Multivariate Verfahren

MSc Psychologie | MSc Klinische Psychologie und Psychotherapie

WiSe 2024/25

Prof. Dr. Dirk Ostwald

(11) Lineare Diskriminanzanalyse

Anwendungsszenario

Modellformulierung

Inferenz und Klassifikation

Lernen

Anwendungsbeispiel

Selbstkontrollfragen

Anwendungsszenario

Modellformulierung

Inferenz und Klassifikation

Lernen

Anwendungsbeispiel

Selbstkontrollfragen

Anwendungsszenario

Psychotherapie Non-Response-Rate wird auf etwa 20 - 30% geschätzt

Vorhersage von Behandlungserfolg basierend auf klinischen Markern wäre hilfreich

- Therapieauswahloptimierung
- Lebensqualitätverbesserung
- Ressourcensensitivität

Digitale Datenbank von Psychotherapieverläufen als Trainingsdatensatz

Prädiktive Modellierung zur Etablierung eines prädiktiven klinischen Markerprofils

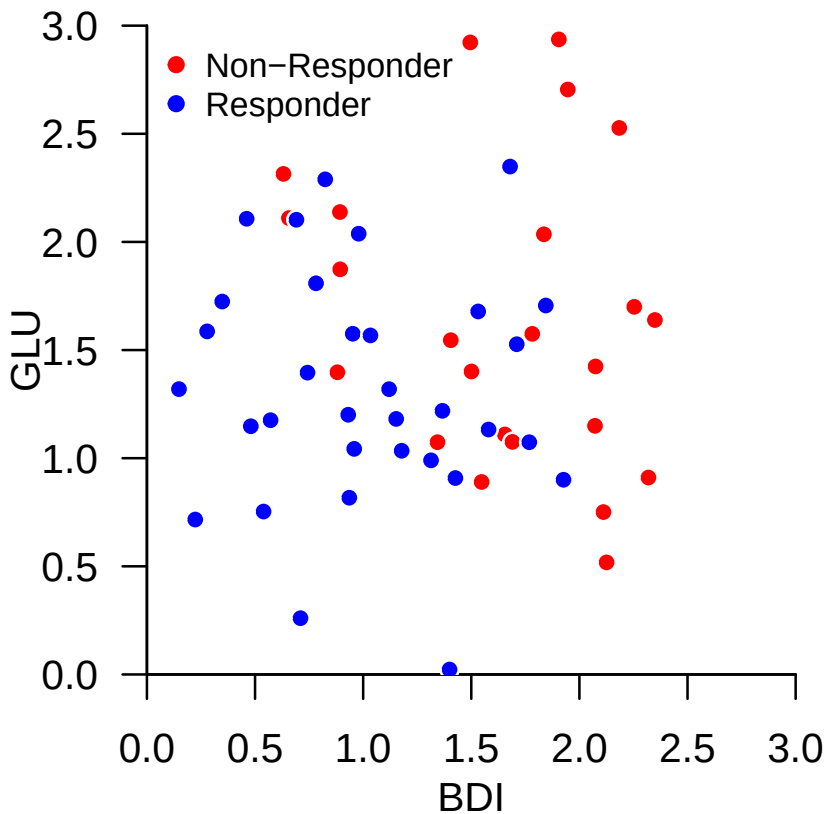
Treatmentsuccessvorhersage für neue Patient:innen

Anwendungsbeispiele

- BDI und GLU Werte bei Depressionsdiagnose als Prädiktoren von CBT Erfolg
- Lineare Diskriminanzanalyse, Logistische Regression, SVM, Neuronale Netze
- Zhao et al. (2024) geben eine aktuelle Übersicht zur Linearen Diskriminanzanalyse

Beispieldatensatz

- BDI und GLU Werte bei Depressionsdiagnose als Prädiktoren von CBT Erfolg



Anwendungsszenario

Modellformulierung

Inferenz und Klassifikation

Lernen

Anwendungsbeispiel

Selbstkontrollfragen

Definition (Bernoulli-Zufallsvariable)

Es sei y eine Zufallsvariable mit Ergebnisraum $\mathcal{Y} := \{0, 1\}$ und WMF

$$p : \mathcal{Y} \to [0, 1], y \mapsto p(y) := \mu^y (1 - \mu)^{1-y} \text{ mit } \mu \in [0, 1]. \tag{1}$$

Dann sagen wir, dass y einer *Bernoulli-Verteilung mit Parameter* $\mu \in [0, 1]$ unterliegt und nennen y eine *Bernoulli-Zufallsvariable*. Wir kürzen dies mit $y \sim \text{Bern}(\mu)$ ab. Die WMF einer Bernoulli-Zufallsvariable bezeichnen wir mit

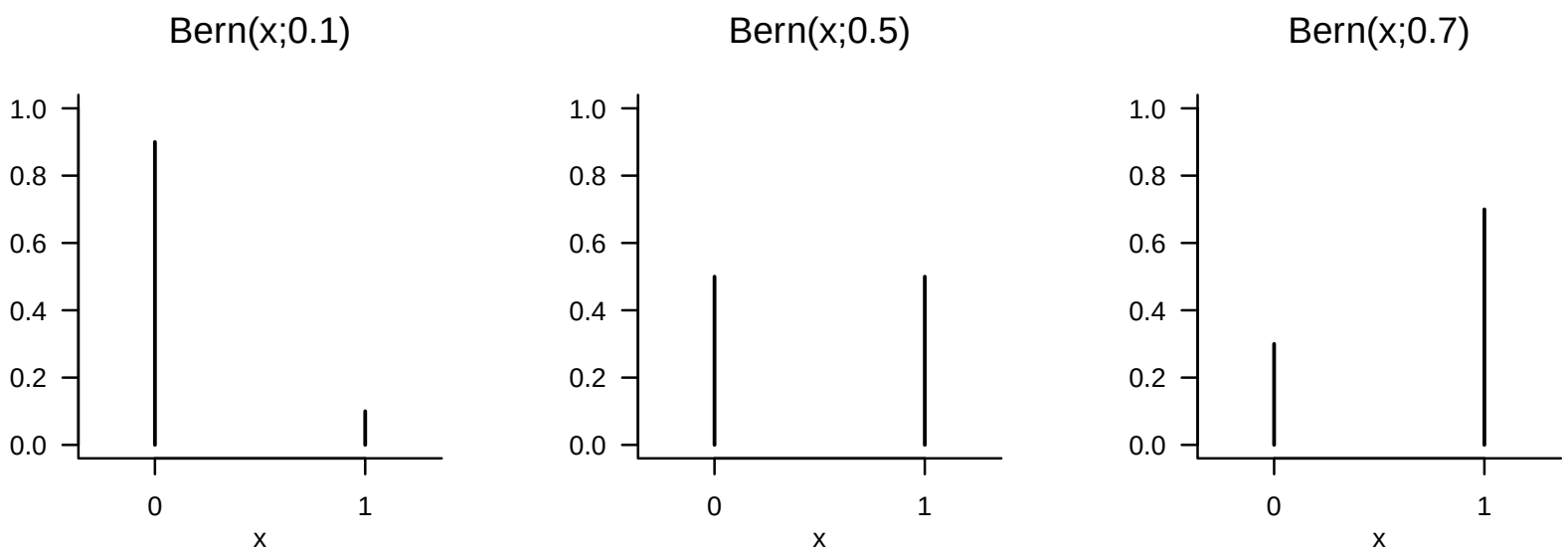
$$\text{Bern}(y; \mu) := \mu^y (1 - \mu)^{1-y}. \tag{2}$$

Bemerkungen

- Eine Bernoulli-Zufallsvariable kann als Modell eines Münzwurfs dienen.
- μ ist die Wahrscheinlichkeit dafür, dass y den Wert 1 annimmt,

$$\mathbb{P}(y = 1) = \mu^1 (1 - \mu)^{1-1} = \mu. \tag{3}$$

Bernoulli-Zufallsvariable



Definition (Modell der Linearen Diskriminanzanalyse)

x sei ein m-dimensionaler Zufallsvektor mit Ergebnisraum $\mathbb{R}^m$ und y sei eine Zufallsvariable mit Ergebnisraum $\{0, 1\}$. Dann ist das *Modell der Linearen Diskriminanzanalyse* die gemeinsame Verteilung

$$\mathbb{P}(x, y) = \mathbb{P}(y)\mathbb{P}(x|y), \tag{4}$$

wobei die diskrete marginale Verteilung $\mathbb{P}(y)$ durch die WMF

$$p(y) = \mathsf{Bern}(y; \mu) \tag{5}$$

mit $\mu \in]0, 1[$ und die kontinuierliche bedingte Verteilung $\mathbb{P}(x|y)$ durch die WDF

$$p(x|y) = N(x; \mu_0, \Sigma)^{1-y} N(x; \mu_1, \Sigma)^y \tag{6}$$

mit $\mu_0, \mu_1 \in \mathbb{R}^m$ und $\Sigma \in \mathbb{R}^{m \times m}$ pd definiert ist. Wir bezeichnen die gemischte WMF und WDF (WMDF) des Modells der Linearen Diskriminanzanalyse mit

$$p(x, y) := p(y)p(x|y) = \mathsf{Bern}(y; \mu) N(x; \mu_0, \Sigma)^{1-y} N(x; \mu_1, \Sigma)^y \tag{7}$$

Modellformulierung

Bemerkung

Aus generativer Sicht wird ein Trainingsdatensatz

$$\left\{\left(x^{(i)}, y^{(i)}\right)\right\}_{i=1}^{n} \text{ mit } x^{(i)} \in \mathbb{R}^m \text{ und } y^{(i)} \in \{0,1\} \tag{8}$$

eines Modells zur Linearen Diskriminanzanalyse wie folgt erzeugt:

(1) $y^{(i)}$ wird zunächst durch Ziehen aus einer Bernoulliverteilung mit Parameter μ erzeugt.

(2) In Abhängigkeit vom Wert von $y^{(i)}$ wird $x^{(i)}$ dann durch Ziehen aus einer multivariaten Normalverteilung mit Kovarianzmatrixparameter Σ und Erwartungswertparameter μ_0 für $y^{(i)} = 0$ oder μ_1 für $y^{(i)} = 1$ erzeugt.

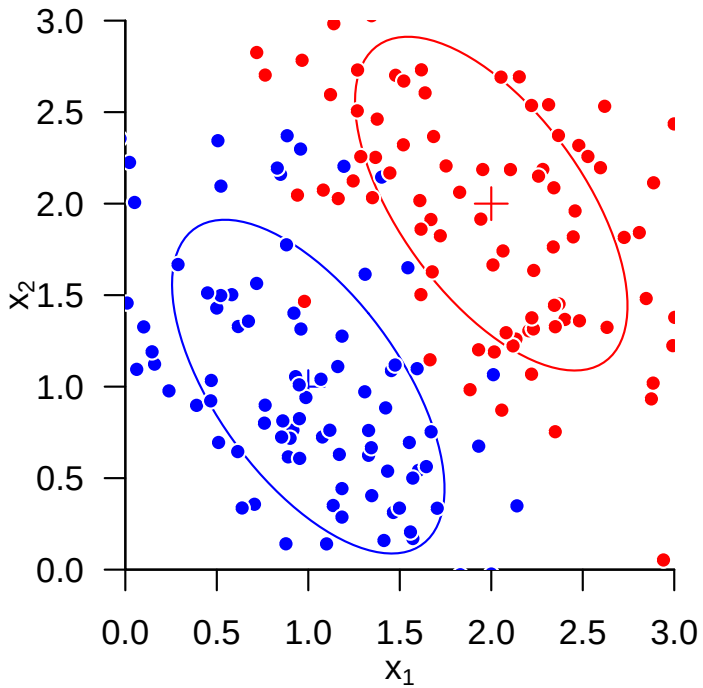
Datengeneration

```
# Modellformulierung
library(mvtnorm)                                  # Multivariate Normalverteilung
set.seed(0)                                       # Zufallszahlengenerator
m      = 2                                        # Featurevektordimension
n      = 2e2                                      # Anzahl Trainingsdatenpunkte
mu     = 0.5                                      # wahrer, aber unbekannter, Bernoulliparameter \mu
mu_0   = c(1,1)                                   # wahrer, aber unbekannter, Normalverteilungsparameter \mu_0
mu_1   = c(2,2)                                   # wahrer, aber unbekannter, Normalverteilungsparameter \mu_1
Sigma  = matrix(c( 0.40, -0.30,                   # Kovarianzmatrixparameter
                  -0.30,  0.60),
                byrow = TRUE,
                nrow = m)

# Modellsampling
y      = matrix(rep(NaN,n)  , nrow = 1)           # Labeldatenarray
x      = matrix(rep(NaN,n*m), nrow = m)           # Featurevektorarray
for(i in 1:n){                                    # Iteration über Datenpunkte
  y[i]  = rbinom(1,1,mu)                          # y^{(i)} \sim Bern(\mu)
  x[,i] = ((rmvnorm(1, mu_0, Sigma)**(1-y[i]))    # x^{i} \sim N(\mu_0,\Sigma)^{1-y}N(\mu_1,\Sigma)^y
          *(rmvnorm(1, mu_1, Sigma)**(y[i])))
}

# Datensatzkonkatenation
D = rbind(x,y)
```

Datengeneration

$$m = 2, n = 200, \mu = 0.5, \mu_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mu_1 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \Sigma = \begin{pmatrix} 0.40 & -0.30 \\ -0.30 & 0.60 \end{pmatrix}$$

$+$ μ_0 • $x^{(i)}$ mit $y^{(i)} = 0$, $+$ μ_1 • $x^{(i)}$ mit $y^{(i)} = 1, i = 1, ..., n$



Anwendungsszenario

Modellformulierung

Inferenz und Klassifikation

Lernen

Anwendungsbeispiel

Selbstkontrollfragen

Theorem (Inferenz bei Linearer Diskriminanzanalyse)

$p(x, y)$ sei die WMDF des Modells einer Linearen Diskriminanzanalyse. Dann gilt

$$p(y = 1|x) = \frac{1}{1 + \exp(-\tilde{x}^T \beta)} \text{ und } p(y = 0|x) = 1 - p(y = 1|x), \tag{9}$$

wobei

$$\tilde{x} := \begin{pmatrix} 1 \\ x \end{pmatrix} \in \mathbb{R}^{m+1} \tag{10}$$

der *erweiterten Featurevektor* und

$$\beta := \begin{pmatrix} \frac{1}{2} \left(\mu_0^T \Sigma^{-1} \mu_0 - \mu_1^T \Sigma^{-1} \mu_1 \right) + \ln \left(\frac{\mu}{1-\mu} \right) \\ -\Sigma^{-1} (\mu_0 - \mu_1) \end{pmatrix} \in \mathbb{R}^{m+1}. \tag{11}$$

der *Inferenzparametervektor* sind.

Bemerkungen

- $p(y|x)$ kann zur Prädiktion der Klasse eines $x \in \mathbb{R}^m$ genutzt werden.
- Diese Prädiktion hängt von den Parameter $\mu, \mu_0, \mu_1, \Sigma$ des Modells der Linearen Diskriminanzanalyse ab.

Beweis

Wir halten zunächst fest, dass

$$\begin{aligned} p(y=1|x) &= \frac{p(x,y=1)}{p(x,y=0)+p(x,y=1)} \\ &= \frac{\frac{p(x,y=1)}{p(x,y=1)}}{\frac{p(x,y=0)}{p(x,y=1)}+\frac{p(x,y=1)}{p(x,y=1)}} \\ &= \frac{1}{1+\frac{p(x,y=0)}{p(x,y=1)}} \\ &= \frac{1}{1+\exp\left(\ln\left(\frac{p(x,y=0)}{p(x,y=1)}\right)\right)} \\ &= \frac{1}{1+\exp\left(-\ln\left(\frac{p(x,y=1)}{p(x,y=0)}\right)\right)} \end{aligned} \tag{12}$$

Mit der Definition des Modells der Linearen Diskriminanzanalyse gilt dann

$$p(x,y=1) = p(x|y=1)p(y=1) = N(x;\mu_1,\Sigma)\mu \tag{13}$$

und

$$p(x,y=0) = p(x|y=0)p(y=0) = N(x;\mu_0,\Sigma)(1-\mu) \tag{14}$$

Beweis (fortgeführt)

Wir erhalten also

$$
\begin{aligned}
&= \ln\left(\frac{p(x, y=1)}{p(x, y=0)}\right) \\
&= \ln\left(\frac{N(x; \mu_1, \Sigma)\mu}{N(x; \mu_0, \Sigma)(1-\mu)}\right) \\
&= \ln\left(N(x; \mu_1, \Sigma)\mu\right) - \ln\left(N(x; \mu_0, \Sigma)(1-\mu)\right) \\
&= \ln(\mu) + \ln N(x; \mu_1, \Sigma) - \ln(1-\mu) - \ln N(x; \mu_0, \Sigma) \\
&= \ln\mu - \ln(1-\mu) - \frac{m}{2}\ln 2\pi - \ln|\Sigma| - \frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1) \\
&\qquad\qquad + \frac{m}{2}\ln 2\pi + \ln|\Sigma| + \frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0) \\
&= -\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1) + \frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0) + \ln\mu - \ln(1-\mu) \\
&= -\frac{1}{2}x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu_1 - \frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 + \frac{1}{2}x^T\Sigma^{-1}x - x^T\Sigma^{-1}\mu_0 + \frac{1}{2}\mu_0^T\Sigma^{-1}\mu_0 + \ln\left(\frac{\mu}{1-\mu}\right) \\
&= \frac{1}{2}\mu_0^T\Sigma^{-1}\mu_0 - \frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 - x^T\Sigma^{-1}(\mu_0 - \mu_1) + \ln\left(\frac{\mu}{1-\mu}\right) \\
&= \begin{pmatrix}1 & x^T\end{pmatrix}\begin{pmatrix}\frac{1}{2}\left(\mu_0^T\Sigma^{-1}\mu_0 - \mu_1^T\Sigma^{-1}\mu_1\right) + \ln\left(\frac{\mu}{1-\mu}\right) \\ -\Sigma^{-1}(\mu_0 - \mu_1)\end{pmatrix} \\
&=: \tilde{x}^T\beta
\end{aligned}
$$

□

Definition (Klassifikationsregel der linearen Diskriminanzanalyse)

$p(x, y)$ sei die WMDF des Modells der Linearen Diskriminanzanalyse. Dann ist die *Klassifikationregel* definiert als

$$\delta : \mathbb{R}^m \to \{0, 1\}, x \mapsto \delta(x) := \begin{cases} 0 & \text{für } p(y = 0|x) \geq p(y = 1|x) \\ 1 & \text{für } p(y = 0|x) < p(y = 1|x) \end{cases} \tag{15}$$

Bemerkung

- Es gilt

$$\delta(x) = 1 \Leftrightarrow p(y = 1|x) > p(y = 0|x) \Leftrightarrow p(y = 1|x) > 0.5. \tag{16}$$

Inferenz und Klassifikation bei bekannten Modellparametern

$$\mu = 0.5, \mu_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mu_1 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \Sigma = \begin{pmatrix} 0.40 & -0.30 \\ -0.30 & 0.60 \end{pmatrix}$$

```
# Inferenz und Klassifikation für die ersten k Datenpunkte
k           = 10                                          # Anzahl Datenpunkte
x_tilde     = rbind(rep(1,k), x[,1:k])                    # erweiterte Featurevektoren
beta        = matrix(                                     # Inferenzparametervektor
              c((1/2)*(  t(mu_0) %*% solve(Sigma) %*% mu_0
                       - t(mu_1) %*% solve(Sigma) %*% mu_1)
                       + log(mu/(1-mu)),
                -solve(Sigma) %*% (mu_0-mu_1)), nrow = 3)
p_y_giv_x   = 1/(1+exp(-t(x_tilde) %*% beta))             # p(y = 1|x)
delta       = as.numeric(p_y_giv_x >= 0.5)                # Klassifikationsregel
```

	1	2	3	4	5	6	7	8	9	10
x_1	1.15	1.83	2.37	0.92	2.37	2.05	0.24	0.89	2.48	1.43
x_2	3.37	2.06	1.45	0.76	2.37	2.69	0.98	0.62	2.32	0.54
p(y = 1|x)	1.00	0.99	0.99	0.00	1.00	1.00	0.00	0.00	1.00	0.01
delta(x)	1.00	1.00	1.00	0.00	1.00	1.00	0.00	0.00	1.00	0.00
y	1.00	1.00	1.00	0.00	1.00	1.00	0.00	0.00	1.00	0.00

Anwendungsszenario

Modellformulierung

Inferenz und Klassifikation

Lernen

Anwendungsbeispiel

Selbstkontrollfragen

Theorem (ML-Schätzer der Linearen Diskriminanzanalyse)

$p(x, y)$ sei die WMDF des Modells einer Linearen Diskriminanzanalyse mit Parametern $\{\mu, \mu_0, \mu_1, \Sigma\}$, $\{(x^{(i)}, y^{(i)})\}_{i=1}^n$ sei ein LDA Trainingsdatensatz, und $1_{\{S\}}$ sei die Indikatorfunktion der Aussage A, d.h. $1_{\{A\}} = 1$, wenn A WAHR ist uns $1_{\{A\}} = 0$, wenn A FALSCH ist. Dann sind die Maximum-Likelihood-Schätzer für μ, μ_0, μ_1 und Σ gegeben durch

$$
\begin{aligned}
\hat{\mu} &= \frac{1}{n} \sum_{i=1}^{n} 1_{\{y^{(i)}=1\}}, \\
\hat{\mu}_0 &= \frac{1}{\sum_{i=1}^{n} 1_{\{y^{(i)}=0\}}} \sum_{i=1}^{n} 1_{\{y^{(i)}=0\}} x^{(i)}, \\
\hat{\mu}_1 &= \frac{1}{\sum_{i=1}^{n} 1_{\{y^{(i)}=1\}}} \sum_{i=1}^{n} 1_{\{y^{(i)}=1\}} x^{(i)}, \\
\hat{\Sigma} &= \frac{1}{n} \sum_{i=1}^{n} \left(x^{(i)} - \hat{\mu}_{y^{(i)}} \right) \left(x^{(i)} - \hat{\mu}_{y^{(i)}} \right)^T .
\end{aligned}
\tag{17}
$$

Bemerkungen

- μ wird als die relative Häufigkeit der 1en im Trainingsdatensatz geschätzt.
- μ_0 und μ_1 werden als Stichprobenmittel aller $x^{(i)}$ mit $y^{(i)} = 0$ bzw. $y^{(i)} = 1$ geschätzt.
- Σ wird durch die empirische Kovarianzmatrix aller $x^{(i)}, i = 1, \ldots, n$ geschätzt.
- Substitution ergibt den Schätzer $\hat{\beta}$

Beweis

(1) Formulierung der Log Likelihood Funktion

$$
\begin{aligned}
\ell(\mu, \mu_0, \mu_1, \Sigma) &:= \ln \prod_{i=1}^{n} p\left(x^{(i)}, y^{(i)}\right) \\
&= \sum_{i=1}^{n} \ln p\left(x^{(i)}, y^{(i)}\right) \\
&= \sum_{i=1}^{n} \ln p\left(x^{(i)} | y^{(i)}\right) p\left(y^{(i)}\right) \\
&= \sum_{i=1}^{n} \ln p\left(x^{(i)} | y^{(i)}\right) + \ln p\left(y^{(i)}\right) \\
&= \sum_{i=1}^{n} \ln \left(N(x^{(i)}; \mu_0, \Sigma)\right)^{1-y^{(i)}} \left(N(x^{(i)}; \mu_1, \Sigma)\right)^{y^{(i)}} + \ln \left(\mu^{y^{(i)}} (1-\mu)^{1-\mu^{y^{(i)}}}\right) \\
&= \sum_{i=1}^{n} \left(1-y^{(i)}\right) \ln N\left(x^{(i)}; \mu_0, \Sigma\right) + y^{(i)} \ln N\left(x^{(i)}; \mu_1, \Sigma\right) + y^{(i)} \ln \mu + \left(1-y^{(i)}\right) \ln(1-\mu) \\
&= \sum_{i=1}^{n} \left(1-y^{(i)}\right) \left(-\frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma| - \frac{1}{2} \left(x^{(i)} - \mu_0\right)^T \Sigma^{-1} \left(x^{(i)} - \mu_0\right)\right) \\
&\quad + \sum_{i=1}^{n} y^{(i)} \left(-\frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma| - \frac{1}{2} \left(x^{(i)} - \mu_1\right)^T \Sigma^{-1} \left(x^{(i)} - \mu_1\right)\right) \\
&\quad + \sum_{i=1}^{n} y^{(i)} \ln \mu + \sum_{i=1}^{n} \left(1-y^{(i)}\right) \ln(1-\mu).
\end{aligned}
$$

Beweis (fortgeführt)

(2) Gradient der Log Likelihood Funktion

Der Gradient der Log Likelihood Funktion des Modells der Linearen Diskriminanzanalyse besteht aus den partiellen Ableitungen von ℓ hinsichtlich von μ, μ_0, μ_1 und Σ. Wie unten gezeigt ergibt er sich als

$$\nabla \ell\left(\mu, \mu_0, \mu_1, \Sigma\right) = \begin{pmatrix} \frac{\partial}{\partial \mu} \ell\left(\mu, \mu_0, \mu_1, \Sigma\right) \\ \frac{\partial}{\partial \mu_0} \ell\left(\mu, \mu_0, \mu_1, \Sigma\right) \\ \frac{\partial}{\partial \mu_1} \ell\left(\mu, \mu_0, \mu_1, \Sigma\right) \\ \frac{\partial}{\partial \Sigma} \ell\left(\mu, \mu_0, \mu_1, \Sigma\right) \end{pmatrix} = \begin{pmatrix} \frac{1}{\mu} \sum_{i=1}^{n} 1_{\{y^{(i)}=1\}} - \frac{1}{1-\mu} \sum_{i=1}^{n} 1_{\{y^{(i)}=0\}} \\ -\frac{1}{2} \sum_{i=1}^{n} 1_{\{y^{(i)}=0\}} \left(\left(x^{(i)} - \mu_0\right)^T \Sigma^{-1} \right) \\ -\frac{1}{2} \sum_{i=1}^{n} 1_{\{y^{(i)}=1\}} \left(\left(x^{(i)} - \mu_1\right)^T \Sigma^{-1} \right) \\ \frac{n}{2} \Sigma - \frac{1}{2} \sum_{i=1}^{n} \left(x^{(i)} - \mu_{1_{\{y^{(i)}=1\}}} \right) \left(x^{(i)} - \mu_{1_{\{y^{(i)}=1\}}} \right)^T \end{pmatrix}.$$

Beweis (fortgeführt)

Für die partielle Ableitung hinsichtlich μ_0 und ähnlich für μ_1 ergibt sich

$$\begin{aligned}
\frac{\partial}{\partial \mu_0} \ell(\mu, \mu_0, \mu_1, \Sigma) &= \frac{\partial}{\partial \mu_0} \sum_{i=1}^{n} \left(1 - y^{(i)}\right) \left(-\frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma| - \frac{1}{2} \left(x^{(i)} - \mu_0\right)^T \Sigma^{-1} \left(x^{(i)} - \mu_0\right)\right) \\
&= \sum_{i=1}^{n} \left(1 - y^{(i)}\right) \frac{\partial}{\partial \mu_0} \left(-\frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma| - \frac{1}{2} \left(x^{(i)} - \mu_0\right)^T \Sigma^{-1} \left(x^{(i)} - \mu_0\right)\right) \\
&= \sum_{i=1}^{n} \left(1 - y^{(i)}\right) \left(-\frac{1}{2} \frac{\partial}{\partial \mu_0} \left(x^{(i)} - \mu_0\right)^T \Sigma^{-1} \left(x^{(i)} - \mu_0\right)\right) \\
&= \sum_{i=1}^{n} \left(1 - y^{(i)}\right) \left(-\frac{1}{2} \left(x^{(i)} - \mu_0\right)^T \Sigma^{-1}\right) \\
&= -\frac{1}{2} \sum_{i=1}^{n} \left(1 - y^{(i)}\right) \left(\left(x^{(i)} - \mu_0\right)^T \Sigma^{-1}\right). \\
&= -\frac{1}{2} \sum_{i=1}^{n} 1_{\{y^{(i)}=0\}} \left(\left(x^{(i)} - \mu_0\right)^T \Sigma^{-1}\right).
\end{aligned}$$

Beweis (fortgeführt)

Für die partielle Ableitung hinsichtlich Σ ergibt sich

$$
\begin{aligned}
\frac{\partial}{\partial \Sigma} \ell(\mu, \mu_1, \mu_0, \Sigma) = \quad & \frac{\partial}{\partial \Sigma} \sum_{i=1}^{n} \left(1 - y^{(i)}\right) \left(-\frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma| - \frac{1}{2} \left(x^{(i)} - \mu_0\right)^T \Sigma^{-1} \left(x^{(i)} - \mu_0\right)\right) \\
& + \frac{\partial}{\partial \Sigma} \sum_{i=1}^{n} y^{(i)} \left(-\frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma| - \frac{1}{2} \left(x^{(i)} - \mu_1\right)^T \Sigma^{-1} \left(x^{(i)} - \mu_1\right)\right) \\
= \quad & \sum_{i=1}^{n} \left(1 - y^{(i)}\right) \left(-\frac{1}{2} \frac{\partial}{\partial \Sigma} \ln |\Sigma| - \frac{1}{2} \frac{\partial}{\partial \Sigma} \left(x^{(i)} - \mu_0\right)^T \Sigma^{-1} \left(x^{(i)} - \mu_0\right)\right) \\
& + \sum_{i=1}^{n} y^{(i)} \left(-\frac{1}{2} \frac{\partial}{\partial \Sigma} \ln |\Sigma| - \frac{1}{2} \frac{\partial}{\partial \Sigma} \left(x^{(i)} - \mu_1\right)^T \Sigma^{-1} \left(x^{(i)} - \mu_1\right)\right) \qquad (18) \\
= \quad & \sum_{i=1}^{n} \left(1 - y^{(i)}\right) \left(-\frac{1}{2} \Sigma - \frac{1}{2} \left(x^{(i)} - \mu_0\right) \left(x^{(i)} - \mu_0\right)^T\right) \\
& + \sum_{i=1}^{n} y^{(i)} \left(-\frac{1}{2} \Sigma - \frac{1}{2} \left(x^{(i)} - \mu_1\right) \left(x^{(i)} - \mu_1\right)^T\right) \\
= \quad & \frac{n}{2} \Sigma - \frac{1}{2} \sum_{i=1}^{n} \left(x^{(i)} - \mu_{1_{\{y^{(i)}=1\}}}\right) \left(x^{(i)} - \mu_{1_{\{y^{(i)}=1\}}}\right)^T .
\end{aligned}
$$

Beweis (fortgeführt)

Für die partielle Ableitung hinsichtlich μ ergibt sich

$$\begin{aligned}\frac{\partial}{\partial \mu}\ell(\mu, \mu_1, \mu_0, \Sigma) &= \frac{\partial}{\partial \mu}\left(\sum_{i=1}^{n} y^{(i)} \ln \mu + \sum_{i=1}^{n}\left(1 - y^{(i)}\right)\ln(1-\mu)\right) \\ &= \sum_{i=1}^{n} y^{(i)} \frac{\partial}{\partial \mu} \ln \mu + \sum_{i=1}^{n}\left(1 - y^{(i)}\right) \frac{\partial}{\partial \mu} \ln(1-\mu) \\ &= \frac{1}{\mu}\sum_{i=1}^{n} 1_{\{y^{(i)}=1\}} - \frac{1}{1-\mu}\sum_{i=1}^{n} 1_{\{y^{(i)}=0\}}.\end{aligned}$$

(4) Auflösen der Maximum Likelihood Gleichungen

Nullsetzen der partiellen Ableitungen des Gradienten der Log Likelihood Funktion und Auflösen der resultierenden Log Likelihood Gleichungen ergibt dann die Maximum-Likelihood-Schätzer des Modells der Linearen Diskriminanzanalyse.

Beweis (fortgeführt)

Nullsetzen der ersten Gradientenkomponente ergibt

$$\begin{aligned}
&\frac{1}{\hat{\mu}} \sum_{i=1}^{n} 1_{\{y^{(i)}=1\}} - \frac{1}{1-\hat{\mu}} \sum_{i=1}^{n} 1_{\{y^{(i)}=0\}} = 0 \\
\Leftrightarrow\ &\frac{1}{\hat{\mu}} \sum_{i=1}^{n} y^{(i)} - \frac{1}{1-\hat{\mu}} \sum_{i=1}^{n} \left(1 - y^{(i)}\right) = 0 \\
\Leftrightarrow\ &\frac{1-\hat{\mu}}{\hat{\mu}} \sum_{i=1}^{n} y^{(i)} - \sum_{i=1}^{n} \left(1 - y^{(i)}\right) = 0 \\
\Leftrightarrow\ &\frac{1-\hat{\mu}}{\hat{\mu}} \sum_{i=1}^{n} y^{(i)} - n + \sum_{i=1}^{n} y^{(i)} = 0 \\
\Leftrightarrow\ &(1-\hat{\mu}) \sum_{i=1}^{n} y^{(i)} - \hat{\mu} n + \hat{\mu} \sum_{i=1}^{n} y^{(i)} = 0 \\
\Leftrightarrow\ &(1-\hat{\mu}) \sum_{i=1}^{n} y^{(i)} - \hat{\mu} n + \hat{\mu} \sum_{i=1}^{n} y^{(i)} = 0 \\
\Leftrightarrow\ &(1-\hat{\mu}+\hat{\mu}) \sum_{i=1}^{n} y^{(i)} = \hat{\mu} n \\
\Leftrightarrow\ &\hat{\mu} n = \sum_{i=1}^{n} y^{(i)} \\
\Leftrightarrow\ &\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} 1_{\{y^{(i)}=1\}}.
\end{aligned}$$

Beweis (fortgeführt)

Nullsetzen der zweiten Gradientenkomponente ergibt

$$\begin{aligned}
&\sum_{i=1}^{n} \left(1 - y^{(i)}\right) \left(\left(x^{(i)} - \hat{\mu}_0\right)^T \Sigma^{-1}\right) = 0 \\
&\quad \Leftrightarrow \sum_{i=1}^{n} 1_{\{y^{(i)}=0\}} \left(x^{(i)} - \hat{\mu}_0\right)^T = 0 \\
\Leftrightarrow &\sum_{i=1}^{n} 1_{\{y^{(i)}\}} x^{(i)} - \sum_{i=1}^{n} 1_{\{y^{(i)}=0\}} \hat{\mu}_0 = 0 \\
&\quad \Leftrightarrow \sum_{i=1}^{n} 1_{\{y^{(i)}=0\}} \hat{\mu}_0 = \sum_{i=1}^{n} 1_{\{y^{(i)}\}} x^{(i)} \\
&\quad \Leftrightarrow \hat{\mu}_0 = \frac{1}{\sum_{i=1}^{n} 1_{\{y^{(i)}=0\}}} \sum_{i=1}^{n} 1_{\{y^{(i)}\}} x^{(i)}.
\end{aligned}$$

Nullsetzen der dritten Gradientenkomponente ergibt dann in ähnlicher Weise den Maximum-Likelihood-Schätzer $\hat{\mu}_1$.

Beweis (fortgeführt)

Nullsetzen der vierten Gradientenkomponente ergibt dann schließlich

$$
\begin{aligned}
0 &= \frac{n}{2}\hat{\Sigma} - \frac{1}{2}\sum_{i=1}^{n}\left(x^{(i)} - \mu_{1_{\{y^{(i)}=1\}}}\right)\left(x^{(i)} - \mu_{1_{\{y^{(i)}=1\}}}\right)^T \\
\Leftrightarrow n\hat{\Sigma} &= \sum_{i=1}^{n}\left(x^{(i)} - \mu_{1_{\{y^{(i)}=1\}}}\right)\left(x^{(i)} - \mu_{1_{\{y^{(i)}=1\}}}\right)^T \\
\hat{\Sigma} &= \frac{1}{n}\sum_{i=1}^{n}\left(x^{(i)} - \mu_{1_{\{y^{(i)}=1\}}}\right)\left(x^{(i)} - \mu_{1_{\{y^{(i)}=1\}}}\right)^T
\end{aligned}
\quad .
$$

Anwendung

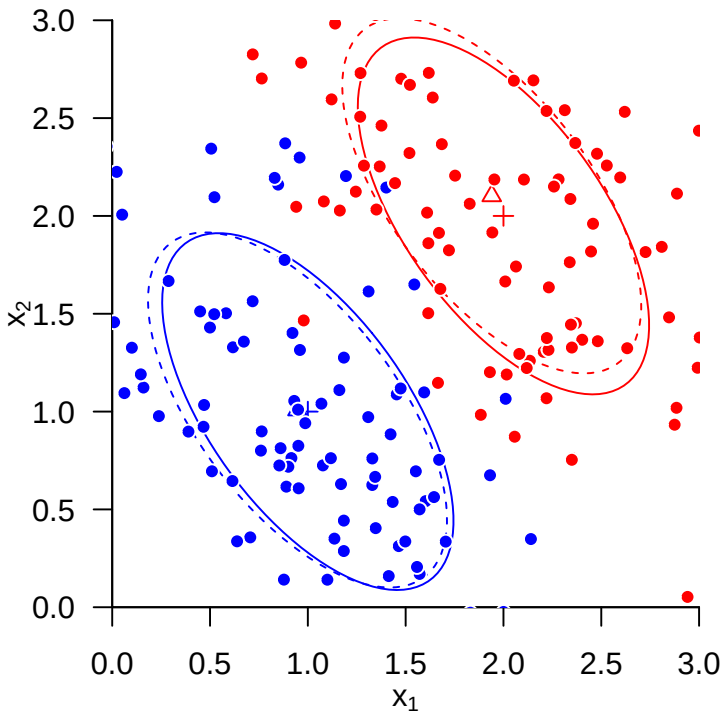
$$m = 2, n = 200, \mu = 0.5, \mu_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mu_1 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \Sigma = \begin{pmatrix} 0.40 & -0.30 \\ -0.30 & 0.60 \end{pmatrix}$$

```
# Parameterlernen
n            = ncol(x)                                                    # n
m            = nrow(x)                                                    # m
mu_hat       = mean(y)                                                    # \hat{\mu}
mu_0_hat     = rowMeans(x[,y == 0])                                       # \hat{\mu}_0
mu_1_hat     = rowMeans(x[,y == 1])                                       # \hat{\mu}_1
Sigma_hat    = matrix(rep(0,m^2), nrow = m)                               # \hat{\Sigma}
for(i in 1:n){
  Sigma_hat  = (Sigma_hat + (1/n)*
               ((y[i] == 0)*(x[,i]-mu_0_hat) %*% t((x[,i]-mu_0_hat))
               +(y[i] == 1)*(x[,i]-mu_1_hat) %*% t((x[,i]-mu_1_hat))))
}
beta_hat     = matrix(c((1/2)*( t(mu_0_hat) %*% solve(Sigma_hat) %*% mu_0_hat   # \hat{beta}
                              - t(mu_1_hat) %*% solve(Sigma_hat) %*% mu_1_hat)
                              + log(mu_hat/(1-mu_hat)),
                        -solve(Sigma_hat) %*% (mu_0_hat-mu_1_hat)),
                      nrow = m+1)
```

Anwendung

$$m = 2, n = 200, \hat{\mu} = 0.52, \hat{\mu}_0 = \begin{pmatrix} 0.95 \\ 1.00 \end{pmatrix}, \hat{\mu}_1 = \begin{pmatrix} 1.94 \\ 2.10 \end{pmatrix}, \hat{\Sigma} = \begin{pmatrix} 0.42 & -0.31 \\ -0.31 & 0.59 \end{pmatrix}$$

$+\ \mu_0$, $\bullet\ x^{(i)}$ mit $y^{(i)} = 0$, $\triangle \hat{\mu}_0$, $+\ \mu_1$, $\bullet\ x^{(i)}$ mit $y^{(i)} = 1$, $\triangle\ \hat{\mu}_1$



Anwendungsszenario

Modellformulierung

Inferenz und Klassifikation

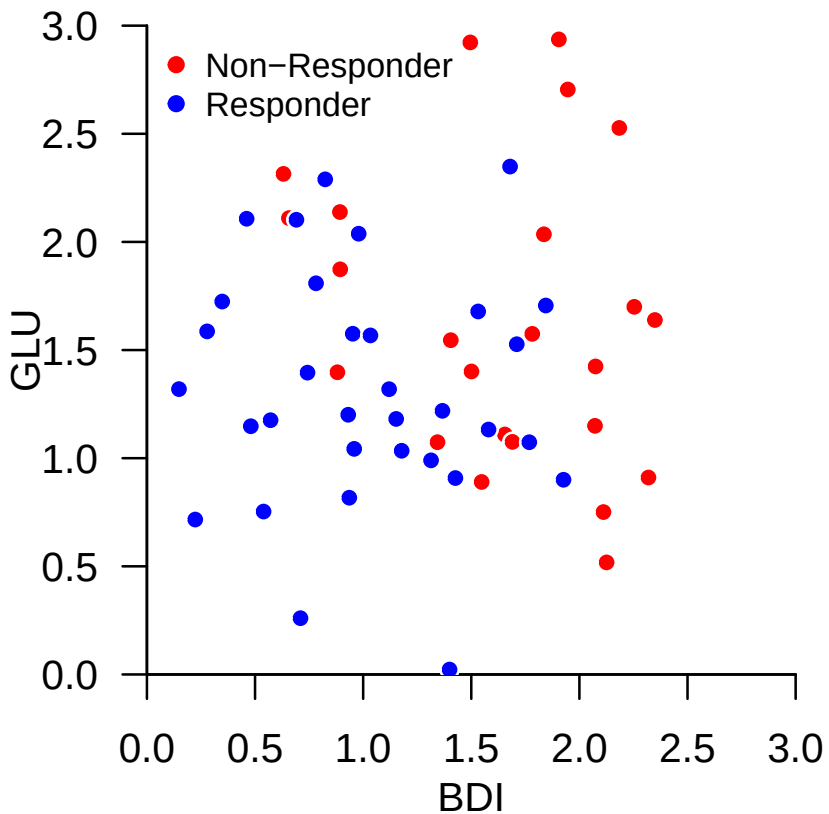
Lernen

Anwendungsbeispiel

Selbstkontrollfragen

Beispieldatensatz

BDI und GLU Werte bei Depressionsdiagnose als Prädiktoren von CBT Erfolg



Beispieldatensatz

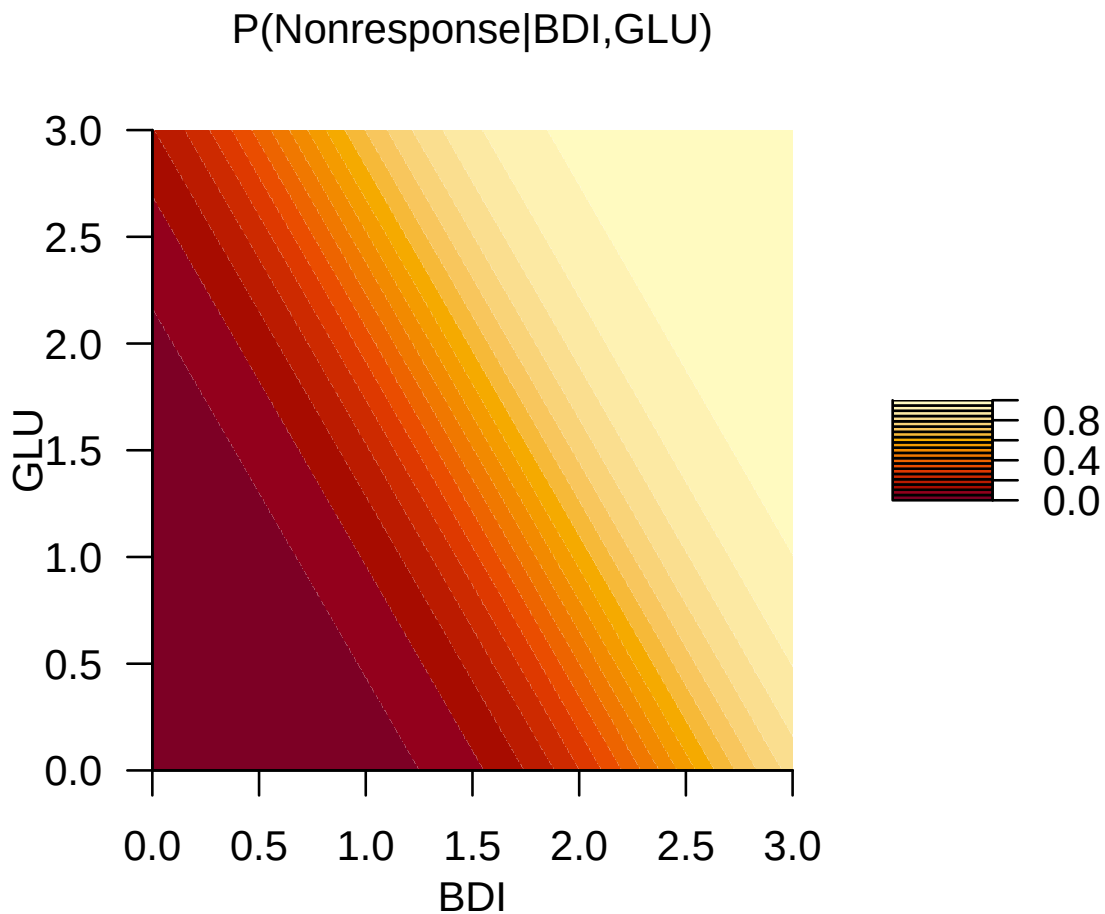
`BDI` und `GLU` Werte bei Depressionsdiagnose als Prädiktoren von CBT Erfolg `RES`

BDI	GLU	RES
0.74	1.40	1
0.22	0.72	1
0.82	2.29	1
2.07	1.15	0
1.71	1.53	1
1.77	1.07	1
1.95	2.70	0
2.18	2.53	0
0.93	1.20	1
1.34	1.07	0
2.35	1.64	0
1.43	0.91	1
1.66	1.11	0
0.28	1.59	1
2.13	0.52	0
1.37	1.22	1
0.89	2.14	0
0.88	1.40	0
0.98	2.04	1
1.93	0.90	1

Evaluation der Non-Response Wahrscheinlichkeit

```
D           = read.csv("./11_Daten/11_Lineare_Diskriminanzanalyse.csv")          # Datensatz
x           = t(D[,1:2])                                                        # Featurevektoren
y           = t(D[,3])                                                          # Label
n           = ncol(x)                                                           # n
m           = nrow(x)                                                           # m
mu_hat      = mean(y)                                                           # \hat{\mu}
mu_0_hat    = rowMeans(x[,y == 0])                                              # \hat{\mu}_0
mu_1_hat    = rowMeans(x[,y == 1])                                              # \hat{\mu}_1
Sigma_hat   = matrix(rep(0,m^2), nrow = m)                                      # \hat{\Sigma}
for(i in 1:n){
  Sigma_hat  = (Sigma_hat + (1/n)*
               ((y[i] == 0)*(x[,i]-mu_0_hat) %*% t((x[,i]-mu_0_hat))
               +(y[i] == 1)*(x[,i]-mu_1_hat) %*% t((x[,i]-mu_1_hat))))}
beta_hat     = matrix(c((1/2)*( t(mu_0_hat) %*% solve(Sigma_hat) %*% mu_0_hat   # \hat{beta}
                              - t(mu_1_hat) %*% solve(Sigma_hat) %*% mu_1_hat)
                              + log(mu_hat/(1-mu_hat)),
                        -solve(Sigma_hat) %*% (mu_0_hat-mu_1_hat)),
                      nrow = m+1)
x_min     = 0                                                                   # GLU/BDI Minimum
x_max     = 3                                                                   # GLU/BDI Maximum
x_res     = 5e2                                                                 # GLU/BDI Auflösung
bdi       = seq(x_min, x_max, length.out = x_res)                               # BDI
glu       = seq(x_min, x_max, length.out = x_res)                               # GLU
p_y       = matrix(rep(NaN, x_res*x_res), nrow = x_res)                         # p(y=1|(BDI, GLU))
for(i in 1:x_res){                                                              # BDI Iterationen
    for(j in 1:x_res){                                                          # GLU Iterationen
        x_tilde   = rbind(1, bdi[i], glu[j])                                    # \tilde{x}
        p_y[i,j]  = 1/(1+exp(-t(x_tilde) %*% beta_hat))}}                       # p(y=1|(BDI, GLU))
```

Evaluation der Non-Response Wahrscheinlichkeit



LOOCV zur Bestimmung der Featureprädiktivität

```
D       = read.csv("./11_Daten/11_Lineare_Diskriminanzanalyse.csv")             # Datensatz
K       = nrow(D)                                                               # Anzahl Cross Folds
p_y     = matrix(rep(NaN, K)  , nrow = 1)                                       # p(y = 1|x)
y_pred  = matrix(rep(NaN, K*2), nrow = K)                                       # Prädiktionsperformancearray
for(k in 1:K){                                                                  # K-fold LOOCV
  x_train     = t(D[-k,1:2])                                                    # Trainingsdatensatzfeatures
  y_train     = t(D[-k,3  ])                                                    # Trainingsdatensatzlabels
  x_test      = t(D[ k,1:2])                                                    # Testdatensatzfeaturevektor
  y_pred[k,1] = t(D[ k,3])                                                      # Testdatensatzfeaturevektorlabel
  n            = ncol(x_train)                                                  # n
  m            = nrow(x_train)                                                  # m
  mu_hat       = mean(y_train)                                                  # \hat{\mu}
  mu_0_hat     = rowMeans(x_train[,y_train == 0])                               # \hat{\mu}_0
  mu_1_hat     = rowMeans(x_train[,y_train == 1])                               # \hat{\mu}_1
  Sigma_hat    = matrix(rep(0,m^2), nrow = m)                                   # \hat{\Sigma}
  for(i in 1:n){
    Sigma_hat  = (Sigma_hat + (1/n)*
                 ((y_train[i] == 0)*(x_train[,i]-mu_0_hat) %*% t((x_train[,i]-mu_0_hat))
                 +(y_train[i] == 1)*(x_train[,i]-mu_1_hat) %*% t((x_train[,i]-mu_1_hat))))}
  beta_hat     = matrix(c((1/2)*( t(mu_0_hat) %*% solve(Sigma_hat) %*% mu_0_hat # \hat{beta}
                                - t(mu_1_hat) %*% solve(Sigma_hat) %*% mu_1_hat)
                                + log(mu_hat/(1-mu_hat)),
                                -solve(Sigma_hat) %*% (mu_0_hat-mu_1_hat)), nrow = m+1)
  x_test_tilde = rbind(1, x_test)                                               # \tilde{x}
  p_y[k]       = 1/(1+exp(-t(x_test_tilde) %*% beta_hat))                       # p(y=1|x)
  y_pred[k,2]  = as.numeric(p_y[k] >= 0.5)}                                     # Prädiktion
rp      = sum(y_pred[y_pred[,1] == 1,2] == 1)                                   # |(1,1)|
rn      = sum(y_pred[y_pred[,1] == 0,2] == 0)                                   # |(0,0)|
fp      = sum(y_pred[y_pred[,1] == 0,2] == 1)                                   # |(0,1)|
fn      = sum(y_pred[y_pred[,1] == 1,2] == 0)                                   # |(1,0)|
ACC     = (rp+rn)/(rp+fp+rn+fn)                                                 # Accuracy
SEN     = rp/(rp+fn)                                                            # Sensitivity
SPE     = rn/(rn+fp)                                                            # Specificity
cat("Accuracy   : "  , ACC, ", Sensitivity: " , SEN, ", Specificity: " , SPE)   # Ergebnisausgabe
```

```
Accuracy   :  0.7166667 , Sensitivity:  0.8235294 , Specificity:  0.5769231
```

Anwendungsszenario

Modellformulierung

Inferenz und Klassifikation

Lernen

Anwendungsbeispiel

Selbstkontrollfragen

Selbstkontrollfragen

1. Geben Sie die Definition einer Bernoullizufallsvariable wieder.
2. Geben Sie die Definition des Modells der Linearen Diskriminanzanalyse wieder.
3. Erläutern Sie die Generation von Daten unter dem Modell der Linearen Diskriminanzanalyse.
4. Geben Sie das Theorem zur Inferenz der Linearen Diskriminanzanalyse wieder.
5. Geben Sie die Definition der Klassifikationsregel der Linearen Diskriminanzanalyse wieder.
6. Geben Sie das Theorem zur Maximum-Likelihood-Schätzung der Linearen Diskriminanzanalyse wieder.
7. Erläutern Sie wie, mithilfe einer Linearen Diskriminanzanalyse die psychotherapeutische Nonresponse-wahrscheinlichkeit geschätzt werden kann.

Referenzen

Zhao, Shuping, Bob Zhang, Jian Yang, Jianhang Zhou, and Yong Xu. 2024. "Linear Discriminant Analysis." *Nature Reviews Methods Primers* 4 (1): 70. https://doi.org/10.1038/s43586-024-00346-y.