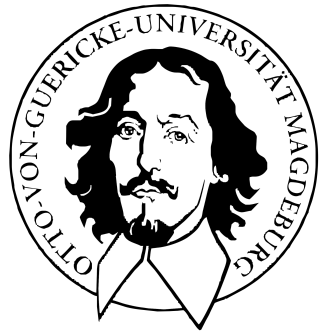


Psychotherapieforschung

MSc Klinische Psychologie und Psychotherapie

SoSe 2025

Prof. Dr. Dirk Ostwald

(3) Das Linear Mixed Model

Überblick

- Das Linear Mixed Model (LMM) ist eine Weiterentwicklung des Allgemeinen Linearen Modells (ALM)
- Durch iterative Schätzverfahren sind LMMs in den letzten 50 Jahren sehr populär geworden
- In **R** sind LMMs durch die Verfügbarkeit der Pakete `nlme` und `lme4` sehr verbreitet
- Klassische Anwendungen von LMMs sind Mehrebenen- und Longitudinalanalysen
- Fixed- und Random-Effects Modelle der Metaanalyse sind spezielle LMMs
- Viele weitere Modelle sind Spezialfälle von LMMs, z.B. die Bayesianische ALM Schätzung
- LMMs sind die State-of-the-Art Inferenzmodelle der Psychotherapiewirksamkeitsforschung

Wir formulieren zunächst das Linear Mixed Model

Zur Schätzung eines Linear Mixed Models betrachten wir dann

- die Generalisierte Kleinste-Quadrate Schätzung der Fixed Effects und ihre Konfidenzintervalle,
- den bedingten Erwartungswert der Random-Effects, sowie
- die Varianzkomponentenschätzung mit Restricted Maximum-Likelihood.

Zur Evaluation eines LMMs betrachten wir dann

- Wald-Konfidenzintervalle

Anwendungsbeispiel

- Multizentren-Parallelgruppendesign mit Treatmentgruppe und Kontrollgruppe
- Je zwei Gruppen randomisierter Proband:innen an fünf verschiedenen HSA-Standorten
- Treatmentfaktor (`TRM`) mit zwei Leveln (`1`: Waitlist Control, `2`: Treatment)
- Hochschulambulanzfaktor (`HSA`) mit vier Leveln (`1`: Magdeburg, `2`: Halle, `3`: Dresden, `4`: Marburg)
- 5 Proband:innen pro Treatmentlevel an jedem HSA-Standort
- Primäres Ergebnismaß: BDI-II Differenz
- Random-Intercept-Modell Analyse

Beispieldatensatz

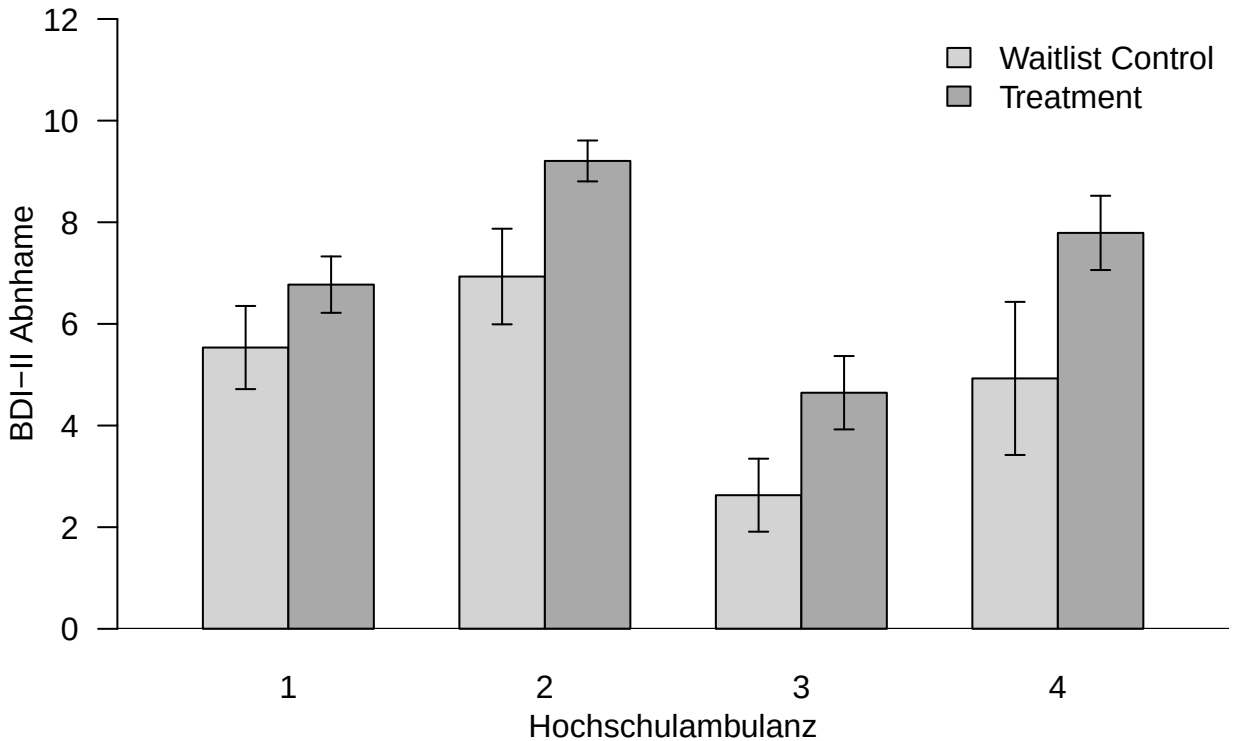
TRM	HSA	BDI
1	1	4.7
1	1	6.2
1	1	4.6
1	1	6.0
1	1	6.2
2	1	7.7
2	1	6.4
2	1	6.6
2	1	6.5
2	1	6.8
1	2	7.0
1	2	5.7
1	2	6.3
1	2	8.1
1	2	7.5
2	2	8.9
2	2	9.8
2	2	9.1
2	2	9.4
2	2	8.8
1	3	1.8
1	3	3.4
1	3	2.1
1	3	3.3
1	3	2.6
2	3	4.7
2	3	4.7
2	3	3.9
2	3	4.2
2	3	5.8
1	4	7.4
1	4	5.0
1	4	4.7
1	4	4.1
1	4	3.5
2	4	7.4
2	4	8.3
2	4	8.3
2	4	6.7
2	4	8.3

Deskriptivstatistik

```r
library(dplyr)                                                    # dplyr für einfache Datengruppierug
D   = read.csv("./3_Daten/mz-anova.csv")                          # Dateneinlesen
DS  = D %>% group_by(TRM, HSA) %>%  summarise(av = mean(BDI, na.rm = TRUE), # Group mean
                                              sd = sd(BDI, na.rm = TRUE),   # Group standard deviation
                                             .groups = "drop")              # Gruppierungsaufhebung
print(DS)
```

```
# A tibble: 8 x 4
    TRM   HSA    av    sd
  <int> <int> <dbl> <dbl>
1     1     1  5.53 0.818
2     1     2  6.93 0.941
3     1     3  2.63 0.718
4     1     4  4.93 1.51
5     2     1  6.77 0.555
6     2     2  9.21 0.402
7     2     3  4.65 0.722
8     2     4  7.79 0.731
```

Visualisierung



Modellformulierung

Modellschätzung

Modellevaluation

Selbstkontrollfragen

Modellformulierung

Modellschätzung

Modellevaluation

Selbstkontrollfragen

Definition (Linear Mixed Model in Clusterdarstellung)

Für $i = 1, \ldots, k$ sei

$$y_i = X_i\beta + Z_i b_i + \varepsilon_i, \tag{1}$$

wobei

- y_i ein n_i-dimensionaler beobachtbarer Zufallsvektor ist, der *Daten des $i$ten Clusters* genannt wird,
- $X_i \in \mathbb{R}^{n_i \times p}$ eine vorgegeben Matrix ist, die *Fixed-Effects-Designmatrix des $i$ten Clusters* genannt wird,
- $\beta \in \mathbb{R}^p$ ein unbekannter fester Vektor ist, der *Fixed Effects* oder *Populationsparameter* genannt wird,
- $Z_i \in \mathbb{R}^{n_i \times q_i}$ eine vorgegeben Matrix ist, die *Random-Effects-Designmatrix des $i$ten Clusters* genannt wird,
- b_i ein q_i-dimensionaler latenter Zufallsvektor ist, der *Random Effects des $i$ten Clusters* genannt wird, mit

$$b_i \sim N(0_{q_i}, \Sigma_{b_i}) \text{ mit } \Sigma_{b_i} \in \mathbb{R}^{q_i \times q_i} \text{ p.d.}, \tag{2}$$

- ε_i ein n-dimensionaler latenter Zufallsvektor ist, der *Zufallsfehler des $i$ten Clusters* genannt wird, mit

$$\varepsilon_i \sim N(0_{n_i}, \Sigma_\varepsilon) \text{ mit } \Sigma_\varepsilon \in \mathbb{R}^{n_i \times n_i} \text{ p.d. und unabhängig von } b_j \text{ für } j = 1, \ldots, k. \tag{3}$$

Dann werden die k Gleichungen (11) *Linear Mixed Model in Clusterdarstellung* genannt.

Bemerkungen

- Häufig gelten $\Sigma_{b_i} := \sigma_b^2 I_{q_i}$ mit $\sigma_b^2 > 0$ und $\Sigma_{\varepsilon_i} := \sigma_\varepsilon^2 I_{n_i}$ mit $\sigma_\varepsilon^2 > 0$.
- Die LME Clusterdarstellung herrscht in der methodischen Literatur und Anwendungssoftware vor.

Definition (Linear Mixed Model in Kompaktdarstellung)

Gegeben sei die Clusterdarstellung eines Linear Mixed Models und mit $n := \sum_{i=1}^{k} n_i$, $q := \sum_{i=1}^{k} q_i$ seien

$$y := \begin{pmatrix} y_1 \\ \vdots \\ y_k \end{pmatrix}, \; X := \begin{pmatrix} X_1 \\ \vdots \\ X_k \end{pmatrix}, \; Z := \begin{pmatrix} Z_1 & \cdots & 0_{n_i k} \\ \vdots & \ddots & \vdots \\ 0_{n_i q_1} & \cdots & Z_k \end{pmatrix}, \; b := \begin{pmatrix} b_1 \\ \vdots \\ b_k \end{pmatrix}, \; \varepsilon := \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_k \end{pmatrix} \tag{4}$$

ein n-dimensionaler beobachtbarer Zufallsvektor, eine $n \times p$ vorgebene Fixed-Effects-Designmatrix, eine $n \times q$ vorgegebene Random-Effects-Designmatrix, ein q-dimensionaler latenter Zufallsvektor bzw. ein n-dimensionaler latenter Zufallsvektor. Dann wird

$$y = X\beta + Zb + \varepsilon \tag{5}$$

Kompaktdarstellung des Linear Mixed Models genannt und es gelten entsprechend

$$b \sim N(0_q, \Sigma_b) \text{ mit } \Sigma_b \in \mathbb{R}^{q \times q} \text{ p.d. und } \varepsilon \sim N(0_n, \Sigma_\varepsilon) \text{ mit } \Sigma_\varepsilon \in \mathbb{R}^{n \times n} \text{ p.d. und unabhängig von } b \tag{6}$$

und mit

$$\Sigma_b := \begin{pmatrix} \Sigma_{b_1} & \cdots & 0_{q_k q_k} \\ \vdots & \ddots & \vdots \\ 0_{q_1 q_1} & \cdots & \Sigma_{b_k} \end{pmatrix} \text{ und } \Sigma_\varepsilon := \begin{pmatrix} \Sigma_{\varepsilon_1} & \cdots & 0_{n_1 n_1} \\ \vdots & \ddots & \vdots \\ 0_{n_k n_k} & \cdots & \Sigma_{\varepsilon_k} \end{pmatrix}. \tag{7}$$

Bemerkungen

- Der Betaparametervektor der Kompaktdarstellung entspricht dem Betaparametervektor der Clusterdarstellung.
- In einem LME-Modell hat die Random-Effects-Designmatrix Z immer eine Blockdiagonalmatrixstruktur.

Modellformulierung

Anwendungsbeispiel

- Grundidee: Modellierung der Daten durch ein Zweistichproben-T-Test-Modell mit zufälligen HSA-Effekt
- Entsprechend des Zweistichproben-T-Test Modells in Effektdarstellung ergibt sich für HSA $i = 1, 2, 3, 4$

$$\begin{aligned} y_{i1j} &= \mu_0 + \mu_i + \varepsilon_{i1j} \qquad \text{mit } \varepsilon_{i1j} \sim N(0, \sigma_\varepsilon^2) \text{ für } j = 1, ..., n_{i_1} \\ y_{i2j} &= \mu_0 + \alpha_2 + \mu_i + \varepsilon_{i1j} \text{ mit } \varepsilon_{i2j} \sim N(0, \sigma_\varepsilon^2) \text{ für } j = 1, ..., n_{i_2} \end{aligned} \tag{8}$$

  und die entsprechende Datenverteilungsform

$$\begin{aligned} y_{i1j} &\sim N(\mu_0 + \mu_i, \sigma_\varepsilon^2) \qquad \text{u.i.v. für } j = 1, ..., n_{i_1} \text{ mit } \mu_0 \in \mathbb{R}, \sigma_\varepsilon^2 > 0 \\ y_{i2j} &\sim N(\mu_0 + \alpha_2 + \mu_i, \sigma_\varepsilon^2) \text{ u.i.v für } j = 1, ..., n_{i_2} \text{ mit } \mu_0, \alpha_2 \in \mathbb{R}, \sigma_\varepsilon^2 > 0 \end{aligned} \tag{9}$$

- Insbesondere sollen die Interceptparameter μ_i hier normalverteilte Zufallsvariablen sein, es soll gelten

$$\mu_i \sim N(0, \sigma_b^2) \text{ u.i.v. für } i = 1, 2, 3, 4. \tag{10}$$

- μ_0 und α_2 sind hier also Fixed Effects, μ_i für $i = 1, 2, 3, 4$ dagegen Random Effects

Anwendungsbeispiel

- Gemäß der Designmatrixform des Zweistichproben-T-Test-Modells in Effektdarstellung ergibt sich für das Linear Mixed Model in Clusterdarstellung für $i = 1, 2, 3, 4$ damit die Form

$$y_i = X_i\beta + Z_i b_i + \varepsilon_i \text{ mit } \varepsilon_i \sim N(0_{n_i}, \sigma^2_{eps} I_{n_i}, \sigma^2_{\varepsilon} I_{n_i}) \tag{11}$$

mit

$$y_i = \begin{pmatrix} y_{i11} \\ \vdots \\ y_{i1n_{i_1}} \\ y_{i21} \\ \vdots \\ y_{i2n_{i_2}} \end{pmatrix} \in \mathbb{R}^{n_i}, X_i := \begin{pmatrix} 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & 1 \end{pmatrix} \in \mathbb{R}^{n_i \times 2}, \beta := \begin{pmatrix} \mu_0 \\ \alpha_2 \end{pmatrix} \in \mathbb{R}^2, \tag{12}$$

sowie

$$Z_i := \begin{pmatrix} 1 \\ \vdots \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^{n_i \times 1} \text{ und } b_i := (\mu_i) \text{ mit } b_i \sim N(0, \sigma^2_b) \tag{13}$$

Anwendungsbeispiel

- Weiterhin ergibt sich für das Linear Mixed Model in Kompaktdarstellung

$$y = X\beta + Zb + \varepsilon \text{ mit } \varepsilon \sim N(0_n, \sigma_\varepsilon^2) \text{ und } b \sim N(0_4, \sigma_b^2 I_4) \tag{14}$$

sowie

$$y = \begin{pmatrix} y_{111} \\ \vdots \\ y_{11n_{1_1}} \\ y_{121} \\ \vdots \\ y_{12n_{1_2}} \\ y_{211} \\ \vdots \\ y_{21n_{2_1}} \\ y_{221} \\ \vdots \\ y_{22n_{2_2}} \\ y_{311} \\ \vdots \\ y_{31n_{3_1}} \\ y_{321} \\ \vdots \\ y_{32n_{3_2}} \\ y_{411} \\ \vdots \\ y_{41n_{4_1}} \\ y_{421} \\ \vdots \\ y_{42n_{4_2}} \end{pmatrix} \in \mathbb{R}^n,\ X = \begin{pmatrix} 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & 1 \\ 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & 1 \\ 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & 1 \\ 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & 1 \end{pmatrix} \in \mathbb{R}^{n\times 2},\ \beta := \begin{pmatrix} \mu_0 \\ \alpha_2 \end{pmatrix} \in \mathbb{R}^2,\ Z := \begin{pmatrix} 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{n\times 4},\ b := \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{pmatrix} \in \mathbb{R}^4.$$

(15)

Definition (Verteilungsdarstellung des Linear Mixed Models)

Gegeben sei ein Linear Mixed Model in Kompaktdarstellung,

$$y = X\beta + Zb + \varepsilon \text{ mit } b \sim N(0_q, \Sigma_b) \text{ und } \varepsilon \sim N(0_n, \Sigma_\varepsilon). \tag{16}$$

Dann nennt man die äquivalente Darstellung dieses Modells mit der marginalen Verteilung

$$b \sim N(0_q, \Sigma_b) \tag{17}$$

und der bedingten Verteilung

$$y \,|\, b \sim N(X\beta + Zb, \Sigma_\varepsilon) \tag{18}$$

die *Verteilungsdarstellung des Linear Mixed Models*.

Bemerkungen

- Die Äquivalenz folgt mit dem Theorem zur linear-affinen Transformation multivariater Normalverteilungen.
- Intuitiv beschreibt der Ausdruck $y = X\beta + Zb + \varepsilon$ eine bedingte Verteilung.
- Die Fehlerkovarianzmatrix Σ_ε ist die Kovarianzmatrix dieser bedingten Verteilung.

Theorem (Gemeinsame Verteilung des Linear Mixed Models)

Gegeben sei ein Linear Mixed Model in Kompaktdarstellung,

$$y = X\beta + Zb + \varepsilon \text{ mit } b \sim N(0_q, \Sigma_b) \text{ und } \varepsilon \sim N(0_n, \Sigma_\varepsilon). \tag{19}$$

Dann gilt für die gemeinsame Verteilung von Daten und Random Effects, dass

$$\begin{pmatrix} b \\ y \end{pmatrix} \sim N\left(\mu_{b,y}, \Sigma_{b,y}\right) \tag{20}$$

mit

$$\mu_{b,y} := \begin{pmatrix} 0_q \\ X\beta \end{pmatrix} \in \mathbb{R}^{q+n} \text{ und } \Sigma_{b,y} := \begin{pmatrix} \Sigma_b & \Sigma_b Z^T \\ Z\Sigma_b & Z\Sigma_b Z^T + \Sigma_\varepsilon \end{pmatrix} \in \mathbb{R}^{(q+n)\times(q+n)} \tag{21}$$

Beweis

Die gemeinsame Verteilung des Linear Mixed Models ergibt sich direkt durch Anwendung des Theorems zu Gemeinsamen Normalverteilungen auf die Verteilungsdarstellung des Linear Mixed Models.

□

Theorem (Marginale Datenverteilung des Linear Mixed Models)

Gegeben sei ein Linear Mixed Model in Kompaktdarstellung,

$$y = X\beta + Zb + \varepsilon \text{ mit } b \sim N(0_q, \Sigma_b) \text{ und } \varepsilon \sim N(0_n, \Sigma_\varepsilon). \tag{22}$$

Dann gilt für die marginale Verteilung der Daten, dass

$$y \sim N\left(\mu_y, \Sigma_y\right) \tag{23}$$

mit

$$\mu_y := X\beta \in \mathbb{R}^n \text{ und } \Sigma_y := Z\Sigma_b Z^T + \Sigma_\varepsilon \in \mathbb{R}^{n \times n} \tag{24}$$

Beweis

Die Aussage ergibt sich direkt aus dem Theorem zur Gemeinsamen Verteilung des LMMs und dem Theorem zu Marginalen Normalverteilungen. □

Bemerkungen

- LMMs erlauben es, nicht-sphärische Kovarianzmatrixstrukturen zu modellieren.
- Gilt speziell $\Sigma_b := \sigma_b^2 I_q, \sigma_b^2 > 0$ und $\Sigma_\varepsilon := \sigma_\varepsilon^2 I_n, \sigma_\varepsilon^2 > 0$, so folgt

$$y \sim N\left(X\beta, \sigma_b^2 ZZ^T + \sigma_\varepsilon^2 I_n\right) \tag{25}$$

- Parameter wie σ_b^2 und σ_ε^2 nennt man deshalb auch *Kovarianzkomponenten*.

Definition (Hierarchische Darstellung des Linear Mixed Models)

Gegeben sei ein Linear Mixed Model in Kompaktdarstellung,

$$y = X\beta + Zb + \varepsilon \text{ mit } b \sim N(0_q, \Sigma_b) \text{ und } \varepsilon \sim N(0_n, \Sigma_\varepsilon). \tag{26}$$

Dann nennt man die äquivalente Darstellung dieses Modells in der Form

$$\begin{aligned} b &= 0_q + \eta \qquad \text{mit } \eta \sim N(0_q, \Sigma_b) \;\Leftrightarrow\; b \sim N(0_q, \Sigma_b) \\ y &= X\beta + Zb + \varepsilon \;\text{ mit } \varepsilon \sim N(0_n, \Sigma_\varepsilon) \Leftrightarrow y \,|\, b \sim N(X\beta + Zb, \Sigma_\varepsilon) \end{aligned} \tag{27}$$

die *hierarchische Darstellung des Linear Mixed Models*

Bemerkung

- Man nennt diese Darstellung auch ein *Mehrebenenmodell*.
- Es ist leicht, sich LMMs mit mehr als den hier spezifizierten zwei Ebenen vorzustellen.
- Die obigen Verteilungsaussagen gelten natürlich auch für die Hierarchische Form des Linear Mixed Models.

Modellformulierung

Modellschätzung

Modellevaluation

Selbstkontrollfragen

Überblick zur Modellschätzung

Wie bereits gesehen, impliziert das Linear Mixed Model mit

$$\Sigma_b := \sigma_b^2 I_q \text{ und } \Sigma_\varepsilon := \sigma_\varepsilon^2 I_n \tag{28}$$

die gemeinsame Verteilung von Datenvektor und Random-Effects-Vektor

$$\begin{pmatrix} b \\ y \end{pmatrix} \sim N\left(\begin{pmatrix} 0_q \\ X\beta \end{pmatrix}, \begin{pmatrix} \sigma_b^2 I_q & \sigma_b^2 Z^T \\ \sigma_b^2 Z & \sigma_b^2 Z Z^T + \sigma_\varepsilon^2 I_n \end{pmatrix} \right), \tag{29}$$

sowie die marginale Datenverteilung

$$y \sim N\left(X\beta, \sigma_b^2 Z Z^T + \sigma_\varepsilon^2 I_n\right). \tag{30}$$

Mithilfe der Definitionen des *Varianzkomponentenvektors* θ und des marginalen Datenkovarianzmatrixparameters V_θ

$$\theta := \left(\sigma_b^2, \sigma_\varepsilon^2\right) \text{ und } V_\theta := \sigma_b^2 Z Z^T + \sigma_\varepsilon^2 I_n, \tag{31}$$

wird die marginale Datenverteilung des Linear Mixed Models häufig auch als

$$y \sim N\left(X\beta, V_\theta\right) \tag{32}$$

geschrieben. Das Schätzproblem für ein Linear Mixed Model hat dann drei zentrale Aspekte:

(1) Die Angabe eines Schätzers $\hat{\beta}$ für den Fixed-Effects-Parameter β.
(2) Die Angabe eines Schätzers $\hat{\theta}$ für den Varianzkomponentenparameter θ.
(3) Die Angabe eines Schätzers $\hat{b}$ für den Random-Effects-Parameter b.

Überblick zur Modellschätzung

Die Lösung dieses Problems ist nicht trivial und Gegenstand aktueller Forschung

Generell werden in der Anwendung iterative Verfahren genutzt

Wir beschreiben hier folgenden, der Funktion `lme()` aus dem R Paket `nlme` nahen Ansatz

(1) Schätzung von β basierend auf der geschätzten Marginalverteilung von y

$\Rightarrow$ V_θ wird durch $V_{\hat{\theta}}$ ersetzt und β durch den *Generalisierten-Kleinste-Quadrate Schätzer* geschätzt.

(2) Schätzung von θ basierend auf der geschätzten Marginalverteilung von y

$\Rightarrow$ θ wird iterativ mit dem *Restricted Maximum-Likelihood* Verfahren geschätzt.

(3) Schätzung von b basierend auf der geschätzten gemeinsamen Verteilung von b und y

$\Rightarrow$ V_θ und β werden durch $V_{\hat{\theta}}$ und $\hat{\beta}$ ersetzt und b durch seinen *bedingten Erwartungswert* geschätzt.

Überblick zur Modellschätzung

Iteratives Verfahren zur LMM Parameterschätzung

(0) Initialisierung

- Wahl eines geeigneten Startwerts $\widehat{\beta}^{(0)}$

(1) Für $k = 1, ..., K$

- ReML-Schätzung $\widehat{\theta}^{(k)}$ basierend auf $\widehat{\beta}^{(k-1)}$
- GLS-Schätzung $\widehat{\beta}^{(k)}$ basierend auf $\widehat{\theta}^{(k)}$

(2) Schätzung von $\widehat{b}$ basierend auf $\widehat{\beta}^{(K)}$ und $\widehat{\theta}^{(K)}$

Generalisierte Kleinste-Quadrate Schätzung der Fixed-Effects

- Kleinste-Quadrate Schätzung heißt auf English "Ordinary Least Squares" (OLS).
- Zur Abgrenzung nennen wir den bekannten Betaparameterschätzer im Folgenden "OLS-Schätzer".
- Generalisierte Kleinste-Quadrate Schätzung heißt auf English "Generalized Least Squares" (GLS).
- Der GLS-Schätzer ist ein Betaparameterschätzer für das ALM

$$y = X\beta + \varepsilon \text{ mit } \varepsilon \sim N\left(0_n, \sigma^2 V\right) \text{ mit } V \neq I_n \tag{33}$$

- Der GLS-Schätzer ist also im Fall nicht-sphärischer Fehlerkovarianzmatrixparameter angezeigt.
- Der GLS-Schätzer stellt sicher, dass T-Statistiken auch im Fall $V \neq I_n$ t-verteilt sind.
- Im Kontext der Fixed-Effects-Schätzung eines LMMs gilt in Hinblick auf obiges ALM speziell

$$X := X, \beta := \beta, \sigma^2 := \sigma_\varepsilon^2 \sigma_b^2, V := \frac{1}{\sigma_\varepsilon^2} Z Z^T + \frac{1}{\sigma_b^2} I_n \text{ und somit } \sigma^2 V = V_\theta. \tag{34}$$

Definition (Generalisierte Kleinste-Quadrate Schätzer)

Gegeben sei ein Allgemeines Lineares Modell der Form

$$y = X\beta + \varepsilon \text{ mit } \varepsilon \sim N\left(0_n, \sigma^2 V\right) \text{ mit} \tag{35}$$

mit $\sigma^2 > 0$ und einer positiv-definiten Matrix $V \in \mathbb{R}^{n \times n}$. Dann heißt

$$\hat{\beta}_{\text{GLS}} := \left(X^T V^{-1} X\right)^{-1} X^T V^{-1} y \tag{36}$$

der *Generalisierte-Kleinste-Quadrate-Schätzer von* β und

$$\hat{\sigma}^2_{\text{GLS}} := \frac{\left(y - X\hat{\beta}_{\text{GLS}}\right)^T V^{-1} \left(y - X\hat{\beta}_{\text{GLS}}\right)}{n - p} \tag{37}$$

der *Generalisierte-Kleinste-Quadrate-Schätzer von* σ^2.

Bemerkungen

- Es muss nicht notwendigerweise $V = I_n$ gelten.
- Die Fehlerkomponenten in ε können unterschiedliche Varianzen haben oder korreliert sein.
- Im Fall $V = I_n$ gilt weiterhin

$$\hat{\beta}_{\text{GLS}} = \left(X^T I_n^{-1} X\right)^{-1} X^T I_n^{-1} y = \left(X^T X\right)^{-1} X^T y =: \hat{\beta}_{\text{OLS}}. \tag{38}$$

Theorem (GLS-Schätzer und OLS-Schätzer)

Gegeben sei ein *untransformiertes ALM* der Form

$$y = X\beta + \varepsilon \text{ mit } \varepsilon \sim N\left(0_n, \sigma^2 V\right) \tag{39}$$

mit $\sigma^2 > 0$ und einer positiv-definiten Matrix $V \in \mathbb{R}^{n \times n}$ und es sei $\hat{\beta}_{\text{GLS}}$ der Generalisierte-Kleinste-Quadrate-Schätzer von β. Weiterhin sei $K \in \mathbb{R}^{n \times n}$ eine Matrix mit den Eigenschaften

$$KK^T = V \text{ und } \left(K^{-1}\right)^T K^{-1} = V^{-1} \tag{40}$$

Schließlich sei

$$y^* = X^*\beta + \varepsilon^* \text{ mit } y^* := K^{-1}y, X^* := K^{-1}X, \varepsilon^* := K^{-1}\varepsilon \tag{41}$$

das *transformierte ALM*. Dann gelten

(1) Der GLS-Schätzer des untransformierten ALMs ist der OLS-Schätzer des transformierten ALMs.

(2) Für den Zufallsfehler im transformierten ALM gilt $\varepsilon^* \sim N(0_n, \sigma^2 I_n)$.

Bemerkungen

- Der zu schätzende wahre, aber unbekannte, Parameter β ist in beiden ALMs identisch.
- Im transformierten ALM ist der Fehlerkovarianzmatrixparameter sphärisch, also T-Statistiken t-verteilt.
- Man nennt die Transformation des ALMs durch K auch eine "Whitening-Transformation".
- K mit den geforderten Eigenschaften kann durch die *Cholesky-Zerlegung* von V gewonnen werden.

Beweis

(1) Für den GLS-Schätzer im untransformierten Modell gilt

$$\begin{aligned}
\hat{\beta}_{\text{GLS}} &= \left(X^T V^{-1} X\right)^{-1} X^T V^{-1} y \\
&= \left(X^T \left(K^{-1}\right)^T K^{-1} X\right)^{-1} X^T \left(K^{-1}\right)^T K^{-1} y \\
&= \left(\left(K^{-1} X\right)^T K^{-1} X\right)^{-1} \left(K^{-1} X\right)^T K^{-1} y \\
&= \left(X^{*T} X^*\right)^{-1} X^{*T} y^*.
\end{aligned} \tag{42}$$

Dies aber entspricht dem OLS-Schätzer im transformierten Modell.

(2) Mit der Tatsache, dass für eine invertierbare Matrix A immer gilt, dass $\left(A^{-1}\right)^T = \left(A^T\right)^{-1}$ und dem Theorem zur linear-affinen Transformation multivariater Normalverteilungen ergibt sich

$$\begin{aligned}
\varepsilon^* &\sim N\left(K^{-1} 0_n, K^{-1}(\sigma^2 V) K^{-1^T}\right) \\
&= N\left(0_n, \sigma^2 K^{-1} V K^{-1^T}\right) \\
&= N\left(0_n, \sigma^2 K^{-1} K K^T K^{-1^T}\right) \\
&= N\left(0_n, \sigma^2 K^{-1} K K^T K^{T^{-1}}\right) \\
&= N\left(0_n, \sigma^2 I_n\right).
\end{aligned} \tag{43}$$

□

Definition (Fixed-Effects-Parameterschätzer)

Gegeben sei die marginale Datenverteilung eines LMMs mit einem Schätzer $\hat{\theta}$ für θ, also

$$y \sim N(X\beta, V_{\hat{\theta}}). \tag{44}$$

Dann ist der GLS-Schätzer

$$\hat{\beta} = \left(X^T V_{\hat{\theta}}^{-1} X\right)^{-1} X^T V_{\hat{\theta}}^{-1} y \tag{45}$$

ein populärer Schätzer für den Fixed-Effects-Parameter β.

Implementation des Fixed-Effects-Parameterschätzers

```
gls = function(y, X, V){
    # Diese Funktion bestimmt den generalisierten Kleinste-Quadrate-Schätzer.
    #
    # Inputs
    #   y           : y x 1 Datenvektor
    #   X           : n x p Designamtrix
    #   V           : n x n marginale Datenkovarianzmatrix
    #
    # Outputs
    #   beta_hat    : p x 1 generalisierter Kleinste-Quadrate-Schätzer
    # ------------------------------------------------------------------------
    Vi          = solve(V)                                          # Inverse
    beta_hat    = solve(t(X) %*% Vi %*% X) %*% t(X) %*% Vi %*% y    # GKQ Schätzer
    return(beta_hat)                                                # Output
}
```

Motivation von Varianzkomponentenschätung und Restricted Maximum-Likelihood

Die Maximum-Likelihood Methode kann auf verzerrte Varianzschätzer führen

Zum Beispiel ist der Maximum-Likelihood-Schätzer des Varianzparameters des Normalverteilungsmodells

$$\hat{\sigma}^2_{\text{ML}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{\mu}_{\text{ML}})^2 \text{ mit } \hat{\mu}_{\text{ML}} := \frac{1}{n}\sum_{i=1}^{n} y_i =: \bar{y}. \tag{46}$$

verzerrt und nur asymptotisch erwartungstreu. Speziell gilt mit der Erwartungstreue der Stichprobenvarianz

$$\mathbb{E}\left(\hat{\sigma}^2\right) = \mathbb{E}\left(\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2\right) = \frac{1}{n-1}\mathbb{E}\left(\sum_{i=1}^{n}(y_i - \bar{y})^2\right) = \sigma^2, \tag{47}$$

dass

$$\mathbb{E}\left(\hat{\sigma}^2_{\text{ML}}\right) = \mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{\mu}_{\text{ML}})^2\right) = \frac{1}{n}\mathbb{E}\left(\sum_{i=1}^{n}(y_i - \bar{y}_n)^2\right) = \frac{n-1}{n}\sigma^2. \tag{48}$$

Da $(n-1)/n < 1$ insbesondere bei kleinem n, unterschätzt der Maximum-Likelihood-Schätzer den Wert von σ^2.

Patterson and Thompson (1971) schreiben "The difference between the two methods [ML und ReML] is analogous to the well-known difference between two methods of estimating the variance σ^2 of a normal distribution [wie oben] (...)." und Harville (1977) merkt an "One criticism of the ML approach to the estimation of [σ^2] is that the ML estimator (...) takes no account of the loss in degrees of freedom that results from estimating [μ] (...) These"deficiencies" are eliminated in the restricted Maximum-Likelihood (REML) approach (...)"

$\Rightarrow$ ReML für Varianzparameterschätzung scheint eine gute Idee zu sein.

Referenzen zur Theorie der Restricted Maximum-Likelihood Schätzung

Patterson and Thompson (1971)

- Fehlerkontrastmotivation der Restricted Maximum-Likelihood Zielfunktion

Harville (1977)

- Übersicht zu Restricted Maximum-Likelihood Methoden und numerischer Auswertung

Searle, Casella, and McCulloch (1992)

- Ausführliche Übersicht zum Problem der Varianzkomponentenschätzung

Bates and DebRoy (2004)

- Integration von Restricted Maximum-Likelihood in Penalized Least Squares

Starke and Ostwald (2017)

- Expectation-Maximization und Restricted Maximum-Likelihood aus der Perspektive von Variational Inference

Maximum-Likelihood und Restricted Maximum-Likelihood

Betrachtet man die marginale Datenverteilung des LMMs

$$y \sim N(X\beta, V_\theta) \tag{49}$$

so ergibt sich für die Log-Likelihood Funktion der Varianzkomponenten θ für einen Schätzer $\hat{\beta}$ von β

$$\begin{aligned}\ell(\theta) &= \ln\left((2\pi)^{-\frac{n}{2}}|V_\theta|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(y-X\hat{\beta})^T V_\theta^{-1}(y-X\hat{\beta})\right)\right) \\ &= -\frac{n}{2}\ln 2\pi - \frac{1}{2}\ln|V_\theta| - \frac{1}{2}(y-X\hat{\beta})^T V_\theta^{-1}(y-X\hat{\beta})\end{aligned} \tag{50}$$

Die (numerische) Maximierung dieser Funktion hinsichtlich θ führt zu einem Maximum-Likelihood-Schätzer von θ,

$$\hat{\theta}_{\text{ML}} := \text{argmax}_\theta\left(-\frac{1}{2}\ln|V_\theta| - \frac{1}{2}(y-X\hat{\beta})^T V_\theta^{-1}(y-X\hat{\beta})\right). \tag{51}$$

Ein zentrales Resultat ist, dass ein Restricted Maximum-Likelihood-Schätzer von θ gegeben ist durch

$$\hat{\theta}_{\text{ReML}} := \text{argmax}_\theta\left(-\frac{1}{2}\ln|V_\theta| - \frac{1}{2}\ln|X^T V_\theta^{-1} X| - \frac{1}{2}(y-X\hat{\beta})^T V_\theta^{-1}(y-X\hat{\beta})\right). \tag{52}$$

Die Zielfunktion der ReML Methode und der ML Methode unterscheiden sich also nur hinsichtlich eines Terms.

Im Folgenden wollen wir die Motivation für die Einführung des Terms $-\frac{1}{2}\ln|X^T V_\theta^{-1} X|$ (sehr) grob skizzieren.

Motivation der Restricted Maximum-Likelihood Funktion durch Fehlerkontraste

Grundidee des von Patterson and Thompson (1971) formulierten Ansatzes ist es, den Effekt von β aus y herauszurechnen und dann die Likelihood-Funktion der so transformierten Daten hinsichtlich von θ zu maximieren.

Genauer ist das Ziel den Datenvektor durch eine lineare Transformation mit einer Matrix $M \in \mathbb{R}^{m \times n}$ in einen anderen Vektor z zu transformieren, dessen Erwartungswert für jeden möglichen Wert von β der Nullvektor ist, also

$$z = My \text{ mit } \mathbb{E}(z) = 0_m \text{ für alle } \beta \in \mathbb{R}^p \tag{53}$$

und dann die Log-Likelihood-Funktion von z zu maximieren.

Eine solche Matrix M muss insbesondere die Bedingung

$$MX = 0_{mp} \tag{54}$$

erfüllen, denn dann gilt

$$\mathbb{E}(z) = \mathbb{E}(My) = \mathbb{E}(MX\beta) = \mathbb{E}(0_{mp}\beta) = 0_m \text{ für alle } \beta \in \mathbb{R}^p. \tag{55}$$

Eine prinzipielle Möglichkeit für die Wahl von M ist die $n \times n$ Matrix

$$M = I_n - P_n \text{ mit } P_n := X(X^T X)^{-1} X^T \in \mathbb{R}^{n \times n} \tag{56}$$

mit der sogenannten *Projektionsmatrix* P.

Motivation der Restricted Maximum-Likelihood Funktion durch Fehlerkontraste

Es gilt dann nämlich

$$MX = (I_n - P_n)X = X - P_n X = X - X(X^T X)^{-1} X^T X = X - X = 0_{np}. \tag{57}$$

Nutzt man also diese Matrix M zur Transformation der Daten, ergibt sich

$$z = My = (I_n - P_n)y = y - X(X^T X)^{-1} X^T y = y - X\hat{\beta} = \hat{\varepsilon} \tag{58}$$

und wir sehen, dass eine solche Matrix M die Daten auf die Residuals, also die Differenz zwischen Daten und Modellvorhersage nach Schätzung der Fixed-Effects projiziert. Die Matrix P_n nennt man dementsprechend auch *Residual-forming matrix* oder *Projektionsmatrix* und die Matrix M *Fehlerkontrastmatrix*. Der Vektor z sind dann die Residuals und ReML wird auch häufig als *Residual Maximum-Likelihood* bezeichnet. Eine Zeile einer solchen Matrix M nennt man auch *Fehlerkontrast*, die Matrix M daher eine *Fehlerkontrastmatrix*.

Prinzipiell würde man nun die Log-Likelihood Funktion von $z \in \mathbb{R}^n$, das aufgrund des Theorems zur linear-affinen Transformation multivariater Normalverteilungen die Verteilung

$$z \sim N(MX\beta, MV_\theta M^T) \tag{59}$$

hat, also

$$\ell(\theta) = \frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |MV_\theta M^T| - \frac{1}{2} (My)^T (MV_\theta M^T)^{-1} My. \tag{60}$$

Motivation der Restricted Maximum-Likelihood Funktion durch Fehlerkontraste

Leider funktioniert die vorgeschlagene Wahl von M in dieser Form nicht, "da rg$(M) = m < n$".

Man wählt daher die ersten $n-p$ Zeilen von M und erhält eine Matrix $K \in \mathbb{R}^{m \times n}$ mit vollem Spaltenrang $m = n-p$.

Dabei gilt weiterhin $\mathbb{E}(z) = \mathbb{E}(Ky) = 0_m$ und man möchte

$$\ell(\theta) = \frac{m}{2} \ln 2\pi - \frac{1}{2} \ln |KV_\theta K^T| - \frac{1}{2}(Ky)^T (KV_\theta K^T)^{-1} Ky \tag{61}$$

maximieren.

Searle, Casella, and McCulloch (1992) beweisen nun, dass

$$\ln |KV_\theta K^T| = \ln |V_\theta| + \ln |XV_\theta^{-1} X| \tag{62}$$

und

$$(Ky)^T (KV_\theta K^T)^{-1} Ky = y^T P_n y = (y - X\hat{\beta})^T V_\theta^{-1} (y - X\hat{\beta}) \tag{63}$$

Dies ist intuitiv zumindest unter dem Aspekt, dass K Teil von P_n ist, einsichtig. Damit ergibt sich für die Log-Likelihood-Funktion von $z = Ky$ aber, dass

$$\ell(\theta) = \frac{m}{2} \ln 2\pi - \frac{1}{2} \ln |V_\theta| - \frac{1}{2} \ln |XV_\theta^{-1} X| - \frac{1}{2}(y - X\hat{\beta})^T V_\theta^{-1} (y - X\hat{\beta}) \tag{64}$$

also identisch mit der ReML Zielfunktion ist.

Modellschätzung

Die hier gegebene Darstellung lässt allerdings viele Fragen offen

- Warum sind ReML Schätzer der Varianzkomponenten unverzerrt (vgl. Foulley (1993))?
- Was sind weitere generelle Eigenschaften der ReML Schätzer (vgl. Harville (1977))
- Was genau ist das Problem bei Residualprojektion mit $M = (I_n - P_n)$?
- Wie und wann funktionieren die Beweise von Searle, Casella, and McCulloch (1992)?

Darüber hinaus ergeben sich zumindest folgende Fragen

- Was verhält sich die Fehlerkontrastmotivation zur Expectation-Maximization Motivation (vgl. Laird (1982))?
- Wie verhält sich die Fehlerkontrastmotivation zur bedingten Verteilungsmotivation (vgl. Verbyla (1990))?
- Welche Algorithmen eignen sich zur Maximierung der ReML Zielfunktion (vgl. Lindstrom and Bates (1990))?
- Wie verhalten sich ReML und Penalized-Least-Squares (vgl. Bates and DebRoy (2004))?

Definition (ReML-Varianzkomponentenschätzer)

Gegeben sei die marginale Datenverteilung eines LMMs basierend auf einem Fixed-Effects-Parameterschätzer $\hat{\beta}$, also

$$y \sim N(X\hat{\beta}, V_\theta) \tag{65}$$

Dann ist der ReML Schätzer in diesem Modell,

$$\hat{\theta}_{\text{ReML}} := \text{argmax}_\theta \left(-\frac{1}{2} \ln |V_\theta| - \frac{1}{2} \ln |X^T V_\theta^{-1} X| - \frac{1}{2} (y - X\hat{\beta})^T V_\theta^{-1} (y - X\hat{\beta}) \right), \tag{66}$$

ein populärer Schätzer für den Varianzkomponentenvektor θ.

Implementation des ReML-Varianzkomponentenschätzers

```
llh_reml = function(theta, y, X, Z, beta_hat, Sigma_eps){
    # Diese Funktion evaluiert die negative restricted log likelihood
    # Zielfunktion für  das Random-Effects-Modell der Metanalyse.
    #
    # Inputs
    #   theta     : k x 1 Varianzkomponentenvektor
    #   y         : n x 1 Datenvektor
    #   X         : n x p Fixed-Effects-Designmatrix
    #   Z         : n x q  Random-Effects-Designmatrix
    #   beta_hat  : p x 1 Fixed-Effects Schätzer
    #
    # Outputs
    #   llh_reml    : 1 x 1 Wert der ReML Zielfunktion
    # ------------------------------------------------------------------------
    n           = nrow(Z)                                          # Datenpunktanzahl
    V           = theta[1]*Z %*% t(Z) + theta[2]*diag(n)           # marginale Datenkovarianzmatrix
    Vi          = solve(V)                                         # Inverse
    R           = y - X%*%beta_hat                                 # Residuals
    T1          = -(1/2)*log(det(V))                               # Erster Term
    T2          = -(1/2)*log(det(t(X) %*% Vi %*% X))               # Zweiter Term
    T3          = -(1/2)*t(R) %*% Vi %*% R                         # Dritter Term
    llh_reml    = T1 + T2 + T3                                     # Restricted Log Likelihood
    return(llh_reml)                                               # Wert der ReML Zielfunktion
}
```

Implementation des ReML-Varianzkomponentenschätzers

```
reml  = function(theta, lmm){
    # Diese Funktion ist eine Wrapperfunktion für l_reml() zum Gebrauch mit
    # der generischen R Optimierungsfunktion optim().
    #
    # Inputs
    #   theta   : k x 1 Varianzkomponentenvektor
    #   lmm     : Liste von LMM Komponenten
    #
    # Output
    #   reml    : Wert der restricted log likelhood Funktion
    # ------------------------------------------------------------------------
    y           = lmm$y                                                  # Datenvektor
    X           = lmm$X                                                  # Fixed-Effects-Designmatrix
    Z           = lmm$Z                                                  # Random-Effects-Designmatrix
    beta_hat    = lmm$beta_hat                                           # Fixed-Effects Schätzer
    l_reml      = llh_reml(theta,y,X,Z,beta_hat)                         # Wert der ReML Zielfunktion
    return(-l_reml)                                                      # Ausgabeargument
}

mcov = function(theta, Z){
    # Diese Funktion schätzt generierte eine marginale Datenkovarianzmatrix
    # basierend auf einer Random-Effects-Designmatrix und der Varianzkomponenten.
    #
    # Inputs:
    #   theta   : c x 1 Varianzkomponentenvektor
    #   Z       : n x q Random-Effects-Designmatrix
    #
    # Outputs
    #   V_theta : n x n marginale Kovarianzmatrix
    # ------------------------------------------------------------------------
    V_theta = theta[1]*(Z%*%t(Z)) + theta[2]*diag(nrow(Z))                   # marginale Datenkovarianzmatrix
    return(V_theta)                                                      # Ausgabeargument
}
```

Motivation zum Random-Effects Parameterschätzer

Das Linear Mixed Model impliziert wie gesehen eine gemeinsame Verteilung von Daten und unbeobachtbarem Random-Effects-Vektor b. Ein Standardvorgehen im Bereich der Linear Mixed Model Schätzung ist es, b durch den Erwartungswert der auf den Daten bedingten Verteilung von b zu schätzen, wobei unbekannte Parameterwerte wiederrum durch ihre Schätzer ersetzt werden.

Dieses Vorgehen entspricht damit letztlich einer Bayesianischen Punktschätzung von b mit marginaler Verteilung ("Prior distribution")

$$b \sim N\left(0_q, \hat{\sigma}_b^2 I_q\right) \tag{67}$$

und bedingter Verteilung ("Likelihood")

$$y \,|\, b \sim N\left(X\hat{\beta} + Zb, \hat{\sigma}_\varepsilon^2 I_n\right) \tag{68}$$

Anwendung des Theorems zur bedingten Normalverteilungen auf die hier relevante gemeinsame Verteilung

$$\begin{pmatrix} b \\ y \end{pmatrix} \sim N\left(\begin{pmatrix} 0_q \\ X\hat{\beta} \end{pmatrix}, \begin{pmatrix} \hat{\sigma}_b^2 I_q & \hat{\sigma}_b^2 Z^T \\ \hat{\sigma}_b^2 Z & V_{\hat{\theta}} \end{pmatrix}\right) \tag{69}$$

ergibt dann als Schätzer für den Random-Effects Parameter den Erwartungswertparameter der Verteilung $b \,|\, y$

$$\hat{b} = \mu_{b|y} = \hat{\sigma}_b^2 Z^T V_{\hat{\theta}}^{-1}(y - X\hat{\beta}). \tag{70}$$

Definition (Random-Effects-Parameterschätzer)

Gegeben sei die gemeinsame Verteilung von Random-Effects und Daten eines LMMs basierend auf einem Fixed-Effects-Parameterschätzer $\hat{\beta}$ und einem Varianzkomponentenschätzer $\hat{\theta} := \left(\hat{\sigma}_b^2, \hat{\sigma}_\varepsilon^2\right)$, also

$$\begin{pmatrix} b \\ y \end{pmatrix} \sim N\left(\begin{pmatrix} 0_q \\ X\hat{\beta} \end{pmatrix}, \begin{pmatrix} \hat{\sigma}_b^2 I_q & \hat{\sigma}_b^2 Z^T \\ \hat{\sigma}_b^2 Z & V_{\hat{\theta}} \end{pmatrix}\right). \tag{71}$$

Dann ist bedingte Erwartungswert von b in diesem Modell, also

$$\hat{b} = \mu_{b|y} = \hat{\sigma}_b^2 Z^T V_{\hat{\theta}}^{-1}(y - X\hat{\beta}). \tag{72}$$

ein populärer Schätzer für den Random-Effects-Parameter.

Implementation des Random-Effects-Parametschätzers

```
rfx = function(lmm){
    # Diese Funktion bestimmt den bedingten Erwartungswert der Random-Effects.
    #   Inputs :
    #     lmm   : R Liste mit Einträgen
    #       $y              : n x 1 Datenvektor
    #       $X              : n x p Fixed-Effects-Designmatrix
    #       $Z              : n x q Random-Effects-Designmatrix
    #       $beta_hat       : p x 1 Fixed-Effects-Parameterschätzer
    #       $s_b_hat        : 1 x 1 Random-Effects-Varianzkomponente
    #       $s_eps_hat      : 1 x 1 Fehler-Varianzkomponente
    #   Outputs :
    #     lmm               : R Liste mit zusätzlichen Einträgen
    #       $b_hat          : q x 1 Random-Effects-Parameterschätzer
    # ------------------------------------------------------------------
    y               = lmm$y                                          # Daten
    Z               = lmm$Z                                          # Random-Effects-Designmatrix
    X               = lmm$X                                          # Fixed-Effects-Designmatrix
    beta_hat        = lmm$beta_hat                                   # Fixed-Effects-Parameterschätzer
    s_b_hat         = lmm$s_b_hat                                    # Random-Effects-Varianzkomponentenschätzer
    s_eps_hat       = lmm$s_eps_hat                                  # Fehlervarianzkomponentenschätzer
    theta_hat       = c(s_b_hat,s_eps_hat)                           # Varianzkomponentenschätzer
    V_theta_hat_i   = solve(mcov(theta_hat, Z))                      # Inverser Datenkovarianzmatrixschätzer
    eps_hat         = (y - X %*% beta_hat)                           # Residuals
    lmm$b_hat       = s_b_hat*t(Z) %*% V_theta_hat_i %*% eps_hat     # Random-Effects-Parameterschätzer
}
```

Überblick zur Modellschätzung

Iteratives Verfahren zur LMM Parameterschätzung

(0) Initialisierung

- Wahl eines geeigneten Startwerts $\widehat{\beta}^{(0)}$

(1) Für $k = 1, ..., K$

- ReML-Schätzung $\widehat{\theta}^{(k)}$ basierend auf $\widehat{\beta}^{(k-1)}$
- GLS-Schätzung $\widehat{\beta}^{(k)}$ basierend auf $\widehat{\theta}^{(k)}$

(2) Schätzung von $\widehat{b}$ basierend auf $\widehat{\theta}^{(K)}$ und $\widehat{\theta}^{(K)}$

Modellschätzung

Iteratives Verfahren zur LMM Parameterschätzung

```
estimate = function(lmm){
    # Diese Funktion schätzt die Parameter eines LMMs.
    #   Inputs :
    #     lmm   : R Liste mit Einträgen
    #       $y              : n x 1 Datenvektor
    #       $X              : n x p Fixed-Effects-Designmatrix
    #       $Z              : n x q Random-Effects-Designmatrix
    #       $c              : 1 x 1 Varianzkomponentenanzahl
    #   Outputs :
    #     lmm   : R Liste mit zusätzlichen Einträgen
    #       $beta_hat       : p x 1 Fixed-Effects-Parameterschätzer
    #       $s_b_hat        : 1 x 1 Random-Effects-Varianzschätzer
    #       $s_eps_hat      : 1 x 1 Datenvarianzschätzer
    # ------------------------------------------------------------------------
    y               = lmm$y                                                  # Datenvektor
    X               = lmm$X                                                  # Fixed-Effects-Designmatrix
    Z               = lmm$Z                                                  # Random-Effects-Designmatrix
    c               = lmm$c                                                  # Anzahl Varianzkomponenten
    n               = nrow(X)                                                # Anzahl Datenpunkte
    p               = ncol(X)                                                # Anzahl Fixed-Effects
    q               = ncol(Z)                                                # Anzahl Random-Effects
    K               = 2^3                                                    # maximale Iterationsanzahl
    theta_hat_k     = matrix(rep(NaN, c*K), nrow = c)                        # Varianzkomponentenschätzerarray
    theta_hat_k[,1] = rep(1,c)                                               # Initialisierung
    beta_hat_k      = matrix(rep(NaN, p*K), nrow = p)                        # Fixed-Effects-Schätzerarray
    V_theta_hat_k   = mcov(theta_hat_k[,1], Z)                               # marginale Datenkovarianzmatrix
    beta_hat_k[,1]  = gls(y,X,V_theta_hat_k)                                 # Fixed-Effects-Parameterschätzer
    for (k in 2:K){                                                          # Iterationen
        lmm$beta_hat    = beta_hat_k[, k-1]                                  # Fixed-Effects-Schätzer k-1
        max_l_reml      = optim(par=theta_hat_k[,k-1],fn=reml,lmm=lmm)       # ReML-Varianzkomponentschätzung
        theta_hat_k[,k] = max_l_reml$par                                     # Varianzkomponentenschätzer k
        V_theta_hat_k   = mcov(theta_hat_k[,k], Z)                           # marginale Datenkovarianzmatrix
        beta_hat_k[,k]  = gls(y,X,V_theta_hat_k)}                            # Fixed-Effects-Parameterschätzer
    lmm$beta_hat    = beta_hat_k[,K]                                         # Fixed-Effects-Parameterschätzer
    lmm$s_b_hat     = theta_hat_k[1,K]                                       # Random-Effects-Varianzkomponente
    lmm$s_eps_hat   = theta_hat_k[2,K]                                       # Fehler-Varianzkomponente
    lmm$b_hat       = rfx(lmm)                                               # Random-Effects-Parameterschätzer
    return(lmm)                                                              # Ausgabe
}
```

Parameterschätzung im Anwendungsbeispiel

```
D       = read.csv("./3_Daten/mz-anova.csv")                    # Dateneinlesen
n_i     = 4                                                     # Anzahl Zentren
n_j     = 2                                                     # Anzahl Bedingungen
n_ij    = 5                                                     # Patient:innen pro Zentrum und Bedingung
n       = n_i*n_j*n_ij                                          # Gesamtanzahl an Patient:innen
X_i     = kronecker(matrix(c(1,1,0,1), ncol = 2), rep(1,n_ij))  # Treatment-Control Designmatrix
X       = kronecker(rep(1,n_i), X_i)                            # Fixed-Effects-Designmatrix
Z       = kronecker(diag(n_i), rep(1,n_j*n_ij))                 # Random-Effects-Designmatrix
y       = D$BDI                                                 # Daten
lmm     = list(y = y, X = X, Z = Z, c = 2)                      # LMM Komponenten
lmm     = estimate(lmm)                                         # Modellschätzung
```

```
beta_hat         : 5.01 2.1
b_hat            : 0.1 1.97 -2.36 0.3
sigsqr_b_hat     : 3.26
sigsqr_eps_hat   : 0.77
```

Parameterschätzung im Anwendungsbeispiel mit `lme()` aus `nlme`

```r
library(nlme)                                          # nlme R Paket
D           = read.csv("./3_Daten/mz-anova.csv", head = T)   # Dataframe
D$TRM       = as.factor(D$TRM)                         # R Faktor Kodierung
D$HSA       = as.factor(D$HSA)                         # R Faktor Kodierung
M           = lme(BDI ~ TRM, data = D, random = ~ 1 | HSA)   # LMM Schätzung
X           = model.matrix(M,D)                        # Fixed-Effects-Designmatrix
Z           = model.matrix(~ M$groups[[1]] - 1)        # Random-Effects-Designmatrix
beta_hat    = M$coefficients$fixed                     # Fixed-Effects-Parameterschätzer
b_hat       = M$coefficients$random$HSA                # Random-Effects-Parameterschätzer
s_eps_hat   = M$sigma**2                               # Datenvarianzkomponentenschätzer
s_b_hat     = diag(getVarCov(M))                       # Random-Effects-Varianzkomponentenschätzer
ci_beta     = intervals(M, which = "fixed", 0.95)      # Konfidenzintervalle
```

```
beta_hat         : 5.01 2.1
b_hat            : 0.1 1.97 -2.36 0.3
sigsqr_b_hat     : 3.26
sigsqr_eps_hat   : 0.77
```

Modellformulierung

Modellschätzung

Modellevaluation

Selbstkontrollfragen

Überblick

- Im Gegensatz zum ALM gibt es nicht für alle LMM Parameterschätzer nicht-iterative Lösungen.
- $\Rightarrow$ Die Frequentistische Verteilungstheorie der LMM Parameter ist komplex.
- Wir sind hier nur an Konfidenzintervallen für die Fixed Effects Parameter interessiert.
- Wir fokussieren dazu auf eine Approximation der Frequentistischen Verteilung von $\hat{\beta}$.
- Diese Approximation wird in Rencher and Schaalje (2008) basierend auf Fuller and Battese (1974) diskutiert.
- Demidenko (2013) erwähnt diese Approximation nur nebenbei.
- Software-Lösungen wie `nlme` und `lme4` implementieren diese Approximation nicht notwendigerweise.
- Alternative Möglichkeiten sind z.B. Profil-Likelihood-Konfidenzintervalle (vgl. Venzon and Moolgavkar (1988)).

Theorem (Approximative Konfidenzintervalle für Fixed-Effect-Parameter)

Für ein LMM sei

$$\hat{\beta} = \left(X^T V_{\hat{\theta}}^{-1} X \right)^{-1} X^T V_{\hat{\theta}}^{-1} y \tag{73}$$

der Fixed-Effects-Parameterschätzer. Dann gilt für $n \to \infty$, dass

$$\hat{\beta} \overset{a}{\sim} N\left(\beta, \left(X^T V_{\hat{\theta}}^{-1} X \right)^{-1} \right). \tag{74}$$

Weiterhin gilt mit der kumulativen Verteilungsfunktion Φ der Standardnormalverteilung,

$$z_\delta := \Phi^{-1}\left(\frac{1+\delta}{2} \right). \tag{75}$$

sowie

$$\lambda_j := \left(\left(X^T V_{\hat{\theta}}^{-1} X \right)^{-1} \right)_{jj}, \text{ dem } j\text{ten Diagonalelement von } \left(X^T V_{\hat{\theta}}^{-1} X \right)^{-1}, \tag{76}$$

dass für $j = 1, \ldots p$

$$\kappa_j := \left[\hat{\beta}_j - \sqrt{\lambda_j} z_\delta, \hat{\beta}_j + \sqrt{\lambda_j} z_\delta \right] \tag{77}$$

ein approximatives δ-Konfidenzintervall für die $j$te Komponente β_j des Fixed-Effects-Parameters ist.

Bemerkungen

- Wir verzichten auf einen Beweis und verweisen auf Rencher and Schaalje (2008), Seiten 490 - 491.

Approximative Fixed Effects Konfidenzintervalle

```
evaluate = function(lmm){

    # Diese Funktion evaluiert ein geschätztes LMMs.
    #   Inputs:
    #       .lmm    : R Liste mit Einträgen
    #        $y          : n x 1 Datenvektor
    #        $X          : n x p Fixed-Effects-Designmatrix
    #        $Z          : n x q Random-Effects-Designmatrix
    #        $c          : 1 x 1 Varianzkomponentenanzahl
    #        $beta_hat   : p x 1 Fixed-Effects-Parameterschätzer
    #        $s_b_hat    : 1 x 1 Random-Effects-Varianzschätzer
    #        $s_eps_hat  : 1 x 1 Datenvarianzschätzer
    #   Outputs
    #       .lmm    : R Liste mit zusätzlichen Einträgen
    #        $C_beta_hat : p x p Fixed-Effects Kovarianzmatrixschätzer
    #        $se_beta_hat: p x 1 Fixed Effects Standardfehlerschätzer
    #        $ci_beta_hat: p x 2 Wald Fixed Effects Wald Konfidenzintervalle
    #
    # ------------------------------------------------------------------------
    y               = lmm$y                                         # Daten
    Z               = lmm$Z                                         # Random-Effects-Designmatrix
    X               = lmm$X                                         # Fixed-Effects-Designmatrix
    beta_hat        = lmm$beta_hat                                  # Fixed-Effects-Parameterschätzer
    s_b_hat         = lmm$s_b_hat                                   # Random-Effects-Varianzkomponentenschätzer
    s_eps_hat       = lmm$s_eps_hat                                 # Fehlervarianzkomponentenschätzer
    theta_hat       = c(s_b_hat,s_eps_hat)                          # Varianzkomponentenschätzer
    V_hat_i         = solve(mcov(theta_hat, Z))                     # inverser Datenkovarianzmatrixschätzer
    lmm$C_beta_hat  = solve(t(X) %*% V_hat_i %*% X)                 # Fixed-Effects Kovarianzmatrixschätzer
    lmm$se_beta_hat = sqrt(diag(lmm$C_beta_hat))                    # Fixed-Effects Standardfehlerschätzer
    lmm$delta       = 0.95                                          # Konfidenzlevel
    lmm$zdelta      = qnorm((1 + lmm$delta)/2)                      # Wald-KI (Rencher (2008) S. 491)
    lmm$kappa_u     = lmm$beta_hat - lmm$zdelta*lmm$se_beta_hat     # untere KI Grenzen
    lmm$kappa_o     = lmm$beta_hat + lmm$zdelta*lmm$se_beta_hat     # obere KI Grenzen
    return(lmm)
}
```

Approximative Fixed-Effects-Konfidenzintervalle im Anwendungsbeispiel

```
lmm     = evaluate(lmm)                                          # Modellevaluation
E       = data.frame("Std.Error"    = lmm$se_beta_hat,           # Ausgabeformatierung
                     "Value"        = lmm$beta_hat,
                     "lower"        = lmm$kappa_u,
                     "upper"        = lmm$kappa_o,
                     row.names = c("mu_hat", "alpha_2"))
print(E, digits = 3)
```

```
        Std.Error Value lower upper
mu_hat      0.923  5.01  3.20  6.82
alpha_2     0.277  2.10  1.56  2.64
```

Modellevaluation

Fixed-Effects Konfidenzintervalle im Anwendungsbeispiel mit `lme()` aus `nlme`

```
library(nlme)                                                    # nlme R Paket
D           = read.csv("./3_Daten/mz-anova.csv", head = T)       # Dataframe
D$TRM       = as.factor(D$TRM)                                   # R Faktor Kodierung
D$HSA       = as.factor(D$HSA)                                   # R Faktor Kodierung
M           = lme(BDI ~ TRM, data = D, random = ~ 1 | HSA)       # LMM Schätzung
se_beta     = summary(M)$tTable[, "Std.Error"]                   # Standardfehlerschätzer
ci_beta     = intervals(M, which = "fixed", 0.95)                # Konfidenzintervalle
print(se_beta, digits = 3)                                       # Ausgabe
```

```
(Intercept)        TRM2
      0.924       0.277
```

```
print(ci_beta, digits = 3)                                       # Ausgabe
```

```
Approximate 95% confidence intervals

 Fixed effects:
            lower est. upper
(Intercept)  3.13 5.01  6.88
TRM2         1.54 2.10  2.66
```

⇒ Offenbar nutzt `lme()` **nicht** die hier diskutierte Approximation (vgl. Pinheiro and Bates (2000), S. 92).

Modellformulierung

Modellschätzung

Modellevaluation

Selbstkontrollfragen

Selbstkontrollfragen

1. Geben Sie die Definition des LMMs in Clusterdarstellung wieder.
2. Geben Sie die Definition des LMMs in Kompaktdarstellung wieder.
3. Geben Sie das Theorem zur Marginalen Datenverteilung des LMMs wieder.
4. Geben Sie die Definition der hierarchischen Darstellung des LMMs wieder.
5. Skizzieren Sie ein iteratives Verfahren zur LMM Parameterschätzung.
6. Geben Sie die Definition des Generalisierten Kleinste-Quadrate Schätzers wieder.
7. Erläutern Sie den Zusammenhang von GLS- und OLS-Schätzern.
8. Geben Sie die Definition des Fixed-Effects-Parameterschätzers im LMM wieder.
9. Geben Sie die Definition des ReML-Varianzkomponentenschätzers im LMM wieder.
10. Geben Sie die Definition des Random-Effects-Parameterschätzers im LMM wieder.
11. Geben Sie das Theorem zu approximativen Konfidenzintervallen für Fixed-Effects-Parameter im LMM wieder.

Referenzen I

Bates, Douglas M, and Saikat DebRoy. 2004. "Linear Mixed Models and Penalized Least Squares." *Journal of Multivariate Analysis* 91 (1): 1–17. https://doi.org/10.1016/j.jmva.2004.04.013.

Demidenko, Eugene. 2013. *Mixed Models: Theory and Applications with R*. 2. ed. Wiley Series in Probability and Statistics. Hoboken, NJ: Wiley.

Foulley, JL. 1993. "A Simple Argument Showing How to Derive Restricted Maximum Likelihood."

Fuller, Wayne A., and George E. Battese. 1974. "Estimation of Linear Models with Crossed-Error Structure." *Journal of Econometrics* 2 (1): 67–78. https://doi.org/10.1016/0304-4076(74)90030-X.

Harville, David A. 1977. "Maximum Likelihood Approaches to Variance Component Estimation and to Related Problems." *Journal of the American Statistical Association* 72 (358): 320. https://doi.org/10.2307/2286796.

Laird, Nan M. 1982. "Computation of Variance Components Using the Em Algorithm." *Journal of Statistical Computation and Simulation* 14 (3-4): 295–303. https://doi.org/10.1080/00949658208810550.

Lindstrom, Mary J., and Douglas M. Bates. 1990. "Nonlinear Mixed Effects Models for Repeated Measures Data." *Biometrics* 46 (3): 673. https://doi.org/10.2307/2532087.

Patterson, H. D., and R. Thompson. 1971. "Recovery of Inter-Block Information When Block Sizes Are Unequal." *Biometrika* 58 (3): 545–54. https://doi.org/10.1093/biomet/58.3.545.

Pinheiro, José C., and Douglas M. Bates. 2000. *Mixed-Effects Models in S and S-PLUS*. Statistics and Computing. New York: Springer.

Rencher, Alvin C., and G. Bruce Schaalje. 2008. *Linear Models in Statistics*. 2nd ed. Hoboken, N.J: Wiley-Interscience.

Searle, Shayle, George Casella, and Charles E. McCulloch. 1992. *Variance Components*. Wiley Series in Probability and Mathematical Statistics. New York: Wiley.

Referenzen II

Starke, Ludger, and Dirk Ostwald. 2017. "Variational Bayesian Parameter Estimation Techniques for the General Linear Model." *Frontiers in Neuroscience* 11 (September). https://doi.org/10.3389/fnins.2017.00504.

Venzon, D. J., and S. H. Moolgavkar. 1988. "A Method for Computing Profile-Likelihood-Based Confidence Intervals." *Applied Statistics* 37 (1): 87. https://doi.org/10.2307/2347496.

Verbyla, A. P. 1990. "A Conditional Derivation of Residual Maximum Likelihood." *Australian Journal of Statistics* 32 (2): 227–30. https://doi.org/10.1111/j.1467-842X.1990.tb01015.x.