeichnen sich im Allgemeinen dadurch aus, dass sie die Effizienz des EM-Algorithmus erhöhen, indem sie Rechenaufwand durch Modifikation der Berechnungen in E-Schritt und M-Schritt reduzieren, oder dass sie die Konvergenz beschleunigen. Dagegen haben die stochastischen Varianten meist als Ziel, Hindernisse, auf die der gewöhnliche EM-Algorithmus stößt, zu umgehen. So kann die Bestimmung der Funktion $Q(\theta|\theta^{(k)})$ in vielen Situationen mit der Berechnung eines Erwartungswert-Integrals verbunden sein, das keine geschlossene Darstellung besitzt. Die Idee in den stochastischen Algorithmen ist dann, die Berechnung dieses Ausdrucks durch eine geeignete Monte Carlo-Simulation zu ersetzen. In Kapitel 4 werden wir dies anhand des MCEM-Algorithmus sehen. Davor werden wir uns in erster Linie mit dem EM-Algorithmus in Standardform auseinandersetzen.

Um den EM-Algorithmus anwenden zu können, ist es nötig, eine Darstellung für die Funktion $Q(\cdot|\cdot)$ zu finden. Dazu muss der nicht-beobachtete Teil der Daten durch eine passende Formulierung repräsentiert werden. Wie dies in einigen wichtigen Beispielen passieren kann, sollen die folgenden Abschnitte veranschaulichen.

3.2 Endliche diskrete Mischungen

Die in Abschnitt 2.1.1 beschriebene Exponentialfamilie zeichnet sich durch ihre vielfältigen Anwendungsmöglichkeiten in statistischen Modellen aus, kann jedoch nicht die Analyse von allen Arten gegebener Daten abdecken. In diesem Abschnitt werden wir den Begriff der *Mischverteilung* einführen, mit dem es möglich ist, diese Klasse von Verteilungen zu erweitern. Eine Mischverteilung besitzt allgemein eine marginale Dichte der Form

$$m(y;\lambda) = \int f(y|z)h(z)\,dz. \tag{44}$$

Dabei ist $f(y|z)$ die bedingte Dichte oder Wahrscheinlichkeitsfunktion, der Zufallsvariable Y, deren Realisierung y wir beobachten. Die Funktion $h(z)$ spezifiziert die Verteilung der nicht zu beobachtenden Zufallsvariable Z und damit die Art der Mischung (vgl. Aitkin et al., 2009, Kap. 7). Alle in $f(\cdot)$ und $h(\cdot)$ enthaltenen Parameter sind vorerst in λ zusammengefasst.

Die Verteilung von Z wird im Folgenden als diskret angenommen mit einer endlichen Menge von Massepunkten $z_1, \ldots, z_L$ und Wahrscheinlichkeiten

$\pi_1, \ldots, \pi_L$. Die marginale Dichte einer Zufallsvariable Y, die der resultierenden *endlichen diskreten Mischung* aus L Komponenten unterliegt, kann dann geschrieben werden als

$$m(y; \theta_1, \ldots, \theta_L, \pi_1, \ldots, \pi_{L-1}, \phi) = \sum_{g=1}^{L} \pi_g f(y; \theta_g, \phi). \tag{45}$$

Die $f(y; \theta_g, \phi) = f(y|Z = z_g)$ sind dabei Dichte- oder Wahrscheinlichkeitsfunktionen, abhängig vom für die Komponente g charakteristischen Parameter θ_g und dem globalen Parameter ϕ, den alle Komponenten gemeinsam haben.[5] Die Massen $\pi_1, \ldots, \pi_L$ sind nicht-negative Werte mit $\sum_g \pi_g = 1$, weshalb nur $L-1$ unbekannte Wahrscheinlichkeits-Parameter vorliegen. Ohne Beschränkung der Allgemeinheit können wir fordern, dass $\pi_g > 0$ für $g = 1, \ldots, L$ gelten soll und damit nur Komponenten betrachtet werden, die tatsächlich durch Elemente der Stichprobe repräsentiert werden. Die diskrete Verteilung der Zufallsvariable Z, die durch $\pi = (\pi_1, \ldots, \pi_L)$ bestimmt ist, wird auch *latente* Verteilung genannt. Könnte man Z beobachten, so wäre eine Identifikation der Komponente möglich und eine Stichprobe vom Umfang n könnte in L Gruppen unterteilt werden.

3.2.1 Direkter Zugang zur ML-Schätzung

Die Likelihood Funktion einer Stichprobe $y_1, \ldots, y_n$ aus der diskreten Mischung, wie sie im vorangegangenen Abschnitt beschrieben wurde, ist für $\theta = (\theta_1, \ldots, \theta_L)^t$ und $\pi = (\pi_1, \ldots, \pi_L)^t$ gegeben durch

$$L(\theta, \pi, \phi, y) = \prod_{i=1}^{n} m_i = \prod_{i=1}^{n} \sum_{g=1}^{L} \pi_g f_{ig}, \tag{46}$$

mit

$$m_i = m(y_i; \theta, \pi, \phi), \quad f_{ig} = f(y_i; \theta_g, \phi).$$

Mit θ, π und ϕ liegen insgesamt $L + (L-1) + 1 = 2L$ unbekannte Parameter vor, die geschätzt werden sollen. Die Log-Likelihood

$$l(\theta, \pi, \phi, y) = \sum_{i=1}^{n} \log m_i = \sum_{i=1}^{n} \log \left(\sum_{g=1}^{L} \pi_g f_{ig} \right)$$

[5] Bei Mitgliedern der Exponentialfamilie wird ϕ meist mit dem Dispersionsparameter assoziiert. Wir werden in diesem Kapitel aber auch Mischungen mit unterschiedlichen Dispersionsparametern betrachten.

liefert für die Ableitungen

$$\frac{\partial l}{\partial \theta_g} = \sum_{i=1}^{n} \frac{\pi_g}{m_i} \frac{\partial f_{ig}}{\partial \theta_g} = \sum_{i=1}^{n} \frac{\pi_g f_{ig}}{m_i} \frac{\partial \log f_{ig}}{\partial \theta_g}, \quad g = 1, \ldots, L.$$

Mit

$$\omega_{ig} := \frac{\pi_g f_{ig}}{m_i} \tag{47}$$

erhalten wir

$$\frac{\partial l}{\partial \theta_g} = \sum_{i=1}^{n} \omega_{ig} \frac{\partial \log f_{ig}}{\partial \theta_g}, \tag{48}$$

was als gewichtete Summe von Score-Funktionen aufgefasst werden kann. Den Gewichten ω_{ig} kommt dabei eine besondere Bedeutung zu. Sei Z_i die Zufallsvariable, die die Information der Gruppenzugehörigkeit von y_i beinhaltet. Dann liefert Anwendung der Bayes-Theorie

$$\omega_{ig} = \frac{\pi_g f_{ig}}{m_i} = \frac{\pi_g f_{ig}}{\sum_{l=1}^{L} \pi_l f_{il}} = P(Z_i = z_g | y_i),$$

womit sie gerade die a posteriori Wahrscheinlichkeiten, dass eine Beobachtung y_i aus Gruppe g stammt, beschreiben. Diese Interpretation wird in der weiteren Betrachtung sehr nützlich sein.

Für die Ableitung der Log-Likelihood nach dem gemeinsamen Parameter ϕ gilt

$$\frac{\partial l}{\partial \phi} = \sum_{i=1}^{n} \frac{1}{m_i} \sum_{g=1}^{L} \pi_g \frac{\partial f_{ig}}{\partial \phi} = \sum_{i=1}^{n} \sum_{g=1}^{L} \omega_{ig} \frac{\partial \log f_{ig}}{\partial \phi}. \tag{49}$$

Für die Differentiation nach den Mischwahrscheinlichkeiten π_g verwendet man wegen der Restriktion $\sum_g \pi_g = 1$ den Lagrange-Multiplikator λ, mit dem für $l^* = l - \lambda(\sum_g \pi_g - 1)$

$$\frac{\partial l^*}{\partial \pi_g} = \sum_{i=1}^{n} \frac{f_{ig}}{m_i} - \lambda = \sum_{i=1}^{n} \frac{\omega_{ig}}{\pi_g} - \lambda$$

folgt. Die Nullstelle dieser Score-Gleichung ist durch

$$\hat{\pi}_g = \frac{1}{\lambda} \sum_{i=1}^{n} \omega_{ig}$$

gegeben, was nach Multiplikation mit λ und Summation über g

$$\hat{\lambda} = \sum_{g=1}^{L} \sum_{i=1}^{n} \omega_{ig} = \sum_{i=1}^{n} \sum_{g=1}^{L} \omega_{ig} = n$$

liefert, also

$$\hat{\pi}_g = \frac{1}{n} \sum_{i=1}^{n} \omega_{ig}. \tag{50}$$

Die ML-Schätzer für θ, ϕ und π_g sind also als die Nullstellen von (48), (49) sowie durch (50) definiert. Die in allen drei Gleichungen auftretenden ω_{ig} sind dabei aber unbekannt und hängen selbst von θ, ϕ und π ab, was die direkte Lösung im Allgemeinen nicht ermöglicht. Um die Likelihood Funktion iterativ mittels EM-Algorithmus maximieren zu können, soll das Problem alternativ formuliert werden.

3.2.2 Diskrete Mischungen als Problem unvollständiger Daten

Wird eine Stichprobe $y = (y_1, \ldots, y_n)^t$ der Mischverteilung mit L Komponenten zugeordnet, dann geschieht dies als Folge davon, dass jede Beobachtung y_i von einer der Komponenten stammt, die Zugehörigkeit jedoch nicht beobachtet werden kann. Wie bereits erwähnt wurde, kann diese Information mit einer Zufallsvariable Z assoziiert werden. Die vollständigen, nichtbeobachteten Daten sind dann durch Paare (y_i, z_i) gegeben. Für die weitere Vorgehensweise können wir annehmen, dass jedes z_i Indikatorvariable für y_i ist und die Werte $1, \ldots, L$ annehmen kann. Es gilt

$$Z_i = g \Leftrightarrow f(y_i) = f(y_i; \theta_g, \phi).$$

Zusätzlich seien zur i-ten Beobachtung die Zufallsvariablen $Z_{ig}, g = 1, \ldots, L$, definiert durch

$$Z_{ig} = \begin{cases} 1 & \text{wenn } y_i \text{ aus Komponente } l \text{ stammt} \Leftrightarrow Z_i = g, \\ 0 & \text{sonst.} \end{cases}$$

Fasst man diese zu L-dimensionalen Indikatorvektoren $G_i = (Z_{i1}, \ldots, Z_{iL})^t$ zusammen, erhält man multinomialverteilte Vektoren $G_i \sim M(1, \pi)$ (vgl. McLachlan und Peel, 2000, Kap. 1.4, 2.8). Die gemeinsame Dichte der Stichprobe hat damit die Form

$$f(y, z; \theta, \pi, \phi) = \prod_{i=1}^{n} \prod_{g=1}^{L} (\pi_g f_{ig})^{z_{ig}},$$

also gilt

$$\log f(y, z; \theta, \pi, \phi) = \sum_{i=1}^{n} \sum_{g=1}^{L} z_{ig} (\log \pi_g + \log f_{ig}).$$

Für feste Werte $\theta^{(k)}, \pi^{(k)}, \phi^{(k)}$ wird im E-Schritt des EM-Algorithmus die Funktion

$$Q(\theta, \pi, \phi | \theta^{(k)}, \pi^{(k)}, \phi^{(k)}) = \mathbb{E}\big(\log f(Y, Z; \theta, \pi, \phi) \big| Y = y; \theta^{(k)}, \pi^{(k)}, \phi^{(k)}\big)$$

$$= \mathbb{E}\left(\sum_{i=1}^{n} \sum_{g=1}^{L} Z_{ig} \big(\log \pi_g + \log f_{ig}\big) \middle| Y = y; \theta^{(k)}, \pi^{(k)}, \phi^{(k)} \right)$$

berechnet, wobei nur die Z_{ig} unbekannte Größen darstellen. Dabei handelt es sich um binäre Zufallsvariablen, für die die konditionalen Erwartungswerte durch

$$\mathbb{E}(Z_{ig} | Y_i = y_i; \theta^{(k)}, \pi^{(k)}, \phi^{(k)}) = P(z_{ig} = 1 | Y_i = y_i; \theta^{(k)}, \pi^{(k)}, \phi^{(k)}) = \omega_{ig}^{(k)}$$

gegeben sind, und damit den a posteriori Wahrscheinlichkeiten ω_{ig} in (47) entsprechen, die nun aber in $\theta^{(k)}, \pi^{(k)}, \phi^{(k)}$ ausgewertet sind. Es resultiert daher

$$\begin{aligned} Q(\theta, \pi, \phi | \theta^{(k)}, \pi^{(k)}, \phi^{(k)}) &= \sum_{i=1}^{n} \sum_{g=1}^{L} \omega_{ig}^{(k)} \Big(\log \pi_g + \log f_{ig} \Big) \\ &= \sum_{i=1}^{n} \sum_{g=1}^{L} \omega_{ig}^{(k)} \Big(\log \pi_g + \log f(y_i; \theta_g, \phi) \Big). \end{aligned} \tag{51}$$

Der M-Schritt besteht nun aus der Maximierung von (51). Die Ableitungen

$$\frac{\partial Q}{\partial \theta_g}, \quad \frac{\partial Q}{\partial \phi}, \quad \frac{\partial Q^*}{\partial \pi_g}$$

mit $Q^* = Q - \lambda(\sum_g \pi_g - 1)$ liefern gerade die Gleichungen (48), (49) und (50), wobei die ω_{ig} nun aber bekannt sind. Die neuen Parameterschätzer $\theta^{(k+1)}, \pi^{(k+1)}, \phi^{(k+1)}$ sind daher definiert als die Nullstellen von (48), (49) bzw. die Lösung von (50) mit $\omega_{ig} = \omega_{ig}^{(k)}$. Insbesondere erhält man für die Wahrscheinlichkeitsmasse der Komponente g

$$\pi_g^{(k+1)} = \frac{1}{n} \sum_{i=1}^{n} \omega_{ig}^{(k)},$$

was dem arithmetischen Mittel der aktuellen a posteriori Wahrscheinlichkeiten für die Zugehörigkeit zu Komponente g entspricht. Wir bemerken außerdem, dass die $\pi^{(k+1)}$ zwar von $\omega_{ig}^{(k)}$ und somit den Schätzern $\pi^{(k)}, \theta^{(k)}$ und $\phi^{(k)}$ abhängen, ihre Berechnung jedoch ohne Kenntnis der neuen Schätzer $\theta^{(k+1)}$ und $\phi^{(k+1)}$ durchgeführt werden kann.

Eine typische Eigenschaft von diskreten Mischungen ist die Multimodalität der marginalen Log-Likelihood Funktion. Finch, Mendell und Thode (1989) diskutieren Strategien zur Verwendung verschiedener Startwerte, um die folglich auftretenden verschiedenen Maxima zu lokalisieren. Eine generelle Diskussion zur Wahl der Startwerte für den EM-Algorithmus findet man in McLachlan (1988).

Die in diesem Abschnitt gewonnenen Resultate zur Parameterschätzung bei endlichen diskreten Mischungen stützen sich nicht auf die Annahme, dass die $f(y; \theta_g, \phi)$ Dichte- oder Wahrscheinlichkeitsfunktionen aus der Exponentialfamilie sind, sondern haben allgemein Gültigkeit. Sogar die Einschränkung, dass sie von derselben Form sind, ist dabei nicht nötig. Diese Arbeit wird sich aber darauf beschränken, Mischungen von zur Exponentialfamilie gehörenden Verteilungen mit derselben Form zu behandeln.

Beispiel: Mischung von Normalverteilungen

Stammt $y = (y_1, \ldots, y_n)^t$ aus der Mischung von L Normalverteilungen mit unterschiedlichen Erwartungswerten $\mu_g = \theta_g$ und gemeinsamer Varianz $\sigma^2 = \phi$, dann gilt

$$f_{ig} = f(y_i; \mu_g, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \mu_g)^2}{2\sigma^2}\right),$$

und man erhält

$$\frac{\partial \log f_{ig}}{\partial \mu_g} = \frac{y_i - \mu_g}{\sigma^2}, \qquad \frac{\partial \log f_{ig}}{\partial \sigma^2} = -\frac{1}{2\sigma^2} + \frac{(y_i - \mu_g)^2}{2\sigma^4}.$$

Es folgt daher für die Score-Gleichungen (48) und (49)

$$\begin{aligned}
\frac{\partial l}{\partial \mu_g} &= \sum_{i=1}^{n} \omega_{ig} \frac{y_i - \mu_g}{\sigma^2}, \\
\frac{\partial l}{\partial \sigma^2} &= \sum_{i=1}^{n} \sum_{g=1}^{L} \omega_{ig} \left(-\frac{1}{2\sigma^2} + \frac{(y_i - \mu_g)^2}{2\sigma^4}\right).
\end{aligned}$$

Der EM-Algorithmus ist in diesem Fall von sehr einfacher Form. Im E-Schritt werden für feste Werte von μ_g, π_g und σ^2 die Gewichte

$$\omega_{ig} = \frac{\pi_g f_{ig}}{\sum_{l=1}^{L} \pi_l f_{il}}$$

berechnet. Mit diesen ω_{ig} bestimmt man im M-Schritt die Lösungen der Score-Gleichungen, also

$$\hat{\mu}_g = \sum_{i=1}^{n} \omega_{ig} y_i \Big/ \sum_{i=1}^{n} \omega_{ig},$$
$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} \sum_{g=1}^{L} \omega_{ig} (y_i - \mu_g)^2,$$

sowie

$$\hat{\pi}_g = \frac{1}{n} \sum_{i=1}^{n} \omega_{ig},$$

um mit den neuen Schätzern in den E-Schritt zurückzukehren und die Iteration zu wiederholen.

3.2.3 Testen auf die Anzahl der Komponenten

Bisher haben wir angenommen, dass es sich bei der Anzahl der Komponenten in der Mischverteilung um eine bekannte Größe L handelt. Ein einfaches Beispiel dafür ist die Messung einer normalverteilten Zufallsvariable Y in einer Population von Lebewesen, wobei die Information der kategorialen Variable Geschlecht mit den Stufen *weiblich* und *männlich* verworfen oder nicht beobachtet wird (vgl. Lindsay, 1995, Kap. 1.1). Man weiß dann aber, dass zwei Stufen vorliegen, für die $L = 2$ unterschiedliche Komponenten angenommen werden, mit unterschiedlichen Erwartungswerten μ_1 und μ_2 aber der Einfachheit halber mit derselben Varianz σ^2. Ohne Information über Geschlecht erhält man dadurch die Stichprobe $y_1, \ldots, y_n$ einer Mischverteilung mit

$$Y_i \overset{iid}{\sim} \pi_1 N(\mu_1, \sigma^2) + (1 - \pi_1) N(\mu_2, \sigma^2).$$

Mit gleich großen Gewichtungen $\pi_1 = 1 - \pi_1 = 1/2$ und einer Differenz der Erwartungswerte von vier Standardabweichungen weist der Graph der entsprechenden Dichtefunktion eine Bimodalität auf, vergleichbar mit der Dichte in Abbildung 1. Diese zeigt die gemischte Dichte für $\mu_1 = -2, \mu_2 = 2, \sigma_1^2 = \sigma_2^2 = 1$ und gleichgroßen Mischwahrscheinlichkeiten. Trifft man die Annahme, dass der größere Erwartungswert zur Population der männlichen

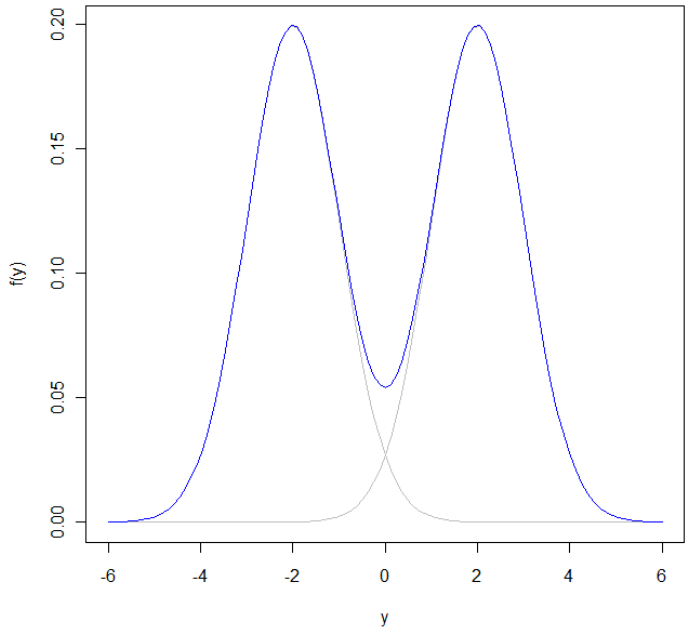

Abbildung 1: Mischung zweier Normalverteilungen mit $|\mu_1 - \mu_2| = 4\sigma$.

Lebewesen gehört, dann kann eine Beobachtung y_i gleichzeitig einen Hinweis auf das nicht-beobachtete Geschlecht liefern. Für eine Klassifizierung ist dies also eine wünschenswerte Situation. Im Vergleich dazu führt eine Mischung, in der sich die gruppenspezifischen Mittelwerte nur um die doppelte Standardabweichung unterscheiden zu einer unimodalen Funktionskurve wie in Abbildung 2. In diesem Fall sind die Komponenten nicht mehr so gut unterscheidbar und die Einteilung in Gruppen wird sich schwieriger gestalten.

Allgemein muss aber auch bei gut identifizierbaren Teilen der marginalen Dichte nicht klar ersichtlich sein, wie viele Komponenten gebraucht werden, um eine adäquate Modellanpassung an vorliegende Daten zu bekommen. Ein naheliegender Ansatz ist, L so lange zu erhöhen, bis der Wert der Log-Likelihood Funktion $l(\theta, \pi, \phi, y)$ sich nicht mehr relevant ändert. Diese Vorgehensweise setzt voraus, dass die LRT-Statistik bekannte, für die Bewertung relevante Eigenschaften aufweist. Die Regularitätsbedingungen, unter denen dies zutrifft, sind für diese Situation jedoch nicht gegeben. Aitkin et al. (2009) führen in diesem Zusammenhang die folgenden Punkte an. Zum einen handelt es sich beim Modell mit $L - 1$ Komponenten nicht um einen Spezi-

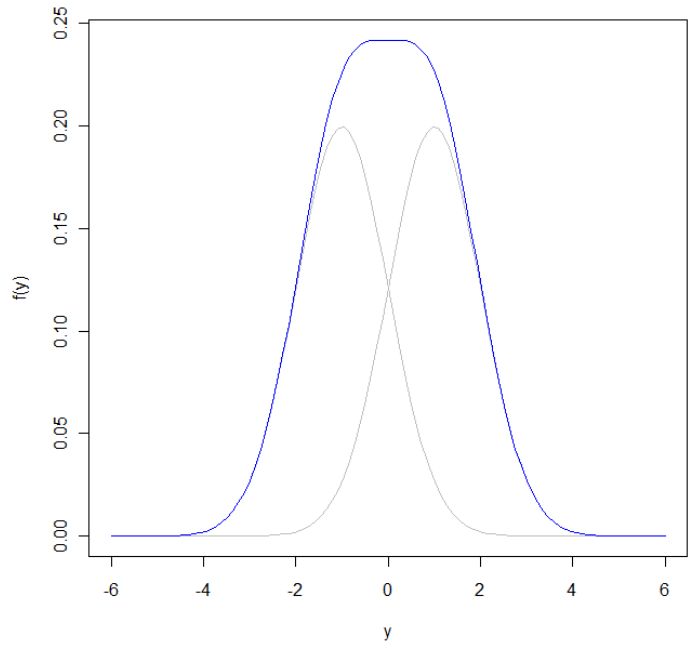

Abbildung 2: Mischung zweier Normalverteilungen mit $|\mu_1 - \mu_2| = 2\sigma$.

alfall des Modells mit L Komponenten. Reduziert man nämlich die Anzahl der Komponenten durch Hinzufügen der Restriktion $\theta_L = \theta_{L-1}$, dann erhält man zwei nicht unterscheidbare Komponenten und die Parameter π_L und π_{L-1} sind nicht mehr getrennt voneinander identifizierbar. Umgekehrt führt die Wahl von $\pi_L = 0$ dazu, dass mit θ_L ebenfalls einer der verbleibenden Parameter nicht mehr identifizierbar ist. Eine zweite Besonderheit besteht darin, dass $l(\theta, \pi, \phi, y)$ bei Mischungen von Verteilungen mit nur einem unbekannten Parameter nicht beliebig groß wird mit wachsendem L, sondern sich irgendwann stabilisiert.
Auch die Interpretation von L als Variable, die dem nicht-beobachteten Teil der Daten zugeschrieben wird, ist im Sinn des EM-Algorithmus nicht möglich, da der Wert von L die Anzahl der zu schätzenden Parameter beeinflusst.
Dies liefert die Motivation, sich einer *Bootstrap*-Methode zu bedienen, wie bei Aitkin et al. (2009, S. 442) beschrieben: Gegeben eine Stichprobe $y = (y_1, \ldots, y_n)^t$, wird zum Test von

$$\begin{aligned} H_0 &: \text{ Mischung besteht aus } L \text{ Komponenten} \\ H_A &: \text{ Mischung besteht aus } L+1 \text{ Komponenten} \end{aligned}$$

eine neue Stichprobe vom Umfang N aus dem gefitteten Modell mit L Komponenten generiert. Zu dieser Stichprobe wird eine Modellschätzung für L und eine für $L+1$ Komponenten vorgenommen. Nun berechnet man für die simulierte Stichprobe die LRT-Statistik λ zu H_0 und vergleicht sie mit λ_y, der LRT-Statistik der beobachteten Stichprobe y. Diesen Vorgang, bestehend aus Simulation und Modellanpassungen, wiederholt man R Mal, um die LRT-Statistik von y mit R Werten vergleichen zu können.
Ist die Nullhypothese wahr, dann wird die Wahrscheinlichkeit, dass λ_y größer als jede der LRT-Statistiken der simulierten Werte ist, ungefähr $1/(R+1)$ betragen. Man benötigt für einen Test vom Level $\alpha = 5\%$ daher zumindest 19 Wiederholungen der Simulation und verwirft H_0, wenn λ_y größer ist als alle simulierten Werte.
Der Umfang des Tests ist nicht als exakt zu werten, da man den ML-Schätzer der Parameter, gegeben y, als wahren Parameter in der Simulation verwendet. Genauere Ausführungen zu diesem Thema liefert McLachlan (1987).
Ob des geringeren Aufwands bevorzugt man in der Praxis jedoch häufig den einfachen Vergleich der maximalen Werte der Likelihood Funktion bei Modellanpassungen mit L und $L+1$ Komponenten, in dem Bewusstsein, dass die Aussagekraft als eingeschränkt anzusehen ist.

3.2.4 Diskrete Mischungen Generalisierter Linearer Modelle

Diskrete Mischungen können in ihrer einfachsten Form als Instrument der Datengruppierung gesehen werden. Das zugrunde liegende Modell ist dabei durch eine Konvexkombination von Verteilungen definiert, von denen jede für eine Komponente steht. Durch das Einbauen unterschiedlichster Modellannahmen in diese einzelnen Komponenten können nach und nach kompliziertere Mischungen entstehen. Eine Möglichkeit, Mischungen auf diese Art zu erweitern, ist für jede Komponente ein GLM anzupassen. In Kapitel 2 wurde die Schätzung der Regressionskoeffizienten β des GLM behandelt. Diese beschreiben den Zusammenhang von erklärenden Variablen $x = (x_1, \ldots, x_p)^t$ und dem Erwartungswert der Response Y. Die Schätzung eines einzigen Vektors von Regressionskoeffizienten kann für Zufallsvariablen $Y_1, \ldots, Y_n$ jedoch unpassend sein, wenn die Beobachtungen aus einer Anzahl nicht bekannter Klassen hervorgehen, in denen der Einfluss der erklärenden Variablen ein unterschiedlicher ist.
Ein ähnlich motivierter Zugang ist die Spezifikation von *Random Coefficient Models*. Dabei wird angenommen, dass die Regressionskoeffizienten der Beobachtungen einer bestimmten Verteilung folgen und auf diese Weise eine Heterogenität in der Population erzeugen. Oft wird dafür eine stetige Verteilung angenommen, die aber zur Vereinfachung durch eine endliche Zahl an

Massepunkten mit entsprechenden Wahrscheinlichkeitsmassen approximiert wird. Dies führt das Problem dann wiederum auf jenes einer endlichen diskreten Mischung zurück (vgl. Hagenaars und McCutcheon, 2002).
Insgesamt ist also die Motivation gegeben, die Ideen des GLM in diskrete Mischungen einzubauen, mit dem wesentlichen Aspekt, dass bei Mischungen von GLMs nicht alle Beobachtungen denselben Regressionskoeffizienten unterliegen müssen. Um dies zu veranschaulichen, betrachten wir die Dichte einer Mischung von L Komponenten

$$m(y;\lambda,\pi) = \sum_{g=1}^{L} \pi_g f_g(y;\lambda_g) \quad \text{mit} \quad \sum_{g=1}^{L} \pi_g = 1,$$

wobei nun in jeder der L Komponenten ein GLM als Basis dienen soll. Die komponentenspezifischen Parameter sind in der marginalen Verteilung zu $\lambda = (\lambda_1, \ldots, \lambda_L)$ zusammengefasst. Wir fordern gemäß des Ansatzes von Kapitel 2, dass ein Zusammenhang zwischen den Erwartungswerten $\mu_g = \mathbb{E}(Y|Z = g)$ und festen Prädiktorvariablen $x = (x_1, \ldots, x_p)^t$ besteht, der durch

$$g_g(\mu_g) = \eta_g = x^t \beta_g,$$

beschrieben wird. Die latente Variable Z ist dabei wie zuvor definiert, erklärt also die Zugehörigkeit zu einer der L Gruppen. Die Funktion $g_g(\cdot)$ ist die zur Komponente g gehörende Linkfunktion und die gruppenspezifischen Regressionskoeffizienten werden mit $\beta_g = (\beta_{g1}, \ldots, \beta_{gp})^t$ bezeichnet. Außerdem sollen die f_g Dichte- oder Wahrscheinlichkeitsfunktionen der Exponentialfamilie sein mit komponentenspezifischem kanonischen Parameter θ_g und dem Dispersionsparameter ϕ_g. Der Zusammenhang zwischen μ_g und θ_g ist durch $\mu_g = b_g'(\theta_g)$ erklärt, wobei $b_g(\cdot)$ die Kumulantenfunktion in Komponente g ist (siehe Abschnitt 2.1.1). Es lässt sich damit der Vektor der unbekannten Parameter λ als die Zusammenfassung von Regressionskoeffizienten und Dispersionsparameter schreiben mit $\lambda_g = (\beta_g, \phi_g)$.
Das Modell ist auf diese Art sehr allgemein formuliert. Wir werden von nun an annehmen, dass die Funktionen f_g denselben Verteilungstyp beschreiben, also $f_g \equiv f$, und auch nur eine globale Linkfunktion für alle Komponenten vorliegt, also $g_g \equiv g$. Diese Annahme ist durchaus sinnvoll, da in Situationen, wo es um Gruppierung von Daten gehen soll, im Allgemeinen a priori keine Information über Unterschiede in den Verteilungen vorliegt (vgl. Grün und Leisch, 2008). Es gilt dann $\mu_g = g^{-1}(x^t \beta_g)$ für feste Prädiktorvariablen x, was bedeutet, dass die Erwartungswerte in den Komponenten sich nur dann unterscheiden, wenn die Regressionskoeffizienten verschieden sind. Für die

marginale Dichte erhält man also die Darstellung

$$m(y;\lambda,\pi)=\sum_{g=1}^{L}\pi_g\exp\left(\frac{y\theta_g-b(\theta_g)}{a(\phi_g)}+c(y,\phi_g)\right).$$

Die Log-Likelihood Funktion der Stichprobe $y=(y_1,\ldots,y_n)^t$ hat wie zuvor die Form

$$l(\lambda,\pi,y)=\sum_{i=1}^{n}\log\left(\sum_{g=1}^{L}\pi_g f_{ig}\right),$$

hier mit

$$f_{ig}=\exp\left(\frac{y_i\theta_g-b(\theta_g)}{a(\phi_g)}+c(y_i,\phi_g)\right).$$

Die Anwendung des EM-Algorithmus erfordert im E-Schritt analog zu (51) die Berechnung von

$$Q(\lambda,\pi|\lambda^{(k)},\pi^{(k)})=\sum_{i=1}^{n}\sum_{g=1}^{L}\omega_{ig}^{(k)}\big(\log\pi_g+\log f_{ig}\big) \tag{52}$$

durch das Bestimmen der Gewichte $\omega_{ig}^{(k)}$ für feste Parameter $\lambda^{(k)}=(\beta^{(k)},\phi^{(k)})$. Der M-Schritt enthält wiederum die Berechnung

$$\pi_g^{(k+1)}=\frac{1}{n}\sum_{i=1}^{n}\omega_{ig}^{(k)}$$

und da die Maximierung bezüglich λ unabhängig von π durchgeführt wird, muss

$$\frac{\partial}{\partial\beta_{gj}}\sum_{i=1}^{n}\sum_{l=1}^{L}\omega_{il}^{(k)}\log f_{il}\stackrel{!}{=}0,\quad j=1,\ldots,p\,,\quad g=1,\ldots,L \tag{53}$$

gelten, wobei

$$\frac{\partial\log f_{il}}{\partial\beta_{gj}}=\begin{cases}\frac{y_i-\mu_{il}}{a_i(\phi_l)V(\mu_{il})}\frac{x_{ij}}{g'(\mu_{il})} & \text{wenn } l=g\\ 0 & \text{sonst.}\end{cases} \tag{54}$$

Es bezeichnet μ_{ig} dabei den Erwartungswert von Y_i, gegeben die Beobachtung stammt aus Komponente g. Die Lösung der Score-Gleichungen (53) ist

damit äquivalent zur gewichteten ML-Schätzung eines GLM basierend auf dem Datensatz

y	ω	β_{11}	$\dots$	β_{1p}	β_{21}	$\dots$	β_{2p}	$\dots$	β_{L1}	$\dots$	β_{Lp}
y_1	ω_{11}	x_{11}	$\dots$	x_{1p}							
$\vdots$	$\vdots$	$\vdots$		$\vdots$		0		$\dots$		0	
y_n	ω_{n1}	x_{n1}	$\dots$	x_{np}							
y_1	ω_{12}				x_{11}	$\dots$	x_{1p}				
$\vdots$	$\vdots$		0		$\vdots$		$\vdots$	$\dots$		0	
y_n	ω_{n2}				x_{n1}	$\dots$	x_{np}				
$\vdots$	$\vdots$		$\vdots$			$\vdots$		$\ddots$		$\vdots$	
y_1	ω_{1L}								x_{11}	$\dots$	x_{1p}
$\vdots$	$\vdots$		0			0		$\dots$	$\vdots$		$\vdots$
y_n	ω_{nL}								x_{n1}	$\dots$	x_{np}

Dabei wird jede Response y_i gemeinsam mit den erklärenden Variablen x_i L-fach betrachtet, um die Zugehörigkeit zu jeder Komponente ins Modell einbeziehen zu können. Die einzige Spalte, die sich im Lauf der Iterationen verändert, ist jene der Gewichte ω_{ig}, deren Werte im jeweiligen E-Schritt zuvor aktualisiert werden. Die Parameterschätzung zerfällt dabei unter der Annahme komponentenspezifischer Dispersionsparameter in jedem M-Schritt in L Teilprobleme, da für die Maximierung bezüglich (β_g, ϕ_g) nur der Teil der Log-Likelihood in Komponente g relevant ist, wie man anhand von (54) sieht. Ein ähnlicher Algorithmus ist in der `R`-Funktion `alldist` implementiert, wo per Default-Einstellung die Existenz eines globalen Dispersionsparameters $\phi = \phi_1 = \ldots = \phi_L$ und $a_i(\phi) = a_i\phi$ angenommen wird, was mittels Spezifikation das Arguments λ korrigiert werden kann (Aitkin et al., 2009, S. 452). Die Aktualisierung des Schätzers dafür lässt sich dann analog zu (30) mit entsprechend angepassten Gewichten durchführen. Als Teil des Pakets `npmlreg` (Einbeck und Hinde, 2006) erlaubt die Funktion `alldist` die Modellierung mit Random Coefficients für eine feste Zahl an Komponenten. Der EM-Algorithmus tritt dort in Form einer Schleife auf, die in jedem Durchlauf die Funktion `glm.fit` aufruft, welche die ML-Schätzer für β_g und ϕ berechnet (vgl. Steinkellner, 2012). Zur Bestimmung von Startwerten wird die Funktion `glm` aufgerufen und eine gewöhnliche ML-Schätzung mit einer Komponente durchgeführt, um die linearen Prädiktoren η_i L-fach zu verwenden, wobei die Lokationen der Prädiktoren $\eta_{ig}^{(0)}$ mit den Massepunkten der Gauss-Quadratur bestimmt werden. Die unterstützten Verteilungen sind Normal-, Gamma-, Poisson-, und Binomialverteilung.

Mit den Parameterschätzern $\hat{\beta}_g = (\hat{\beta}_{g1}, \ldots, \hat{\beta}_{gp})^t$ lassen sich im konditionalen Modell Vorhersagen für $\mu_{ig} = \mathbb{E}(Y_i|Z_i = g)$ machen. Man erhält

$$\hat{\mu}_{ig} = g^{-1}(\hat{\eta}_{ig}) = g^{-1}(x_i\hat{\beta}_g),$$

aber auch das marginale Modell $\mu_i = \mathbb{E}(Y_i)$ kann geschätzt werden. Es gilt

$$\begin{aligned}\mathbb{E}(Y_i) &= \int y_i \sum_{g=1}^{L} \pi_g f(y_i|Z_i = g)\, dy_i = \sum_{g=1}^{L} \pi_g \int y_i f(y_i|Z_i = g)\, dy_i \\ &= \sum_{g=1}^{L} \pi_g \mathbb{E}(Y_i|Z_i = g) = \sum_{g=1}^{L} \pi_g \mu_{ig} \qquad (55)\end{aligned}$$

und damit resultiert als Schätzer die Summe der gewichteten Schätzer für die Erwartungswerte

$$\widehat{\mathbb{E}}(Y_i) = \sum_{g=1}^{L} \hat{\pi}_g \hat{\mu}_{ig}. \qquad (56)$$

Wie wir schon zu Beginn des Abschnitts festgehalten haben, wird in vielen Situationen eine Klassifizierung der Daten gefordert. Dabei ist jedem y_i ein $g \in \{1, \ldots, L\}$ zuzuordnen unter Verwendung einer Entscheidungsregel $G : y \to \{1, \ldots, L\}$. Die optimale oder Bayes'sche Entscheidungsregel ist dabei definiert über die Bedingung

$$G(y_i) = g \;\text{ wenn }\; \omega_{ig} \geq \omega_{il} \;\text{ für }\; l = 1, \ldots, L.$$

Demnach wird y_i jener Komponente zugewiesen, für die die maximale a posteriori Wahrscheinlichkeit geschätzt wird. Dies muss aber keineswegs eindeutig sein, da das Maximum für mehrere Komponenten angenommen werden kann (McLachlan und Peel, 2000).
Auf verschiedene Arten lässt sich der Begriff Residuen definieren. Die Funktion `alldist` berechnet zu einem Model Fit Residuen der Form

$$r_i = y_i - \widehat{\mathbb{E}}(Y_i).$$

Eine andere Möglichkeit bestünde darin, gewichtete Residuen zu berechnen. Genaue Ausführungen zu Residuen in Mischmodellen liefern Lindsay und Roeder (1992).

Bei L Komponenten und Prädiktorvariablen $x_i = (x_{i1}, \ldots, x_{ip})^t$ werden also bis zu Lp Regressionsparameter geschätzt. Will man diese Zahl reduzieren, kann man die Schätzung der Random Coefficients auf eine Teilmenge

der erklärenden Variablen einschränken. Für die restlichen Variablen wird dann angenommen, dass in allen Klassen der selbe Regressionskoeffizient wirkt. Die Funktion `alldist` bietet dazu das Argument `random` an, in dem jene Variablen gekennzeichnet werden, für die unterschiedliche Parameter geschätzt werden sollen. Der Spezialfall, in dem nur der zum konstanten Teil der Prädiktorvariablen gehörende Intercept zufällig ist, während alle anderen Koeffizienten fest sind, ist auch als *Random Intercept Model* oder *Random Effect Model* bekannt und wird in Abschnitt 3.3 behandelt.

Information über die Relevanz der Parameterschätzer liefern beim GLM die Standardfehler (Kapitel 2.3). Die Funktion `alldist` gibt zu den Regressionskoeffizienten jene Standardfehler aus, die sich auf die letzte Anpassung eines GLM durch `glm.fit` beziehen und somit zu klein sind, da sie auf der konditionalen Informationsmatrix der kompletten Daten basieren. Eine Möglichkeit der Schätzung der Standardfehler für globalen Koeffizienten findet man bei Aitkin et al. (2009, Kap. 7.5).

3.2.5 Likelihood Spikes

Mischmodelle mit verschiedenen Dispersionsparametern weisen eine Eigenschaft auf, die rechnerische Schwierigkeiten mit sich bringen kann. Wenn eine Beobachtung y_j von den übrigen abweicht, ist es möglich, dass sie im Zuge der ML-Schätzung als die einzige Realisierung einer Komponente identifiziert wird. In diesem Fall wird die geschätzte Varianz der Komponente gegen Null gehen und der Erwartungswert wird durch y_j geschätzt. Der Wert der Dichtefunktion einer solchen, degenerierten Komponente kann für y_j dann beliebig groß werden und lässt die Likelihood Funktion an den entsprechenden Stellen gegen ∞ gehen. Man bezeichnet diese Stellen als *Likelihood Spikes*. Lindsay (1995, Kap. 3) führt dazu folgendes Beispiel an: Für eine Stichprobe $y_1, \ldots, y_n$ zu

$$Y_i \overset{iid}{\sim} \pi_1 N(\mu_1, \sigma_1^2) + (1 - \pi_1) N(\mu_2, \sigma_2^2)$$

ist das globale Maximum der Likelihood Funktion ∞ und wird z.B. angenommen, wenn $\mu_1 = y_1$ und $\sigma_1^2 = 0$ erfüllt ist. Die globale Maximierung der Likelihood Funktion, die per Definition einer Wahrscheinlichkeit entspricht, ist somit nicht zulässig. Das Problem tritt als Folge davon auf, dass man den Wert der Likelihood Funktion berechnet, ohne dabei die Angabe einer Messgenauigkeit zu machen. Sei diese durch δ bezeichnet, dann folgt

$$L_\delta(\theta, y) = P(y - \delta/2 < Y < y + \delta/2 \,|\, \theta).$$

Betrachtet man in diesem Zusammenhang eine einzige Beobachtung y aus einer $N(\mu, \sigma^2)$-verteilten Population, erhält man damit

$$L_\delta(\mu, \sigma^2, y) = \Phi\left(\frac{y + \delta/2 - \mu}{\sigma}\right) - \Phi\left(\frac{y - \delta/2 - \mu}{\sigma}\right), \tag{57}$$

wobei $\Phi(\cdot)$ die Verteilungsfunktion der Standard-Normalverteilung ist. Für einen sehr kleinen Wert von σ ist die Approximation der Likelihood Funktion durch die Dichtefunktion $\phi(\cdot)$ nicht mehr adäquat. Setzt man den ML-Schätzer des Erwartungswerts, der im Fall einer einzigen Beobachtung durch $\hat{\mu} = y$ gegeben ist, in (57) ein, erhält man die Profile-Likelihood Funktion

$$PL_\delta(\sigma) = \Phi\left(\frac{\delta}{2\sigma}\right) - \Phi\left(-\frac{\delta}{2\sigma}\right) = 2\Phi\left(\frac{\delta}{2\sigma}\right) - 1,$$

die nicht von y abhängt, sondern nur von δ. Eine einzelne Beobachtung hat somit ohne zusätzlich erhältliche Information über μ keine Aussagekraft über die Varianz, was nicht überraschend ist, da von Variabilität ja nur dann gesprochen werden kann, wenn mehrere Beobachtungen vorhanden sind (vgl. Aitkin et al., 2009, Kap. 7.7).
In der Praxis löst man das Problem bei diskreten Mischungen für gewöhnlich dadurch, dass man die Likelihood Funktion mit der Restriktion $\sigma_g^2 > \sigma_0^2$, $g = 1, \ldots, L$ für einen positiven Wert σ_0^2 maximiert. Dies lässt sich einfach implementieren, macht den Wert der Likelihood Funktion aber abhängig von σ_0^2. Solange von jeder Komponente mehrere Beobachtungen stammen, muss davon kein Gebrauch gemacht werden. Natürlich ist es möglich, dass eine Beobachtung Merkmale, die in der Stichprobe einzigartig sind, aufweist, und daher wirklich einer eigenen Komponente zugeordnet werden soll. In diesem Fall kann sie von der Berechnung der Likelihood Funktion ausgeschlossen werden. Für die Modellanpassung an die verbleibenden Daten wird die Maximierung der Likelihood Funktion dann zulässig sein, und für die ausgeschlossene Beobachtung ist eine zusätzliche Komponente in der Mischung erlaubt.

3.3 Modelle mit zufälligen Effekten

3.3.1 Überdispersion

In Kapitel 2 wurde die Anpassung eines GLM, basierend auf linearen Prädiktoren

$$\eta_i = g(\mu_i) = x_i^t \beta$$

behandelt. Es kann dabei vorkommen, dass eine Modellanpassung nicht adäquat erscheint, weil sie die Variabilität der vorliegenden Daten nicht ausreichend erklärt. Die Ursache dafür kann eine unzulängliche Spezifikation des Regressionsmodells oder auch eine unpassende Verteilungsannahme der Responses Y_i sein. So enthält beispielsweise die Poissonverteilung die Annahme, dass ein Zusammenhang von Erwartungswert und Varianz der Form $\mathbb{E}(Y_i) = \phi \operatorname{var}(Y_i)$ mit $\phi = 1$ besteht (siehe Kapitel 2.1.2). In vielen Situationen drängt sich aber aufgrund der Daten die Vermutung $\phi > 1$ auf.
Eine Möglichkeit, mit derartigen Problemen umzugehen, ist, die Annahme zu treffen, dass in x_i bestimmte wichtige Variablen fehlen, die, wenn sie verfügbar wären, eine solche *Überdispersion* erklären würden.
Wir gehen also ab jetzt davon aus, dass neben den beobachteten Variablen x_i eine Menge nicht-beobachteter Variablen $u_i = (u_1, \ldots, u_{p'})^t$ existiert und die wahren linearen Prädiktoren sich als

$$\eta_i = g(\mu_i) = x_i^t \beta + u_i^t \gamma$$

schreiben lassen, wobei γ der Vektor der Regressionskoeffizienten zu den nicht-beobachtbaren Variablen ist. Diese Schreibweise kann als Verallgemeinerung der Definition der Prädiktoren aus Abschnitt 2.1.3 gesehen werden, wo die Menge der nicht-beobachteten Variablen leer war. Da die Vektoren u_i nicht gemessen werden können, liefern sie keine Information und können als Zufallsvektoren aufgefasst werden. Weil auch γ unbekannt ist, erhalten wir mit $z_i = u_i^t \gamma$ einen skalaren, nicht-beobachteten Wert oder zufälligen Effekt und lineare Prädiktoren der Form

$$\eta_i = g(\mu_i) = x_i^t \beta + z_i.$$

Das Modell wird *Random Effect Model* oder Modell mit zufälligem Effekt genannt. In dieser Form fließt z_i additiv in das Modell ein, steht also nicht in Interaktion mit x und erzeugt dadurch nur einen *Random Intercept*. Ohne Beschränkung der Allgemeinheit kann angenommen werden, dass die z_i Realisierung von unabhängig, identisch verteilten Zufallsvariablen Z_i mit Dichte oder Wahrscheinlichkeitsfunktion $h(z)$ sind, und $\mathbb{E}(Z_i) = 0$, da ein Erwartungswert ungleich 0 einem Intercept im Modell zugeschrieben werden könnte.
Für die weitere Vorgehensweise wird auch angenommen, dass keine Abhängigkeit zwischen z_i und x_i besteht.[6] Die Verteilung von Y ist dann als zusammengesetzte oder gemischte Verteilung schreibbar mit Dichte

$$m(y) = \int f(y|z)h(z)\, dz. \tag{58}$$

[6] Aitkin et al. (2009, Kap. 8) zeigen, dass auch eine Abhängigkeit hier keine Einschränkung für die Modellierung bedeutet, wenn sie linearer Natur ist.

Die konditionale Verteilung von Y, gegeben Z, ist Mitglied der Exponentialfamilie mit Dichte oder Wahrscheinlichkeitsfunktion $f(y|z)$. Man erhält damit die allgemeinen Eigenschaften

$$\begin{aligned}\mathbb{E}(Y) &= \mathbb{E}\big(\mathbb{E}(Y|Z)\big),\\ \operatorname{var}(Y) &= \mathbb{E}\big(\operatorname{var}(Y|Z)\big) + \operatorname{var}\big(\mathbb{E}(Y|Z)\big).\end{aligned}$$

Verwendet man die Notation $\mu(Z)$ und $v(Z)$ für Erwartungswert und Varianz von Y, gegeben Z, resultiert

$$\begin{aligned}\mathbb{E}(Y) &= \mathbb{E}\big(\mu(Z)\big),\\ \operatorname{var}(Y) &= \mathbb{E}\big(v(Z)\big) + \operatorname{var}\big(\mu(Z)\big).\end{aligned}$$

Die ursprüngliche Beziehung von Erwartungswert und Varianz, $\operatorname{var}(Y) = a(\phi)V(\mu)$, geht durch den Einfluss des zufälligen Effektes verloren, wie wir am Beispiel der zu Beginn des Abschnitts erwähnten Poissonverteilung sehen. Sei $Y|Z \sim P(\mu(Z))$ mit $v(Z) = \mu(Z)$, dann gilt für das loglineare Modell $\log \mu(Z) = x^t\beta + Z$

$$\begin{aligned}\mathbb{E}(Y) &= \mathbb{E}\big(\exp(x^t\beta + Z)\big) = \exp(x^t\beta)\mathbb{E}\big(\exp(Z)\big) = \exp(x^t\beta)M_Z(1),\\ \operatorname{var}(Y) &= \mathbb{E}\big(\exp(x^t\beta + Z)\big) + \operatorname{var}\big(\exp(x^t\beta + Z)\big)\\ &= \exp(x^t\beta)M_Z(1) + \mathbb{E}\big(\exp(2(x^t\beta + Z))\big) - \mathbb{E}^2\big(\exp(x^t\beta + Z)\big)\\ &= \exp(x^t\beta)M_Z(1) + \exp(2x^t\beta)[M_Z(2) - M_Z^2(1)],\end{aligned}$$

wobei $M_Z(t)$ die Momentenerzeugende Funktion von Z ist. Damit erhalten wir

$$\operatorname{var}(Y) = \mathbb{E}(Y) + \phi\,\mathbb{E}^2(Y),$$

mit $\phi = [M_Z(2) - M_Z^2(1)] - 1$ für eine beliebige Verteilung von Z.

Bevor man zur ML-Schätzung übergehen kann, sind gewisse Annahmen über die Verteilung von Z zu treffen. Wir werden dazu zwei Ansätze diskutieren: Im ersten (Abschnitt 3.3.2) wird von einem normalverteilten zufälligen Effekt ausgegangen, während der zweite (Abschnitt 3.3.3) sich dem allgemeinsten Fall, in dem $h(z)$ nichtparametrisch geschätzt wird, widmet.

3.3.2 Normalverteilte zufällige Effekte

Zu jeder Verteilung F der Exponentialfamilie existiert eine spezielle Verteilung für den kanonischen Parameter θ, genannt die zu F *konjugierte Verteilung*. Mit dieser erhält man eine zusammengesetzte Verteilung, die als *konjugierte Erweiterung* bezeichnet wird und für die eine geschlossene Darstellung

der Dichte oder Wahrscheinlichkeitsfunktion verfügbar ist (Lindsey, 1997). Wird die Verteilungsannahme für den zufälligen Effekt Z daran angepasst, lässt sich auch

$$m(y;\lambda) = \int f(y|z;\lambda)h(z)\,dz$$

analytisch berechnen, wobei die resultierende Verteilung der konjugierten Erweiterung im Allgemeinen kein Mitglied der Exponentialfamilie mehr ist. Die Normalverteilung tritt in diesem Zusammenhang als ihre eigene konjugierte Verteilung auf, wenn ein Effekt Z auf den Erwartungswert einer normalverteilten Zufallsvariable Y wirkt. Für $Y|Z \sim N(\theta + Z, \sigma^2)$ und $Z \sim N(\mu, \rho^2)$ erhält man als entsprechendes marginales Modell mit $Y \sim N(\theta + \mu, \sigma^2 + \rho^2)$ aber wiederum eine Normalverteilung, in der die Parameter nicht mehr identifizierbar sind. Somit lässt sich die Normalverteilung bezüglich des Effekts auf den Erwartungswert nicht konjugiert erweitern.[7] Unter der Annahme, $h(z)$ beschreibe die Dichte einer normalverteilten Zufallsvariable, ist (58) daher nur dann analytisch berechenbar, wenn auch Y normalverteilt angenommen wird. Ansonsten wird zur Approximation die Gauss-Quadratur, mit der man

$$m(y;\lambda) \approx \sum_{g=1}^{L} f(y|z_g;\lambda)\pi_g \tag{59}$$

erhält, verwendet. Dabei sind $z_1, \ldots, z_L$ bekannte Massestellen, die mit der Wahl von L automatisch festgelegt sind und $\pi_1, \ldots, \pi_L$ die dazugehörenden bekannten Massen. Aufgrund der Notwendigkeit dieser numerischen Integration war die Verwendung normalverteilter Random Effects nicht immer beliebt. Wie wir aber nun sehen werden, lässt sich die Modellschätzung dadurch auf eine Vereinfachung des Mischungsproblems zurückführen, was wiederum die Anwendung des EM-Algorithmus ermöglicht. Die Aufnahme des nicht-beobachteten, normalverteilten Effekts Z in den linearen Prädiktor des GLM kann ohne Verlust der Allgemeinheit durch

$$\eta_i = x_i^t\beta + \sigma Z_i \quad \text{mit } Z_i \overset{iid}{\sim} N(0,1)$$

ausgedrückt werden. Für die Likelihood Funktion folgt

$$\begin{aligned} L(\beta,\sigma,y) &= \prod_{i=1}^{n} \int f(y_i|z_i;\beta,\sigma)\phi(z_i)\,dz_i \\ &\approx \prod_{i=1}^{n} \sum_{g=1}^{L} \pi_g f(y_i|z_g;\beta,\sigma). \end{aligned} \tag{60}$$

[7] Es lässt sich zeigen, dass eine Erweiterung bezüglich eines Effekts auf die Varianz sehr wohl möglich ist (Aitkin et al., 2009, Kap. 8.2).

Sie entspricht damit approximativ der Likelihood Funktion der Stichprobe aus einer diskreten Mischung von Verteilungen der Exponentialfamilie mit bekannten Massepunkten z_g und Mischwahrscheinlichkeiten π_g. In diesem Zusammenhang gilt für den linearen Prädiktor der i-ten Beobachtung, gegeben die Beobachtung stammt aus Komponente g

$$\eta_{ig} = x_i^t \beta + \sigma z_g.$$

Für die Zielfunktion Q erhält man analog zu (52) in Abschnitt 3.2.4

$$Q(\beta, \sigma | \beta^{(k)}, \sigma^{(k)}) \approx \sum_{i=1}^{n} \sum_{g=1}^{L} \omega_{ig}^{(k)} \Big(\log \pi_g + \log f(y_i | z_g; \beta, \sigma) \Big), \qquad (61)$$

wobei hier

$$\omega_{ig}^{(k)} = \frac{\pi_g f(y_i | z_g; \beta^{(k)}, \sigma^{(k)})}{\sum_{l=1}^{L} \pi_l f(y_i | z_l; \beta^{(k)}, \sigma^{(k)})}$$

gilt. Der M-Schritt erfordert hierbei keine Berechnung von $\hat{\pi}_g$, da die Mischwahrscheinlichkeiten bekannt sind. Die Ableitung der Funktion Q nach β_j enthält wie in Abschnitt 3.2.4 die doppelte Summation über gewichtete Score-Funktionen, was die Maximierung bezüglich β und σ äquivalent zu einer gewichteten ML-Schätzung eines GLM macht, basierend auf dem Datensatz

y	ω	β_1	$\dots$	β_p	σ
y_1	ω_{11}	x_{11}	$\dots$	x_{1p}	z_1
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
y_n	ω_{n1}	x_{n1}	$\dots$	x_{np}	z_1
y_1	ω_{12}	x_{11}	$\dots$	x_{1p}	z_2
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
y_n	ω_{n2}	x_{n1}	$\dots$	x_{np}	z_2
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
y_1	ω_{1L}	x_{11}	$\dots$	x_{1p}	z_L
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
y_n	ω_{nL}	x_{n1}	$\dots$	x_{np}	z_L

Gegenüber Abschnitt 3.2.4 stellt dies eine wesentliche Vereinfachung dar. Die hier vorliegende endliche Mischung entsteht nämlich durch die Approximation der Normalverteilung mittel Gauss-Quadratur, basierend auf L Massestellen und resultiert nicht aus der Annahme unterschiedlicher Regressionsparameter. Man spricht auch von einem Überdispersionsmodell, für dessen ML-Schätzer wiederum die R-Funktion `alldist` verwendet werden kann. Das

Argument `random` beinhaltet dabei nur den Intercept und die Verwendung der Gauss-Quadratur zur Approximation der Normalverteilung wird durch Spezifikation des Arguments `random.distribution` mit `gq` erreicht.
Eine Besonderheit der Gauss-Quadratur ist die Tatsache, dass die Likelihood Funktion nicht monoton wächst mit Erhöhung der Anzahl der Massepunkte. Man beachte, dass die Anzahl der Parameter im Modell von L unabhängig ist, da Massepunkte und Wahrscheinlichkeiten bekannte Größen sind. Fehlende Monotonie der Likelihood Funktion kann deshalb als Folge davon auftreten, dass die tatsächliche Varianz des zufälligen Effekts nicht unbedingt besser repräsentiert sein muss, wenn L vergrößert wird. Die Schwankungen der Likelihood Funktion werden mit zunehmender Anzahl natürlich geringer. Lesaffre und Spiessens (2001) untersuchen den Einfluss der Größe von L am Beispiel der logistischen Regression.

3.3.3 Beliebige zufällige Effekte

Wir beschäftigen uns nun mit dem allgemeinen Fall, in dem keine Annahme für $h(z)$ getroffen wird. Während für die Normalverteilung die Gauss-Quadratur eine diskrete Approximation mit bekannten Werten $z_1, \ldots, z_L$ und $\pi_1, \ldots, \pi_L$ liefert, werden die Massepunkte und Wahrscheinlichkeiten im Falle einer beliebigen Verteilung als eine Menge unbekannter Parameter angenommen, deren Schätzer den *nichtparametrischen Maximum Likelihood* (NPML)-Schätzer $\hat{h}(z)$ ergeben. Man erhält damit lineare Prädiktoren

$$\eta_i = x_i^t\beta + z_i \quad \text{mit} \quad z_i \overset{iid}{\sim} F_Z \quad \text{und} \quad \mathbb{E}(Z_i) = 0,$$

also

$$\eta_{ig} = x_i^t\beta + z_g.$$

Es resultiert damit der in Abschnitt 3.2.4 erwähnte Spezialfall der Mischung von GLMs, in dem nur der Intercept komponentenspezifischen Wert hat. Während im E-Schritt wiederum die Gewichte ω_{ig} aktualisiert werden, erfordert der M-Schritt hier wieder die Berechnung der $\pi_g^{(k+1)}$ und eine gewichtete

GLM-Schätzung zu

y	ω	β_1	$\dots$	β_p	z_1	z_2	$\dots$	z_L
y_1	ω_{11}	x_{11}	$\dots$	x_{1p}	1	0	$\dots$	0
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$
y_n	ω_{n1}	x_{n1}	$\dots$	x_{np}	1	0	$\dots$	0
y_1	ω_{12}	x_{11}	$\dots$	x_{1p}	0	1	$\dots$	0
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$
y_n	ω_{n2}	x_{n1}	$\dots$	x_{np}	0	1	$\dots$	0
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$
y_1	ω_{1L}	x_{11}	$\dots$	x_{1p}	0	0	$\dots$	1
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$
y_n	ω_{nL}	x_{n1}	$\dots$	x_{np}	0	0	$\dots$	1

Z kann in dieser Form als L-stufiger Faktor aufgefasst werden. Falls x bereits einen Intercept enthält, ist dieser von z_1 in der Modellschätzung nicht mehr unterscheidbar. Die Assoziation der z_g mit latenten Klassen kann sich in bestimmten Situationen als direkte Interpretation aufdrängen. So ergibt sich manchmal die Möglichkeit, Outliers zu identifizieren. Unterscheidet sich eine Beobachtung stark von den übrigen, kann sie als Realisierung einer eigenen Komponente mit Masse $\hat{\pi}_g = 1/n$ identifiziert werden.[8] Latente Klassen sind im Allgemeinen bei einem hohen Anteil an Werten 0 und 1 in den a posteriori Wahrscheinlichkeiten zu identifizieren, ihre Bedeutung ist aber als eingeschränkt anzusehen, da die z_g doch in erster Linie die diskrete Form des zufälligen Effekts Z repräsentieren (Aitkin et al., 2009, S. 488).
Analog zu (56) kann eine Prädiktion für den marginalen Erwartungswert von Y_i gemacht werden durch

$$\widehat{\mathbb{E}}(Y_i) = \sum_{g=1}^{L} \hat{\pi}_g \hat{\mu}_{ig}. \tag{62}$$

Bemerkung: Diese Art der Prädiktion ist auch für normalverteilte Effekte, approximiert mittels Gauss-Quadratur, möglich und auch häufig notwendig, da eine analytische Darstellung nur bei speziellen Verteilungsannahmen für Y verfügbar ist (vgl. Aitkin et al., 2009, S. 481).

Während die z_i in Abschnitt 3.2.4 als Indizierung der Komponenten fungierten, sind sie hier Realisierungen eines Random Effects, für den sich der a

[8] Solange wir von einem gemeinsamen Dispersionsparameter ausgehen, ist das Auftreten von Likelihood Spikes (Abschnitt 3.2.5) unwahrscheinlich.

posteriori Erwartungswert berechnen lässt. Es gilt

$$\begin{aligned}\mathbb{E}(Z_i|y_i) &= \int z_i f(z_i|y_i)\, dz_i \\ &= \int z_i \frac{f(y_i|z_i)f(z_i)}{\int f(y_i|z)f(z)\, dz}\, dz_i \\ &= \sum_{g=1}^{L} z_g \frac{f_{ig}\pi_g}{\sum_{l=1}^{L} f_{il}\pi_l} \\ &= \sum_{g=1}^{L} z_g \omega_{ig}, \end{aligned} \tag{63}$$

wofür der *empirische Bayes Schätzer* durch

$$\widetilde{\mathbb{E}}(Z_i|y_i) = \sum_{g=1}^{L} \hat{z}_g \hat{\omega}_{ig}$$

bestimmt ist. Außerdem lässt sich hier der lineare Prädiktor $\tilde{\eta}_i$ bestimmen. Man erhält dafür

$$\begin{aligned}\widetilde{\eta}_i = x_i^t\hat{\beta} + \widetilde{\mathbb{E}}(Z_i|y_i) &= x_i^t\hat{\beta}\overbrace{\sum_{g=1}^{L}\hat{\omega}_{ig}}^{=1} + \sum_{g=1}^{L}\hat{z}_g\hat{\omega}_{ig} \\ &= \sum_{g=1}^{L}\hat{\omega}_{ig}\left(x_i^t\hat{\beta} + \hat{z}_g\right) = \sum_{g=1}^{L}\hat{\omega}_{ig}\hat{\eta}_{ig}.\end{aligned}$$

3.3.4 Shared Random Effects

Die Motivation für die Hinzunahme eines zufälligen Effekts in ein Modell kann auch von einer anderen Art der Stichprobenerhebung kommen. Werden n unabhängige Datengruppen $y_i = (y_{i1}, \ldots, y_{in_i})^t$ beobachtet, dann spricht man von zweistufigem Sampling. Die erste Ebene bildet dabei die Stichprobe $y_1, \ldots, y_n$ und in jeder ihrer sogenannten *Primary Sampling Units* werden Datenpunkte y_{ij}, $j = 1, \ldots, n_i$, erhoben, die als *Secondary Sampling Units* bezeichnet werden und die zweite Ebene des Samplings darstellen. In den meisten Populationen impliziert für Beobachtungen y_{ij} und $y_{ij'}$ die Zugehörigkeit zur selben Einheit eine stärkere Homogenität als sie für Beobachtungen aus unterschiedlichen Einheiten gegeben ist. Eine naheliegende Idee, dies in ein Modell einzubauen, ist die Aufnahme eines zufälligen Effekts

Z, der auf alle Elemente einer Gruppe die gleiche Wirkung haben soll. Es entstehen dadurch Prädiktoren der Form

$$\eta_{ij} = x_{ij}^t \beta + z_i.$$

Der Effekt z_i, der somit alle y_{ij} einer Einheit betrifft, wird deshalb auch *Shared Random Effect* genannt. Dieser Effekt induziert eine Abhängigkeit der Variablen innerhalb einer Gruppe, während zwischen Variablen unterschiedlicher Gruppen Unabhängigkeit besteht. Dadurch erreicht man einen Zugang zu einem großen Bereich von Modellen, genannt *Variance Component Models*, die eng in Zusammenhang mit Überdispersionsmodellen stehen. Diese Arbeit wird sich jedoch nicht intensiver damit auseinandersetzen, da der Nutzen für die hier behandelte Modellklasse kein wesentlicher ist.

3.4 Zensierte Daten

3.4.1 Überblick

Bis jetzt haben wir Modelle betrachtet, in denen der nicht-beobachtete Teil der Daten mit einer latenten Variable Z assoziiert wurde, die entweder die diskrete Verteilung einer Mischung beschrieb (Abschnitt 3.2), oder den Einfluss nicht-beobachtbarer Prädiktorvariablen modellieren sollte (Abschnitt 3.3). In diesem Abschnitt widmen wir uns einer Situation, in der die eingeschränkte Beobachtbarkeit der Daten direkt die Responses betrifft.
Wird die Realisierung einer Zufallsvariable Y nicht exakt beobachtet, kann man die Beobachtung schlicht und einfach als fehlend bezeichnen. Ist aber zumindest eine Information über den möglichen Wertebereich verfügbar, spricht man für gewöhnlich von einer *zensierten Beobachtung*. Wir unterscheiden dabei zwischen drei Arten:

- links-zensierte Beobachtung: $Y \leq \tau$
- rechts-zensierte Beobachtung: $Y \geq \tau$
- intervall-zensierte Beobachtung: $Y \in [a, b]$

Die Größen τ sowie a und b sind jeweils bekannt und beeinflussen die Messbarkeit der Zufallsvariable τ. Im Fall einer links-zensierten Beobachtung kann τ als Schwellwert gesehen werden. Wird er nicht überschritten, ist keine exakte Messung mehr möglich und anstelle von $Y = y$ liegt nur die Information $Y \leq \tau$ vor. Ein Beispiel dafür sind Nachweisbarkeitsgrenzen bei Messungen von Konzentrationen verschiedener Chemikalien. Liegt ein entsprechender Wert unter diesem Niveau, ist er zwar nicht exakt zu beobachten, man gewinnt aber die Information, dass er nicht darüber liegt.

Ähnliche Aussagen lassen sich für rechts-zensierte Daten treffen, deren Auftreten eine typische Eigenschaft der Analyse von Lebensdauern (engl.: *Survival Analysis*) ist. Dabei lassen sich zu einem für die Erhebung der Daten vorher festgelegten Zeitpunkt τ nur die Lebensdauern jener Individuen beobachten, die den Zeitpunkt τ nicht überleben. Alle übrigen liefern keine exakte, sondern eine rechts-zensierte Beobachtung, also die Information $Y \geq \tau$.

Bemerkung: In diesem Fall bietet sich die Verwendung der Information $Y > \tau$ an, wenn ein Ableben genau zum Zeitpunkt τ ausgeschlossen werden kann. Da es sich bei Lebenszeiten meist um stetig verteilte Zufallsvariablen handelt, ändert dies aber nichts am Modell.

Schließlich kommt es zu einer intervall-zensierten Beobachtung, wenn Y einen nicht-beobachtbaren Wert aus dem Intervall $[a, b]$ annimmt. Diese Situation kann aus dem Vorliegen einer rechts- oder links-zensierten Beobachtung hervorgehen, wenn zusätzlich eine zweite Grenze beobachtet wird, die den Wertebereich dann auf das Intervall $[a, b]$ einschränkt.
Diese Arbeit beschäftigt sich in erster Linie mit links-zensierten Daten, deren Existenz die Motivation für die Betrachtung einer speziellen Modellklasse in Kapitel 4 liefert. Wie wir aber sehen werden, lassen sich viele der gewonnenen Ergebnisse für rechts-zensierte Daten analog verwenden.

Nehmen wir an, dass für Elemente $y_1, \ldots, y_n$ einer Stichprobe die Möglichkeit besteht, dass ihre Werte links-zensiert sind, nämlich dann, wenn $y_i \leq \tau_i$ gilt. Dabei ist τ_i ein für die i-te Beobachtung charakteristischer, fester Wert, der deshalb als beobachtbar bezeichnet werden kann. Wir können damit den Vektor der beobachtbaren Daten Y^* definieren mit

$$Y_i^* = \max(Y_i, \tau_i), \quad i = 1, \ldots, n$$

sowie den Vektor C der Indikatorvariablen mit

$$C_i = \begin{cases} 1 & \text{wenn } Y_i \leq \tau_i, \\ 0 & \text{wenn } Y_i > \tau_i, \end{cases} \quad i = 1, \ldots, n.$$

Diese geben Auskunft darüber, ob eine Beobachtung exakt oder zensiert ist. Kann ein Wert y_i in der Praxis nicht exakt gemessen werden, weil er unter einem Messbarkeitsniveau τ_i liegt, dann ist es nicht selbstverständlich, dass beim Durchführen der Messung ein Wert kleiner τ_i in den Datenvektor eingetragen wird. Manchmal wird der Wert 0 gewählt, in vielen Fällen wird auch einfach der Wert von τ_i als eine obere Abschätzung gespeichert. Um diese Situation als das Vorliegen einer links-zensierten Beobachtung zu

kennzeichnen, ist es also durchaus sinnvoll, $C_i = 1$ auch für den Fall $Y_i = \tau_i$ zu definieren.

3.4.2 Die Likelihood Funktion

Wir nehmen an, dass $Y_1, \ldots, Y_n$ unabhängig verteilte Zufallsvariablen sind, wobei die Verteilung von Y_i durch die Dichte f_{Y_i} und die Verteilungsfunktion F_{Y_i} bestimmt ist. Wenn ein Teil der Beobachtungen zensiert ist, wirkt sich das auf die ML-Schätzung aus. Die Dichte-Funktion einer einzelnen Beobachtung (y_i^*, c_i) ist gegeben durch

$$f(y_i^*, c_i) = \big[f_{Y_i}(y_i^*)\big]^{1-c_i}\big[F_{Y_i}(y_i^*)\big]^{c_i} = \big[f_{Y_i}(y_i)\big]^{1-c_i}\big[F_{Y_i}(\tau_i)\big]^{c_i}.$$

Sie stimmt für eine unzensierte Beobachtung mit der Dichte f_{Y_i} überein und ist im Fall einer links-zensierten Beobachtung gleich der Wahrscheinlichkeit, dass $Y_i \leq \tau_i$ gilt. Analog folgt

$$f(y_i^*, c_i) = \big[f_{Y_i}(y_i)\big]^{1-c_i}\big[1 - F_{Y_i}(\tau_i)\big]^{c_i},$$

wenn eine Beobachtung rechts-zensiert sein kann. Schreibt man die Wahrscheinlichkeit, dass die Beobachtung zensiert ist als $S_{Y_i}(\tau_i)$, erhält man für beide Fälle die Darstellung

$$f(y_i^*, c_i) = \big[f_{Y_i}(y_i^*)\big]^{(1-c_i)}\big[S_{Y_i}(y_i^*)\big]^{c_i}.$$

Für die Likelihood Funktion der Stichprobe $y_1^*, \ldots, y_n^*$ folgt damit

$$L(y^*, c) = \prod_{i=1}^{n}\big[f_{Y_i}(y_i^*)\big]^{1-c_i}\big[S_{Y_i}(y_i^*)\big]^{c_i}. \tag{64}$$

Je nachdem, welche Verteilungsannahmen für die Y_i getroffen wurden, kann die Likelihood Funktion in dieser Form angenehme oder unangenehme Eigenschaften aufweisen. In der Survival Analysis spielt die Exponentialverteilung eine wesentliche Rolle, da sie zu einer einfachen Form der Likelihood Funktion für rechts-zensierte Daten führt. Auch wenn die Lebenszeiten nur in seltenen Fällen tatsächlich exponentialverteilt sind, stehen viele dort betrachtete Verteilungen eng mit der Exponentialverteilung in Zusammenhang (Aitkin et al., 2009, Kap. 6).

3.4.3 Links-zensierte normalverteilte Daten

Im Folgenden sollen $\phi(\cdot)$ und $\Phi(\cdot)$ Dichte und Verteilungsfunktion der Standard-Normalverteilung bezeichnen:

$$\phi(z) = \frac{1}{\sqrt{2\pi}}\exp(-z^2/2), \quad \Phi(z) = \int_{-\infty}^{z}\phi(t)\,dt$$

Dichte und Verteilungsfunktion einer $N(\mu, \sigma^2)$-verteilten Zufallsvariable lassen sich damit schreiben als

$$f(y) = \frac{1}{\sigma}\phi\left(\frac{y-\mu}{\sigma}\right) \quad \text{und} \quad F(y) = \Phi\left(\frac{y-\mu}{\sigma}\right).$$

Liegt eine Stichprobe $y_1^*, \ldots, y_n^*$ vor, wobei wie zuvor $y_i^* = \max(y_i, \tau_i)$ gilt mit $Y_i \overset{iid}{\sim} N(\mu, \sigma^2)$, dann folgt für die Likelihood Funktion

$$L(\mu, \sigma, y^*, c) = \prod_{i=1}^{n} \left[\frac{1}{\sigma}\phi\left(\frac{y_i^*-\mu}{\sigma}\right)\right]^{1-c_i} \left[\Phi\left(\frac{y_i^*-\mu}{\sigma}\right)\right]^{c_i}. \tag{65}$$

Für den Fall, dass zensierte Beobachtungen vorliegen, wird die ML-Schätzung also komplizierter und motiviert die Anwendung des EM-Algorithmus.
Dazu wird wieder die Log-Likelihood Funktion der vollständigen, zum Teil nicht-beobachteten Daten y betrachtet. Ihre konditionale Erwartung, gegeben y^* und c, sowie aktuelle Parameterschätzer $\mu^{(k)}, \sigma^{2(k)}$, ist

$$\begin{aligned} Q(\mu, \sigma^2 | \mu^{(k)}, \sigma^{2(k)}) &= \mathbb{E}\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\mu)^2 - \frac{n}{2}\log 2\pi\sigma^2 \,\middle|\, y^*, c; \mu^{(k)}, \sigma^{2(k)} \right) \\ &= -\frac{n}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\mathbb{E}\Big((y_i-\mu)^2 \Big| y^*, c; \mu^{(k)}, \sigma^{2(k)}\Big). \end{aligned} \tag{66}$$

Entscheidend für den E-Schritt ist, welche Erwartungswerte zur Bestimmung der Zielfunktion Q berechnet werden müssen. Für die Normalverteilung ist die Log-Likelihood Funktion linear in y_i und y_i^2. Das bedeutet, dass die entsprechenden Werte für zensierte Beobachtungen durch die konditionalen Erwartungswerte ersetzt werden müssen. Es gilt

$$\mathbb{E}(y|y \le \tau) = \frac{\int_{-\infty}^{\tau} y\frac{1}{\sigma}\phi\left(\frac{y-\mu}{\sigma}\right)\, dy}{\Phi\left(\frac{\tau-\mu}{\sigma}\right)} = \mu - \sigma\frac{\phi\left(\frac{\tau-\mu}{\sigma}\right)}{\Phi\left(\frac{\tau-\mu}{\sigma}\right)} = \mu + \sigma\alpha\left(\frac{\tau-\mu}{\sigma}\right). \tag{67}$$

Dabei beschreibt $\alpha(z)$ die Sterblichkeitsfunktion für die Normalverteilung. Definiert als die negative Ableitung der Wahrscheinlichkeit, dass eine Beobachtung zensiert ist, also

$$\alpha(z) = -\frac{\partial}{\partial z}\log\big(S(z)\big) = \begin{cases} -\frac{\phi(z)}{\Phi(z)} & \text{wenn } Y < z\sigma + \mu, \\ \frac{\phi(z)}{1-\Phi(z)} & \text{wenn } Y > z\sigma + \mu, \end{cases}$$

kann $\alpha(z)$ gleichzeitig bei rechts-zensierten Daten verwendet werden, da in diesem Fall

$$\mathbb{E}(y|y \ge \tau) = \frac{\int_{\tau}^{\infty} y\frac{1}{\sigma}\phi\left(\frac{y-\mu}{\sigma}\right)\, dy}{1-\Phi\left(\frac{\tau-\mu}{\sigma}\right)} = \mu + \sigma\frac{\phi\left(\frac{\tau-\mu}{\sigma}\right)}{1-\Phi\left(\frac{\tau-\mu}{\sigma}\right)} \tag{68}$$

gilt. Für rechts-zensierte Daten bezeichnet man $\alpha(z)$ als Überlebensfunktion. Die Resultate in diesem Abschnitt halten damit ebenso für rechts-zensierte Beobachtungen, wenn die Bedingung $Y \leq \tau$ durch $Y \geq \tau$ ersetzt wird. Für das zweite Moment gilt

$$\mathbb{E}(y^2|y \leq \tau) = \frac{\int_{-\infty}^{\tau} y^2 \frac{1}{\sigma}\phi\left(\frac{y-\mu}{\sigma}\right)\, dy}{\Phi\left(\frac{\tau-\mu}{\sigma}\right)} = \mu^2 + \sigma^2 + \sigma(\mu+\tau)\alpha\left(\frac{\tau-\mu}{\sigma}\right). \quad (69)$$

Damit können wir im E-Schritt zu den aktuellen Parameterschätzern $\mu^{(k)}$ und $\sigma^{2(k)}$ die erwarteten Beobachtungen und zweiten Momente

$$\tilde{y}_i^{(k)} = (1-c_i)y_i + c_i\left[\mu^{(k)} + \sigma^{(k)}\alpha\left(\frac{\tau_i - \mu^{(k)}}{\sigma^{(k)}}\right)\right], \quad (70)$$

$$\widetilde{y^2}_i^{(k)} = (1-c_i)y_i^2 + c_i\left[\mu^{2(k)} + \sigma^{2(k)} + \sigma^{(k)}(\mu^{(k)}+\tau_i)\alpha\left(\frac{\tau_i - \mu^{(k)}}{\sigma^{(k)}}\right)\right] \quad (71)$$

konstruieren. Für (66) folgt dann

$$Q(\mu, \sigma^2|\mu^{(k)}, \sigma^{2(k)}) = -\frac{n}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(\widetilde{y^2}_i^{(k)} - 2\mu\tilde{y}_i^{(k)} + \mu^2\right) \quad (72)$$

und man erhält

$$\frac{\partial Q}{\partial \mu} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(\tilde{y}_i^{(k)} - \mu),$$

$$\frac{\partial Q}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{n}\left(\widetilde{y^2}_i^{(k)} - 2\mu\tilde{y}_i^{(k)} + \mu^2\right),$$

also

$$\mu^{(k+1)} = \frac{1}{n}\sum_{i=1}^{n}\tilde{y}_i^{(k)}, \quad (73)$$

$$\sigma^{2(k+1)} = \frac{1}{n}\sum_{i=1}^{n}\left(\widetilde{y^2}_i^{(k)} - 2\mu^{(k+1)}\tilde{y}_i^{(k)} + \mu^{2(k+1)}\right). \quad (74)$$

Dieses Ergebnis findet man bei Aitkin et al. (2009, S. 413). Um den neuen Parameterschätzer der Varianz in Termen der konstruierten Beobachtungen $\tilde{y}_i$ auszudrücken, bemerken wir, dass für unzensierte Beobachtungen $\widetilde{y^2}_i = \tilde{y}_i^2$ gilt. Mit $\tilde{\tau}_i^{(k)} = (\tau_i - \mu^{(k)})/\sigma^{(k)}$ folgt für eine zensierte Beobachtung aus (71)

und (71)

$$\begin{aligned}
\widetilde{y^2}_i^{(k)} &= \mu^{2(k)} + \sigma^{2(k)} + \sigma^{(k)}(\mu^{(k)} + \tau_i)\alpha(\tilde{\tau}_i^{(k)}) \\
&= \mu^{2(k)} + 2\mu^{(k)}\sigma^{(k)}\alpha(\tilde{\tau}_i^{(k)}) + \sigma^{2(k)}\alpha^2(\tilde{\tau}_i^{(k)}) - 2\mu^{(k)}\sigma^{(k)}\alpha(\tilde{\tau}_i^{(k)}) \\
&\quad - \sigma^{2(k)}\alpha^2(\tilde{\tau}_i^{(k)}) + \sigma^{2(k)} + \sigma^{(k)}(\mu^{(k)} + \tau_i)\alpha(\tilde{\tau}_i^{(k)}) \\
&= \left(\tilde{y}_i^{(k)}\right)^2 - \sigma^{2(k)}\alpha^2(\tilde{\tau}_i^{(k)}) + \sigma^{2(k)} + \sigma^{(k)}(-\mu^{(k)} + \tau_i)\alpha(\tilde{\tau}_i^{(k)}) \\
&= \left(\tilde{y}_i^{(k)}\right)^2 + \sigma^{2(k)}\left(1 + \tilde{\tau}_i^{(k)}\alpha(\tilde{\tau}_i^{(k)}) - \alpha^2(\tilde{\tau}_i^{(k)})\right).
\end{aligned}$$

Es resultiert damit

$$\sigma^{2(k+1)} = \frac{1}{n}\sum_{i=1}^{n}\left[(\tilde{y}_i^{(k)} - \mu^{(k+1)})^2 + c_i\sigma^{2(k)}\left(1 + \tilde{\tau}_i^{(k)}\alpha(\tilde{\tau}_i^{(k)}) - \alpha^2(\tilde{\tau}_i^{(k)})\right)\right]. \tag{75}$$

Damit unterscheidet sich dieses Resultat von einem anderen Iterationsschema, das man bei Lawless (1982, S. 224) findet. Als aktualisierter Schätzer der Varianz wird dort

$$\sigma^{2(k+1)} = \sum_{i=1}^{n}(\tilde{y}_i^{(k)} - \mu^{(k+1)})^2 \Bigg/ \left(n - c - \sum_{i=1}^{n} c_i\left(\tilde{\tau}_i^{(k)}\alpha(\tilde{\tau}_i^{(k)}) - \alpha^2(\tilde{\tau}_i^{(k)})\right)\right) \tag{76}$$

angegeben, wobei c die Anzahl der zensierten Beobachtungen ist. Man erhält dieses Resultat, wenn man $\sigma^{2(k)}$ in (75) durch $\sigma^{2(k+1)}$ ersetzt. Obwohl er bei der Herleitung dieses Schemas von einem Ansatz ausgeht, der von Aitkin et al. (2009) abweicht, bezeichnet es Lawless als eine Prozedur, die mehr oder weniger ungewollt einem EM-Algorithmus entspricht.

3.4.4 Das Tobit-Modell

Das Tobit-Modell bringt links-zensierte Daten in Zusammenhang mit linearer Regression. Wie zuvor lautet die zentrale Annahme, dass beobachtbare Zufallsvariablen $Y_i^* = \max(Y_i, \tau_i)$ vorliegen, wobei die Variablen Y_i nun als Responses des linearen Regressionsmodells

$$Y_i = x_i^t\beta + \varepsilon_i \text{ mit } \varepsilon_i \overset{iid}{\sim} N(0, \sigma^2)$$

auftreten. Seinen Namen hat das Tobit-Modell von James Tobin, der es (1958) vorstellte mit dem Spezialfall $\tau_i = 0$ für $i = 1, \ldots, n$. Die Likelihood

Funktion ist in diesem Fall gegeben durch

$$L(\beta,\sigma,y^*,c) = \prod_{i=1}^{n}\left[\frac{1}{\sigma}\phi\left(\frac{y_i^*-x_i^t\beta}{\sigma}\right)\right]^{1-c_i}\left[\Phi\left(\frac{y_i^*-x_i^t\beta}{\sigma}\right)\right]^{c_i}$$
$$= \prod_{i=1}^{n}\left[\frac{1}{\sigma}\phi\left(\frac{y_i-x_i^t\beta}{\sigma}\right)\right]^{1-c_i}\left[1-\Phi\left(\frac{x_i^t\beta}{\sigma}\right)\right]^{c_i}, \qquad (77)$$

woraus

$$l(\beta,\sigma,y^*,c) = \log L(\beta,\sigma,y^*,c) = \sum_{i=1}^{n}\left\{(1-c_i)\left[-\log\sigma+\log\phi\left(\frac{y_i^*-x_i^t\beta}{\sigma}\right)\right]\right.$$
$$\left. + c_i\log\left[1-\Phi\left(\frac{x_i^t\beta}{\sigma}\right)\right]\right\}$$

folgt. Die Log-Likelihood Funktion besteht aus zwei Teilen. Der erste Teil stimmt mit der Log-Likelihood einer einfachen linearen Regression für die unzensierten Beobachtungen überein, während der zweite mit den Wahrscheinlichkeiten, dass die Beobachtungen zensiert sind, korrespondiert. Man erhält

$$l(\beta,\sigma,y^*,c) = -\frac{1}{2}\sum_{i=1}^{n}(1-c_i)\left[\log(2\pi\sigma^2)+\frac{(y_i-x_i^t\beta)^2}{\sigma^2}\right]$$
$$+\sum_{i=1}^{n}c_i\log\left[1-\Phi\left(\frac{x_i^t\beta}{\sigma}\right)\right],$$

zu dessen Maximierung Tobin (1958) eine Reparametrisierung der Form $\beta' = \beta/\sigma$ und $\sigma' = 1/\sigma$ vorschlägt, was

$$l(\beta',\sigma',y^*,c) = -\frac{1}{2}\sum_{i=1}^{n}(1-c_i)\Big[\log(2\pi)-\log(\sigma'^2)+(\sigma' y_i - x_i^t\beta')^2\Big]$$
$$+\sum_{i=1}^{n}c_i\log\Big[1-\Phi(x_i^t\beta')\Big],$$

liefert. Die Log-Likelihood Funktion weist in dieser Form keine lokalen Maxima auf (Bierens, 2004) und die ML-Schätzer $\hat{\beta}$ und $\hat{\sigma}^2$ können mittels Newton-Verfahren bestimmt werden.
Bevor wir den Ansatz des EM-Algorithmus zur ML-Schätzung diskutieren, soll auf die Bedeutung der Regressionsparameter näher eingegangen werden. Um diese richtig zu interpretieren, ist es sinnvoll, die verschiedenen Erwartungswerte, die sich im Tobit-Modell berechnen lassen, zu betrachten (vgl.

Sigelman und Zeng, 1999): Für den Erwartungswert der Zufallsvariable Y_i gilt entsprechend dem einfachen LM

$$\mathbb{E}(Y_i) = x_i^t\beta,$$

und der Effekt der j-ten erklärenden Variable ist durch

$$\frac{\partial \mathbb{E}(Y_i)}{\partial x_{ij}} = \beta_j$$

gegeben. Betrachtet man dagegen den konditionalen Erwartungswert von Y_i, gegeben Y_i ist unzensiert, erhält man mit (68)

$$\begin{aligned}\mathbb{E}(Y_i|Y_i \geq 0) &= \mu_i + \sigma\frac{\phi\left(\frac{-\mu_i}{\sigma}\right)}{1-\Phi\left(-\frac{\mu_i}{\sigma}\right)} = x_i^t\beta + \sigma\frac{\phi\left(\frac{-x_i^t\beta}{\sigma}\right)}{1-\Phi\left(\frac{-x_i^t\beta}{\sigma}\right)} \\ &= x_i^t\beta + \sigma\alpha\left(-\frac{x_i^t\beta}{\sigma}\right). \end{aligned} \tag{78}$$

Dieser Zusammenhang macht deutlich, dass ein lineares Modell ausschließlich für die positiven Y_i zu einem verzerrten Schätzer für β führt (vgl. Bierens, 2004). Für den Effekt von x_{ij} auf die unzensierte Variable Y_i folgt

$$\begin{aligned}\frac{\partial \mathbb{E}(Y_i|Y_i \geq 0)}{\partial x_{ij}} &= \beta_j + \sigma\frac{\frac{\beta_j}{\sigma}\phi'\left(\frac{x_i^t\beta}{\sigma}\right)\Phi\left(\frac{x_i^t\beta}{\sigma}\right) - \frac{\beta_j}{\sigma}\phi^2\left(\frac{x_i^t\beta}{\sigma}\right)}{\Phi^2\left(\frac{x_i^t\beta}{\sigma}\right)} \\ &= \beta_j\left[1 + \frac{-\frac{x_i^t\beta}{\sigma}\phi\left(\frac{x_i^t\beta}{\sigma}\right)\Phi\left(\frac{x_i^t\beta}{\sigma}\right) - \phi^2\left(\frac{x_i^t\beta}{\sigma}\right)}{\Phi^2\left(\frac{x_i^t\beta}{\sigma}\right)}\right] \\ &= \beta_j\left[1 - \frac{x_i^t\beta}{\sigma}\alpha\left(-\frac{x_i^t\beta}{\sigma}\right) - \alpha^2\left(-\frac{x_i^t\beta}{\sigma}\right)\right]. \end{aligned} \tag{79}$$

Schließlich kann mit $\mathbb{E}(Y_i^*)$ noch ein dritter Erwartungswert berechnet werden. Der Ausdruck

$$\mathbb{E}(Y_i^*) = \mathbb{E}(\max(Y_i, \tau_i)) = P(Y_i \geq \tau_i)\mathbb{E}(Y_i|Y_i \geq \tau_i) + P(Y_i < \tau_i)\tau_i$$

vereinfacht sich im klassischen Tobit-Modell wegen $\tau_i = 0$, $i = 1, \ldots, n$, zu

$$\mathbb{E}(Y_i^*) = P(Y_i \geq 0)\mathbb{E}(Y_i|Y_i \geq 0) \overset{(78)}{=} \Phi\left(\frac{x_i^t\beta}{\sigma}\right)\left[x_i^t\beta + \sigma\alpha\left(-\frac{x_i^t\beta}{\sigma}\right)\right]. \tag{80}$$

Dies macht deutlich, dass man auch in einem LM, in dem Werte von Y^* gleich Null als reguläre Responses betrachtet werden, verzerrte Parameterschätzer, erhält. Für den Effekt von x_{ij} auf die beobachtbare Variable Y_i^* folgt mit (79) und (80)

$$\begin{aligned}
\frac{\partial \mathbb{E}(Y_i^*)}{\partial x_{ij}} &= P(Y_i \geq 0)\frac{\partial \mathbb{E}(Y_i|Y_i \geq 0)}{\partial x_{ij}} + \mathbb{E}(Y_i|Y_i \geq 0)\frac{\partial P(Y_i \geq 0)}{\partial x_{ij}} \\
&= \Phi\left(\frac{x_i^t\beta}{\sigma}\right)\beta_j\left[1 - \frac{x_i^t\beta}{\sigma}\alpha\left(-\frac{x_i^t\beta}{\sigma}\right) - \alpha^2\left(-\frac{x_i^t\beta}{\sigma}\right)\right] \\
&\quad + \left[x_i^t\beta + \sigma\alpha\left(-\frac{x_i^t\beta}{\sigma}\right)\right]\frac{1}{\sigma}\beta_j\phi\left(\frac{x_i^t\beta}{\sigma}\right) \\
&= \Phi\left(\frac{x_i^t\beta}{\sigma}\right)\beta_j - \beta_j\frac{x_i^t\beta}{\sigma}\phi\left(-\frac{x_i^t\beta}{\sigma}\right) - \beta_j\alpha\left(-\frac{x_i^t\beta}{\sigma}\right)\phi\left(-\frac{x_i^t\beta}{\sigma}\right) \\
&\quad + \beta_j\phi\left(\frac{x_i^t\beta}{\sigma}\right)\frac{x_i^t\beta}{\sigma} + \beta_j\alpha\left(-\frac{x_i^t\beta}{\sigma}\right)\phi\left(\frac{x_i^t\beta}{\sigma}\right) \\
&= \Phi\left(\frac{x_i^t\beta}{\sigma}\right)\beta_j.
\end{aligned}$$

Dieses Ergebnis würde sich auch intuitiv vermuten lassen, da x_{ij} im Fall einer zensierten Beobachtung keinen Einfluss auf den erwarteten Wert von $Y_i^* = 0$ hat, während der Effekt für für eine unzensierte Beobachtung wie im gewöhnlichen Regressionsmodell β_j ist. Somit entspricht der Effekt auf $\mathbb{E}(Y_i^*)$ insgesamt dem Koeffizienten β_j, multipliziert mit der Wahrscheinlichkeit, dass Y_i unzensiert beobachtet wird.

Das klassische Tobit-Modell ist an die Annahme geknüpft, dass normalverteilte Zufallsvariablen vorliegen, deren Beobachtung immer nur dann möglich ist, wenn positive Werte angenommen werden. Diese Annahme kann unpassend sein, wenn die Responses in Wirklichkeit aus einer Population stammen, in denen nur positive Werte angenommen werden können. Wie wir in Abschnitt 3.4.1 am Beispiel von Konzentrationen bereits erwähnt haben, kann dann das mehrfache Auftreten des Werts Null zwar bedeuten, dass die entsprechenden Werte nicht gemessen werden konnten, die Vermutung, dass die wahren Werte negativ sind, wäre aber falsch.
Die Funktion `tobit` aus dem Paket `AER` ermöglicht in `R` Regression mit zensierten Daten (Kleiber und Zeileis, 2008, Kap. 5.4). Sie stellt dabei eine Art Interface zur Funktion `survreg` aus dem Paket `survival` (Lumley, 2004) dar, in der auch andere Verteilungen als die Normalverteilung für die Response-Variablen angenommen werden können (Kleiber und Zeileis, 2008, S. 143). Die Spezifikation erfolgt dabei nicht durch das Argument `family`, sondern

durch das Argument `dist`, das nur den Identitäts-Link erlaubt.[9]

Wir werden uns die Eigenschaften der Normalverteilung in Kapitel 4 zunutze machen, um das Schema eines EM-Algorithmus anzuführen, der die Parameterschätzung für lineare Regressionsmodelle mit links-zensierten Daten durchführt. Für das klassische Tobit-Modell lassen sich die entsprechenden Score-Gleichungen im M-Schritt einfach herleiten. Analog zu (72) erhält man mit $\mathbb{E}(Y_i) = \mu_i = x_i^t\beta$

$$Q(\beta, \sigma^2|\beta^{(k)}, \sigma^{2(k)}) = -\frac{n}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(\widetilde{y^2}_i^{(k)} - 2\mu_i\tilde{y}_i^{(k)} + \mu_i^2\right),$$

also

$$\begin{aligned}\frac{\partial Q}{\partial \beta_j} &= \frac{1}{\sigma^2}\sum_{i=1}^{n}(\tilde{y}_i^{(k)} - \mu_i)\frac{\partial \mu_i}{\partial \beta_j}, \\ &= \frac{1}{\sigma^2}\sum_{i=1}^{n}(\tilde{y}_i^{(k)} - \mu_i)x_{ij}, \quad j = 1, \ldots, p.\end{aligned}$$

Damit ist die Aktualisierung des Schätzers der Regressionsparameter äquivalent zur Berechnung von

$$\beta^{(k+1)} = \Big(\sum_{i=1}^{n} x_i x_i^t\Big)^{-1}\Big(\sum_{i=1}^{n} x_i\tilde{y}_i^{(k)}\Big), \tag{81}$$

wie man bei (Aitkin et al., 2009, S. 413) nachlesen kann. Der Schätzer $\sigma^{2(k+1)}$ wird dabei wiederum wie in (74) bestimmt.
Diese Ergebnisse lassen sich herleiten, da für die Funktion Q unter der Annahme normalverteilter Responses ohne Schwierigkeiten eine geschlossene Darstellung gefunden werden kann. Um zu sehen dass dies nicht immer so einfach ist, betrachten wir im nächsten Abschnitt eine andere Verteilung.

3.4.5 Gammaverteilte zensierte Daten

In der Analyse von Lebenszeiten spielt die Gammaverteilung eine wichtige Rolle. Wenn eine Lebensdauer aus r exponentialverteilten Abschnitten mit

[9] Das Paket `survival` bietet auch einen nicht-parametrischen Ansatz an, bei dem die Funktion `survfit` zur Approximation der gemeinsamen Verteilungsfunktion zensierter und unzensierter Beobachtungen den *Kaplan-Meier-Schätzer* (Kaplan und Meier, 1958) berechnet.

demselben Erwartungswert θ besteht, dann ist sie als Summe davon gammaverteilt mit Erwartungswert $r\theta$ und Dispersionsparameter $1/r$. Wenn dabei zensierte Beobachtungen auftreten, möchte man zur Parameterschätzung wie in Abschnitt 3.4.3 den EM-Algorithmus verwenden. Für $Y \sim \Gamma(\mu, \nu)$ ist die Dichtefunktion aus (11) gegeben durch

$$f(y;\mu,\nu) = \exp\left(-\frac{\nu}{\mu}y + \nu\log\nu - \nu\log\mu + (\nu-1)\log y - \log\Gamma(\nu)\right).$$

Für eine links-zensierte Beobachtung folgt damit hier

$$\begin{aligned}\mathbb{E}\big(\log f(Y;\mu,\nu)\big|Y \le \tau;\mu^{(k)},\nu^{(k)}\big) &= -\nu\log\mu + \nu\log\nu - \log\Gamma(\nu)\\ -\frac{\nu}{\mu}\mathbb{E}\big(Y\big|Y \le \tau;\mu^{(k)},\nu^{(k)}\big) &+ (\nu-1)\mathbb{E}\big(\log Y\big|Y \le \tau;\mu^{(k)},\nu^{(k)}\big).\end{aligned}$$

Die Durchführung des E-Schritts erfordert also die Berechnung der konditionalen Erwartungswerte von Y und $\log Y$. Verglichen mit der Normalverteilung bedeutet dies einen erhöhten Rechenaufwand, wie wir dem Ansatz von Aitkin et al. (2009, Kap. 6.9) folgend, sehen werden.[10] Wir schreiben die Dichtefunktion zuerst um zu

$$f(y;\mu,\nu) = \exp\left(-\frac{\nu}{\mu}y\right)\left(\frac{\nu}{\mu}\right)^{\nu} y^{\nu-1}\frac{1}{\Gamma(\nu)}.$$

Für die Verteilungsfunktion gilt dann

$$\begin{aligned}F(y;\mu,\nu) &= \int_0^y \exp\left(-\frac{\nu}{\mu}u\right)\left(\frac{\nu}{\mu}\right)^{\nu} u^{\nu-1}\frac{1}{\Gamma(\nu)}\,du\\ &= \int_0^y \exp\left(-\frac{\nu}{\mu}u\right)\left(\frac{\nu}{\mu}u\right)^{\nu-1}\frac{\nu}{\mu\,\Gamma(\nu)}\,du\\ &= \int_0^{\frac{\nu}{\mu}y} \exp(-z)\,z^{\nu-1}\frac{1}{\Gamma(\nu)}\,dz\\ &= F_\nu(\nu y/\mu),\end{aligned}$$

wobei $F_\nu(\cdot)$ die Verteilungsfunktion der *standardisierten Gammaverteilung* mit Erwartungswert ν und Dispersionsparameter $1/\nu$ ist. Der konditionale Erwartungswert von Y einer links-zensierten Beobachtung ist dann definiert

[10] Im Kontext der Survial Analysis basiert der Ansatz dort auf rechts-zensierten Beobachtungen, die Berechnung der Erwartungswerte erfolgt aber für links-zensierte Daten analog.

durch

$$\begin{aligned}
\mathbb{E}\big(Y\big|Y \le \tau\big) &= \frac{1}{F_\nu(\nu\tau/\mu)} \int_0^\tau u f(u;\mu,\nu)\, du \\
&= \frac{1}{F_\nu(\nu\tau/\mu)} \int_0^\tau \exp\left(-\frac{\nu}{\mu}u\right) \left(\frac{\nu}{\mu}u\right)^\nu \frac{1}{\Gamma(\nu)}\, du \\
&= \frac{\mu}{F_\nu(\nu\tau/\mu)} \int_0^\tau \exp\left(-\frac{\nu}{\mu}u\right) \left(\frac{\nu}{\mu}u\right)^\nu \frac{\nu}{\mu\,\Gamma(\nu+1)}\, du \\
&= \frac{\mu}{F_\nu(\nu\tau/\mu)} \int_0^{\frac{\nu}{\mu}\tau} \exp(-z)\, z^\nu \frac{1}{\Gamma(\nu+1)}\, dz \\
&= \mu \frac{F_{\nu+1}(\nu\tau/\mu)}{F_\nu(\nu\tau/\mu)}.
\end{aligned}$$

Um den konditionalen Erwartungswert von $\log Y$ zu bestimmen, betrachtet man die entsprechende Momentenerzeugende Funktion. Mit dieser erhält man

$$\begin{aligned}
M_{\log Y}(t) &= \mathbb{E}\big(\mathrm{e}^{t\log Y}\big|Y \le \tau\big) = \mathbb{E}\big(Y^t\big|Y \le \tau\big) \\
&= \frac{1}{F_\nu(\nu\tau/\mu)} \int_0^\tau u^t f(u;\mu,\nu)\, du \\
&= \frac{1}{F_\nu(\nu\tau/\mu)} \int_0^\tau \exp\left(-\frac{\nu}{\mu}u\right) \left(\frac{\nu}{\mu}\right)^\nu u^{\nu+t-1} \frac{1}{\Gamma(\nu)}\, du \\
&= \frac{1}{F_\nu(\nu\tau/\mu)} \frac{\mu^t}{\nu^t} \frac{\Gamma(\nu+t)}{\Gamma(\nu)} \int_0^\tau \exp\left(-\frac{\nu}{\mu}u\right) \left(\frac{\nu}{\mu}u\right)^{\nu+t-1} \frac{\nu}{\mu\,\Gamma(\nu+t)}\, du \\
&= \frac{1}{F_\nu(\nu\tau/\mu)} \frac{\mu^t}{\nu^t} \frac{\Gamma(\nu+t)}{\Gamma(\nu)} \int_0^{\frac{\nu}{\mu}\tau} \exp(-z)\, \frac{z^{\nu+t-1}}{\Gamma(\nu+t)}\, dz \\
&= \frac{\mu^t}{\nu^t} \frac{\Gamma(\nu+t)}{\Gamma(\nu)} \frac{F_{\nu+t}(\nu\tau/\mu)}{F_\nu(\nu\tau/\mu)}.
\end{aligned}$$

Die Kumulantenerzeugende Funktion ist damit von der Form

$$\begin{aligned}
K_{\log Y}(t) = \log M_{\log Y}(t) = {} & t(\log\mu - \log\nu) + \log\Gamma(\nu+t) \\
& + \log F_{\nu+t}(\nu\tau/\mu) + c(\mu,\nu,\tau),
\end{aligned}$$

und es folgt

$$\mathbb{E}\big(\log Y|Y \le \tau\big) = \frac{\partial}{\partial t} K_{\log Y}(t)\Big|_{t=0} = \log\mu - \log\nu + \psi(\nu) + \frac{a(0\,|\,\nu\tau/\mu)}{F_\nu(\nu\tau/\mu)},$$

wobei $\psi(t) = \frac{\partial}{\partial t}\log\Gamma(t)$ die Digamma-Funktion bezeichnet und

$$a(t\,|\,\nu\tau/\mu) = \frac{\partial}{\partial t} F_{\nu+t}(\nu\tau/\mu).$$

Um den Ausdruck $a(0 \mid \nu\tau/\mu)$ auszuwerten, wendet man numerische Differentiation an, also

$$a(0 \mid \nu\tau/\mu) = \big[F_{\nu+\delta}(\nu\tau/\mu) - F_{\nu-\delta}(\nu\tau/\mu)\big]/2\delta$$

für einen kleinen Wert von δ.
Im E-Schritt des EM-Algorithmus ersetzt man für jede zensierte Beobachtung y_i^* durch

$$\mu_i \frac{F_{\nu+1}(\nu\tau_i/\mu_i)}{F_\nu(\nu\tau_i/\mu_i)}$$

und $\log y_i^*$ durch

$$\log \mu_i - \log \nu + \psi(\nu) + \frac{a(0 \mid \nu\tau_i/\mu_i)}{F_\nu(\nu\tau_i/\mu_i)}.$$

ausgewertet jeweils in den aktuellen Parameterschätzern $\mu_i^{(k)}$ und $\nu^{(k)}$. Mit diesen erwarteten Daten kann die Lösung der Score-Gleichungen im M-Schritt unter der Annahme $\mu_i = g^{-1}(x_i^t\beta)$ wiederum auf eine ML-Schätzung eines GLM zurückgeführt werden.

4 Endliche Mischungen mit links-zensierten Daten

4.1 Motivation

Basierend auf den Ergebnissen von Kapitel 3 wollen wir uns nun der speziellen Modellklasse der endlichen Mischmodelle für Daten mit links-zensierten Beobachtungen widmen. Eine in der Praxis auftretende Situation, die dies motiviert, stellen beispielsweise Schadstoffkonzentrationen dar, deren Messung nur dann exakt erfolgen kann, wenn die Konzentrationen eine für das Messinstrument charakteristische Nachweisbarkeitsgrenze übersteigen. Typischerweise möchte man bei der Stichprobenerhebung zusätzliche Daten gewinnen, die in einem Regressionsmodell die Rolle der erklärenden Variablen übernehmen können. In der Praxis ist die Möglichkeit, diese vollständig zu erfassen, nicht immer gegeben, was ein Modell mit einer nicht-beobachtbaren Größe nahelegt. Häufig ist man daran interessiert, ob sich die Population in mehrere Gruppen unterteilten lässt, in denen die Verteilungen der Responses unterschiedliche Eigenschaften aufweisen. Das dazu passende Modell ist jenes einer endlichen Mischung aus Kapitel 3.2, in dem eine latente Zufallsvariable dazu dient, die gemeinsame Verteilung der Population als diskrete Mischung einer endlichen Anzahl von Komponenten zu beschreiben. Die Besonderheiten, die dabei auftreten, wenn ein Teil der Beobachtungen links-zensiert ist, sollen nun in diesem Kapitel diskutiert werden, wobei sich die folgenden Ausführungen an Booth und Friedl (2005) orientieren.

4.2 Das Modell

Für das Modell der diskreten endlichen Mischung von L Komponenten nehmen wir an, dass die Indikatorvariable Z eine diskret verteilte Zufallsvariable ist mit Wahrscheinlichkeitsmassen $\pi = (\pi_1, \ldots, \pi_L)$, so dass

$$P(Z = g) = \pi_g, \quad g = 1, \ldots, L.$$

Gegeben Z, sei die konditionale Wahrscheinlichkeits- oder Dichtefunktion von Y bestimmt durch

$$f(y|Z = g) = f_g(y; \lambda_g).$$

Handelt es sich bei Y um eine zu messende Konzentration, dann ist es sinnvoll, vorauszusetzen, dass eine Mischung von stetigen Verteilungen vorliegt, weshalb wir die f_g in diesem Kapitel ausschließlich als Dichtefunktionen bezeichnen werden. Wie in Kapitel 3.2 nehmen wir an, dass wir nicht beobachten können, aus welcher Komponente der Mischung Y stammt. Zusätzlich

bleibt nun auch der wahre Wert von Y unbeobachtet, wenn er einen bekannten Schwellwert τ nicht überschreitet. Der beobachtbare Teil der Daten ist dann durch Paare (Y^*, C) gegeben, wobei $Y^* = \max(Y, \tau)$ und $C = I(Y \leq \tau)$ der Indikator dafür ist, ob die Beobachtung zensiert ist.
Ausgehend von diesem Modell nehmen wir an, dass ein Datensatz aus n unabhängigen Paaren (y_i^*, c_i) besteht, die der vorangegangenen Beschreibung entsprechend generiert werden. Der Schwellwert darf dabei von Beobachtung zu Beobachtung verschieden sein, was für Werte von Konzentrationen in der Praxis eine Folge davon sein kann, dass Messgeräte mit unterschiedlichen Nachweisbarkeitsgrenzen verwendet werden. Bezeichne τ_i den zur i-ten Beobachtung gehörenden Schwellwert und sei $\lambda = (\lambda_1, \ldots, \lambda_L)$, dann gilt für die zu maximierende Likelihood Funktion einer Stichprobe vom Umfang n gemäß (46) und (64)

$$\begin{aligned} L(\lambda, \pi, y^*, c) &= \prod_{i=1}^{n} \sum_{g=1}^{L} \pi_g \big[f_g(y_i^*; \lambda_g) \big]^{1-c_i} \big[F_g(y_i^*; \lambda_g) \big]^{c_i} \\ &= \prod_{i=1}^{n} \sum_{g=1}^{L} \pi_g \Big[(1-c_i) f_g(y_i; \lambda_g) + c_i F_g(\tau_i; \lambda_g) \Big], \end{aligned} \tag{82}$$

weshalb sich die marginale Log-Likelihood Funktion schreiben lässt als

$$l(\lambda, \pi, y^*, c) = \sum_{i=1}^{n} \log \left(\sum_{g=1}^{L} \pi_g \Big[(1-c_i) f_g(y_i; \lambda_g) + c_i F_g(\tau_i; \lambda_g) \Big] \right). \tag{83}$$

4.3 Der Monte Carlo EM-Algorithmus

Zur Maximierung von (83) betrachten wir die Log-Likelihood Funktion der vollständigen, zum Teil nicht-beobachteten Daten (y_i, z_i). Analog zu Abschnitt 3.2.2 können wir Indikatorvariablen Z_{ig} definieren durch

$$Z_{ig} = \begin{cases} 1 & \text{wenn } y_i \text{ aus Komponente } g \text{ stammt } \Leftrightarrow Z_i = g, \\ 0 & \text{sonst,} \end{cases}$$

mit denen

$$\log f(y, z; \lambda, \pi) = \sum_{i=1}^{n} \sum_{g=1}^{L} z_{ig} \Big(\log \pi_g + \log f_g(y_g; \lambda_g) \Big) \tag{84}$$

folgt. Sei $(\lambda^{(k)}, \pi^{(k)})$ der Schätzer des Parametervektors, der nach k Iterationen berechnet wurde. Gemäß der Iterationsvorschrift des EM-Algorithmus

(Abschnitt 3.1.1) ist der Schätzer $(\lambda^{(k+1)}, \pi^{(k+1)})$ definiert als der Vektor, der die Zielfunktion

$$\begin{aligned} Q(\lambda, \pi | \lambda^{(k)}, \pi^{(k)}) &= \mathbb{E}\Big(\log f(Y, Z; \lambda, \pi) \Big| y^*, c; \lambda^{(k)}, \pi^{(k)} \Big) \\ &= \mathbb{E}\left(\sum_{i=1}^{n} \sum_{g=1}^{L} Z_{ig} \Big(\log \pi_g + \log f_g(Y_i; \lambda_g) \Big) \right| y^*, c; \lambda^{(k)}, \pi^{(k)} \Bigg) \end{aligned}$$

maximiert, die sich ebenso schreiben lässt als

$$Q(\lambda, \pi | \lambda^{(k)}, \pi^{(k)}) = \sum_{i=1}^{n} \mathbb{E}\left(\log \pi_{Z_i} + \log f_{Z_i}(Y_i; \lambda_{Z_i}) \Big| y^*, c; \lambda^{(k)}, \pi^{(k)} \right). \quad (85)$$

Die a posteriori Wahrscheinlichkeiten, dass y_i zu Komponente g gehört, sind gegeben durch

$$\begin{aligned} P(Z_i = g | y_i^*, c_i; \lambda, \pi) &= \frac{\pi_g \Big((1 - c_i) f_g(y_i; \lambda_g) + c_i F_g(\tau_i; \lambda_g) \Big)}{\sum_{l=1}^{L} \pi_l \Big((1 - c_i) f_l(y_i; \lambda_l) + c_i F_l(\tau_i; \lambda_l) \Big)} \\ &= \begin{cases} \dfrac{\pi_g f_g(y_i; \lambda_g)}{\sum_{l=1}^{L} \pi_l f_l(y_i; \lambda_l)} & \text{wenn } c_i = 0 \\ \dfrac{\pi_g F_g(\tau_i; \lambda_g)}{\sum_{l=1}^{L} \pi_l F_l(\tau_i; \lambda_l)} & \text{wenn } c_i = 1 \end{cases} \\ &= \begin{cases} p_{ig}(\lambda, \pi) & \text{wenn } c_i = 0, \\ q_{ig}(\lambda, \pi) & \text{wenn } c_i = 1. \end{cases} \end{aligned} \quad (86)$$

Für unzensierte Beobachtungen stimmt dies mit den Gewichten ω_{ig} aus Abschnitt 3.2.2 ein, während bei zensierten Beobachtungen die Auswertung der Verteilungsfunktionen an der Stelle des Schwellwerts τ_i für die Berechnung relevant ist. Im Gegensatz zu Abschnitt 3.2.2 erfordert der E-Schritt hier nicht nur die Berechnung der a posteriori Wahrscheinlichkeiten (87), die den konditionalen Erwartungswerten der Z_{ig} entsprechen. Für zensierte Beobachtungen liegt mit $\log f_g(Y_i; \lambda_g)$ ebenfalls eine unbekannte Größe vor. Es

gilt

$$\begin{aligned}
&\mathbb{E}\Big(\log f(Y_i, Z_i; \lambda, \pi)\Big| Z_i = g, y_i^*, c_i; \lambda^{(k)}, \pi^{(k)}\Big) \\
&= \mathbb{E}\Big(\log \pi_g + \log f_g(Y_i; \lambda_g)\Big| y_i^*, c_i; \lambda_g^{(k)}\Big) \\
&= \begin{cases} \log \pi_g + \log f_g(y_i; \lambda_g) & \text{wenn } c_i = 0, \\ \log \pi_g + H_{ig}(\lambda_g, \lambda_g^{(k)}) & \text{wenn } c_i = 1, \end{cases}
\end{aligned} \tag{87}$$

wobei

$$\begin{aligned}
H_{ig}(\lambda_g, \lambda_g^{(k)}) &= \mathbb{E}\Big(\log f_g(Y_i; \lambda_g)\Big| Y_i \le \tau_i; \lambda_g^{(k)}\Big) \\
&= \frac{1}{F_g(\tau_i; \lambda_g^{(k)})} \int_{-\infty}^{\tau_i} \log f_g(y; \lambda_g) f_g(y; \lambda_g^{(k)})\, dy.
\end{aligned} \tag{88}$$

Wir erhalten mit den Gleichungen (85)-(88)

$$\begin{aligned}
&Q(\lambda, \pi | \lambda^{(k)}, \pi^{(k)}) = \sum_{i=1}^{n} \mathbb{E}\left(\log \pi_{Z_i} + \log f_{Z_i}(Y_i; \lambda_{Z_i})\Big| y^*, c; \lambda^{(k)}, \pi^{(k)}\right) \\
&= \sum_{i=1}^{n} \sum_{g=1}^{L} P(Z_i = g | y_i^*, c_i; \lambda^{(k)}, \pi^{(k)}) \mathbb{E}\Big(\log f(Y_i, Z_i; \lambda, \pi)\Big| Z_i = g, y_i^*, c_i; \lambda^{(k)}, \pi^{(k)}\Big) \\
&= \sum_{i=1}^{n} \sum_{g=1}^{L} \Big[(1 - c_i) p_{ig}^{(k)} + c_i q_{ig}^{(k)}\Big] \log \pi_g \\
&+ \sum_{i=1}^{n} \sum_{g=1}^{L} \Big[(1 - c_i) p_{ig}^{(k)} \log f_g(y_i; \lambda_g) + c_i q_{ig}^{(k)} H_{ig}(\lambda_g, \lambda_g^{(k)})\Big]
\end{aligned} \tag{89}$$

mit $p_{ig}^{(k)} = p_{ig}(\lambda^{(k)}, \pi^{(k)})$ und $q_{ig}^{(k)} = q_{ig}(\lambda^{(k)}, \pi^{(k)})$. Da diese Funktionen in $(\lambda^{(k)}, \pi^{(k)})$ ausgewertet sind, stellen sie im E-Schritt bekannte Größen dar. Die einzigen Terme in (89), bei deren Auswertung Schwierigkeiten auftreten können, sind die $H_{ig}(\lambda_g, \lambda_g^{(k)})$, die nicht für jede Dichtefunktion f_g eine geschlossene Darstellung besitzen. Allgemein sollen sie deshalb durch die Monte Carlo-Mittel

$$\tilde{H}_{ig}(\lambda_g, \lambda_g^{(k)}) = \frac{1}{S} \sum_{s=1}^{S} \log f_g(y_{isg}^{(k)}; \lambda_g) \tag{90}$$

ersetzt werden. Dabei bezeichnet S den Monte Carlo-Simulationsumfang, der umso größer gewählt werden muss, je genauer die Approximation sein soll. Für $S \to \infty$ wird das Gesetz der großen Zahlen wirksam, und $\tilde{H}_{ig}(\lambda_g, \lambda_g^{(k)})$

strebt gegen $H_{ig}(\lambda_g, \lambda_g^{(k)})$. Die Beobachtungen $y_{i1g}^{(k)}, \ldots, y_{iSg}^{(k)}$ sind Responses vom linken Rand der Komponente g, deren Simulation den Algorithmus zu einem *Monte Carlo EM-Algorithmus (MCEM-Algorithmus)* macht.

Der MCEM-Algorithmus gehört, wie in Abschnitt 3.1.3 erwähnt, zu den stochastischen Varianten des EM-Algorithmus (vgl. Roche, 2003). Die allgemein zugrundeliegende Idee ist dabei, die Funktion $Q(\theta|\theta^{(k)})$ im E-Schritt durch

$$\tilde{Q}(\theta|\theta^{(k)}) = \frac{1}{S}\sum_{s=1}^{S} \log f(y, w_s^{(k)}; \theta) \tag{91}$$

zu ersetzen, wobei $w_1^{(k)}, \ldots, w_S^{(k)}$ eine zufallsgenerierte Stichprobe aus der durch den aktuellen Parameterschätzer $\theta^{(k)}$ beschriebenen, konditionalen Verteilung der unbeobachteten Daten ist. Im M-Schritt wird dann die rechte Seite von (91) maximiert (vgl. Wei und Tanner, 1990).
Im Hinblick auf die Konvergenz des MCEM-Algorithmus spielt die Wahl von S eine wesentliche Rolle. Natürlich ist die Approximation für $Q(\theta|\theta^{(k)})$ umso genauer, je größer S ist. Mit einem sehr großen Wert von S zu starten, kann aber unnötig und vor allem ineffizient sein, wenn $\theta^{(k)}$ noch weit entfernt vom ML-Schätzer $\hat{\theta}$ ist. Dagegen sollte die Approximation in der Nähe des ML-Schätzers sehr gut sein. Wei und Tanner (1990) empfehlen, das Konvergenzverhalten des Algorithmus im Auge zu behalten und auf eine Phase der Stabilisierung zu warten. Aufgrund der Zufälligkeit von $w_1^{(k)}, \ldots, w_S^{(k)}$ muss der Wert der marginalen Log-Likelihood Funktion nicht monoton wachsen, sondern kann einer zufälligen Fluktuation unterliegen. Beobachtet man dies im Laufe der Iterationen, kann man es zum Anlass nehmen, den Simulationsumfang S zu erhöhen. Im Zusammenhang damit entsteht der für stochastische Varianten des EM-Algorithmus typische Nebeneffekt, dass sie eine geringere Tendenz aufweisen, gegen lokale Maxima zu konvergieren (Roche, 2003). Intensiver mit Konvergenzverhalten von MCEM-Algorithmen haben sich Sherman, Yu-Yun und Dalal (1999) auseinandergesetzt.

Durch die Verwendung von $H_{ig} \approx \tilde{H}_{ig}$ im hier betrachteten Modell lässt

sich (89) in eine Form ähnlich zu (91) bringen:

$$\begin{aligned}
Q(\lambda,\pi|\lambda^{(k)},\pi^{(k)}) &\approx \tilde{Q}(\lambda,\pi|\lambda^{(k)},\pi^{(k)}) \\
&= \sum_{i=1}^{n}\sum_{g=1}^{L}\Big[(1-c_i)p_{ig}^{(k)} + c_i q_{ig}^{(k)}\Big]\log\pi_g \\
&+ \sum_{i=1}^{n}\sum_{g=1}^{L}\Big[(1-c_i)p_{ig}^{(k)}\log f_g(y_i;\lambda_g) + c_i q_{ig}^{(k)}\frac{1}{S}\sum_{s=1}^{S}\log f_g(y_{isg}^{(k)};\lambda_g)\Big] \\
&= \frac{1}{S}\sum_{s=1}^{S}\Bigg(\sum_{i=1}^{n}\sum_{g=1}^{L}\Big[(1-c_i)p_{ig}^{(k)} + c_i q_{ig}^{(k)}\Big]\log\pi_g \\
&+ \sum_{i=1}^{n}\sum_{g=1}^{L}\Big[(1-c_i)p_{ig}^{(k)}\log f_g(y_i;\lambda_g) + c_i q_{ig}^{(k)}\log f_g(y_{isg}^{(k)};\lambda_g)\Big]\Bigg). \quad (92)
\end{aligned}$$

Wir bemerken dabei aber, dass die Monte Carlo-Simulation nur für die zensierten Beobachtungen durchgeführt wird, während für die Indikatorvariablen Z_i, die ebenfalls nicht beobachtet werden, entsprechend dem E-Schritt des Standard-EM-Algorithmus die konditionalen Erwartungswerte $\mathbb{E}(Z_{ig}) = (1-c_i)p_{ig} + c_i q_{ig}$ in die Zielfunktion einfließen. Die Stichproben $y_{i1g}^{(k)},\ldots,y_{iSg}^{(k)}$ lassen sich mittles Inversionsmethode generieren, also

$$y_{isg}^{(k)} = F_g^{-1}(U_s F_g(\tau_i;\lambda_g^{(k)});\lambda_g^{(k)}) \text{ mit } U_1,\ldots,U_S \overset{iid}{\sim} \mathcal{U}[0,1],$$

wobei $\mathcal{U}[0,1]$ die Gleichverteilung auf dem Intervall $[0,1]$ bezeichnet.

Damit ist der E-Schritt abgeschlossen und die Iteration wird mit dem M-Schritt fortgesetzt. Die Maximierung von (92) bezüglich der Mischwahrscheinlichkeiten π ist dabei wieder unabhängig von den übrigen, unbekannten Parametern λ durchführbar. Mit der Restriktion $\sum_{g=1}^{L}\pi_g = 1$ folgt analog zu Abschnitt 3.2.2

$$\pi_g^{(k+1)} = \frac{1}{n}\sum_{i=1}^{n}\tilde{\omega}_{ig}^{(k)}, \quad (93)$$

wobei die Gewichte hier durch

$$\tilde{\omega}_{ig}^{(k)} = (1-c_i)p_{ig}^{(k)} + c_i q_{ig}^{(k)} \quad (94)$$

gegeben sind. Wie die Maximierung bezüglich λ durchgeführt wird, hängt von den getroffenen Verteilungsannahmen ab.

4.4 Links-zensierte Daten in Mischungen von GLMs

Wir nehmen nun wie in Abschnitt 3.2.4 an, dass ein Zusammenhang zwischen dem konditionalen Erwartungswert $\mu_{ig} = \mathbb{E}(Y_i|Z_i = g)$ und einem linearen Prädiktor besteht, gegeben durch

$$g(\mu_{ig}) = x_i^t \beta_g$$

mit der bekannten Linkfunktion $g(\cdot)$. Die Response-Variablen Y_i sollen außerdem einer Mischung von Verteilungen folgen, in der die Dichte- oder Wahrscheinlichkeitsfunktion der Komponente g von der Form

$$f_g(y_i; \lambda_g) = \exp\left(\frac{y_i\theta_{ig} - b(\theta_{ig})}{\phi_g} + c(y_i, \phi_g)\right) \tag{95}$$

ist. Die Beziehung des kanonischen Parameters θ_{ig} zum konditionalen Erwartungswert μ_{ig} ist bestimmt durch $\mu_{ig} = b'(\theta_{ig})$ (vgl. Abschnitt 2.1.1). Um den M-Schritt zu komplettieren ist die Funktion $Q(\lambda, \pi|\lambda^{(k)}, \pi^{(k)})$ also bezüglich $\lambda_g = (\beta_g, \phi_g)$, $g = 1, \ldots, L$, zu maximieren. Der dafür relevante Teil ist, wie die Gleichung (92) zeigt, von der Form

$$\tilde{Q}_\lambda^{(k)} = \frac{1}{S}\sum_{s=1}^{S}\sum_{i=1}^{n}\sum_{g=1}^{L}\Big[(1-c_i)p_{ig}^{(k)}\log f_g(y_i;\lambda_g) + c_i q_{ig}^{(k)}\log f_g(y_{isg}^{(k)};\lambda_g)\Big]. \tag{96}$$

Mit $y_{i0g}^{(k)} \equiv y_i^*$ für $k \geq 0$, $i = 1, \ldots, n$ und $g = 1, \ldots, L$ lässt sich dies schreiben als

$$\begin{aligned}\tilde{Q}_\lambda^{(k)} &= \sum_{i=1}^{n}\sum_{g=1}^{L}\sum_{s=1}^{S}\left[\frac{(1-c_i)p_{ig}^{(k)}}{S}\log f_g(y_i;\lambda_g) + \frac{c_i q_{ig}^{(k)}}{S}\log f_g(y_{isg}^{(k)};\lambda_g)\right] \\ &= \sum_{i=1}^{n}\sum_{g=1}^{L}\sum_{s=0}^{S}\omega_{isg}^{*(k)}\log f_g(y_{isg}^{(k)};\lambda_g) \end{aligned} \tag{97}$$

mit den Gewichten

$$\omega_{isg}^{*(k)} = (1-c_i)\delta_{s0}p_{ig}^{(k)} + (1-\delta_{s0})c_i\frac{q_{ig}^{(k)}}{S},$$

wobei $\delta_{rs} = I[r = s]$ gilt. Somit folgt unter der Annahme, dass es sich bei f_g um Dichte- oder Wahrscheinlichkeitsfunktionen einer Verteilung der Exponentialfamilie (95) handelt,

$$\tilde{Q}_\lambda^{(k)} = \sum_{i=1}^{n}\sum_{g=1}^{L}\sum_{s=0}^{S}\omega_{isg}^{*(k)}\left(\frac{y_{isg}^{(k)}\theta_{ig} - b(\theta_{ig})}{\phi_g} + c(y_{isg}^{(k)}, \phi_g)\right). \tag{98}$$

Dies entspricht der Summe von Likelihood-Termen eines GLM mit Responses $y_{isg}^{(k)}$ und Gewichten $\omega_{isg}^{*(k)}$, womit sich die Maximierung von Q bezüglich (β, ϕ) auf eine gewichtete ML-Schätzung für ein GLM reduziert, dessen Datensatz nun aus L Blöcken besteht mit bis zu Sn Zeilen. In einer Implementierung lässt sich diese Zahl verringern, da die Zeilen mit $\omega_{isg}^{*(k)} = 0$ nicht berücksichtigt werden müssen. Unter der Annahme komponentenspezifischer Regressionsparameter $\beta_1, \ldots, \beta_L$ und Dispersionsparameter $\phi_1, \ldots, \phi_L$ zerfällt (98) in die Terme

$$L_g^{(k)} = L^{(k)}(\beta_g, \phi_g) = \sum_{i=1}^{n} \sum_{s=0}^{S} \omega_{isg}^{*(k)} \left(\frac{y_{isg}^{(k)} \theta_{ig} - b(\theta_{ig})}{\phi_g} + c(y_{isg}^{(k)}, \phi_g) \right) \tag{99}$$

und der M-Schritt besteht aus L gewichteten Modellanpassungen für die Komponenten.

Ein Spezialfall liegt vor, wenn das Modell außer einer Konstante keine erklärenden Variablen enthält. Es gilt dann $\mu_g \equiv \mu_{ig}$, und der Wert von $\mu_g^{(k+1)}$, der (99) maximiert, lässt sich explizit angeben. Wegen

$$\begin{aligned} \frac{\partial L_g^{(k)}}{\partial \mu_g} &= \sum_{i=1}^{n} \sum_{s=0}^{S} \omega_{isg}^{*(k)} \left(\frac{y_{isg}^{(k)} - b'(\theta_g)}{\phi_g} \right) \frac{\partial \mu_g}{\partial \theta_g} \\ &= \sum_{i=1}^{n} \sum_{s=0}^{S} \omega_{isg}^{*(k)} \frac{y_{isg}^{(k)} - \mu_g}{\phi_g V(\mu_g)} \end{aligned}$$

erhält man in diesem Fall

$$\mu_g^{(k+1)} = \frac{\sum_{i=1}^{n} \sum_{s=0}^{S} \omega_{isg}^{*(k)} y_{isg}^{(k)}}{\sum_{i=1}^{n} \sum_{s=0}^{S} \omega_{isg}^{*(k)}}.$$

4.4.1 Schätzung der Dispersionsparameter

Für $\phi_g^{(k+1)}$ lässt sich im M-Schritt ebenfalls nur in speziellen Situationen eine geschlossene Darstellung angeben. Bei normalverteilten Responses erhält man mit $\mu_{ig} = \theta_{ig}$ und $\phi_g = \sigma_g^2$

$$\begin{aligned} \frac{\partial L_g^{(k)}}{\partial \sigma_g^2} &= \sum_{i=1}^{n} \sum_{s=0}^{S} \omega_{isg}^{*(k)} \left(\frac{1}{2\sigma_g^4} (y_{isg}^{(k)} - \mu_{ig})^2 - \frac{1}{2\sigma_g^2} \right) \\ &= \frac{1}{2\sigma_g^4} \sum_{i=1}^{n} \sum_{s=0}^{S} \omega_{isg}^{*(k)} (y_{isg}^{(k)} - \mu_{ig})^2 - \frac{1}{2\sigma_g^2} \sum_{i=1}^{n} \sum_{s=0}^{S} \omega_{isg}^{*(k)}, \end{aligned}$$

also

$$\sigma_g^{2(k+1)} = \frac{\sum_{i=1}^{n}\sum_{s=0}^{S}\omega_{isg}^{*(k)}(y_{isg}^{(k)} - \mu_{ig}^{(k+1)})^2}{\sum_{i=1}^{n}\sum_{s=0}^{S}\omega_{isg}^{*(k)}}, \tag{100}$$

was der gewichteten Fehlerquadratsumme entspricht. Nimmt man dagegen gammaverteilte Responses an, ist der Dispersionsschätzer für $\nu_g = 1/\phi_g$ als die Lösung von

$$\sum_{i=1}^{n}\sum_{s=0}^{S}\omega_{isg}^{*(k)}\left[\log\frac{y_{isg}^{(k)}}{\mu_{ig}^{(k+1)}} - \frac{y_{isg}^{(k)} - \mu_{ig}^{(k+1)}}{\mu_{ig}^{(k+1)}} + \log\nu_g - \psi(\nu_g)\right] = 0 \tag{101}$$

definiert, wobei $\psi(\nu_g)$ wie in Abschnitt 3.4.5 die Digamma-Funktion bezeichnet. Um einen Schätzer für den aktualisierten Dispersionsparameter $\phi_g^{(k+1)}$ allgemein mit geringem Aufwand zu bestimmen, kann man die Fehlerquadrate in (100) durch die quadrierten Pearson Residuen (Abschnitt 2.2.4) ersetzen. Der MCEM-Algorithmus, der in der Funktion `glmm.lc` (Kapitel 5) implementiert ist, wird die Dispersionsschätzer auf diese Art aktualisieren.

4.4.2 Der deterministische EM-Algorithmus für normalverteilte Responses

Die Verwendung der Monte Carlo-Simulation in Abschnitt 4.3 ist dadurch motiviert, dass die Zielfunktion Q die Terme H_{ig} enthält (88). Dabei handelt es sich um die konditionalen Erwartungswerte von $\log f_g(y_i;\lambda_g)$, gegeben die Beobachtung ist zensiert. Für normalverteilte Responses ist die Verwendung der Monte Carlo-Simulation nicht zwingend nötig, da man für H_{ig} in diesem Fall mit den Resultaten aus Abschnitt 3.4.3 eine geschlossene Darstellung erhält. Es bietet sich dadurch die Möglichkeit, den MCEM-Algorithmus mit einem deterministischen EM-Algorithmus zu vergleichen, dessen Schema nun betrachtet wird:
Der für die Maximierung von Q bezüglich $\lambda = (\beta, \phi)$ relevante Teil der Zielfunktion kann in die Form

$$\begin{aligned} Q_\lambda^{(k)} &= \sum_{i=1}^{n}\sum_{g=1}^{L}\Big[(1-c_i)p_{ig}^{(k)}\log f_g(y_i;\lambda_g) + c_i q_{ig}^{(k)} H_{ig}(\lambda_g, \lambda_g^{(k)})\Big] \\ &= \sum_{i=1}^{n}\sum_{g=1}^{L}\tilde{\omega}_{ig}^{(k)}\Big[(1-c_i)\log f_g(y_i;\lambda_g) + c_i H_{ig}(\lambda_g, \lambda_g^{(k)})\Big] \end{aligned} \tag{102}$$

gebracht werden mit den in (94) definierten Gewichten $\tilde{\omega}_{ig}^{(k)}$. Unter der Annahme $Y_i|g \sim N(\mu_{ig}, \sigma_g^2)$, $g = 1, \ldots, L$, folgt für (88)

$$\begin{aligned} H_{ig}(\lambda_g, \lambda_g^{(k)}) &= \mathbb{E}\Big(\log f_g(Y_i; \lambda_g)\Big| Y_i \leq \tau_i; \lambda_g^{(k)}\Big) \\ &= \mathbb{E}\left(-\frac{1}{2\sigma_g^2}(Y_i - \mu_{ig})^2 - \frac{1}{2}\log 2\pi\sigma_g^2 \middle| Y_i \leq \tau_i; \lambda_g^{(k)}\right), \end{aligned} \tag{103}$$

und wie in Abschnitt 3.4.3 gezeigt, lässt sich dieser Erwartungswert analytisch berechnen. Mit den analog zu (71) definierten Variablen

$$\begin{aligned} \tilde{y}_{ig}^{(k)} &= (1 - c_i)y_i + c_i\mathbb{E}(Y_i|Y_i \leq \tau_i; \lambda_g^{(k)}) \\ &= (1 - c_i)y_i + c_i\left[\mu_{ig}^{(k)} + \sigma_g^{(k)}\alpha\left(\frac{\tau_i - \mu_{ig}^{(k)}}{\sigma_g^{(k)}}\right)\right], \end{aligned} \tag{104}$$

$$\begin{aligned} \widetilde{y^2}_{ig}^{(k)} &= (1 - c_i)y_i^2 + c_i\mathbb{E}(Y_i^2|Y_i \leq \tau_i; \lambda_g^{(k)}) \\ &= (1 - c_i)y_i^2 + c_i\left[\mu_{ig}^{2(k)} + \sigma_g^{2(k)} + \sigma_g^{(k)}(\mu_{ig}^{(k)} + \tau_i)\alpha\left(\frac{\tau_i - \mu_{ig}^{(k)}}{\sigma_g^{(k)}}\right)\right] \end{aligned} \tag{105}$$

und $\mu_{ig}^{(k)} = g^{-1}(x_i^t\beta_g^{(k)})$ folgt für (102)

$$Q_\lambda^{(k)} = \sum_{i=1}^{n}\sum_{g=1}^{L}\tilde{\omega}_{ig}^{(k)}\Big[-\frac{1}{2}\log 2\pi\sigma_g^2 - \frac{1}{2\sigma_g^2}\left(\widetilde{y^2}_{ig}^{(k)} - 2\mu_{ig}\tilde{y}_{ig}^{(k)} + \mu_{ig}^2\right)\Big]. \tag{106}$$

Wegen

$$\frac{\partial Q_\lambda}{\partial\beta_{gj}} = \sum_{i=1}^{n}\tilde{\omega}_{ig}^{(k)}\frac{\tilde{y}_{ig}^{(k)} - \mu_{ig}}{\sigma_g^2}\frac{x_{ij}}{g'(\mu_{ig})}, \quad g = 1, \ldots, L,\ j = 1, \ldots, p$$

ist die Maximierung bezüglich β_g äquivalent zur ML-Schätzung eines GLM mit Responses $\tilde{y}_{ig}^{(k)}$ und Gewichten $\tilde{\omega}_{ig}^{(k)}$. Nimmt man an, dass keine echte Mischung vorliegt, sondern $L = 1$ gilt, vereinfachen sich die Gleichungen, wegen $\tilde{\omega}_{ig}^{(k)} = 1$, $i = 1, \ldots, n$, und die Verwendung des Identitäts-Links $g(\mu) = \mu$ führt auf die Aktualisierung der Parameterschätzer im Tobit-Modell (81) zurück.

Um den M-Schritt zu komplettieren, muss noch der Schätzer für $\sigma^{2(k+1)}$ berechnet werden. Dazu schreiben wir die Zielfunktion um zu

$$\begin{aligned} Q_\lambda^{(k)} = &\sum_{i=1}^{n}\sum_{g=1}^{L}\tilde{\omega}_{ig}^{(k)}\Big[-\frac{1}{2}\log 2\pi\sigma_g^2 \\ &- \frac{1}{2\sigma_g^2}\Big((\tilde{y}_{ig}^{(k)} - \mu_{ig})^2 + c_i\sigma_g^{2(k)}\big(1 + \tilde{\tau}_{ig}^{(k)}\alpha(\tilde{\tau}_{ig}^{(k)}) - \alpha^2(\tilde{\tau}_{ig}^{(k)})\big)\Big)\Big], \end{aligned} \tag{107}$$

mit $\tilde{\tau}_{ig}^{(k)} = (\tau_i - \mu_{ig}^{(k)})/\sigma_g^{(k)}$ und wobei $\alpha(\cdot)$ wie in Abschnitt 3.4.3 definiert ist. Daraus resultiert

$$\begin{aligned}\frac{\partial Q_\lambda^{(k)}}{\partial \sigma_g^2} =& \sum_{i=1}^{n} \tilde{\omega}_{ig}^{(k)} \Big[-\frac{1}{2\sigma_g^2} \\ &+ \frac{1}{2\sigma_g^4} \Big((\tilde{y}_{ig}^{(k)} - \mu_{ig})^2 + c_i \sigma_g^{2(k)} \big(1 + \tilde{\tau}_{ig}^{(k)} \alpha(\tilde{\tau}_{ig}^{(k)}) - \alpha^2(\tilde{\tau}_{ig}^{(k)})\big) \Big) \Big] \\ =& -\frac{1}{2\sigma_g^2} \sum_{i=1}^{n} \tilde{\omega}_{ig}^{(k)} + \frac{1}{2\sigma_g^4} \sum_{i=1}^{n} \Big[\tilde{\omega}_{ig}^{(k)} (\tilde{y}_{ig}^{(k)} - \mu_{ig})^2 \\ &+ c_i q_{ig}^{(k)} \sigma_g^{2(k)} \left(1 + \tilde{\tau}_{ig}^{(k)} \alpha(\tilde{\tau}_{ig}^{(k)}) - \alpha^2(\tilde{\tau}_{ig}^{(k)})\right) \Big] \end{aligned} \tag{108}$$

und wir erhalten den aktualisierten Schätzer der Varianz in Komponente g durch

$$\begin{aligned}\sigma^{2(k+1)} =& \sum_{i=1}^{n} \Big[\tilde{\omega}_{ig}^{(k)} (\tilde{y}_{ig}^{(k)} - \mu_{ig}^{(k+1)})^2 \\ &+ c_i q_{ig}^{(k)} \sigma_g^{2(k)} \left(1 + \tilde{\tau}_{ig}^{(k)} \alpha(\tilde{\tau}_{ig}^{(k)}) - \alpha^2(\tilde{\tau}_{ig}^{(k)})\right) \Big] \Big/ \sum_{i=1}^{n} \tilde{\omega}_{ig}^{(k)}. \end{aligned} \tag{109}$$

Das nächste Kapitel wird sich der Implementierung des MCEM-Algorithmus in der `R`-Funktion `glmm.lc` widmen. Für die Normalverteilung wird darin zusätzlich der in diesem Abschnitt beschriebene deterministische EM-Algorithmus zur Verfügung stehen, der einen Vergleich mit dem MCEM-Algorithmus möglich macht. Wie wir in Abschnitt 3.4.5 gesehen haben, existiert auch für die Gammaverteilung ein Ansatz, mit dem man den Erwartungswert der konditionalen Log-Likelihood Funktion einer zensierten Beobachtung berechnen kann und somit auch in diesem Fall ohne Simulation zu einer geschlossenen Darstellung der H_{ig} gelangt. Dies erfordert, wie dort gezeigt, jedoch mehr Rechenaufwand (etwa numerisches Differenzieren der Verteilungsfunktion), weshalb für die Gammaverteilung vorerst kein deterministischer EM-Algorithmus implementiert wurde.

5 Die R-Funktion glmm.lc

5.1 Überblick

Nachdem nun die Theorie der Mischmodelle mit links-zensierten Daten behandelt wurde, widmet sich dieses Kapitel der Implementierung des MCEM-Algorithmus in der R-Funktion `glmm.lc`. Der Name `glmm.lc` soll darauf hinweisen, dass die Parameterschätzung eines Generalisierten Linearen Mischmodells (engl.: **G**eneralized **L**inear **M**ixed **M**odel) durchgeführt wird, unter Berücksichtigung der Tatsache, dass ein Teil der Beobachtungen linkszensiert (engl.: **l**eft **c**ensored) ist. Die Funktion ermöglicht eine ML-Schätzung unter der Annahme, dass die Response-Variablen entweder normal- oder gammaverteilt sind, wobei verschiedene Link-Funktionen verwendet werden können. Die Art der Implementierung erlaubt es, den Programmcode dahingehend zu erweitern, dass auch andere Verteilungen der Exponentialfamilie zugänglich werden. Diese Arbeit beschränkt sich jedoch darauf, dass von den in Abschnitt 2.1 beschriebenen Verteilungen die beiden stetigen Mitglieder der Exponentialfamilie verfügbar sind, um dem Datentyp von Konzentrationen gerecht zu werden. Für die Normalverteilung bietet die Funktion zusätzlich die Option an, anstelle des MCEM-Algorithmus den deterministischen EM-Algorithmus (Abschnitt 4.4.2) zu nützen.
Dieses Kapitel wird einen Überblick über die Argumente und Arbeitsweise der Funktion `glmm.lc` geben und sich einigen Besonderheiten des Algorithmus widmen, um anschließend sein Verhalten anhand von Simulationsstudien zu untersuchen. Der gesamte Programmcode ist im Anhang B zu finden.

Die Implementierung der Funktion `glmm.lc` orientiert sich an einem Schema, welches sich für Modellierungsfunktionen in R zu einem Standard entwickelt hat.

Argumente der Funktion `glmm.lc`

```
glmm.lc (formula, family = gaussian, data, threshold, L = 2,
          random, na.action, prstart, s2start, proc = "MCEM",
          simul = "standard", maxsim = 500, s2update = "EM",
          spike.protect = 1e-05, EMmaxit = 500, eps = 1e-07,
          trace = 0, ...)
```

Detaillierte Erläuterungen zu Argumenten wie `formula` und `family`, die bei derartigen Funktionen zur Spezifikation von Modellformel bzw. Verteilungsannahme vorgesehen sind, findet man bei Weisberg und Fox (2010, Kap. 4 und 5) und Steinkellner (2012). Hier sollen in erster Linie einige Argumente näher betrachtet werden, die für unser Modell von besonderer Bedeutung sind.

5.2 Der Algorithmus

Das Argument `proc` erlaubt die Auswahl des Algorithmus. Der Default-Wert `"MCEM"` steht dabei für den MCEM-Algorithmus und ist für beide Verteilungen möglich. Die Verwendung des deterministischen EM-Algorithmus wird für den Fall, dass normalverteilte Responses angenommen werden, durch die Spezifikation `proc = "detEM"` erreicht. Je nachdem, welche der beiden Prozeduren verwendet wird, ruft `glmm.lc` entweder die Funktion `lcMCEM.fit` oder die Funktion `lcdetEM.fit` intern auf und übergibt die jeweils relevanten Argumente.
Dazu zählt in jedem Fall das Argument `threshold`, das für den Vektor der Schwellwerte steht. Die Funktion `glmm.lc` verlangt bei einem Stichprobenumfang n dafür einen Vektor $\tau = (\tau_1, \ldots, \tau_n)^t$, der zu jeder Response die Information enthält, ab welchem Wert die Realisierung unzensiert und somit exakt beobachtet wird. Wird die Funktion ohne eine Angabe für `threshold` ausgeführt, werden alle Beobachtungen als unzensiert betrachtet.
Die Anzahl der Komponenten des Mischmodells wird durch `L` bestimmt und mit der Spezifikation von `random` wird festgelegt, zu welchen erklärenden Variablen komponentenspezifische Regressionsparameter geschätzt werden sollen (vgl. Steinkellner, 2012). Fehlt die Spezifikation, so werden zu jeder erklärenden Variable der Modellformel L Parameter geschätzt.
Die Option `s2update` ist nur dann relevant, wenn der deterministische EM-Algorithmus gewählt wird. Sie bezieht sich auf die Aktualisierung der Dispersionsschätzer im M-Schritt, die entweder auf die in Abschnitt 4.4.2 hergeleitete Methode (`s2update = "EM"`) durchgeführt werden kann, oder dem an das Mischmodell angepassten Schema von Lawless (1982, S.224) folgt (`s2update = "LA"`). Bei den Simulationsstudien in den Abschnitten 5.3.1 und 5.3.2 werden wir nur den Fall `s2update = "EM"` betrachten, da die Ergebnisse der beiden Methoden asymptotisch gesehen keine nennenswerten Unterschiede aufweisen.
Im Gegensatz dazu sind alle Parameter, die im Zusammenhang mit der Monte Carlo-Simulation stehen, nur für den MCEM-Algorithmus von Bedeutung (siehe: Abschnitt 5.2.2).

5.2.1 Startwerte

Beide EM-Algorithmen benötigen Startwerte als Ausgangspunkt für die Parameterschätzung. Die Prozedur, die diese generiert, ist ähnlich jener, die die Funktion `alldist` (Abschnitt 3.2.4) verwendet. Ausgehend vom vorliegenden Datensatz wird zuerst der Vektor der beobachtbaren Werte $y^* = (y_1^*, \ldots, y_n^*)^t$ konstruiert mit $y_i^* = \max(y_i, \tau_i)$. Anschließend wird die Funktion `glm.fit`

aufgerufen und führt die ML-Schätzung für das GLM

$$g(\mu_i^*) = \eta_i^*, \quad i = 1, \ldots, n \tag{110}$$

durch, wobei $\mu_i^* = \mathbb{E}(y_i^*)$ ist und in den Prädiktoren η_i^* alle Variablen der Modellformel enthalten sind. Unter der Annahme, dass für die wahren konditionalen linearen Prädiktoren der vollständigen Daten y

$$\eta_{ig} = \eta_i^* + \zeta_g, \quad i = 1, \ldots, n, \quad g = 1, \ldots, L$$

gilt, können daraus mit geeigneter Wahl der ζ_g adäquate Startwerte $\eta_{ig}^{(0)}$ und $\mu_{ig}^{(0)}$ konstruiert werden. Für normalverteilte Responses wählen wir dazu $\zeta_g = z_g\sigma_0$, wobei

$$\sigma_0 = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\Big(g(y_i^*) - \eta_i^*\Big)^2} \tag{111}$$

gilt und die Werte von $z_1, \ldots, z_l$ den Massestellen bei Verwendung der Gauss-Quadratur zur Approximation der Standard-Normalverteilung entsprechen. Unter der Annahme, dass die Response-Variablen gammaverteilt sind, kann diese Prozedur dazu führen, dass unzulässige Prädiktoren bestimmt werden, weshalb in diesem Fall

$$\eta_{ig}^{(0)} = \eta_i^* \left(1 + \frac{\sigma_0}{|\eta_i^*|}\right)^{z_g}, \quad i = 1, \ldots, n, \quad g = 1, \ldots, L$$

gewählt wird. Die Massestellen z_g können in R von der Funktion `gqz` bestimmt werden, die genau wie die Funktion `alldist` zum Paket npmlreg (Einbeck und Hinde, 2006) gehört. Die Ausnahme ist der Spezialfall $L = 1$, wo `gqz` nicht aufgerufen wird und $z_g = 0$ gewählt wird. Für die Startwerte der konditionalen Erwartungswerte erhält man $\mu_{ig}^{(0)} = g^{-1}(\eta_{ig}^{(0)})$.
Für die Mischwahrscheinlichkeiten $\pi_1, \ldots, \pi_L$ kann man Startwerte explizit angeben, indem man dem Argument `prstart` einen entsprechenden Vektor der Länge L zuweist. Andernfalls wird $\pi_g^{(0)} = 1/L$ gesetzt für $g = 1, \ldots, L$. Anfangswerte für die gruppenspezifischen Dispersionsparameter $\phi_1, \ldots, \phi_L$ können ebenfalls angegeben werden (`s2start`). Verzichtet man darauf, berechnet der Algorithmus $\phi_g^{(0)} = \hat{\phi}^*/L$ für $g = 1, \ldots, L$, wobei $\hat{\phi}^*$ der Dispersionsschätzer zu (110) ist.

5.2.2 Monte Carlo-Simulation

Das Argument `simul` dient beim MCEM-Algorithmus der Bestimmung der Simulationsgröße S in jedem E-Schritt. Wie in Abschnitt 4.3 erwähnt wird,

kann es von Vorteil sein, zu Beginn einen niedrigen Wert zu wählen, diesen aber im Lauf der Iterationen zu vergrößern. Zur Bestimmung von $S^{(k)}$, der Simulationsgröße zum Zeitpunkt der k-ten Iteration, sind deshalb mehrere Einstellungen möglich:

- Default-Einstellung: Der MCEM-Algorithmus beginnt mit einer Simulationsgröße von $S^{(1)} = 10$ und erhöht diesen Wert nach jeder 20. Iteration um 10, also $S^{(k)} = 10\,(\lfloor k/20 \rfloor + 1)\,,\ k \geq 1$.
- konstante Größe: Die Simulationsgröße wird zu keinem Zeitpunkt erhöht und es gilt $S^{(k)} = S$, $k \geq 1$ (z.B. `simul = 100`).
- automatische Anpassung: Der MCEM-Algorithmus beginnt mit einer Simulationsgröße, die dem Anteil der zensierten Beobachtungen angepasst ist. Der Wert von S wird nicht nur in regelmäßigen Abständen erhöht, sondern zusätzlich auch dann, wenn der Algorithmus eine Verkleinerung des Wertes der Log-Likelihood Funktion feststellt (`simul = "automatic"`).

Unabhängig von der Spezifikation von `simul` ist bei Verwendung der Funktion `glmm.lc` eine obere Schranke für die Simulationsgröße $S^{(k)}$ anzugeben, die nicht überschritten werden soll. Dazu wird dem Argument `maxsim` ein Wert $S_{\max}$ zugewiesen und als Simulationsgröße wird dann in jeder Iteration der Wert $\min(S_{\max}, S^{(k)})$ gewählt.

Bei Verwendung des deterministischen EM-Algorithmus werden die in diesem Abschnitt beschriebenen Argumente ignoriert.

5.2.3 Weitere Optionen zur Kontrolle des Algorithmus

In Abschnitt 3.2.5 wurde darauf hingewiesen, dass bei Mischungen mit verschiedenen Dispersionsparametern Likelihood Spikes auftreten können. Um den Algorithmus davor zu schützen, ermöglicht die Funktion `glmm.lc` mit dem Argument `spike.protect` die Wahl eines Werts ϕ_0, den die Schätzer $\phi_g^{(k)}$, $g = 1, \ldots, L$, $k \geq 1$, nicht unterschreiten dürfen. Die Standardeinstellung dafür lautet $\phi_0 = 10^{-5}$.
Ein weiterer wichtiger Punkt ist die Definition einer Abbruchbedingung. Anstatt das Lack Of Progress Kriterium (Abschnitt 3.1.1) zu verwenden, wird die relative Änderung der Log-Likelihood Funktion betrachtet und der Algorithmus bricht ab, wenn

$$\frac{\left| l(\lambda^{(k)}, \pi^{(k)}, y^*, c) - l(\lambda^{(k-1)}, \pi^{(k-1)}, y^*, c) \right|}{\left| l(\lambda^{(k-1)}, \pi^{(k-1)}, y^*, c) \right|} < \varepsilon \tag{112}$$

erfüllt ist, wobei der gewünschte Wert für ε dem Parameter `eps` zugewiesen wird.
Schließlich besteht die Möglichkeit, die Zahl an Iterationen, die der gewählte EM-Algorithmus durchlaufen kann, nach oben hin zu beschränken (`EMmaxit`).

5.3 Simulationen

5.3.1 Mischung von Normalverteilungen

Wir beginnen die Untersuchung des Verhaltens des MCEM-Algorithmus am Beispiel einer Mischung von zwei Komponenten mit normalverteilten Responses. Im Gegensatz zu den Beispielen aus Abschnitt 3.2.3 erlauben wir hier unterschiedliche Varianzen und erhalten vorerst ohne zusätzliche erklärende Variablen eine Mischverteilung mit

$$Y_i \overset{iid}{\sim} \pi_1 N(\mu_1, \sigma_1^2) + (1 - \pi_1) N(\mu_2, \sigma_2^2), \quad i = 1, \ldots, n.$$

Für die erste Studie treffen wir folgende Wahl von Parametern:

μ_1	μ_2	σ_1^2	σ_2^2	π_1
0	4	1.5	0.5	0.8

Der Graph, der die Dichtefunktion der resultierenden Mischung beschreibt, ist in Abbildung 3 dargestellt und lässt zwei optisch gut unterscheidbare Komponenten erkennen. Wir generieren nun daraus eine Zufallsstichprobe $y_1, \ldots, y_n$ für $n = 100$ und setzen $\tau_i = 0$ für alle i, sodass alle Werte, die links der roten vertikalen Linie liegen, nicht beobachtet werden. Für eine Realisierung aus Komponente 1 beträgt die Wahrscheinlichkeit, zensiert zu sein genau 1/2, während diese Wahrscheinlichkeit für Beobachtungen aus Komponente 2 geringer als 10^{-8} ist. Wir erwarten damit insgesamt, dass etwa 40 Beobachtungen zensiert sind. Wie sich dies auf den Vektor der beobachteten Daten y^* auswirken kann, sehen wir am Beispiel einer einzelnen Stichprobe in Abbildung 4.

Wir testen nun den MCEM-Algorithmus und vergleichen ihn mit dem deterministischen EM-Algorithmus, indem wir die zuvor beschriebene Prozedur $R = 200$ mal wiederholen und für jede der generierten Stichproben beide EM-Algorithmen eine Parameterschätzung durchführen lassen. Der Wert der Monte Carlo-Simulationsgröße soll hierbei konstant $S = 100$ für alle Iterationen sein. Beide Algorithmen brechen ab, wenn die Bedingung (112) für $\varepsilon = 10^{-7}$ erreicht ist, oder die maximal erlaubte Zahl von 500 EM-Iterationen erreicht wird.
Die Resultate sind in folgender Tabelle dargestellt, wobei die Abkürzung SE

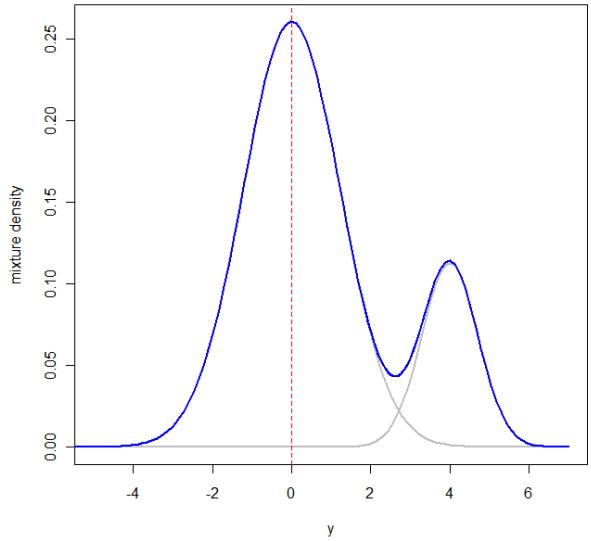

Abbildung 3: Dichte der Mischverteilung für die Simulationsstudie.

wie auch in den folgenden Abschnitten für den Standardfehler steht.

det. EM	$\hat{\mu}_1$	$\hat{\mu}_2$	$\hat{\sigma}_1^2$	$\hat{\sigma}_2^2$	$\hat{\pi}_1$	l	It.
Minimum	-0.612	2.630	0.396	0.033	0.585	-172.5	13
1. Quartil	-0.156	3.860	1.153	0.311	0.770	-161.6	34
Median	-0.021	4.010	1.527	0.430	0.807	-157.2	49
Mittelwert	-0.022	3.998	1.569	0.502	0.802	-156.2	58
3. Quartil	0.095	4.170	1.870	0.610	0.844	-151.0	73
Maximum	0.597	4.751	4.363	2.298	0.930	-132.8	231
SE	0.195	0.271	0.604	0.300	0.055	7.7	35
MCEM	$\hat{\mu}_1$	$\hat{\mu}_2$	$\hat{\sigma}_1^2$	$\hat{\sigma}_2^2$	$\hat{\pi}_1$	l	It.
Minimum	-0.610	2.642	0.395	0.033	0.586	-172.5	12
1. Quartil	-0.154	3.860	1.139	0.310	0.770	-161.6	56
Median	-0.017	4.014	1.508	0.431	0.807	-157.2	79
Mittelwert	-0.022	3.996	1.567	0.504	0.802	-156.2	102
3. Quartil	0.093	4.173	1.879	0.604	0.844	-151.0	124
Maximum	0.598	4.751	4.374	2.279	0.930	-133.0	367
SE	0.195	0.272	0.607	0.300	0.055	7.7	71
wahre Parameter	0.000	4.000	1.500	0.500	0.800		

Wir stellen fest, dass die Parameterschätzer des MCEM-Algorithmus sich im Mittel nicht wesentlich von denen des deterministischen EM-Algorithmus unterscheiden. Die Zahl der benötigten Iterationen stellt den einzigen we-

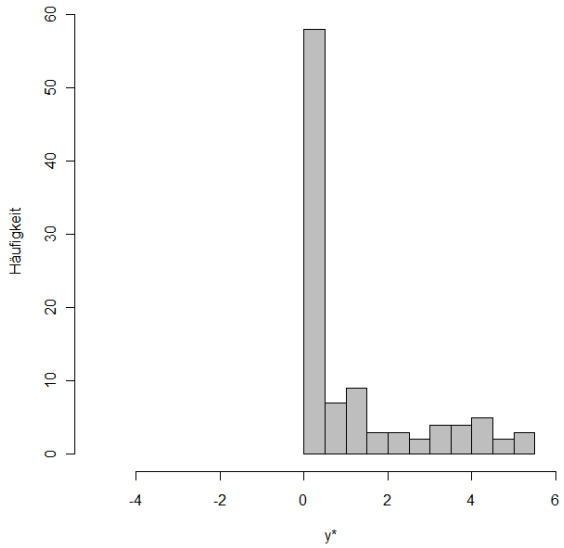

Abbildung 4: Häufigkeiten der beobachteten Daten aus der Population dargestellt in Abbildung 3.

sentlichen Unterschied dar. Während der deterministische EM-Algorithmus im Mittel 58 Iterationen benötigt, braucht der MCEM-Algorithmus 102 Iterationen. Die Schranke 500 wurde in keinem Rechenvorgang erreicht.
Obwohl im Mittel nur 60 der 100 Datenpunkte exakt zu beobachten sind, unterscheiden sich die arithmetischen Mittel der $R = 200$ Schätzer von den wahren Parametern nur geringfügig.

Um zu sehen, ob sich dies ändert, wenn der Anteil der zensierten Beobachtungen zunimmt, führen wir eine zweite Studie durch, in der wir $\tau_i = 0.8$ für $i = 1, \ldots, n$ wählen. Wir erwarten somit, dass der Anteil der zensierten Beobachtungen auf 0.59 wächst. Die folgende Tabelle zeigt die Resultate, basierend auf $R = 200$ Wiederholungen, in denen beide Algorithmen das Konvergenzkriterium erfüllen. Um diese zu erhalten, waren insgesamt 205 Wiederholungen nötig, da der MCEM-Algorithmus die maximal erlaubten

500 Iterationen fünf Mal überschritten hat:

det. EM	$\hat{\mu}_1$	$\hat{\mu}_2$	$\hat{\sigma}_1^2$	$\hat{\sigma}_2^2$	$\hat{\pi}_1$	l	It.
Minimum	-1.481	2.642	0.041	0.075	0.562	-147.8	16
1. Quartil	-0.205	3.862	0.863	0.304	0.763	-129.7	58
Median	0.045	4.014	1.379	0.418	0.801	-123.3	86
Mittelwert	-0.006	3.992	1.669	0.471	0.798	-123.2	108
3. Quartil	0.235	4.141	2.320	0.568	0.832	-116.7	131
Maximum	0.586	4.848	5.234	1.957	0.921	-99.1	499
SE	0.339	0.265	1.107	0.266	0.053	9.7	81
MCEM	$\hat{\mu}_1$	$\hat{\mu}_2$	$\hat{\sigma}_1^2$	$\hat{\sigma}_2^2$	$\hat{\pi}_1$	l	It.
Minimum	-1.297	2.706	0.071	0.075	0.575	-147.8	14
1. Quartil	-0.194	3.850	0.833	0.304	0.761	-129.7	76
Median	0.045	4.001	1.308	0.422	0.799	-123.3	121
Mittelwert	0.003	3.982	1.595	0.482	0.796	-123.2	146
3. Quartil	0.229	4.137	2.089	0.575	0.832	-116.7	185
Maximum	0.574	4.848	5.097	1.935	0.921	-99.1	480
SE	0.310	0.272	1.031	0.288	0.053	9.7	95
wahre Parameter	0.000	4.000	1.500	0.500	0.800		

Wie in der ersten Studie liefern auch jetzt beide Algorithmen sehr ähnliche Resultate. Die mittlere Anzahl der Iterationen ist beim deterministischen EM-Algorithmus von 58 auf 108, beim MCEM-Algorithmus von 102 auf 146 gestiegen. Da durch die Verschiebung der Threshold-Werte auf $\tau_i = 0.8, i = 1, \ldots, 100$, der Anteil der nicht-beobachteten Realisierungen erhöht wurde, ist dies nicht überraschend. Für Realisierungen, die aus der zweiten Komponente stammen, wirkt sich dies jedoch kaum aus, da die Wahrscheinlichkeit, unter den Schwellwert zu fallen immer noch weniger als 10^{-5} beträgt. Wir sehen aus diesem Grund anhand der Standardfehler, dass sich die Verteilung der Schätzer $\hat{\mu}_2$ und $\hat{\sigma}_2^2$ gegenüber der ersten Studie kaum verändert hat, während die zur ersten Komponente gehörenden Schätzer $\hat{\mu}_1$ und $\hat{\sigma}_1^2$ nun stärkeren Schwankungen unterliegen.
Die Abbildungen 5-7 stellen die Werte der errechneten Parameterschätzer sowie die Werte der Log-Likelihood Funktion gegenüber und ermöglichen so einen genaueren Vergleich der beiden Algorithmen. Während diese Resultate nur in Ausnahmefällen deutliche Unterschiede aufweisen, ist eine Übereinstimmung bei den benötigten Iterationen nicht erkennbar, wie Abbildung 8 zeigt.

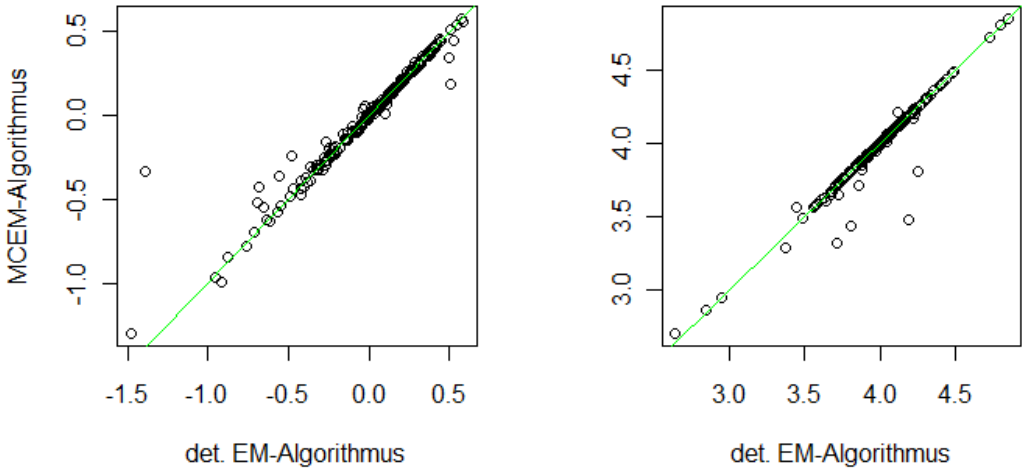

Abbildung 5: Vergleich der 200 Schätzer für μ_1 (links) und μ_2 (rechts).

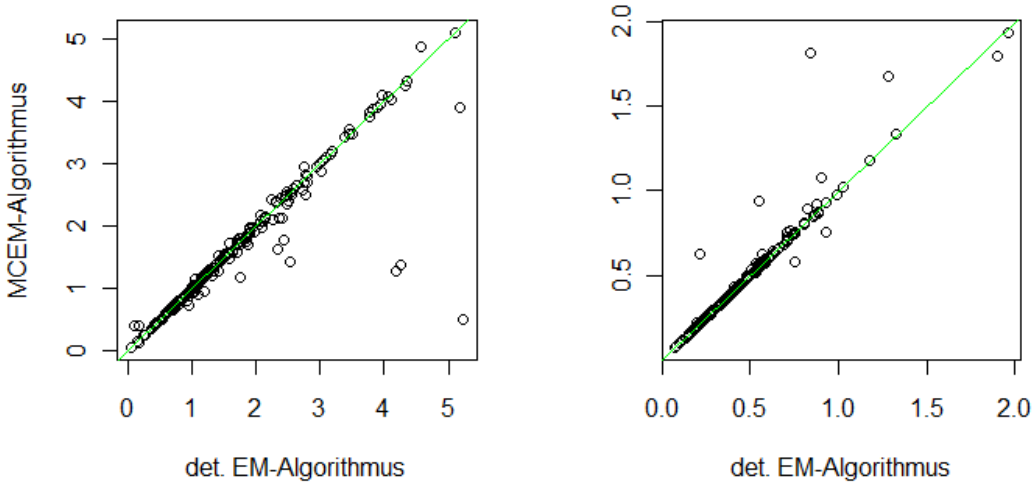

Abbildung 6: Vergleich der 200 Schätzer für σ_1^2 (links) und σ_2^2 (rechts).

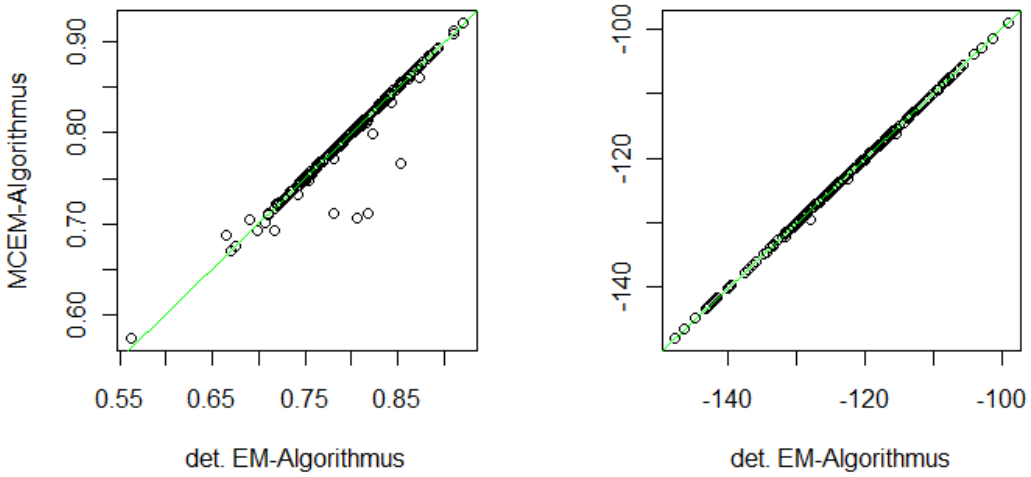

Abbildung 7: Vergleich der 200 Schätzer für π_1 (links) und Werte der Log-Likelihood Funktion (rechts).

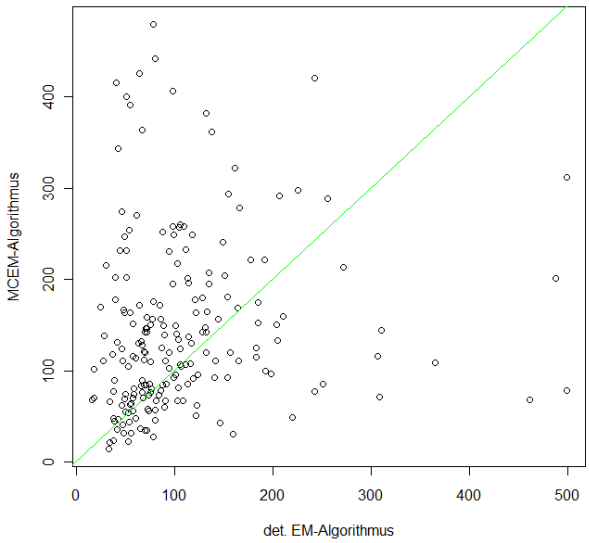

Abbildung 8: Benötigte Iterationen.

5.3.2 Normalverteilte Responses in einer Mischung von Regressionsmodellen

Um das Verhalten des MCEM-Algorithmus bei Modellen mit erklärenden Variablen zu studieren, definieren wir eine Variable x und betrachten das lineare Modell $\mu_g(x) = \beta_{g1} + \beta_{g2}x$ für $g \in \{1, 2\}$. Für die folgende Studie setzen wir $x = (0.1, 0.2, \ldots, 1)$ und verwenden 10 Kopien des Vektors x, um eine Stichprobe vom Umfang $n = 100$ zu generieren, sodass

$$Y_i \overset{ind}{\sim} \pi_1 N(\mu_1(x_i), \sigma_1^2) + (1 - \pi_1) N(\mu_2(x_i), \sigma_2^2), \quad i = 1, \ldots, 100.$$

Wir wählen die Parameterwerte

β_{11}	β_{21}	β_{12}	β_{22}	σ_1^2	σ_2^2	π_1
-0.4	0.5	1	3	0.5	0.25	0.75

wobei π_1, π_2 wie zuvor die Mischwahrscheinlichkeiten sind und $\pi_2 = 1 - \pi_1$ gilt. Weiters soll $\tau_i = 0$ für $i = 1, \ldots, 100$ gelten, was zur Folge hat, dass etwa ein Drittel der Beobachtungen zensiert ist. Abbildung 9 zeigt ein Beispiel für einen generierten Vektor y zur erklärenden Variable x. Es stellt sich heraus, dass die meisten zensierten Beobachtungen aus der Komponente 1 stammen, wo negative Werte wesentlich häufiger auftreten. Es werden $R = 200$ Wiederholungen der Simulation durchgeführt, und zu jedem generierten Response-Vektor werden Parameterschätzer von beiden EM-Algorithmen errechnet. Die MC-Simulationsgröße im MCEM-Algorithmus wird dabei wieder konstant $S = 100$ in allen Iterationen gewählt und auch die Abbruchbedingung ist wie zuvor definiert. Die folgende Tabelle zeigt die Resultate:

det. EM	$\hat{\beta}_{11}$	$\hat{\beta}_{21}$	$\hat{\beta}_{12}$	$\hat{\beta}_{22}$	$\hat{\sigma}_1^2$	$\hat{\sigma}_2^2$	$\hat{\pi}_1$	l	It.
Minimum	-1.387	-1.954	-0.477	1.803	0.098	0.033	0.520	-141.5	15
1. Quartil	-0.512	0.158	0.693	2.621	0.370	0.156	0.704	-130.1	27
Median	-0.360	0.447	0.921	3.021	0.473	0.215	0.743	-126.3	36
Mittelwert	-0.367	0.426	0.927	3.069	0.499	0.241	0.739	-126.2	43
3. Quartil	-0.183	0.711	1.146	3.458	0.585	0.290	0.791	-122.8	53
Maximum	0.422	1.453	2.413	6.401	1.319	1.212	0.908	-108.8	154
SE	0.284	0.436	0.398	0.599	0.203	0.142	0.071	5.9	23
MCEM	$\hat{\beta}_{11}$	$\hat{\beta}_{21}$	$\hat{\beta}_{12}$	$\hat{\beta}_{22}$	$\hat{\sigma}_1^2$	$\hat{\sigma}_2^2$	$\hat{\pi}_1$	l	It.
Minimum	-1.412	-1.975	-0.474	1.803	0.010	0.033	0.522	-141.5	13
1. Quartil	-0.518	0.154	0.695	2.621	0.368	0.156	0.704	-130.1	61
Median	-0.360	0.449	0.920	3.024	0.476	0.214	0.743	-126.3	115
Mittelwert	-0.366	0.420	0.927	3.077	0.503	0.239	0.740	-126.2	146
3. Quartil	-0.177	0.710	1.149	3.456	0.592	0.287	0.791	-122.8	201
Maximum	0.421	1.454	2.444	6.427	1.540	1.216	0.909	-108.8	497
SE	0.283	0.446	0.398	0.615	0.214	0.142	0.071	5.9	107
wahre P.	-0.400	0.500	1.000	3.000	0.500	0.250	0.750		

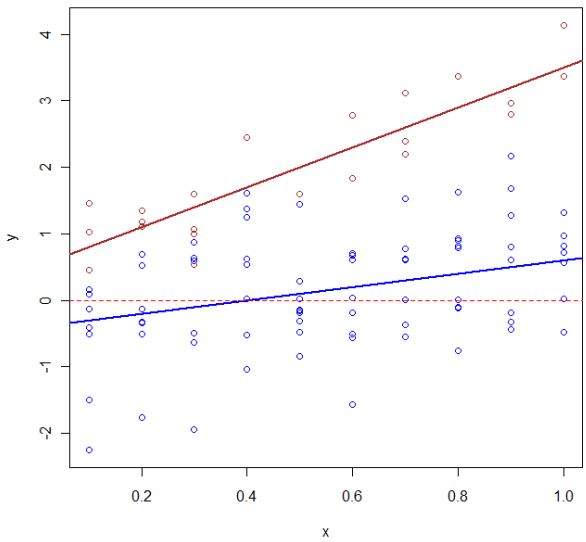

Abbildung 9: Beispiel einer Stichprobe aus der Mischung der Modelle $\mu_1 = -0.4 + x$ (braun) und $\mu_2 = 0.5 + 3x$ (blau) mit eingezeichneten Thresholds (strichliert).

Elf Mal wurden die maximal erlaubten 500 Iteration vom MCEM-Algorithmus erreicht und die entsprechenden Ergebnisse durch ML-Schätzungen für neue Datensätze ersetzt.
Um zu sehen, wie der Anteil der zensierten Beobachtungen das Verhalten der Algorithmen beeinflusst, betrachten wir die Werte der Log-Likelihood Funktion und die Anzahl der Iterationen in Abhängigkeit davon (Abbildung 10). Während eine leichte Tendenz zu erkennen ist, dass mit wachsendem Anteil an zensierten Beobachtungen die Berechnung von höheren Log-Likelihood Werten möglich ist, scheint die Iterationszahl davon bei beiden Algorithmen nicht beeinflusst.

5.3.3 Mischung von Gammaverteilungen

Basierend auf der Erkenntnis, dass der MCEM-Algorithmus bei normalverteilten Response-Variablen Resultate liefert, die sehr gut mit dem deterministischen EM-Algorithmus vergleichbar sind, untersuchen wir nun das Verhalten des MCEM-Algorithmus bei gammaverteilten Responses. Wir beginnen mit einer Mischung von $L = 2$ Komponenten, aus der eine Stichprobe

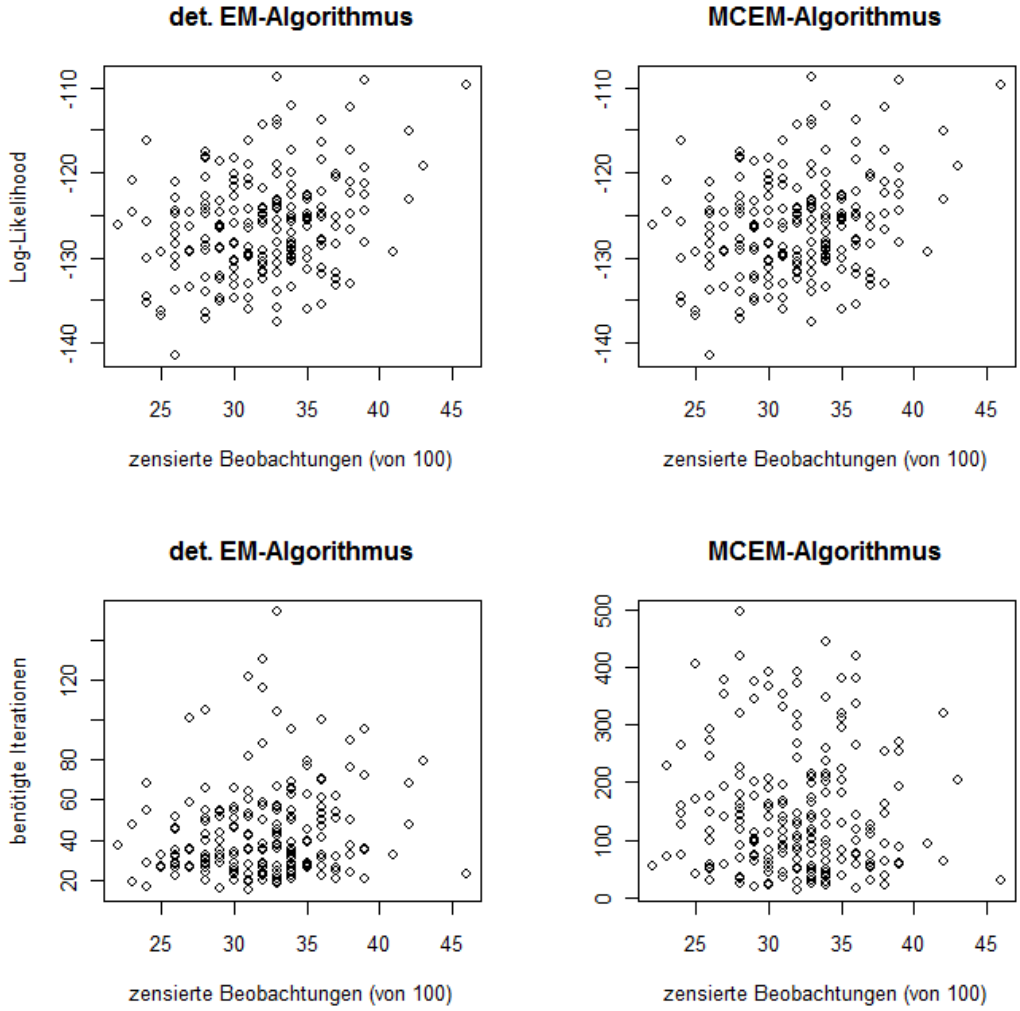

Abbildung 10: Log-Likelihood Werte und EM-Iterationen in Abhängigkeit der Anzahl zensierter Beobachtungen.

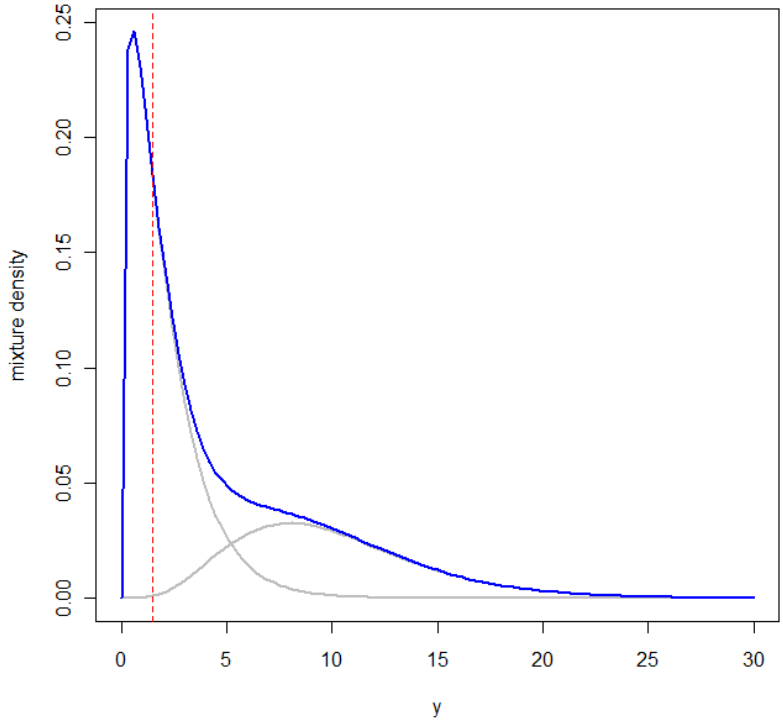

Abbildung 11: Mischung der zwei Gammaverteilungen aus Abschnitt 5.3.3.

$y_1, \ldots, y_n$ gezogen wird, in der y_i die Realisierung der Zufallsvariable Y_i ist mit

$$Y_i \overset{iid}{\sim} \pi_1 \Gamma(\mu_1, \nu_1) + (1 - \pi_1)\Gamma(\mu_2, \nu_2), \quad i = 1, \ldots, n.$$

Gegeben g, ist die konditionale Dichtefunktion von Y_i wie in (11) definiert mit Erwartungswert μ_g und Dispersionsparameter $\phi_g = 1/\nu_g$. Unter Verwendung des kanonischen Links der Gammaverteilung erhalten wir das Modell $1/\mu_g = \beta_g$, $g \in \{1, 2\}$.
Wir wählen für die Studie die Parameter

β_1	β_2	ϕ_1	ϕ_2	π_1
1/10	1/2	1/5	3/4	1/3

und erhalten damit eine marginale Dichtefunktion, deren Graph in Abbildung 11 zu sehen ist. Es resultiert damit $\mu_1 = 10$, $\mu_2 = 2$, sowie $\nu_1 = 5$, $\nu_2 = 4/3$. Weiters wählen wir $\tau_i = 3/2$, $i = 1, \ldots, n$, womit wir in einer Stichprobe vom Umfang $n = 100$ im Mittel 33 zensierte Beobachtungen erhalten, die großteils aus Komponente 2 stammen. Es gilt $P(Y_i \leq \tau_i | g = 2) = 0.492$ und $P(Y_i \leq$

$\tau_i|g = 1) = 0.001$. Wir generieren nun $R = 200$ solcher Stichproben und lassen vom MCEM-Algorithmus zwei Parameterschätzungen durchführen, zuerst unter Verwendung der MC-Simulationsgröße $S = 100$, dann für $S = 200$. Die Ergebnisse dieser Studie sind in folgender Tabelle zusammengefasst:

$S = 100$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\phi}_1$	$\hat{\phi}_2$	$\hat{\pi}_1$	l	It.
Minimum	0.059	0.269	0.011	0.118	0.059	-268.0	4
1. Quartil	0.087	0.410	0.108	0.504	0.271	-243.0	34
Median	0.100	0.521	0.178	0.722	0.342	-236.1	62
Mittelwert	0.102	0.522	0.207	0.741	0.344	-236.0	117
3. Quartil	0.114	0.601	0.254	0.917	0.426	-228.5	167
Maximum	0.172	0.999	0.781	2.593	0.653	-198.2	488
SE	0.021	0.137	0.138	0.329	0.113	11.0	121
$S = 200$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\phi}_1$	$\hat{\phi}_2$	$\hat{\pi}_1$	l	It.
Minimum	0.061	0.256	0.011	0.109	0.072	-268.1	4
1. Quartil	0.088	0.417	0.112	0.521	0.269	-242.9	32
Median	0.100	0.516	0.180	0.725	0.352	-236.1	69
Mittelwert	0.103	0.524	0.209	0.744	0.346	-236.0	101
3. Quartil	0.114	0.607	0.256	0.913	0.427	-228.3	122
Maximum	0.172	0.932	0.745	2.604	0.629	-198.2	495
SE	0.021	0.129	0.134	0.328	0.108	11.0	98
wahre Parameter	0.100	0.500	0.200	0.750	0.333		

Wie wir sehen können, liegen die Schätzer im Mittel in beiden Fällen sehr nahe an den wahren Werten der Parameter und weisen für $S = 200$ meist etwas geringere Standardfehler auf. Fälle in denen die maximal erlaubten 500 Iterationen überschritten wurden, sind dabei nicht berücksichtigt, sondern wurden durch neue Parameterschätzungen ersetzt. Mit der Wahl von $S = 100$ ist dies 31 Mal passiert, mit $S = 200$ nur 16 Mal.
Abbildung 12 zeigt die schrittweise Maximierung der Log-Likelihood Funktion in beiden Fällen am Beispiel einer einzelnen Stichprobe y.

Mischung dreier Gammaverteilungen

Wir setzen die Untersuchung des MCEM-Algorithmus fort, indem wir nun eine Mischung mit $L = 3$ Komponenten betrachten, also

$$Y_i \overset{iid}{\sim} \pi_1\Gamma(\mu_1, \nu_1) + \pi_2\Gamma(\mu_2, \nu_2) + \pi_3\Gamma(\mu_3, \nu_3), \quad i = 1, \ldots, n$$

mit $\sum_{g=1}^{3} \pi_g = 1$. Gegenüber der vorangegangenen Studie bedeutet dies die Schätzung von drei zusätzlichen Parametern.
Wir verwenden nun den Log-Link, also das Modell $\log \mu_g = \beta_g, g \in \{1, 2, 3\}$,

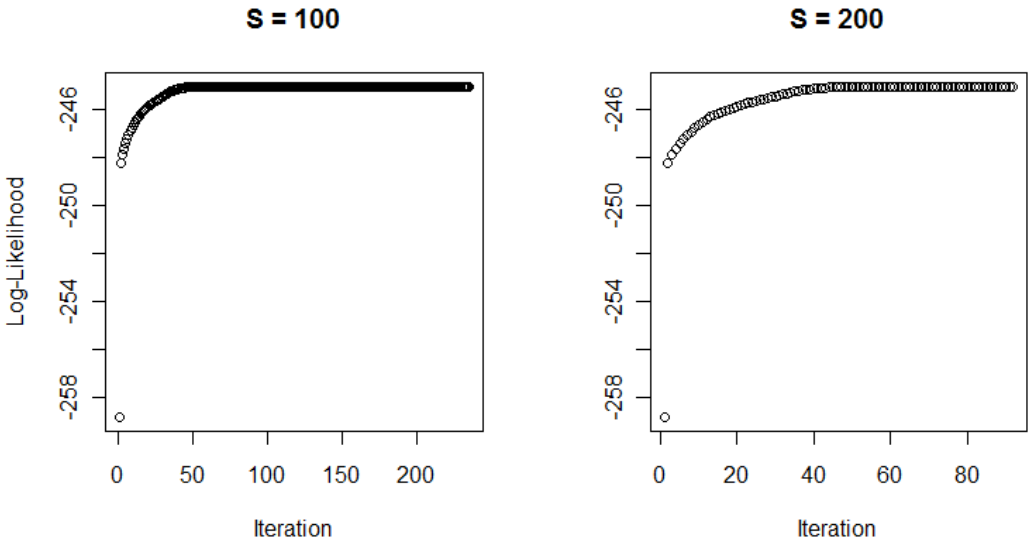

Abbildung 12: Entwicklung der Log-Likelihood Werte am Beispiel einer einzelnen Stichprobe y aus der ersten Studie in Abschnitt 5.3.3.

und wählen die Werte

β_1	β_2	β_3	ϕ_1	ϕ_2	ϕ_3	π_1	π_2
1/4	3/2	5/2	9/10	1/8	1/10	2/5	2/5

wobei wieder $\phi_g = 1/\nu_g$ gilt. Es resultiert damit eine marginale Dichtefunktion, deren Graph in Abbildung 13 zu sehen ist. Weiters sei $\tau_i = 3/2$, $i = 1, \ldots, n$, womit im Mittel jede fünfte Beobachtung zensiert ist. Wie zuvor soll der MCEM-Algorithmus für Stichproben der Größe $n = 100$ je zwei Parameterschätzungen durchführen, wobei wieder die Ergebnisse unter Verwendung von $S = 100$ mit denen für $S = 200$ verglichen werden.
Der typische Fortschritt der Maximierung einer Log-Likelihood Funktion bei Verwendung des MCEM-Algorithmus wird in Abbildung 12 recht anschaulich dargestellt. Durch die simulierten Werte im E-Schritt erreicht man die gewünschte Stabilität in vielen Fällen erst sehr spät, obwohl sich der Wert der Zielfunktion schon lange davor nicht mehr wesentlich ändert, sondern der in Abschnitt 4.3 beschriebenen Fluktuation unterliegt. Mit diesem Wissen wählen wir, um von nun an Rechenzeit zu sparen, $\varepsilon = 10^{-6}$ und erhalten damit folgende Resultate, die wiederum auf $R = 200$ Wiederholungen basie-

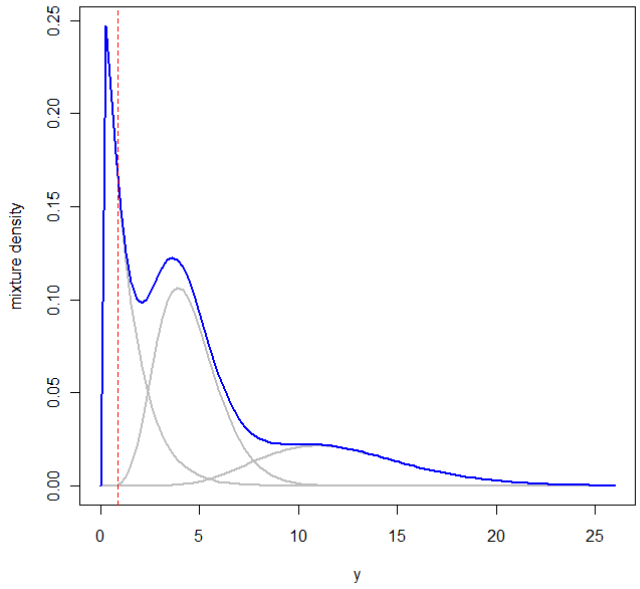

Abbildung 13: Mischung dreier Gammaverteilungen.

ren.

S = 100	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\phi}_1$	$\hat{\phi}_2$	$\hat{\phi}_3$	$\hat{\pi}_1$	$\hat{\pi}_2$	l	It.
Min.	-0.790	1.160	2.258	0.107	0.011	0.001	0.215	0.144	-275.0	8
1. Qu.	0.152	1.468	2.513	0.718	0.099	0.032	0.358	0.389	-258.5	29
Med.	0.289	1.557	2.570	1.078	0.161	0.060	0.397	0.444	-252.1	46
Mittel.	0.294	1.564	2.584	1.276	0.194	0.068	0.399	0.443	-251.8	57
3. Qu.	0.442	1.662	2.663	1.502	0.269	0.089	0.435	0.506	-244.0	73
Max.	1.181	1.945	3.083	8.247	0.657	0.317	0.706	0.686	-221.1	237
SE	0.258	0.136	0.125	0.917	0.123	0.051	0.066	0.087	9.9	39
S = 200	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\phi}_1$	$\hat{\phi}_2$	$\hat{\phi}_3$	$\hat{\pi}_1$	$\hat{\pi}_2$	l	It.
Min.	-0.791	1.160	2.262	0.202	0.011	0.001	0.214	0.143	-275.7	5
1. Qu.	0.142	1.466	2.514	0.731	0.101	0.032	0.358	0.395	-258.9	30
Med.	0.292	1.561	2.568	1.077	0.164	0.060	0.397	0.451	-252.1	48
Mittel.	0.285	1.564	2.584	1.314	0.195	0.068	0.397	0.445	-251.9	57
3. Qu.	0.422	1.652	2.666	1.530	0.274	0.089	0.434	0.507	-244.6	75
Max.	1.181	1.944	3.083	8.351	0.653	0.314	0.706	0.717	-221.1	237
SE	0.253	0.137	0.124	0.967	0.121	0.051	0.066	0.086	9.9	38
wahre P.	0.250	1.500	2.500	0.900	0.125	0.100	0.400	0.400		

Durch das Entschärfen des Konvergenzkriteriums wurde dafür gesorgt, dass beide Algorithmen im Mittel wesentlich weniger Iterationen benötigen als in der Studie davor, obwohl nun drei zusätzliche Parameter zu schätzen waren. In vier der ersten 200 Wiederholungen traten `NaNs` als simulierte Werte auf. Diese Fälle lieferten keine Konvergenz und wurden durch neue Parameterschätzungen ersetzt.
Auffällig ist hierbei, dass bei der Anzahl der benötigten Iterationen keine wesentlichen Unterschiede erkennbar sind. Außerdem bemerken wir, dass der Dispersionsparameter der ersten Komponente im Mittel überschätzt wird. Es handelt sich dabei um die Komponente, in der die Wahrscheinlichkeit, dass eine Beobachtung zensiert ist, mit 0.508 am größten ist. In den beiden anderen Komponenten ist die Wahrscheinlichkeit dafür geringer als 10^{-3}.

5.3.4 Gammaverteilte Responses im Regressionsmodell

Um auch die Schätzung von Regressionsparametern zu untersuchen, wenn für die Response-Variablen die Gammaverteilung angenommen wird, betrachten wir das Modell

$$\log \mu_g = \beta_{g1} + \beta_{g1} x, \quad g \in \{1, 2\}.$$

Wir definieren x dabei wie in Abschnitt 5.3.2 und wählen die Parameter

β_{11}	β_{21}	β_{12}	β_{22}	ϕ_1	ϕ_2	π_1
-0.25	0.5	1.5	3	0.5	0.2	0.75

um abermals Stichproben vom Umfang $n = 100$ zu generieren. Ein Beispiel für eine derartige Stichprobe zeigt Abbildung 14. Weiters definieren wir Schwellwerte $\tau_i = 1$ für $i = 1, \ldots, 100$, womit die Anzahl der zensierten Beobachtungen im Mittel etwa 25 beträgt.

Wir vergleichen nun die Ergebnisse unter Verwendung der konstanten Simulationsgröße $S = 100$ mit den Ergebnissen, die erzielt werden, wenn S in Abhängigkeit des Verhaltens der Log-Likelihood Funktion erhöht wird, wobei der Wert nach oben durch $S_{\max} = 250$ beschränkt sein soll.
Wie zuvor wählen wir $\varepsilon = 10^{-6}$ und erhalten nach $R = 200$ Wiederholungen

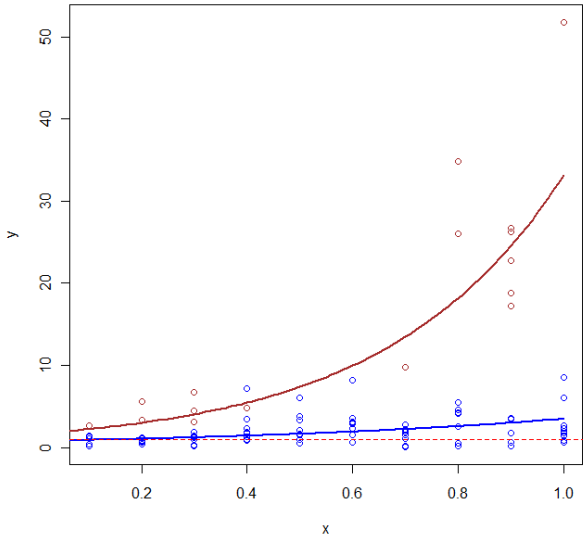

Abbildung 14: Beispiel für gammaverteilte Responses aus der Mischung der Regressionsmodelle $\mu_1 = \exp(-0.25 + 1.5x)$ (blau) und $\mu_2 = \exp(0.5 + 3x)$ (braun).

die folgenden Resultate:

S = 100	$\hat{\beta}_{11}$	$\hat{\beta}_{21}$	$\hat{\beta}_{12}$	$\hat{\beta}_{22}$	$\hat{\phi}_1$	$\hat{\phi}_2$	$\hat{\pi}_1$	l	It.
Minimum	-1.153	-1.177	-0.520	1.855	0.012	0.003	0.154	-246.9	3
1. Quartil	-0.443	0.320	1.199	2.706	0.375	0.080	0.697	-223.3	26
Median	-0.250	0.516	1.482	2.992	0.465	0.132	0.751	-215.2	46
Mittelwert	-0.217	0.483	1.480	3.018	0.496	0.187	0.746	-215.4	64
3. Quartil	-0.016	0.732	1.776	3.280	0.593	0.212	0.804	-209.1	78
Maximum	1.140	1.378	3.399	5.031	0.978	2.268	0.932	-180.9	439
SE	0.333	0.396	0.478	0.495	0.164	0.231	0.092	11.6	59
autom.	$\hat{\beta}_{11}$	$\hat{\beta}_{21}$	$\hat{\beta}_{12}$	$\hat{\beta}_{22}$	$\hat{\phi}_1$	$\hat{\phi}_2$	$\hat{\pi}_1$	l	It.
Minimum	-1.152	-1.040	-0.290	1.854	0.129	0.003	0.288	-246.9	7
1. Quartil	-0.424	0.324	1.185	2.710	0.376	0.083	0.696	-223.6	28
Median	-0.247	0.514	1.487	2.986	0.462	0.135	0.751	-215.3	44
Mittelwert	-0.220	0.490	1.480	3.007	0.492	0.190	0.745	-215.4	58
3. Quartil	-0.023	0.724	1.768	3.284	0.589	0.217	0.806	-208.8	74
Maximum	0.891	1.613	2.928	4.919	0.962	2.239	0.932	-180.9	285
SE	0.325	0.384	0.467	0.480	0.160	0.236	0.094	11.7	47
wahre P.	-0.250	0.500	1.500	3.000	0.500	0.200	0.750		

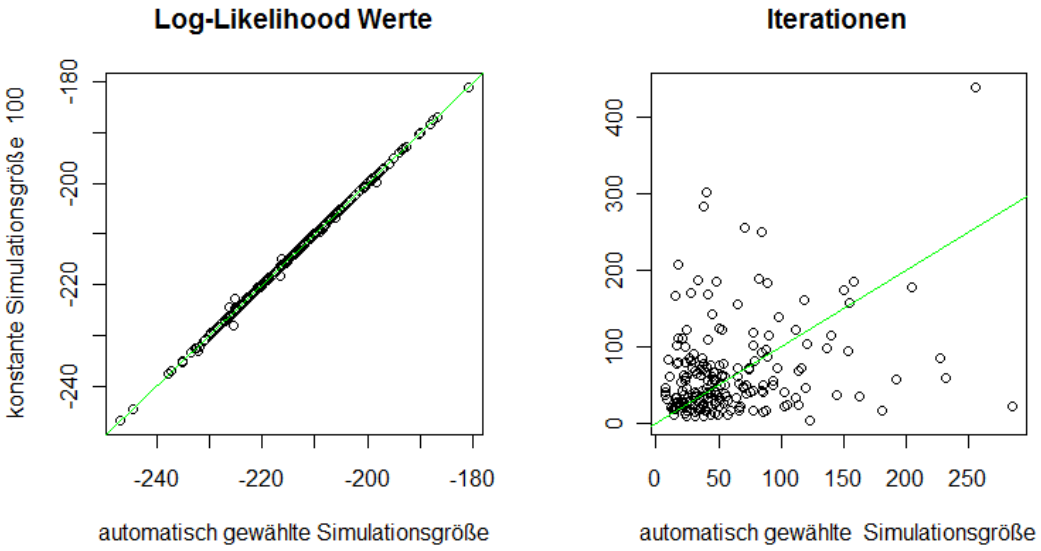

Abbildung 15: Vergleich der Log-Likelihood Werte und der benötigten Iterationen im Gamma-Regressionsmodell mit zwei Komponenten für 200 Datensätze.

Die automatische Anpassung des MC-Stichprobenumfangs bewirkt dabei gegenüber der konstanten Wahl von $S = 100$ nur unwesentlich geringere Iterationszahlen und auch die Werte der Standardfehler unterscheiden sich nur geringfügig. Einen Vergleich dazu sowie die errechneten Werte der marginalen Log-Likelihood Funktion zeigt Abbildung 15.

5.4 Verwendung der Funktion `glmm.lc`

5.4.1 Generieren von Datensätzen

Um die Vorgangsweise in den Simulationsstudien nachvollziehbar zu machen, betrachten wir das Beispiel eines generierten Datensatzes aus Abschnitt 5.3.4. Bevor dieser erzeugt wird, wählen wir eine Zahl p und bestimmen mit dem Befehl `set.seed(p)`, auf welche Weise alle anschließend generierten Zufallszahlen zustande kommen. Die neuerliche Eingabe des Befehls `set.seed(p)` mit der selben Wahl für p macht es also zu jedem Zeitpunkt möglich, die folgenden Ergebnisse zu rekonstruieren. Zum Generieren der Zufallsstichprobe dient die Funktion `lcmixsim`, deren Code im Anhang B zu finden ist:

Generieren einer Stichprobe

```
>  p <- 71
>  set.seed(p)
>  pr <- c(0.75,0.25)
>  L <- length(pr)
>  s2 <- c(0.5, 0.2)
>  b1 <- c(-0.25, 1.5)
>  b2 <- c(0.5,  3)
>  beta <- rbind(b1,b2)
>  fam <- Gamma(log)
>  N <- 100
>  thr <- rep(1,N)
>  x1 <- seq(0.1, 1, 0.1)
>  x1 <- rep(x1, N/length(x1))
>  form1 <- ~ x1
>  form2 <- paste(c("rep(17,N)", form1), collapse = "")
>  sim1 <- lcmixsim(form2, beta = beta , family = fam, pi = pr,
+       sigma2 = s2, threshold = thr)
>  Y <- sim1$Ys
```

In den Zeilen 3-8 werden die in Abschnitt 5.3.4 gewählten Parameterwerte zugewiesen. Die in den R-Befehlen verwendeten Bezeichnungen sind dabei `pr` für die Wahrscheinlichkeitsmassen, `s2` für die Dispersionsparameter und `b1` und `b2` für die Regressionsparameter (β_{11}, β_{12}) und (β_{21}, β_{22}). Danach folgt die Spezifikation der Verteilungsannahme (Zeile 9) und die Wahl von Stichprobenumfang und Thresholds (Zeilen 10-11). Die Zeilen 12-16 erzeugen die erklärende Variable und konstruieren eine gültige Modellformel, die gemeinsam mit den zuvor definierten Variablen an die Funktion `lcmixsim` übergeben wird. Aus dem erzeugten Objekt wird hier nur der Vektor `Ys` entnommen, der die generierten Beobachtungen y^* enthält.

5.4.2 Output

Der zuvor generierte Datensatz wird nun verwendet, um zu zeigen, in welcher Form der Output, der bei einem einzelnen Aufruf der Funktion `glmm.lc` erzeugt wird, erscheint. Um diesen an einen Standard für Modellierungsfunktionen in `R` anzupassen, wurden die Funktionen `print.lcMCEM`, und `print.lcdetEM`, sowie `summary.lcMCEM` und `summary.lcdetEM` implementiert, die von den *generischen* Funktionen `print` und `summary` genau dann aufgerufen werden, wenn die Bezeichnung hinter dem Punkt mit dem Namen der Klasse des Objekts übereinstimmt (vgl. Weisberg und Fox, 2010, Kap. 1.4). Je nachdem, welcher Algorithmus aufgerufen wird, bekommt das von

glmm.lc erzeugte Objekt entweder lcdetEM oder lcMCEM als Klasse zugewiesen. In dem hier betrachteten Beispiel nehmen wir an, dass die Responses gammaverteilt sind, also werden die zum MCEM-Algorithmus gehörenden Funktionen print.lcMCEM und summary.lcMCEM aktiv.

Output der Funktion glmm.lc

```
>  fit <- glmm.lc(Y ~ x1, family = Gamma(log), L = 2,
+                 threshold = thr, simul = "automatic",
+                 maxsim = 250, eps = 1e-06)
>  fit
MCEM algorithm met convergence criteria at iteration #  96

Call:  glmm.lc(formula = Y ~ x1, family = Gamma(log),
                threshold = thr, L = 2, simul = "automatic",
                maxsim = 250, eps = 1e-06)

Observations:
        total left censored    uncensored
          100             21             79

Coefficients:
   Group1     Group2  Group1:x1  Group2:x1
  -0.3051     0.4869     1.5511     2.9123

Mixture proportions:
Group1  Group2
0.7029  0.2971

Dispersion:
Group1  Group2
0.3701  0.1976

-2 log L:           434.85
```

Der Output enthält zunächst die Information, ob und nach wie vielen Iterationen das Konvergenzkriterium erfüllt war. Danach folgt die Anzeige des Funktionsaufrufs, eine Übersicht über die Anzahl der Beobachtungen und schließlich die Auflistung der Parameterschätzer. Im Gegensatz zu anderen Funktionen wird hier keine Referenzgruppe erzeugt, das heißt, dass die Koeffizienten, die zur zweiten Gruppe gehören, die tatsächlichen Intercept- und Slope-Parameter beschreiben und nicht nur den Unterschied zur Gruppe 1. Abschließend wird noch das Doppelte des Werts der Log-Likelihood Funktion angegeben, die es zu maximieren galt.

Summary des Model Fit

```
> summary(fit)
MCEM algorithm met convergence criteria at iteration #  96

Call:  glmm.lc(formula = Y ~ x1, family = Gamma(log),
                threshold = thr, L = 2, simul = "automatic",
                maxsim = 250, eps = 1e-06)

Observations:
        total left censored    uncensored
          100             21             79

Coefficients:
             Estimate Std. Error   t value
Group1    -0.3051091 0.01428753 -21.35493
Group2     0.4868537 0.02103315  23.14697
Group1:x1  1.5510679 0.02301853  67.38345
Group2:x1  2.9123319 0.03392559  85.84469

Mixture proportions:
Group1  Group2
0.7029  0.2971

Dispersion:
Group1 Group2
0.3701 0.1976

-2 log L:           434.85
Monte Carlo sample size increased to  250
Convergence at iteration  96
```

Die Funktion `summary`, die bei Modellierungsfunktionen allgemein eine etwas detailliertere Information liefern soll, enthält zusätzlich die Information über Standardfehler der Schätzer der Regressionskoeffizienten. Wie bei der Funktion `alldist` handelt es sich dabei aber um jene Standardfehler, die sich auf den letzten M-Schritt beziehen und somit zu klein sind (vgl. Abschnitt 3.2.4). Auf eine symbolische Kennzeichnung der Signifikanz wird deshalb verzichtet. Zusätzlich ist in der Summary noch die Information enthalten, wie groß der MC-Simulationsumfang zum Zeitpunkt der letzten Iteration war.

6 Resumee

Diese Arbeit hat sich mit der Theorie des Generalisierten Linearen Modells beschäftigt und ausgehend davon Erweiterungen betrachtet, die als Modelle für unvollständige Daten Anwendung finden. Zur ML-Schätzung wird dabei der EM-Algorithmus verwendet, der die marginale Log-Likelihood Funktion iterativ maximiert. Für die spezielle Modellklasse der Mischungen mit links-zensierten Responses wurde der EM-Algorithmus an die Besonderheiten des Problems angepasst, indem die Durchführung des E-Schritts durch eine Monte Carlo-Simulation unterstützt wird. Der resultierende MCEM-Algorithmus ist für Mischungen von GLMs in der `R`-Funktion `glmm.lc` implementiert und wird am Ende dieser Arbeit mehreren Tests unterzogen.
Eine Weiterführung dieser Arbeit könnte darin bestehen, den Programmcode zu erweitern, sodass zusätzliche Verteilungsannahmen möglich sind. Dabei ist aber zu beachten, dass Zensur, wie sie in dieser Arbeit beschrieben wurde typischerweise auf Beobachtungen stetig verteilter Zufallsvariablen wirkt. Eine weitere Möglichkeit, den MCEM-Algorithmus mächtiger zu machen, ist sich nicht auf links-zensierte Daten zu beschränken, sondern auch die Handhabung von rechts- oder intervall-zensierten Beobachtungen anzubieten.
Verschiedene Möglichkeiten der Verbesserung gibt es bei der Monte Carlo-Simulation im MCEM-Algorithmus, nicht nur was die numerische Stabilität betrifft. Die Entscheidung dafür, wie der Stichprobenumfang im Lauf der Iterationen gewählt werden soll, kann von Problem zu Problem variieren, wobei in bestimmten Situationen auch der Wunsch vorhanden sein kann, nicht für alle zensierten Beobachtungen den selben Wert für S zu verwenden, sondern Werte $S_i^{(k)}$, die an beobachtungsspezifische Merkmale angepasst sind.
Ein Umstand, auf den in der Arbeit des Öfteren hingewiesen wurde, ist das Fehlen der korrekten Standardfehler. Die Berechnung davon, basierend auf der beobachteten Informationsmatrix, stellt ebenfalls eine Möglichkeit dar, diese Arbeit auszubauen, sowohl hinsichtlich der Theorie, als auch bei der Implementierung des Algorithmus.

A Eigenschaften der Score-Funktion

Beweis von Satz 2.1

(6): Für die Score-Funktion gilt

$$\frac{\partial \log f(Y;\theta)}{\partial \theta} = \frac{1}{f(Y;\theta)} \frac{\partial f(Y;\theta)}{\partial \theta} \quad \text{und} \quad \int f(y;\theta)\, dy = 1.$$

Damit folgt

$$\begin{aligned} \mathbb{E}\left(\frac{\partial \log f(Y;\theta)}{\partial \theta}\right) &= \mathbb{E}\left(\frac{1}{f(Y;\theta)} \frac{\partial f(Y;\theta)}{\partial \theta}\right) = \int \frac{\partial f(y;\theta)}{\partial \theta}\, dy \\ &= \frac{\partial}{\partial \theta} \int f(y;\theta)\, dy = 0. \end{aligned}$$

(7): Die Anwendung der Kettenregel liefert

$$\begin{aligned} \mathbb{E}\left(\frac{\partial^2 \log f(Y;\theta)}{\partial \theta^2}\right) &= \mathbb{E}\left(\frac{1}{f(Y;\theta)} \frac{\partial^2 f(Y;\theta)}{\partial \theta^2} - \frac{\partial f(Y;\theta)}{\partial \theta} \frac{\partial f(Y;\theta)}{\partial \theta} \frac{1}{f(Y;\theta)^2}\right) \\ &= \frac{\partial^2}{\partial \theta^2} \int f(y;\theta)\, dy - \mathbb{E}\left(\frac{\partial \log f(Y;\theta)}{\partial \theta} \frac{\partial \log f(Y;\theta)}{\partial \theta}\right) \\ &= -\mathbb{E}\left(\frac{\partial \log f(Y;\theta)}{\partial \theta}\right)^2, \end{aligned}$$

also

$$\mathbb{E}\left(\frac{\partial \log f(Y;\theta)}{\partial \theta}\right)^2 + \mathbb{E}\left(\frac{\partial^2 \log f(Y;\theta)}{\partial \theta^2}\right) = 0 \qquad \square$$

Bemerkung: Der Beweis setzt voraus, dass ein Vertauschen von Differentiation und Integration möglich ist. Die Regularitätsbedingungen, unter denen dies erlaubt ist, werden in dieser Arbeit vorausgesetzt.

B Programmcode

Dieser Anhang enthält den Programmcode, der zur Verwendung der Funktion `glmm.lc` benötigt wird. Neben den Funktionen `lcdetEM.fit` und `lcMCEM.fit` sind auch die Funktion `MCss.t` und die in Abschnitt 5.4.2 erwähnten Funktionen zur Präsentation des Outputs, sowie die Funktion `lcmixsim` hier angeführt. In den folgenden Darstellungen sind den Code-Zeilen Kommentare in englischer Sprache hinzugefügt, die am Symbol `#`, welches ihnen vorangestellt ist, zu erkennen sind. Die Anzeige des Outputs kann bei Verwendung dieses Programmcodes gegenüber Abschnitt 5.4.2 geringfügige Unterschiede aufweisen, wenn ein Zeilenumbruch, der zur Darstellung hier notwendig geworden ist, einen entsprechenden Befehl betrifft. Alle Teile dieses Anhangs sind unter Verwendung der Software `R`, Version 2.13.0 entstanden und auch online über das Institut für Statistik der TU Graz erhältlich.

glmm.lc

```
glmm.lc <-
function (formula, family = gaussian, data, threshold, L = 2,
          random, na.action, prstart, s2start, proc = "MCEM",
          simul = "standard", s2update = "EM", maxsim = 500,
          spike.protect = 1e-05, EMmaxit = 500, eps = 1e-07,
          trace = 0, ...)
{
    call <- match.call()
  # save function call

    if (is.character(family))
        family <- get(family, mode = "function",
        envir = parent.frame())
    if (is.function(family))
        family <- family()
    if (is.null(family$family)) {
        print(family)
        stop("family' not recognized")
    }
  # check family argument

    if (!family$family == "gaussian" && !family$family == "Gamma")
        stop("only Normal or Gamma distribution are supported")
  # only normal and Gamma distribution are available
  # in this version

    if (proc == "detEM" && !family$family == "gaussian")
```

```
28        stop("deterministic EM procedure is only available
              for Normal distribution")
30    # deterministic EM algorithm can only be used if
      # response variables are assumed to be normal distributed
32
      if (missing(data))
34        data <- environment(formula)
      # data should be extracted from the environment if not
36    # explicitly specified

38    mf <- match.call(expand.dots = FALSE)
      m <- match(c("formula", "data", "threshold", "na.action"),
40                 names(mf), 0L)
      # check which arguments are present
42
      mf <- mf[c(1L, m)]
44    mf$drop.unused.levels <- TRUE
      mf[[1L]] <- as.name("model.frame")
46    mf <- eval(mf, parent.frame())
      # create model frame
48
      mt <- attr(mf, "terms")
50    # extract model terms

52    Y <- model.response(mf, "any")
      # extract response
54
      X <- if (!is.empty.model(mt))
56        model.matrix(mt, mf, contrasts)
      else matrix(, NROW(Y), 0L)
58    # extract design matrix

60    intercept <- attr(mt, "intercept") > 0L
      # check if intercept is part of the model
62
      threshold <- model.extract(mf, "threshold")
64    # extract threshold vector from model frame

66    if(missing(s2start)) s2start <- NULL
      if(missing(prstart)) prstart <- NULL
68    # assign the default value of the EM procedures for
      # initial proportions and dispersion parameter
70
```

```
    if(missing(random)) {
72      rv <- colnames(X)
        random.int <- intercept
74  }
    else {
76      mfr <- model.frame(random)
        mtr <- attr(mfr, "terms")
78      XR <- model.matrix(mtr, mfr)
        rv <- colnames(XR)
80      random.int <- (intercept && attr(mtr, "intercept") > 0L)
    }
82  # prepare information about the design matrix
    # used in the EM iterations
84
    if (proc == "MCEM") {
86  fit <-  lcMCEM.fit(x = X, y = Y, threshold = threshold, L = L,
              prstart = prstart, family = family, simul = simul,
88            maxsim = maxsim, rv = rv, random.int = random.int,
              s2start = s2start, intercept = intercept,
90            EMmaxit = EMmaxit, eps = eps,
              spike.protect = spike.protect, trace = trace)
92  fit <- c(fit, list(call = call, family = family,
              formula = formula, terms = mt, data = data,
94            contrasts = attr(X, "contrasts"),
              xlevels = .getXlevels(mt, mf)))
96  fit$model <- mf
    fit$na.action <- attr(mf, "na.action")
98  class(fit) <- "lcMCEM"
    }
100 # call Monte Carlo EM procedure if selected, construct the
    # object fit which is assigned the class "lcMCEM" to obtain
102 # appropriate output when calling summary or
    # print function
104
    if (proc == "detEM") {
106 fit <-  lcdetEM.fit(x = X, y = Y, threshold = threshold, L = L,
              prstart = prstart, family = family, rv = rv,
108           random.int = random.int, s2start = s2start,
              s2update = s2update, intercept = intercept,
110           EMmaxit = EMmaxit, eps = eps,
              spike.protect = spike.protect, trace = trace)
112 fit <- c(fit, list(call = call, family = family,
              formula = formula, terms = mt,
```

```
                data = data, contrasts = attr(X, "contrasts"),
                xlevels = .getXlevels(mt, mf)))
    fit$model <- mf
    fit$na.action <- attr(mf, "na.action")
    class(fit) <- "lcdetEM"
    }
  # call deterministic EM procedure if selected, construct the
  # object fit which is assigned the class "lcdetEM" to obtain
  # appropriate output when calling summary or
  # print function

    fit
}
```

Als nächstes folgen die Funktionen `lcdetEM.fit` und `lcMCEM.fit`, die von `glmm.lc` aufgerufen werden können.

lcdetEM.fit

```
lcdetEM.fit <-
function (x, y, threshold = NULL, L = 2, family = gaussian(),
      rv = colnames(x), s2update = "EM", mustart = NULL,
      s2start = NULL, prstart = NULL, random.int = TRUE,
      EMmaxit = 500, eps = 1e-07, spike.protect = 1e-05,
      intercept = TRUE, trace = 1, ...)
    # x must already have the shape of a model matrix
    # (usually created by glmm.lc before)
{
     if (!family$family == "gaussian")
     stop("deterministic procedure is only available
          for Normal distribution")
   # check the family specification,
   # only gaussian is available here

     xvars <- ncol(x)
     xv <- colnames(x)
   # save the number of variables and their names

     if (xvars == 0)
        stop("Empty model is not supported.")
   # no support for empty model

     if ("" %in% xv | is.null(xv)) {
          x <- as.matrix(as.data.frame(x))
```

```
26      }
        else {
28           x <- as.matrix(x)
        }
30    # a matrix without column names or an column name ""
      # has to be adapted to avoid errors when formula is
32    # created and paste() is used.
      # this will not happen if x is prepared by glmm.lc
34
        if (!is.null(prstart) && (any(prstart <= 0) |
36           length(prstart) != L | sum(prstart) != 1))
           stop("invalid initial vector of mixture proportions")
38    # check validity of specified initial proportions

40      loglik <- rep(NA, EMmaxit + 1)
      # prepare vector to record log likelihood values
42
        th <- if(!is.null(threshold)) threshold
44           else rep(-Inf,NROW(y))
      # the value -Inf is assigned to the threshold vector
46    # if not specified, so that any response in that case
      # will be treated as uncensored
48
        cen <- as.numeric(y <= th)
50    # indication of censure

52      y <- apply(cbind(y, th),1,max)
      # create vector of observed data y*
54
        n.cen <- sum(cen)
56      N <- length(y)
        uncen <- N - n.cen
58    # number of observations (censored, total and uncensored)

60      if (sum(rv %in% xv) == 0) L <- 1
      # if none of the "random variables" is part of x
62    # L is set to 1 as there are no variables
      # with group specific parameters
64
        pr <- if (is.null(prstart)) rep(1/L,N*L)
66           else rep(prstart,rep(N,L))
      # create vector of initial proportions, length N*L
68
```

```
  g0 <- rep(1:L, rep(N, L))
  Group <- as.factor(g0)
# initial group indices, length: N*L

  datat <- x
  datatinit <- x[rep(1:N,L), , drop = FALSE]
# create matrix by expanding x
# use drop = FALSE to avoid coercing to a vector if only one
# column exists

  if (L > 1) {
# formula with group interaction terms has to be created
# only if L > 1

          rv <- rv[rv %in% xv]
        # remove variables in rv that aren't present in x

          n1 <- 1 + intercept
        # determine index of first non - intercept column in x

          n2 <- 1 + random.int
        # determine first element of rv that is not
        # representing the intercept

          if (n1 <= xvars) {
              xv <- xv[n1:xvars]
              nrv <- xv[xv %in% rv == FALSE]
          }
        # if there are variables beside the intercept in x,
        # they are used in the formula where xv can be pasted
        # however if the appear in rv, the go into the formula
        # with the group interaction term
        # so xv is reduced to nrv
        # ("non random variables" =
        # only one coefficient estimated)

          rvars <- length(rv)
          xf <- if(n1 <= xvars && length(nrv) > 0) paste(nrv,
                                              collapse = " + ")
                else NULL
          ran <-  if(n2 <= rvars) paste(if(random.int) "Group +",
              "Group:(",paste(rv[n2:rvars], collapse = "+"), ")")
                  else paste("Group")
```

```
112         # if there are variables in rv beside the intercept their
            # names are inserted and have the interaction term
114         # (  "Group:(",paste(rv[n2:rvars], collapse = "+")  )
            # if intercept should be estimated for each group
116         # ( random.int = TRUE ) the group vector is inserted
            # as well
118         # note that n1 > xvars only if xv includes
            # nothing but an intercept
120         # note that n2 > rvars only if rv includes
            # nothing but an intercept
122
              ran.xf <- paste(c(ran, xf), collapse = " + ")
124         # combine the two parts

126           loop.f <- formula(paste("~ ", ran.xf,
                        if(random.int | !intercept) " - 1"))
128         # create formula to be used in the loop

130       }
          ys1 <- rep(y,L)
132       th.s1 <- rep(th, L)
          cen.s1 <- rep(cen, L)
134     # expand response, threshold and censure index vector

136       init.matrix <- if (L > 1) model.matrix(loop.f,
                                  data = as.data.frame(datatinit))
138                     else x
        # create matrix, here matrix doesn't have to be extended
140
          if (family$link == "inverse") mustart <- y + (y == 0)*0.1
142       if (family$link == "log") mustart <- (y <= 0)*exp(y) +
                                                 (y > 0)*y
144     # create valid starting values for log link and inverse link

146       fit <- glm.fit(x, y, family = family, mustart = mustart)
          Mu <- fit$fitted
148     # use initial model fit to extract fitted values which
        # can be used as starting values for EM procedure
150     # (length: N*L)

152       r <- y - Mu
        # compute residuals
154
```

```
  betanew <- coef(fit)
# initial coefficients

  Eta <- fit$linear.predictors
# extract linear predictors from the initial fit

  s2 <- if (is.null(s2start)) sum(r^2)/(N*L) +
                              rep(spike.protect, L)
        else s2start
# initial dispersion parameters

  if (L > 1) {
         tmp <- gqz(L, minweight = 1e-50)
         z0 <- -tmp$l
         z <- rep(z0, rep(N, L))
         sdev <- sqrt(sum((family$linkfun(y) - Eta)^2)/N)
         Eta <- Eta + z*sdev
  }
# extend the vector of linear predictors using the mass
# points of Gaussian quadrature to shift original predictor
# gqz can be found in the package npmlreg

  Mu <- family$linkinv(Eta)
# initial means

  V <- function(z) {
         return(-dnorm(z)/pnorm(z))
       }
# define alpha(z) (for left censored observations only)

  pr <- apply(cbind(pr,1e-10), 1, max)
# avoid proportions of 0 in the beginning of the
# algorithm to stabilize

  P <- pnorm(th.s1, mean=Mu, sd=sqrt(s2[g0]))
# compute probabilities of being censored (length: N*L)

  pi.f <- pr*((cen.s1 == 0)*dnorm(ys1, mean=Mu, sd=sqrt(s2[g0]))
                + (cen.s1 == 1)*P)
# compute pi*f, which will be used as nominator in
# computation of wig (length: N*L)

  fyi <- tapply(pi.f, rep(1:N, L), sum)
```

```
198   # sum of pi*f will become denominator in computation of wig

200     ll.prev <- -Inf
      d.ll <- Inf
202     it <- 0
      converged <- FALSE
204   # initializations

206     ll <- sum(log(fyi))
      loglik[it+1] <- ll
208   # compute log likelihood for the first time

210     d.ll <- ll - ll.prev
      ll.prev <- ll
212   # store change of the log likelihood

214     if (trace==2){
          cat("Iteration  :", it, "\n")
216       cat("Converged  :", converged, "\n")
          cat("marg loglik:", ll, "\n")
218       cat("delta(ll)  :", d.ll, "\n")
          cat("Parameters :", betanew, "\n")
220       cat("Dispersion :", s2, "\n")
          cat("Mix.Prob's :", pr[N*1:L], "\n")
222       cat("--------------------", "\n")
      }
224   # optional output

226     while (!converged && it < EMmaxit) {
      # begin iteration
228
        wig <-  pi.f/rep(fyi, L)
230     # compute weights (length: N*L)

232       pr <- (tapply(wig, g0, sum)/N)[g0]
      # update proportion vector (length: N*L)
234
        z <- ifelse(cen.s1 == 1, (th.s1 - Mu)/sqrt(s2)[g0], 0)
236     z <- apply(cbind(z, -20), 1, max)
      # create standardized threshold values and make sure
238   # that they don't get to small, so that the computation of
      # alpha(z) is possible
240
```

```
      rig <- (cen.s1 == 0)*ys1 +
242           (cen.s1 == 1)*(Mu + sqrt(s2)[g0]*V(z))
    # create pseudo responses
244
      lstart <- if (family$link == "log") Mu
246             else (Mu + rig)/2
    # create valid starting value for glm.fit which is
248 # part of the M step

250   fit <- glm.fit(x = init.matrix, y = as.vector(rig),
                     family = family, weights = wig,
252                  mustart = as.vector(lstart))
    # complete M step by using glm.fit
254 # to obtain new parameter estimates

256   Mu <- fit$fitted.values
      if (s2update == "EM"){
258       noms2 <- tapply(wig*(rig - Mu)^2 +
              s2[g0]*cen.s1*wig*(1 + z*V(z) - V(z)^2), g0, sum)
260       denoms2 <- tapply(wig, g0, sum)
      }
262   if (s2update == "LA") {
          noms2 <- tapply(wig*(rig - Mu)^2, g0, sum)
264       denoms2 <- tapply(wig*( (1-cen.s1) -
                          cen.s1*(z*V(z) - V(z)^2)), g0, sum )
266   }
      s2 <- (noms2/denoms2)
268 # update dispersion parameters, EM procedure and
    # Lawless/Atkinson procedure are available
270
      s2 <- apply(cbind(s2,spike.protect), 1, max)
272 # guarantee that dispersion parameters cannot become zero
    # and algorithm gets caught in a spike
274
      betanew <- coef(fit)
276 # save new coefficients

278   pr <- apply(cbind(pr,1e-10/(it + 1)^2), 1, max)
    # control proportions during the first iterations
280 # to stabilize algorithm

282   P <- pnorm(th.s1, mean=Mu, sd=sqrt(s2[g0]))
      pi.f <- pr*((cen.s1 == 0)*dnorm(ys1, mean=Mu,
```

```
284             sd=sqrt(s2[g0])) + (cen.s1 == 1)*P)
        fyi <- tapply(pi.f, rep(1:N, L), sum)
286     ll <- sum(log(fyi))
        it <- it+1
288     loglik[it+1] <- ll
        d.ll <- ll - ll.prev
290     converged <- (abs(d.ll)/abs(ll.prev) < eps)
        ll.prev <- ll
292   # the same computations that have been made before
      # the loop are repeated for the current parameters
294
        if (trace==2){
296       cat("Iteration  :", it, "\n")
          cat("Converged  :", converged, "\n")
298       cat("marg loglik:", ll, "\n")
          cat("delta(ll)  :", d.ll, "\n")
300       cat("Parameters :", betanew, "\n")
          cat("Dispersion :", s2, "\n")
302       cat("Mix.Prob's :", pr[N*1:L], "\n")
          cat("--------------------", "\n")
304     }
      # optional output
306
      }
308 # end of iteration

310   cond.predictors <- fit$linear.predictors
      cond.predictors.matrix <- matrix(cond.predictors, N, L,
312                                      byrow = FALSE)
      cond.fitted <-  matrix(Mu, N, L, byrow = FALSE)
314   cond.res <- rig - fit$fitted.values
      cond.res.matrix <- matrix(cond.res, N, L, byrow = FALSE)
316   marg.predictors <- tapply(wig*cond.predictors,
                                    rep(1:N, L), sum)
318   marg.fitted <- tapply(pr*Mu, rep(1:N, L), sum)
      marg.res <- tapply(pr*cond.res, rep(1:N, L), sum)
320   pr <- pr[N*1:L]
    # save important values
322
      if(trace==1){
324       cat("Iteration  :", it, "\n")
          cat("Converged  :", converged, "\n")
326       cat("marg loglik:", ll, "\n")
```

```
          cat("delta(ll)  :", d.ll, "\n")
          cat("Parameters :", betanew, "\n")
          cat("Dispersion :", s2, "\n")
          cat("Mix.Prob's :", pr, "\n")
          cat("--------------------", "\n")
      }
    # optional output

      names(pr) <- names(s2) <- paste("Group", 1:L, sep = "")
      names(N) <- "total"
      names(n.cen) <- "left censored"
      names(uncen) <- "uncensored"
      list(coefficients = betanew, fitted.values = marg.fitted,
         linear.predictors = marg.predictors, residuals = marg.res,
         conditional.fitted.values = cond.fitted,
         conditional.linear.predictors = cond.predictors.matrix,
         conditional.residuals = cond.res.matrix,
         mixture.proportions = pr, Groups = L, obs = N,
         df.residual = N - length(beta) - L + 1, dispersion = s2,
         EMiter = it, leftc.obs = n.cen, unc.obs = uncen,
         threshold = th, weights = wig, logL = ll,
         delta.ll  = d.ll, loglik = loglik[1:(it+1)],
         EMmaxit = EMmaxit, y = y, rig = rig, mat = XZ,
         EMconverged = converged, eps = eps, lastglm = fit)
    # output, to be presented in an appropriate
    # way by the function glmm.lc
}
```

lcMCEM.fit

```
lcMCEM.fit <-
function (x, y, threshold = NULL, L = 2, family = gaussian(),
      rv = colnames(x), simul = "standard", maxsim = 500,
      mustart = NULL, s2start = NULL, prstart = NULL,
      random.int = TRUE, EMmaxit = 500, eps = 1e-07,
      spike.protect = 1e-05, intercept = TRUE, trace = 1, ...)
    # x must already have the shape of a model matrix
    # (usually created by glmm.lc before)

{
     xvars <- ncol(x)
     xv <- colnames(x)
   # save the number of variables and their names
```

```
14
     if (xvars == 0)
16      stop("Empty model is not supported.")
   # no support for empty model
18
     if ("" %in% xv | is.null(xv)) {
20        x <- as.matrix(as.data.frame(x))
     }
22
     else {
24        x <- as.matrix(x)
     }
26 # a matrix without column names or an column name ""
   # has to be adapted to avoid errors when formula is
28 # created and paste() is used.
   # this will not happen if x is prepared by glmm.lc
30
     if (!is.null(prstart) &&  (any(prstart <= 0) |
32     length(prstart) != L |  sum(prstart) != 1))
        stop("invalid initial vector of mixture proportions")
34 # check validity of specified initial proportions

36   loglik <- rep(NA, EMmaxit + 1)
   # prepare vector to record log likelihood values
38
     simsize <- rep(NA, EMmaxit)
40 # prepare vector to record MC sample size

42   th <- if(!is.null(threshold)) threshold
          else rep(-Inf,NROW(y))
44 # the value -Inf is assigned to the threshold vector if not
   # specified, so that any response in that case
46 # will be treated as uncensored

48   cen <- as.numeric(y <= th)
   # indication of censure
50
     y <- apply(cbind(y, th),1,max)
52 # create vector of observed data y*

54   n.cen <- sum(cen)
     N <- length(y)
56   uncen <- N - n.cen
```

```
  # number of observations (censored, total and uncensored)
58
    if (sum(rv %in% xv) == 0) L <- 1
60  # if none of the "random variables" is part of x,
  # L is set to 1 as there are no variables with
62  # group specific parameters

64    pr <- if (is.null(prstart)) rep(1/L,N*L)
           else rep(prstart,rep(N,L))
66  # create vector of initial proportions, length N*L

68    g0 <- rep(1:L, rep(N, L))
    if (n.cen == 0) simul <- 1
70  # if no censored observations have occured simul is
  # assigned a number for simplicity
72
    if (simul == "automatic")
74       S <- max(10, floor(50 * n.cen/N))
  # if MC simulation size is to be computed automatically
76  # the starting value will be half of the percentage of
  # censored observations
78
    Group <- as.factor(g0)
80  # initial group indices, length: N*L

82    datat <- x
    if (L > 1) {
84  # formula with group interaction terms has to be created
  # only if L > 1
86
          rv <- rv[rv %in% xv]
88       # remove variables in rv that aren't present in x

90          n1 <- 1 + intercept
       # determine index of first non - intercept column in x
92
          n2 <- 1 + random.int
94       # determine first element of rv that is not
       # representing the intercept
96
          if (n1 <= xvars) {
98            xv <- xv[n1:xvars]
              nrv <- xv[xv %in% rv == FALSE]
```

```
100           }
          # if there are variables beside the intercept in x,
102       # they are used in the formula where xv can be pasted
          # however if the appear in rv, the go into the formula
104       # with the group interaction term
          # so xv is reduced to nrv
106       # ("non random variables" =
          # only one coefficient estimated)
108
            rvars <- length(rv)
110         xf <- if(n1 <= xvars && length(nrv) > 0) paste(nrv,
                      collapse = " + ")
112               else NULL
            ran <-  if(n2 <= rvars) paste(if(random.int) "Group +",
114             "Group:(",paste(rv[n2:rvars], collapse = "+"), ")")
                   else paste("Group")
116       # if there are variables in rv beside the intercept their
          # names are inserted and have the interaction term
118       # (  "Group:(",paste(rv[n2:rvars], collapse = "+")  )
          # if intercept should be estimated for each group
120       # ( random.int = TRUE ) the group vector is
          # inserted as well.
122       # note that n1 > xvars only if xv includes
          # nothing but an intercept
124       # note that n2 > rvars only if rv includes
          # nothing but an intercept
126
            ran.xf <- paste(c(ran, xf), collapse = " + ")
128       # combine the two parts

130         loop.f <- formula(paste("~ ", ran.xf,
                      if(random.int | !intercept) " - 1"))
132       # create formula to be used in the loop

134     }
        ys1 <- rep(y,L)
136     th.s1 <- rep(th, L)
        cen.s1 <- rep(cen, L)
138   # expand response, threshold and censure index vector

140     if (family$link == "inverse") mustart <- y + (y == 0)*0.1
        if (family$link == "log") mustart <- (y <= 0)*exp(y) +
142                                          (y > 0)*y
```

```
  # create valid starting values for log link and inverse link
144
    fit <- glm.fit(x, y, family = family, mustart = mustart)
146   Mu <- fit$fitted
  # initial fit to extract fitted values which can be
148 # used as starting values for EM procedure (length: N*L)

150   r <- switch(family$family,
              Gamma = (y - Mu)/Mu,
152           gaussian = y - Mu
              )
154 # compute residuals

156   Eta <- fit$linear.predictors
  # extract linear predictors from the initial fit
158
    betanew <- coef(fit)
160 # initial coefficients

162   s2 <- if (is.null(s2start)) sum(r^2)/(N*L) +
                        rep(spike.protect, L)
164         else s2start
  # initial dispersion parameters
166
    if (L > 1) {
168    tmp <- gqz(L, minweight = 1e-50)
       z0 <- -tmp$l
170    z <- rep(z0, rep(N, L))
       sdev <- sqrt(sum((family$linkfun(y) - Eta)^2)/N)
172    Eta <- switch(family$family,
                  gaussian = Eta + z*sdev,
174               Gamma =  Eta * (1 + sdev/abs(Eta))^z
                  )
176   }
  # extend the vector of linear predictors using the mass
178 # points of Gaussian quadrature to shift original predictor
  # gqz can be found in the package npmlreg
180
    Mu <- family$linkinv(Eta)
182 # initial means

184   pr <- apply(cbind(pr,1e-10), 1, max)
  # avoid proportions of 0 in the beginning of
```

```
186   # the algorithm to stabilize

188     P <- switch(family$family,
                Gamma =  pgamma(th.s1, shape=1/s2[g0],
190                                scale=Mu*s2[g0]),
                gaussian = pnorm(th.s1, mean=Mu,
192                                 sd=sqrt(s2[g0]))
      )
194   # compute probabilities of being censored (length: N*L)

196     pi.f <- switch(family$family,
              Gamma = pr*((cen.s1 == 0)* dgamma(ys1,
198                           shape = 1/s2[g0], scale = Mu*s2[g0])
                       + (cen.s1 == 1)* P),
200           gaussian =  pr*((cen.s1 == 0)*dnorm(ys1, mean=Mu,
                             sd=sqrt(s2[g0]))
202                     + (cen.s1 == 1)*P)
            )
204   # compute pi*f, which will be used as nominator in
      # computation of wig (length: N*L)
206
      fyi <- tapply(pi.f, rep(1:N, L), sum)
208   # sum of pi*f will become denominator in computation of wig

210     ll.prev <- -Inf
        d.ll <- Inf
212     it <- 0
        converged <- FALSE
214   # initializations

216     ll <- sum(log(fyi))
        loglik[it+1] <- ll
218   # compute log likelihood for the first time

220     d.ll <- ll - ll.prev
        ll.prev <- ll
222   # store change of the log likelihood

224     if (trace==2){
           cat("Iteration  :", it, "\n")
226        cat("Converged  :", converged, "\n")
           cat("marg loglik:", ll, "\n")
228        cat("delta(ll)  :", d.ll, "\n")
```

```
        cat("Parameters :", betanew, "\n")
        cat("Dispersion :", s2, "\n")
        cat("Mix.Prob's :", pr[N*1:L], "\n")
        cat("--------------------", "\n")
    }
  # optional output

    while (!converged && it < EMmaxit) {
  # begin iteration

      wig <-  pi.f/rep(fyi, L)
     # compute weights (length: N*L)

      pr <- (tapply(wig, g0, sum)/N)[g0]
     # update proportion vector  (length: N*L)

      if (simul == "automatic") {
          S <- ifelse(d.ll < 0,
                min(floor(S * (1.2 + 0.75^(it + 1))), maxsim),
                min(S + 5 * as.numeric(it/10 == floor((it/10))),
                    maxsim))
      }
      else {
      S <- min(maxsim, MCss.t(it + 1, simul))
      }
    # determine current MC sample size

      simsize[it+1] <- S
      ind.r <- NULL
      ind.s <- NULL
      for (i in 1:N) {
        if (cen[i] == 0) {
            ind.r <- c(ind.r, i)
            ind.s <- c(ind.s, 0)
        }
        else {
              ind.r <- c(ind.r,rep(i,S))
              ind.s <- c(ind.s, 1:S)
        }
      }
     # create list of indices which will be used to
     # generat simulated responses
```

```
272         size.t <- n.cen * S + (uncen)
          # compute number of rows per group of the
274       # extended data frame

276         ind.r <- rep(ind.r, L)
            ind.s <- rep(ind.s, L)
278         ys <- y[ind.r]
            th.s <- th[ind.r]
280         cen.s <- cen[ind.r]
          # extend vectors and list of indices according
282       # to the new data frame

284         i.s <- (cen.s == 1)
            g <- rep(1:L, rep(size.t, L))
286       # extension of group indices

288         ind.w <-  N*(g-1) + ind.r
          # new list to extend vector of weights correctly
290       # (length: size.t * L)
          # and also vectors with L*N different elements such as Mu
292
            w <- (1-cen.s)*wig[ind.w] + cen.s*wig[ind.w]/S
294       # weights, w_isg in thesis

296         U <- runif(S*L*n.cen)
          # uniform sample
298
            ys[i.s] <- switch(family$family,
300                       Gamma = ifelse(P[ind.w][i.s] != 0,
                           qgamma(U*P[ind.w][i.s], shape=1/s2[g][i.s],
302                        scale=Mu[ind.w][i.s]*s2[g][i.s]), th.s[i.s]),
                          gaussian = ifelse(P[ind.w][i.s] != 0,
304                        qnorm(U*P[ind.w][i.s], mean=Mu[ind.w][i.s],
                           sd=sqrt(s2[g][i.s])), th.s[i.s])
306         )
          # applying inversion method to simulate from the left tail
308
            while(any(is.nan(ys))) {
310               print("NaNs produced in ys")
                  i.nan <- is.nan(ys)
312               n.nan <- sum(as.numeric(i.nan))
                  cat("n.nan=", n.nan, "\n")
314               U <- runif(n.nan)
```

```
            ys[i.nan] <- switch(family$family,
316         Gamma = qgamma(U*P[ind.w][i.nan],
                    shape=1/s2[g][i.nan],
318                 scale=Mu[ind.w][i.nan]*s2[g][i.nan]),
            gaussian = qnorm(U*P[ind.w][i.nan],
320                 mean=Mu[ind.w][i.nan],
                    sd=sqrt(s2[g][i.nan]))
322         )
            if (n.nan > S/2) return("no convergence!")
324       }
      # repetition of simulation for those indices
326   # where NaN s were generated

328       Group <- as.factor(g)
          datat <- x[ind.r, , drop = FALSE]
330   # extend design matrix

332       if (L > 1) {
              XZ <- model.matrix(loop.f, data = as.data.frame(datat))
334       }
          else {
336           XZ <- datat
          }
338   # create model matrix to be used in the GLM fit
      # of the M step. as the MC sample size increases
340   # the matrix has to be rebuilt in each iteration

342       lstart <- if (family$family == "gaussian" &&
                        family$link == "log") Mu[ind.w]
344                 else (Mu[ind.w] + ys)/2
      # create valid starting value for glm.fit
346
          if  (family$family == "Gamma")  ys <- ifelse(ys == 0,
348         th.s/2,ys)
          if  (family$family == "gaussian") ys <- ifelse(ys == -Inf,
350         th.s,ys)
      # replace invalid simulated values by adequate values
352   # depending on the threshold

354       fit <- glm.fit(x = XZ, y = ys, family = family,
                    weights = as.vector(w), mustart = lstart)
356   # complete M step by using glm.fit to
      # obtain new parameter estimates
```

```
358
        r <- switch(family$family,
360             Gamma = (ys - fit$fitted.values)/fit$fitted.values,
                gaussian = ys - fit$fitted.values
362             )
      # compute residuals
364
        s2 <- (tapply(w*r^2, g, sum)/ (tapply(w, g, sum) ) )
366     s2 <- apply(cbind(s2,spike.protect), 1, max)
      # update dispersion parameters
368
        Mu <- fit$fitted.values[ind.s <= 1]
370     betanew <- coef(fit)
      # save new coefficients
372
        pr <- apply(cbind(pr,1e-10/(it + 1)^2), 1, max)
374   # control proportions during the first
      # iterations to stabilize algorithm
376
        P <- switch(family$family,
378             Gamma =  pgamma(th.s1, shape=1/s2[g0],
                             scale=Mu*s2[g0]),
380             gaussian = pnorm(th.s1, mean=Mu,
                             sd=sqrt(s2[g0]))
382     )
        pi.f <- switch(family$family,
384               Gamma = pr*((cen.s1 == 0)* dgamma(ys1,
                          shape = 1/s2[g0], scale = Mu*s2[g0])
386                       + (cen.s1 == 1)* P),
                gaussian =  pr*((cen.s1 == 0)*dnorm(ys1,
388                         mean=Mu, sd=sqrt(s2[g0]))
                        + (cen.s1 == 1)*P)
390           )
        fyi <- tapply(pi.f, rep(1:N, L), sum)
392     ll <- sum(log(fyi))
        it <- it+1
394     loglik[it+1] <- ll
        d.ll <- ll - ll.prev
396     converged <- (abs(d.ll)/abs(ll.prev) < eps)
        ll.prev <- ll
398   # the same computations that have been made before the loop
      # are repeated for the current parameters
400
```

```
      if (trace==2){
402         cat("Iteration  :", it, "\n")
            cat("Converged  :", converged, "\n")
404         cat("marg loglik:", ll, "\n")
            cat("delta(ll)  :", d.ll, "\n")
406         cat("Parameters :", betanew, "\n")
            cat("Dispersion :", s2, "\n")
408         cat("Mix.Prob's :", pr[N*1:L], "\n")
            cat("--------------------", "\n")
410     }
    # optional output
412
    }
414 # end of iteration

416   cond.predictors <- fit$linear.predictors[ind.s <= 1]
    cond.predictors.matrix <- matrix(cond.predictors, N, L,
418                                          byrow = FALSE)
    cond.fitted <-  matrix(Mu, N, L, byrow = FALSE)
420 cond.MCres <- tapply(r, ind.w, sum)/(1 +(S - 1)*cen)
    cond.MCres.matrix <- matrix(cond.MCres, N, L, byrow = FALSE)
422 marg.predictors <- tapply(wig*cond.predictors,
                              ind.r[ind.s <=1] , sum)
424 marg.fitted <- tapply(pr*Mu, ind.r[ind.s <= 1], sum)
    marg.MCres <- tapply(pr*cond.MCres, ind.r[ind.s <= 1], sum)
426 pr <- pr[N*1:L]
  # save important values
428
    if(trace==1){
430       cat("Iteration  :", it, "\n")
          cat("Converged  :", converged, "\n")
432       cat("marg loglik:", ll, "\n")
          cat("delta(ll)  :", d.ll, "\n")
434       cat("Parameters :", betanew, "\n")
          cat("Dispersion :", s2, "\n")
436       cat("Mix.Prob's :", pr, "\n")
          cat("--------------------", "\n")
438   }
  # optional output
440
    names(pr) <- names(s2) <- paste("Group", 1:L, sep = "")
442 names(N) <- "total"
    names(n.cen) <- "left censored"
```

```
444       names(uncen) <- "uncensored"
      list(coefficients = betanew, fitted.values = marg.fitted,
446           linear.predictors = marg.predictors,
          residuals = marg.MCres,
448           conditional.fitted.values = cond.fitted,
          conditional.linear.predictors = cond.predictors.matrix,
450           conditional.residuals = cond.MCres.matrix,
          mixture.proportions = pr, Groups = L, obs = N,
452           df.residual = N - length(beta) - L + 1, dispersion = s2,
          EMiter = it, MCsample.size = S,
454           simsize = simsize[1:it], maxsim = maxsim,
          leftc.obs = n.cen, unc.obs = uncen, threshold = th,
456           weights = w, logL = ll, delta.ll  = d.ll,
          loglik = loglik[1:(it+1)], EMmaxit = EMmaxit, y = y,
458           ys = ys, mat = XZ, EMconverged = converged,
          eps = eps, lastglm = fit)
460     # output, to be presented in an appropriate
    # way by the function glmm.lc
462
}
```

Die Funktion `MCss.t` wird intern von `lcMCEM.fit` aufgerufen, sofern der MC-Simulationsumfang nicht automatisch gewählt werden soll.

MCss.t

```
MCss.t <- function(it, simul = "standard")
2 {
    if (is.numeric(simul))
4       ss <- floor(simul)
  # numeric argument leads to constant sample size
6
    else {
8       ss <- 10 * (floor(it/20) + 1)
        }
10  # compute standard sample size depending on
  # iteration number
12
    ss
14 }
```

Die folgenden Teile des Codes gehören zu den Funktionen, die für die adäquate Darstellung des Outputs zuständig sind:

print.lcdetEM

```
print.lcdetEM <-
function (x, digits = max(3, getOption("digits") - 3), ...)
{
    if (x$EMconverged) {
        cat("EM algorithm met convergence criteria at
                    iteration # " ,x$EMiter,  "\n")
    }
    else {
        cat("EM algorithm failed to meet
            convergence criteria","\n")
    }
  # show information about convergence behaviour

    cat("\nCall: ", deparse(x$call), "\n\n")
  # display function call

    cat("Observations:", "\n")
    observed <- c(x$obs, x$leftc.obs, x$unc.obs)
    print.default(format(observed, digits = digits),
                          print.gap = 1, quote = FALSE)
  # general information about number of censored
  # and uncensored observations

    cat("\n")
    ncoef <- length(x$coefficients)
    if (ncoef > 0) {
        cat("Coefficients:", " \n")
        print.default(format(x$coefficients, digits = digits),
                              print.gap = 2, quote = FALSE)
    }
    else {
        cat("\nCall: ", deparse(x$call), "\n\n")
        cat("No coefficients", "\n")
    }
  # report estimated coefficients

    cat("\n")
    cat("Mixture proportions:","\n" )
    print.default(format(x$mixture.proportions, digits = digits),
                          print.gap = 2, quote = FALSE)
  # report estimated mixture proportions

```

```
    cat("\n")
        cat("Dispersion:","\n" )
    print.default(format(x$dispersion, digits = digits),
                          print.gap = 2, quote = FALSE)
  # report estimated dispersion parameters

    cat("\n")
    cat("-2 log L:\t   ", format(round(x$logL * -2,
                                  digits = 2)),"\n")
  # report -2l where l is the final log Likelihood value

    invisible(x)
}
```

print.lcMCEM

```
print.lcMCEM <-
function (x, digits = max(3, getOption("digits") - 3), ...)
{
    if (x$EMconverged) {
        cat("MCEM algorithm met convergence criteria at
        iteration # " ,x$EMiter,  "\n")
    }
    else {
        cat("MCEM algorithm failed to meet
        convergence criteria","\n")
    }
  # show information about convergence behaviour

    cat("\nCall: ", deparse(x$call), "\n\n")
  # display function call

    cat("Observations:", "\n")
    observed <- c(x$obs, x$leftc.obs, x$unc.obs)
    print.default(format(observed, digits = digits),
        print.gap = 1, quote = FALSE)
  # general information about number of censored
  # and uncensored observations

    cat("\n")
    ncoef <- length(x$coefficients)
    if (ncoef > 0) {
        cat("Coefficients:", " \n")
```

```
        print.default(format(x$coefficients, digits = digits),
            print.gap = 2, quote = FALSE)
    }
    else {
        cat("\nCall: ", deparse(x$call), "\n\n")
        cat("No coefficients", "\n")
    }
  # report estimated coefficients

    cat("\n")
    cat("Mixture proportions:","\n" )
    print.default(format(x$mixture.proportions, digits = digits),
        print.gap = 2, quote = FALSE)
  # report estimated mixture proportions

    cat("\n")
        cat("Dispersion:","\n" )
    print.default(format(x$dispersion, digits = digits),
        print.gap = 2, quote = FALSE)
  # report estimated dispersion parameters

    cat("\n")
    cat("-2 log L:\t   ", format(round(x$logL * -2,
            digits = 2)),"\n")
  # report -2l where l is the final log Likelihood value

    invisible(x)
}
```

summary.lcdetEM

```
summary.lcdetEM <-
function (object, digits = max(3, getOption("digits") - 3), ...)
{
    if (object$EMconverged) {
        cat("EM algorithm met convergence criteria at
                iteration # " , object$EMiter,  "\n")
    }
    else {
        cat("EM algorithm failed to meet
                convergence criteria","\n")
    }
```

```
12  # show information about convergence behaviour

14    cat("\nCall: ", deparse(object$call), "\n\n")
    # display function call
16
      cat("Observations:", "\n")
18    observed <- c(object$obs, object$leftc.obs, object$unc.obs)
      print.default(format(observed, digits = digits),
20            print.gap = 1, quote = FALSE)
    # general information about number of censored
22  # and uncensored observations

24    cat("\n")
      ncoef <- length(object$coefficients)
26    if (ncoef > 0) {
          cat("Coefficients:", " \n")
28        lastglmsumm <- summary.glm(object$lastglm)
        # if coefficients are available the summary of
30      # the model fit in the last  M-step is used

32        fitcoef <- matrix(lastglmsumm$coeff[, 1:3], ncol = 3,
              dimnames = list(dimnames(lastglmsumm$coef)[[1]],
34            c(dimnames(lastglmsumm$coeff)[[2]][1:2], "t value")))
        # coefficients are extracted together with their
36      # standard errors and t values. As the standard
        # errors are based on the conditional information
38      # matrix no symbols (***,**,...) are used to
        # describe significance
40
          print(fitcoef)
42    }
      else {
44        cat("\nCall: ", deparse(object$call), "\n\n")
          cat("No coefficients", "\n")
46    }
      cat("\n")
48    cat("Mixture proportions:","\n" )
      print.default(format(object$mixture.proportions,
50        digits = digits),print.gap = 2, quote = FALSE)
    # report estimated mixture proportions
52
      cat("\n")
54        cat("Dispersion:","\n" )
```

```
    print.default(format(object$dispersion, digits = digits),
56      print.gap = 2, quote = FALSE)
  # report estimated dispersion parameters
58
    cat("\n")
60  cat("-2 log L:\t   ", format(round(object$logL * -2,
      digits = 2)),"\n")
62  # report -2l where l is the final log Likelihood value

64  if (object$EMconverged)
      cat("Convergence at iteration ", round(object$EMiter, 0))
66  # additional information about number of iterations

68  cat("\n")
    keep <- match(c("call", "family", "terms", "coefficients",
70              "fitted.values", "linear.predictors", "residuals",
                "Groups", "mixture.proportions", "dispersion",
72              "df.residual", "obs", "leftc.obs", "unc.obs",
                "threshold", "EMiter", "logL", "EMconverged"),
74              names(object), 0L)
  # save important parts of the object to keep
76  # them being available

78  invisible(object[keep])
}
```

summary.lcMCEM

```
summary.lcMCEM <-
2 function (object, digits = max(3, getOption("digits") - 3), ...)
  {
4     if (object$EMconverged) {
        cat("MCEM algorithm met convergence criteria at
6           iteration # " ,object$EMiter,  "\n")
      }
8     else {
        cat("MCEM algorithm failed to meet
10          convergence criteria","\n")
      }
12  # show information about convergence behaviour

14    cat("\nCall: ", deparse(object$call), "\n\n")
```

```
  # display function call

    cat("Observations:", "\n")
    observed <- c(object$obs, object$leftc.obs, object$unc.obs)
    print.default(format(observed, digits = digits),
      print.gap = 1, quote = FALSE)
  # general information about number of censored
  # and uncensored observations

    cat("\n")
    ncoef <- length(object$coefficients)
    if (ncoef > 0) {
        cat("Coefficients:", " \n")
        lastglmsumm <- summary.glm(object$lastglm)
      # if coefficients are available the summary of
      # the model fit in the last  M-step is used

        fitcoef <- matrix(lastglmsumm$coeff[, 1:3], ncol = 3,
            dimnames = list(dimnames(lastglmsumm$coef)[[1]],
            c(dimnames(lastglmsumm$coeff)[[2]][1:2], "t value")))
      # coefficients are extracted together with their
      # standard errors and t values. As the
      # standard errors are based on the conditional
      # information matrix no symbols (***,**,...)
      # are used to describe significance

        print(fitcoef)
    }
    else {
        cat("\nCall: ", deparse(object$call), "\n\n")
        cat("No coefficients", "\n")
    }
    cat("\n")
    cat("Mixture proportions:","\n" )
    print.default(format(object$mixture.proportions,
            digits = digits),print.gap = 2, quote = FALSE)
  # report estimated mixture proportions

    cat("\n")
        cat("Dispersion:","\n" )
    print.default(format(object$dispersion, digits = digits),
              print.gap = 1, quote = FALSE)
  # report estimated dispersion parameters
```

```
58
    cat("\n")
60  cat("-2 log L:\t   ", format(round(object$logL * -2,
        digits = 2)),"\n")
62  # report -2l where l is the final log Likelihood value

64  if (object$leftc.obs > 0)
    cat("Monte Carlo sample size increased to ",
66      object$MCsample.size, "\n")
    # show the MC sample size used in the last E-step
68
    if (object$EMconverged)
70      cat("Convergence at iteration ", round(object$EMiter, 0))
    # additional information about number of iterations
72
    cat("\n")
74  keep <- match(c("call", "family", "terms", "coefficients",
          "fitted.values", "linear.predictors", "residuals",
76        "Groups", "mixture.proportions", "dispersion",
          "df.residual", "obs", "leftc.obs", "unc.obs",
78        "threshold", "EMiter", "MCsample.size",
          "logL", "EMconverged"), names(object), 0L)
80  # save important parts of the object to
    # keep them being available
82
    invisible(object[keep])
84 }
```

Schließlich wird noch der Code der Funktion `lcmixsim` angeführt, die zum Generieren der Datensätze in den Simulationsstudien verwendet wurde (vgl. Abschnitt 5.4.1).

lcmixsim

```
lcmixsim <- function(formula, family = gaussian,
2               data = list(), pi, beta, sigma2,
                threshold, ...)
4 {
    call <- match.call()
6   if (is.character(family))
        family <- get(family)
8   if (is.function(family))
        family <- family()
10  if (is.null(family$family)) {
```

```
        print(family)
12      stop("'family' not recognized")
    }
14  # check family specification

16  if (missing(data))
       data <- environment(formula)
18  mf <- match.call(expand.dots = FALSE)
    m <- match(c("formula", "data", "threshold"),
20                names(mf), 0L)
    mf <- mf[c(1,m)]
22  mf[[1]] <- as.name("model.frame")
    mf <- eval(mf, parent.frame())
24  mt <- attr(mf, "terms")
  # create model frame and terms
26
    X <- if (!is.empty.model(mt))
28     model.matrix(mt, mf, contrasts)
    Y <- model.response(mf, "numeric")
30  beta <- as.matrix(beta)
  # extract design matrix
32
    L <- length(pi)
34  # determine number of components

36  N <- length(Y)
    mu <- Y <- Ys <- C <- 1:N
38  # initialize values according to sample size

40  if (nrow(beta) != L)
        stop("parameter matrix incomplete")
42  # checking if the number of parameters is correct

44  rownames(beta) <- paste("beta", 1:nrow(beta))
    if (missing(threshold))
46      threshold <- rep(-Inf, N)

48  G <- sample(L, size=N, prob=pi, replace=TRUE)
  # generate N group indicators
50
    for (i in 1:N) {
52      mu[i] <- family$linkinv(X[i, ] %*% beta[G[i], ])
      # compute predictors first, then apply the function which is
```

```
54      # inverse to the link function to obtain the means

56        Y[i] <- switch(family$family,
                    Gamma = rgamma(1, shape=1/sigma2[G[i]],
58                          scale=mu[i]*sigma2[G[i]]),
                    gaussian = rnorm(1, mean=mu[i],
60                          sd=sqrt(sigma2[G[i]]))
                    )
62      # simulate responses

64        Ys[i] <- max(Y[i], threshold[i])
        # generate observed data by applying a
66      # simple censoring mechanism

68        C[i] <- as.numeric(Y[i] < threshold[i])
        # assign logical values to censoring indicator
70
    }
72    result <- list(X=X, mean=mu, G=G, Y=Y, Ys=Ys, beta = beta,
            C = C, L = L, pi = pi, threshold = threshold)
74  # return generated data and important values

76 }
```

Literatur

Aitkin, M. A., Francis, B., Hinde, J. und Darnell, R. (2009). *Statistical Modelling in R.* Oxford: Oxford University Press.

Böhning, D., Dietz, E., Schaub, R., Schlattmann, P. und Lindsay, B. G. (1994). The distribution of the likelihood ratio for mixtures of densities from the one-parameter exponential family. *Annals of the Institute of Statistical Mathematics*, *46*, 373-388.

Bierens, H. J. (2004). *The Tobit Model.* Verfügbar unter `http://econ.la.psu.edu/~hbierens/EasyRegTours/TOBIT_Tourfiles/TOBIT.PDF`

Booth, J. und Friedl, H. (2005). *Fitting finite mixture models to left censored data using a Monte Carlo EM algorithm.* (University of Florida; Graz, University of Technology, Department of Statistics,)

Box, G. E. P. und Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society. Series B*, *26*, 211-252.

Davison, A. C. (2003). *Statistical Models.* Cambridge: Cambridge University Press.

Dempster, A. P., Laird, N. M. und Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B*, *39*, 1-38.

Einbeck, J. und Hinde, J. (2006). A note on NPML estimation for exponential family regression models with unspecified dispersion parameters. *Austrian Journal of Statistics*, *35*, 233-243.

Finch, S. J., Mendell, N. R. und Thode, H. C. (1989). Probabilistic measures of adequacy of a numerical search for a global maximum. *Journal of the American Statistical Association*, *84*, 1020-1023.

Grün, B. und Leisch, F. (2008). Finite Mixtures of Generalized Linear Regression Models. In *Recent Advances in Linear Models and Related Areas* (S. 205-230). Heidelberg: Physica-Verlag HD.

Hagenaars, J. A. und McCutcheon, A. L. (2002). *Applied Latent Class Analysis.* Cambridge: Cambridge University Press.

Hardin, J. W. und Hilbe, J. (2007). *Generalized Linear Models and Extensions.* Texas: Stata Press.

Johnson, V. E. und Albert, J. H. (1999). *Ordinal Data Modeling.* New York: Springer.

Kaplan, E. L. und Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, *53*, 457-481.

Kleiber, C. und Zeileis, A. (2008). *Applied Econometrics with R.* New York: Springer.

Lawless, J. F. (1982). *Statistical Models and Methods for Lifetime Data*. New York: Wiley.

Lehmann, E. L. und Romano, J. P. (2005). *Testing Statistical Hypotheses*. New York: Springer.

Lesaffre, E. und Spiessens, B. (2001). On the effect of the number of quadrature points in a logistic random-effects model: An example. *Journal of the Royal Statistical Society. Series C*, *50*, 325-335.

Lindsay, B. G. (1995). *Mixture Models: Theory, Geometry, and Applications* (Bd. 5). Hayward: Institute of Mathematical Statistics.

Lindsay, B. G. und Roeder, K. (1992). Residual diagnostics for mixture models. *Journal of the American Statistical Association*, *87*, 785-794.

Lindsey, J. K. (1997). *Applying Generalized Linear Models*. New York: Springer.

Louis, T. A. (1982). Finding the observed information matrix when using the EM algorithm. *Journal of the Royal Statistical Society. Series B*, *44*, 226-233.

Lumley, T. (2004). The survival package. *The Newsletter of the R Project*, *4*, 26-28.

McCullagh, P. und Nelder, J. A. (1989). *Generalized Linear Models* (2. Aufl.). London: Chapman, Hall/CRC.

McLachlan, G. J. (1987). On bootstrapping the likelihood ratio test stastistic for the number of components in a normal mixture. *Journal of the Royal Statistical Society. Series C*, *36*, 318-324.

McLachlan, G. J. (1988). On the choice of starting values for the EM algorithm in fitting mixture models. *Journal of the Royal Statistical Society. Series D*, *37*, 417-425.

McLachlan, G. J. und Peel, D. (2000). *Finite Mixture Models*. New York: Wiley.

Pregibon, D. (1980). Goodness of link tests for generalized linear models. *Journal of the Royal Statistical Society. Series D*, *29*, 15-24.

Rawlings, J. O., Pantula, S. G. und Dickey, D. A. (1998). *Applied Regression Analysis: A Research Tool*. New York: Springer.

Roche, A. (2003). *EM algorithm and variants: An informal tutorial* (Bericht). Orsay, France: CEA.

Sherman, R. P., Yu-Yun, K. H. und Dalal, S. R. (1999). Conditions for convergence of Monte Carlo EM sequences with an application to product diffusion modeling. *Econometrics Journal*, *2*, 248-267.

Sigelman, L. und Zeng, L. (1999). Analyzing censored and sample-selected data with tobit and heckit models. *Political Analysis*, *8*, 167-182.

Steinkellner, C. (2012). *Implementierung einer Modellierungsfunktion in R* (Bericht). Graz, University of Technology, Department of Statistics.

Tobin, J. (1958). Estimation of relationships for limited dependent variables. *Econometrica*, *26*, 24-36.

Watanabe, M. und Yamaguchi, K. (2003). *The EM Algorithm and Related Statistical Models*. New York: Marcel Dekker.

Wei, G. C. G. und Tanner, M. A. (1990). A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, *85*, 699-704.

Weisberg, S. und Fox, J. (2010). *An R Companion to Applied Regression*. Thousand Oaks: SAGE Publications.

Wu, C. F. J. (1983). On the convergence properties of the EM algorithm. *The Annals of Statistics*, *11*, 95-103.