Tutorium Allgemeines Lineares Modell

BSc Psychologie SoSe 2022

14. Termin: (12) Multiple Regression

Belinda Fleischmann

Selbstkontrollfragen

1. Erläutern Sie das Anwendungsszenario und die Ziele der multiplen Regression.
2. Definieren Sie das Modell der multiplen Regression.
3. Erläutern Sie die Begriffe Regressor, Prädiktor, Kovariate und Feature im Rahmen der multiplen Regression.
4. Erläutern Sie, warum $\hat{\beta} \approx$ Regressorkovarianz^{-1}Regressordatenkovarianz gilt.
5. Erläutern Sie den Zusammenhang zwischen Betaparameterschätzern und partieller Korrelation in einem multiplen Regressionmodell mit Interzeptprädiktor und zwei kontinuierlichen Prädiktoren anhand der Formel

$$\hat{\beta}_1 = r_{y,x_1|x_2} \sqrt{\frac{1 - r^2_{y,x_2}}{1 - r^2_{x_1,x_2}}} \frac{s_y}{s_{x_1}}. \tag{1}$$

6. $X \in \mathbb{R}^{n \times 2}$ sei die Designmatrix eines multiplen Regressionsmodells mit zwei Prädiktoren und Betaparametervektor $\beta := (\beta_1, \beta_2)^2$. Geben Sie den Kontrastgewichtsvektor an, um die Nullhypothese $H_0 : \beta_1 = \beta_2$ mithilfe der T-Statistik zu testen.
7. Simulieren Sie einen Datensatz eines multiplen Regressionsmodells mit Interzept und zwei kontinuierlichen Regressoren $x_1, x_2 \in \mathbb{R}^n$, wobei $x_{i2} := ax_{i1} + \xi_i$ mit $\xi_i \sim N(0, \sigma^2_\xi)$ für $i = 1, ..., n$ sein soll. Wählen Sie für die Simulation des Datensatzes $y \in \mathbb{R}^n$ den wahren, aber unbekannten, Betaparametervektor $\beta = (0, 1, 0)^T$ und testen Sie die Nullhypothesen $H_0 : \beta_j = 0$ für $j = 0, 1, 2$. Erläutern Sie Ihre Ergebnisse. Wiederholen Sie Analyse für den wahren, aber unbekannten, Betaparametervektor $\beta = (0, 0, 1)^T$.

Erläutern Sie das Anwendungsszenario und die Ziele der multiplen Regression.

Anwendungsszenario

- Generalisierung der einfachen linearen Regression zu mehr als einer unabhängigen Variable.
- Eine univariate abhängige Variable bestimmt an randomisierten experimentellen Einheiten.
- Zwei oder mehr "kontinuierliche" unabhängige Variablen.
- Die unabhängigen Variablen heißen Regressoren, Prädiktoren, Kovariaten oder Features.

Ziele

- Quantifizierung des Erklärungspotentials der Variation der AV durch die Variation der UVs.
- Quantifizierung des Einflusses einzelner UVs auf die AV im Kontext anderer UVs.
- Prädiktion von AV Werten aus UV Werten nach Parameterschätzung.

Anwendungsbeispiel

- BDI Differenzwerte in Abhängigkeit von Therapiedauer und Alter

Definieren Sie das Modell der multiplen Regression.

Definition (Modell der multiplen Regression)

y_i mit $i = 1, \dots, n$ sei die Zufallsvariable, die den $i$ten Wert einer abhängigen Variable modelliert. Dann hat das *Modell der multiplen Regression* die strukturelle Form

$$y_i = x_{i1}\beta_1 + \cdots + x_{ip}\beta_p + \varepsilon_i \text{ mit } \varepsilon_i \sim N(0, \sigma^2) \text{ u.i.v. für } i = 1, \dots, n \text{ und } \sigma^2 > 0, \tag{2}$$

wobei $x_{ij} \in \mathbb{R}$ mit $1 \leq i \leq n$ und $1 \leq j \leq p$ den $i$ten Wert der $j$te unabhängigen Variable bezeichnet. Die unabhängigen Variablen werden auch *Regressoren*, *Prädiktoren*, *Kovariaten* oder *Features* genannt. Mit

$$x_i := (x_{i1}, \dots, x_{ip})^T \in \mathbb{R}^p \text{ und } \beta := (\beta_1, \dots, \beta_p)^T \in \mathbb{R}^p \tag{3}$$

hat das Modell der multiplen Regression die Datenverteilungsform

$$y_i \sim N(\mu_i, \sigma^2) \text{ u.i.v. für } i = 1, \dots, n, \text{ wobei } \mu_i := x_i^T \beta. \tag{4}$$

In diesem Zusammenhang wird $x_i \in \mathbb{R}^p$ auch als $i$ter *Featurevektor* bezeichnet. Die Designmatrixform des Modells der multiplen Regression schließlich ist gegeben durch

$$y = X\beta + \varepsilon \text{ mit } \varepsilon \sim N(0_n, \sigma^2 I_n) \tag{5}$$

mit

$$y := (y_1, \dots, y_n)^T, X := (x_{ij})_{1 \leq i \leq n, 1 \leq j \leq p} \in \mathbb{R}^{n \times p}, \beta := (\beta_1, \dots, \beta_p)^T \in \mathbb{R}^p \text{ und } \sigma^2 > 0. \tag{6}$$

Erläutern Sie die Begriffe Regressor, Prädiktor, Kovariate und Feature im Rahmen der multiplen Regression.

Regressor, **Prädiktor**, **Kovariate** und **Feature** sind synonyme Bezeichnungen für die unabhängige Variable im Modell der multiplen Regression.

Erläutern Sie, warum $\hat{\beta} \approx \text{Regressorkovariation}^{-1}\text{Regressordatenkovariation}$ gilt.

Der Betaparameterschätzer hat bekanntlich die Form

$$\hat{\beta} := (X^T X)^{-1} X^T y$$

Dabei quantifizieren in sehr grober Auflösung

- $X^T y \in \mathbb{R}^p$ die Kovariation der Regressoren (Spalten der Designmatrix) mit den Daten y und
- $X^T X \in \mathbb{R}^{p \times p}$ die Kovariation der Regressoren untereinander.

Damit ergibt sich für die Betaparameterschätzer also eine Interpretation als “regressorkovariationnormalisierte Regressordatenkovariation”.

Erläutern Sie den Zusammenhang zwischen Betaparameterschätzern und partieller Korrelation in einem multiplen Regressionmodell mit Interzeptprädiktor und zwei kontinuierlichen Prädiktoren anhand der Formel

$$\hat{\beta}_1 = r_{y,x_1|x_2}\sqrt{\frac{1-r^2_{y,x_2}}{1-r^2_{x_1,x_2}}}\frac{s_y}{s_{x_1}}.$$

- Im Allgemeinen gilt für $1 \le i, l \le k$, dass $\hat{\beta}_k \neq r_{y,x_k|x_l}$.
- Betaparameterschätzer sind also im Allgemeinen keine partiellen Stichprobenkorrelationen.
- $\hat{\beta}_k = r_{y,x_k|x_l}$ für $1 \le i, l \le k$ gilt genau dann, wenn $s_y = s_{x_1} = s_{x_2}$ und zudem
    - $r_{y,x_l} = r_{x_k,x_l} = 0$, wenn also die Stichprobenkorrelationen der Daten und der Werte des zweiten Regressors, sowie die Stichprobenkorrelation der Werte der beiden Regressoren gleich Null sind. Dies kann der Fall sein, wenn einer der Regressoren die Daten “sehr gut erklärt” und der andere Regressor von dem ersten “sehr verschieden” ist.
    - $|r_{y,x_l}| = |r_{x_k,x_l}|$, wenn also die obige Stichprobenkorrelationen dem Betrage nach gleich sind. Dies ist vermutlich selten der Fall.

$X \in \mathbb{R}^{n \times 2}$ sei die Designmatrix eines multiplen Regressionsmodells mit zwei Prädiktoren und Betaparametervektor $\beta := (\beta_1, \beta_2)^2$. Geben Sie den Kontrastgewichtsvektor an, um die Nullhypothese $H_0 : \beta_1 = \beta_2$ mithilfe der T-Statistik zu testen.

$c = (1, -1)^T$

Simulieren Sie einen Datensatz eines multiplen Regressionsmodells mit Interzept und zwei kontinuierlichen Regressoren $x_1, x_2 \in \mathbb{R}^n$, wobei $x_{i2} := ax_{i1} + \xi_i$ mit $\xi_i \sim N(0, \sigma_\xi^2)$ für $i = 1, ..., n$ sein soll. Wählen Sie für die Simulation des Datensatzes $y \in \mathbb{R}^n$ den wahren, aber unbekannten, Betaparametervektor $\beta = (0, 1, 0)^T$ und testen Sie die Nullhypothesen $H_0 : \beta_j = 0$ für $j = 0, 1, 2$. Erläutern Sie Ihre Ergebnisse. Wiederholen Sie Analyse für den wahren, aber unbekannten, Betaparametervektor $\beta = (0, 0, 1)^T$.

Anwendungsszenario - SKF 7

für $\beta = (0, 1, 0)^T$

```r
# Modellformulierung und Datensimulation
library(MASS)                                              # Multivariate Normalverteilung
set.seed(1)                                                # reproduzierbare Daten
n           = 100                                          # Anzahl Datenpunkte
p           = 3                                            # Anzahl Parameter
a           = 10                                           # Regressortransformationsparameter
sigsqr_xi   = 1e0                                          # Regressorvarianzparameter
I_n         = diag(n)                                      # Identitätsmatrix
x_1         = round(runif(n,0,10))                         # Regressorwerte x_1
x_2         = mvrnorm(1, a*x_1, sigsqr_xi*I_n)             # eine Realisierung eines n-dimensionalen ZVs
X           = matrix(c(rep(1,n),x_1,x_2), nrow = n)        # Designmatrix
beta        = matrix(c(0,1,0), nrow = p)                   # Betaparametervektor
sigsqr      = 1e0                                          # Varianzparameter
y           = mvrnorm(1, X %*% beta, sigsqr*I_n)           # eine Realisierung eines n-dimensionalen ZVs
n           = length(y)                                    # Anzahl Datenpunkte
p           = ncol(X)                                      # Anzahl Parameter
beta_hat    = solve(t(X) %*% X) %*% t(X) %*% y             # Betaparameterschätzer
eps_hat     = y - X %*% beta_hat                           # Residuenvektor
sigsqr_hat = (t(eps_hat) %*% eps_hat) /(n-p)               # Varianzparameterschätzer

# Modellevaluation | Parameterinferenz
C           = diag(p)                                      # Kontrastgewichtsvektoren
ste         = rep(NaN, ncol(C))                            # Konstraststandardfehler
tee         = rep(NaN, ncol(C))                            # T-Statistiken
pvals       = rep(NaN, ncol(C))                            # p-Werte
for(i in 1:ncol(C)){
    c         = C[,i]                                      # Kontrastgewichtsvektor
    t_num     = t(c)%*%beta_hat                            # Zähler der T-Statistik
    ste[i]    = sqrt(sigsqr_hat*t(c)%*%solve(t(X)%*%X)%*%c) # Kontraststandardfehler/Nenner der T-Statistik
    tee[i]    = t_num/ste[i]                               # T-Statistik
    pvals[i] = 2*(1 - pt(abs(tee[i]),n-p))                 # p-Wert
}
```

Anwendungsszenario - SKF 7

für $\beta = (0, 1, 0)^T$

```
# Ausgabe
R             = data.frame(beta_hat,ste, tee, pvals)
rownames(R)  = c("(Intercept)", "x_1", "x_2")
colnames(R)  = c("Estimate", "Std. Error", "t value", "Pr(>|t|)")
print(R)
```

```
>             Estimate Std. Error t value Pr(>|t|)
> (Intercept)  0.00378      0.229  0.0165    0.987
> x_1          0.34445      1.131  0.3046    0.761
> x_2          0.06594      0.113  0.5831    0.561
```

Anwendungsszenario - SKF 7

für $\beta = (0, 0, 1)^T$

```
# Modellformulierung und Datensimulation
library(MASS)                                                   # Multivariate Normalverteilung
set.seed(1)                                                     # reproduzierbare Daten
n           = 100                                               # Anzahl Datenpunkte
p           = 3                                                 # Anzahl Parameter
a           = 10                                                # Regressortransformationsparameter
sigsqr_xi   = 1e0                                               # Regressorvarianzparameter
I_n         = diag(n)                                           # Identitätsmatrix
x_1         = round(runif(n,0,10))                              # Regressorwerte x_1
x_2         = mvrnorm(1, a*x_1, sigsqr_xi*I_n)                  # eine Realisierung eines n-dimensionalen ZVs
X           = matrix(c(rep(1,n),x_1,x_2), nrow = n)             # Designmatrix
beta        = matrix(c(0,0,1), nrow = p)                        # Betaparametervektor
sigsqr      = 1e0                                               # Varianzparameter
y           = mvrnorm(1, X %*% beta, sigsqr*I_n)                # eine Realisierung eines n-dimensionalen ZVs
n           = length(y)                                         # Anzahl Datenpunkte
p           = ncol(X)                                           # Anzahl Parameter
beta_hat    = solve(t(X) %*% X) %*% t(X) %*% y                  # Betaparameterschätzer
eps_hat     = y - X %*% beta_hat                                # Residuenvektor
sigsqr_hat = (t(eps_hat) %*% eps_hat) /(n-p)                    # Varianzparameterschätzer
# Modellevaluation | Parameterinferenz
C           = diag(p)                                           # Kontrastgewichtsvektoren
ste         = rep(NaN, ncol(C))                                 # Konstraststandardfehler
tee         = rep(NaN, ncol(C))                                 # T-Statistiken
pvals       = rep(NaN, ncol(C))                                 # p-Werte
for(i in 1:ncol(C)){
    c         = C[,i]                                           # Kontrastgewichtsvektor
    t_num     = t(c)%*%beta_hat                                 # Zähler der T-Statistik
    ste[i]    = sqrt(sigsqr_hat*t(c)%*%solve(t(X)%*%X)%*%c)     # Kontraststandardfehler/Nenner der T-Statistik
    tee[i]    = t_num/ste[i]                                    # T-Statistik
    pvals[i] = 2*(1 - pt(abs(tee[i]),n-p))                      # p-Wert
}
```

Anwendungsszenario - SKF 7

für $\beta = (0, 0, 1)^T$

```
# Ausgabe
R            = data.frame(beta_hat,ste, tee, pvals)
rownames(R)  = c("(Intercept)", "x_1", "x_2")
colnames(R)  = c("Estimate", "Std. Error", "t value", "Pr(>|t|)")
print(R)
```

```
>             Estimate Std. Error t value Pr(>|t|)
> (Intercept)  0.00378      0.229  0.0165 9.87e-01
> x_1         -0.65555      1.131 -0.5797 5.63e-01
> x_2          1.06594      0.113  9.4259 2.22e-15
```