

8. Übungsblatt zur Vorlesung Ökonometrie

Aufgabe 1: In der Vorlesung haben wir das lineare Regressionsproblem als statistisches Problem formuliert: Gegeben seien die Datenvektoren $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_p$ und \vec{x}_0 mit $\vec{x}_j = (x_{1,j}, \dots, x_{n,j})$ und \vec{x}_0 ist der konstante Vektor mit lauter Einsen, $\vec{x}_0 = (1, \dots, 1)$. Dann haben wir gesagt: Die ypsilons $\vec{y} = (y_1, \dots, y_n)$ sind nicht ebenfalls einfach gegebene Daten, sondern sie kommen "in Wirklichkeit" von dem Modell

$$\vec{y} = \beta_0 \vec{x}_0 + \beta_1 \vec{x}_1 + \dots + \beta_p \vec{x}_p + \vec{\varepsilon} \quad (1)$$

her, wobei die $\vec{\varepsilon} = (\varepsilon_1, \dots, \varepsilon_n)$ normal-verteilte Zufallszahlen sind mit Mittelwert 0 und Standardabweichung σ , und die $\vec{\beta} = (\beta_0, \beta_1, \dots, \beta_p)$ sind gegebene Zahlen, die wir aber nicht kennen. Jedes einzelne y_i ist dann also gegeben durch

$$y_i = \beta_0 + \beta_1 x_{i,1} + \dots + \beta_p x_{i,p} + \varepsilon_i, \quad (2)$$

insbesondere sind die y_i dann selbst normalverteilte Zufallszahlen mit Mittelwert

$$\mu_i := \beta_0 + \beta_1 x_{i,1} + \dots + \beta_p x_{i,p} \quad (3)$$

und Standardabweichung σ . Jetzt nehmen wir an: Es wurden n Zahlen $(y_1, \dots, y_n) =: \vec{y}$ gemäss des Modells (2) generiert, wie können wir dann aus den gegebenen Daten \vec{y} und $\vec{x}_1, \dots, \vec{x}_p$ die Zahlen $\beta_0, \beta_1, \dots, \beta_p$ (und das σ , was man 'bei dieser Gelegenheit' gleich mitberechnet) zurückgewinnen?

Dazu hatten wir die Likelihood Funktion hingeschrieben,

$$L(\vec{\beta}, \sigma) = \prod_{i=1}^n \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y_i - \mu_i)^2}{2\sigma^2}\right] dy_i \right\} \quad (4)$$

und dann den Logarithmus berechnet (mit μ_i gegeben durch die Gleichung (3) von oben):

$$\log L(\vec{\beta}, \sigma) = -n \log \sigma - \frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \mu_i)^2 + \text{const} \quad (5)$$

wobei die Konstante

$$\text{const} := -\frac{n}{2} \log(2\pi) + \sum_{i=1}^n \log(dy_i)$$

nur Terme enthält, die nicht von den Modellparametern $\vec{\beta}$ und σ abhängen.

..bitte wenden

Durch Maximieren von $\log L(\vec{\beta}, \sigma)$ haben wir dann die Maximum Likelihood Schätzer für die Regressionskoeffizienten $\vec{\beta}$ und die Standardabweichung σ (oder genauer für die Varianz σ^2) hergeleitet:

$$\hat{\beta}_{\text{ML}} = (X^T X)^{-1} X^T \vec{y} \quad (6)$$

$$\hat{\sigma}_{\text{ML}}^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2. \quad (7)$$

Dabei ist die Matrix X wieder gegeben durch die Matrix der Regressoren,

$$X = \begin{pmatrix} | & | & \cdots & | \\ \vec{x}_0 & \vec{x}_1 & & \vec{x}_p \\ | & | & & | \end{pmatrix} \in \mathbb{R}^{n \times (p+1)}$$

und das $\hat{y} = (\hat{y}_1, \dots, \hat{y}_n)$ auf der rechten Seite von Gleichung (7) ist gegeben durch den Regression-fit,

$$\hat{y} = \hat{\beta}_0 \vec{x}_0 + \hat{\beta}_1 \vec{x}_1 + \cdots + \hat{\beta}_p \vec{x}_p$$

wobei die $\hat{\beta}_j = \hat{\beta}_{\text{ML},j}$ durch die Werte der Maximum Likelihood Schätzer gegeben sind. Die Formel für die Regressionskoeffizienten in Gleichung (6) ist genau dieselbe wie die, die wir auch schon im Kapitel 4 durch Minimierung der Fehlerquadrate $(\vec{y} - \sum_{j=0}^p \beta_j \vec{x}_j)^2 \rightarrow \min$ hergeleitet hatten.

In dieser Aufgabe wollen wir uns nun in einer R-Simulation mit den relevanten Grössen vertraut machen. Dazu betrachten wir das Regressionsproblem

$$y_i = 3 - 0.5 x_i + \varepsilon_i, \quad 1 \leq i \leq n \quad (8)$$

wobei wir die Anzahl n der Datenpunkte $(x_i, y_i)_{1 \leq i \leq n}$ variieren lassen wollen, etwa

$$n \in \{ 10, 100, 1000 \}.$$

- a) Öffnen Sie einen R-Editor und, da wir mit variablen n rechnen wollen, starten Sie Ihren R-Code etwa folgendermassen:

```
# Waehlen Sie einen Wert fuer die
# Anzahl der Datenpunkte:

n = 10
# n = 100
# n = 1000

beta0 = 3
beta1 = -0.5
sigma = 2

x = seq(from=-5,to=5,length=n)
eps = rnorm(n,mean=0,sd=sigma)
y = beta0 + beta1*x + eps
plot(x,y)
```

..weiter auf dem nächsten Blatt

b) Wir ignorieren die Konstante const in Gleichung (5) und definieren dementsprechend

$$\log \tilde{L}(\beta_0, \beta_1, \sigma) := -n \log \sigma - \frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - [\beta_0 + \beta_1 x_i])^2 \quad (9)$$

$$\tilde{L}(\beta_0, \beta_1, \sigma) := \prod_{i=1}^n \left\{ \frac{1}{\sigma} \exp \left[-\frac{(y_i - [\beta_0 + \beta_1 x_i])^2}{2\sigma^2} \right] \right\} \quad (10)$$

Wir wollen uns zunächst ein Überblick über die Größenverhältnisse verschaffen. Berechnen Sie dazu die numerischen Werte von

$$\log \tilde{L}(\beta_0 = 3, \beta_1 = -0.5, \sigma = 2)$$

und

$$\tilde{L}(\beta_0 = 3, \beta_1 = -0.5, \sigma = 2)$$

für $n \in \{10, 100, 1000\}$. Achten Sie dabei darauf, dass Sie den Logarithmus von \tilde{L} durch die rechte Seite von (9) berechnen und nicht durch logarithmieren des Resultats aus Gleichung (10). Warum?

c) Legen Sie jetzt die Matrix X der Regressoren an und berechnen Sie dann mit Hilfe der Gleichungen (6) und (7) die numerischen Werte der Maximum Likelihood Schätzer

$$\hat{\beta}_0 = \hat{\beta}_{0,ML}, \quad \hat{\beta}_1 = \hat{\beta}_{1,ML} \quad \text{und} \quad \hat{\sigma} := \sqrt{\hat{\sigma}_{ML}^2}.$$

Beim Anlegen der Matrix X könnte etwa der `cbind()`-Befehl hilfreich sein, mit dem Sie Spaltenvektoren zu einer Matrix zusammensetzen können. Weiterhin können Sie etwa mit dem `repeat`-Befehl `rep()` den Vektor \vec{x}_0 mit lauter Einsen erzeugen.

d) Da wir in Teil (b) gesehen haben, dass die numerische Darstellung von \tilde{L} schwierig oder nicht möglich sein kann, betrachten wir jetzt nur noch $\log \tilde{L}$. Wir wollen uns davon überzeugen, dass die Zahlen aus Teil (c) tatsächlich die Log-Likelihood Funktion maximieren tun. Dazu legen wir wieder, wie in Aufgabe 1c von Übungsblatt 4, einige 'Test-Betas' und 'Test-Sigas' an, etwa

$$\begin{aligned} \text{tbeta0} &= \text{seq}(\text{from} = 0, \text{to} = 6, \text{by} = 0.01) \\ \text{tbeta1} &= \text{seq}(\text{from} = -3, \text{to} = 3, \text{by} = 0.01) \end{aligned}$$

und, da das σ nicht so wichtig ist, mit etwas geringerer Auflösung

$$\text{tsigma} = \text{seq}(\text{from} = 0.1, \text{to} = 4, \text{by} = 0.1)$$

und berechnen dann $\log \tilde{L}(\beta_0, \beta_1, \sigma)$ etwa mit folgendem Code, wo Sie an den mit ... gekennzeichneten Stellen noch ein paar Sachen ergänzen müssen:

..*bitte wenden*

```

tbeta0 = seq(from=0,to=6,by=0.01)
tbeta1 = seq(from=-3,to=3,by=0.01)
tsigma = seq(from=0.1,to=4,by=0.1)

n0 = length(tbeta0)
n1 = length(tbeta1)
nsig = length(tsigma)

logL = array( 0 , dim=c(n0,n1,nsig) )

# takes about 75 seconds for n=1000:
for(i in 1:n0)
{
  for(j in ... )
  {
    for(... 1:nsig)
    {
      b0 = tbeta0[i]           # beta0
      b1 = ... [j]           # beta1
      sig = ...
      mu = ...
      logL[i,j,k] = ...
    }
  }
}

# Ueberblick ueber alle Werte:
summary(logL)
# nur das Maximum:
max(logL)

```

- e) Wir wissen jetzt zwar, was das Maximum von $\log \tilde{L}$ ist, nämlich $\max(\log L)$, aber wir wollen ja die Werte von $(\beta_0, \beta_1, \sigma)$ wissen, an denen das Maximum angenommen wird. Diese Werte können Sie mit folgendem Code erhalten:

```

# wo wird das Maximum angenommen?
indices = which(logL == max(logL), arr.ind = TRUE)
indices
b0max = tbeta0[indices[1]]
b1max = tbeta1[indices[2]]
sigmax = tsigma[indices[3]]

b0max      # -> beta0_ML
b1max      # -> beta1_ML
sigmax     # -> sigma_ML

```

Vergleichen Sie diese Zahlen mit den Zahlen aus Aufgabe 1c. Bis auf 'Auflösegenauigkeit' von 0.01 bzw. 0.1 sollten Ihre Werte übereinstimmen.

- f) Schliesslich schauen wir uns noch ein paar Bilder an: Da $\log \tilde{L} = \log \tilde{L}(\beta_0, \beta_1, \sigma)$ eine Funktion von 3 Variablen ist, ist es schon nicht mehr ganz so einfach, sich einen Überblick über die Funktion zu verschaffen. Plotten wir etwa die Höhenlinien in der (β_0, β_1) -Ebene, der (β_0, σ) -Ebene und der (β_1, σ) -Ebene. Dabei setzen wir den dritten Parameter jeweils auf den Maximum Likelihood Wert:

..weiter auf dem nächsten Blatt

```

# Schauen wir uns noch ein paar Bilder an:

# a) logL in der (beta0,beta1)-Ebene, mit sigma = sigmax:
logLa = logL[, , indices[3]]
contour(tbeta0,tbeta1,logLa,nlevels=50)
contour(tbeta0,tbeta1,logLa,nlevels=100)

# b) logL in der (beta0,sigma)-Ebene, mit beta1 = b1max:
logLb = logL[, indices[2], ]
contour(tbeta0,tsigma,logLb,nlevels=50)
contour(tbeta0,tsigma,logLb,nlevels=50,zlim=c(-10*n,0))
contour(tbeta0,tsigma,logLb,nlevels=100,zlim=c(-4*n,0))
contour(tbeta0,tsigma,logLb,nlevels=150,zlim=c(-2*n,0))

# c) logL in der (beta1,sigma)-Ebene, mit beta0 = b0max:
logLc = ...
contour( ... )
contour( ... )
contour( ... )

```

An den mit ... gekennzeichneten Stellen müssen Sie wieder etwas Code ergänzen.