Multivariate Datenanalyse

MSc Psychologie WiSe 2021/22

Prof. Dr. Dirk Ostwald

(7) Lineare Diskriminanzanalyse und Logistische Regression

Vorbemerkungen

Lineare Diskriminanzanalyse

Logistische Regression

Selbstkontrollfragen

Appendix

Vorbemerkungen

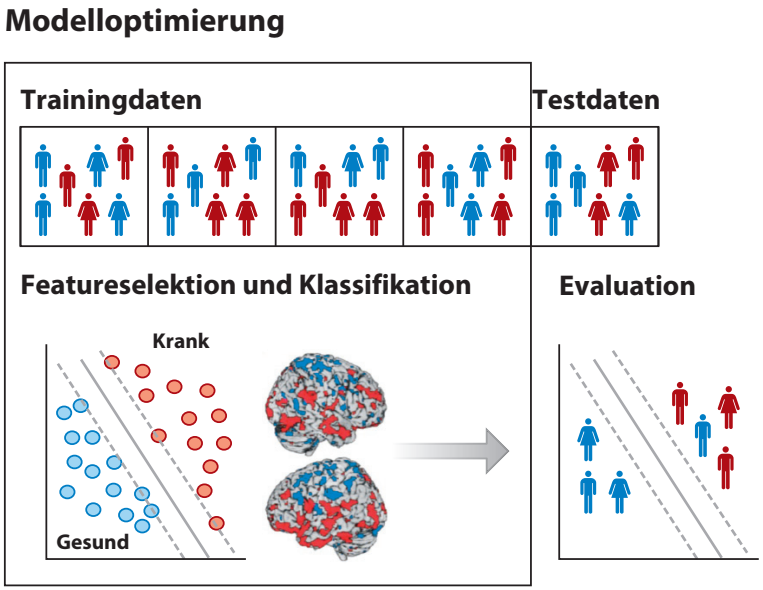
Lineare Diskriminanzanalyse

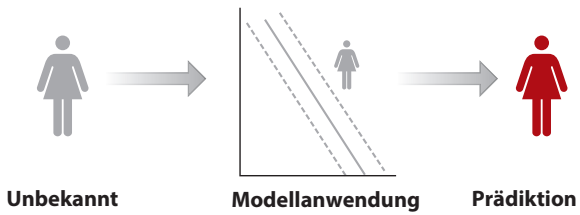
Logistische Regression

Selbstkontrollfragen

Appendix

Struktur der Prädiktiven Modellierung





nach Dwyer, Falkai, and Koutsouleris (2018)

Rhethorik der Prädiktiven Modellierung

Daten	Trainingsdaten und Testdaten
Statistisches Modell	Modell, Machine Learning Algorithmus
Schätzen von Parametern	Trainieren des Modells, Lernen von Parametern, Supervised Learning

Definition (Binärer Klassifikationstrainingdatensatz)

Ein *binärer Klassifikationsdatensatz*

$$\mathcal{D} := \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ..., (x^{(n)}, y^{(n)})\} \tag{1}$$

ist eine Menge von n *Trainingsdatenpunkten*

$$(x^{(i)}, y^{(i)}) \text{ mit } x^{(i)} \in \mathbb{R}^m \text{ und } y^{(i)} \in \{0, 1\} \text{ for } i = 1, ..., n, \tag{2}$$

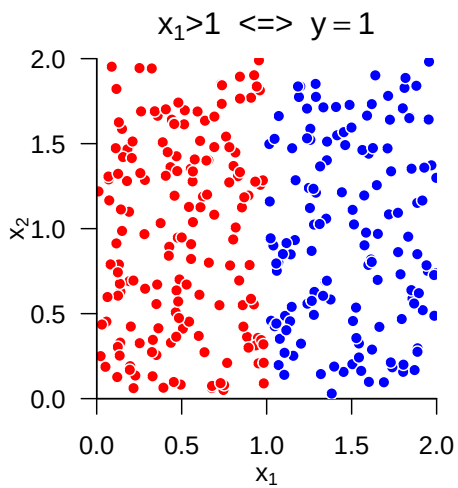
wobei $x^{(i)}$ m-*dimensionaler Featurevektor* und $y^{(i)}$ *Label* genannt wird

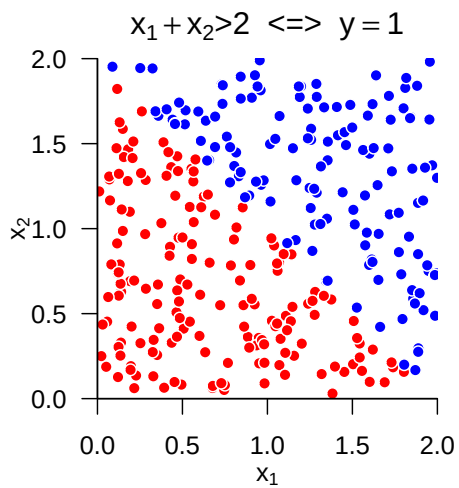
Bemerkungen

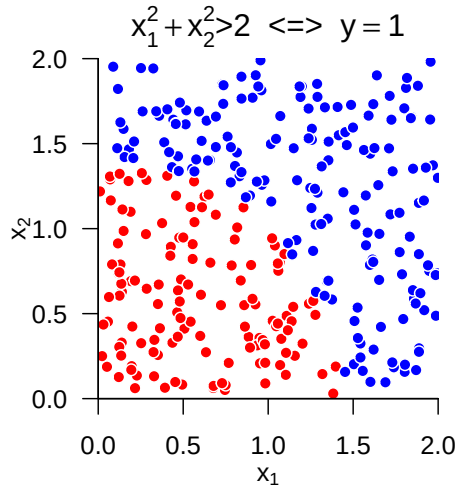
- $y^{(i)} \in \{0, 1\}$ bezeichnet die Klassenzugehörigkeit des Featurevektors $x^{(i)} \in \mathbb{R}^m$.
- Man beachte, dass hier $y^{(i)} \in \{0, 1\}$ gilt, wohingegen bei SVMs $y^{(i)} \in \{-1, 1\}$ ist.

Bivariate Featureplots mit $x^{(i)} \in \mathbb{R}^2, y^{(i)} \in \{0, 1\}, i = 1, ..., n$

$\bullet\ y = 0 \qquad \bullet\ y = 1$







Überblick

	Lineare Diskriminanzanalyse	Logistische Regression
Zufallsvektoren	$x \in \mathbb{R}^m, y \in \{0, 1\}$	$y \in \{0, 1\}$
Modell	$p(x, y) := B(y; \mu)N(x; \mu_0, \Sigma)^{1-y} N(x; \mu_1, \Sigma)^y$	$p(y) := B\left(y; \frac{1}{1+\exp(-x^T\beta)}\right)$
Inferenz	$p(y\|x) = \frac{1}{1+\exp(-\tilde{x}^T\beta)}$	Keine
Klassifikation	$\delta(x) = 1$ für $p(y = 1\|x) > p(y = 0\|x)$	$\delta(x) = 1$ für $p(y = 1) > p(y = 0)$
Diskriminanz	$f(x) = w^T x + x_0, (w_0, w) = \beta$	$f(x) = w^T x + x_0, (w_0, w) = \beta$
Parameterlernen	Analytische Likelihood Maximierung	Numerische Likelihood Maximierung

Vorbemerkungen

Lineare Diskriminanzanalyse

Logistische Regression

Selbstkontrollfragen

Appendix

Definition (Multivariate Normalverteilung)

X sei ein m-dimensionaler Zufallsvektor mit Ergebnisraum $\mathbb{R}^m$ und WDF

$$p : \mathbb{R}^n \to \mathbb{R}_{>0},\ x \mapsto p(x) := (2\pi)^{-\frac{m}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right). \tag{3}$$

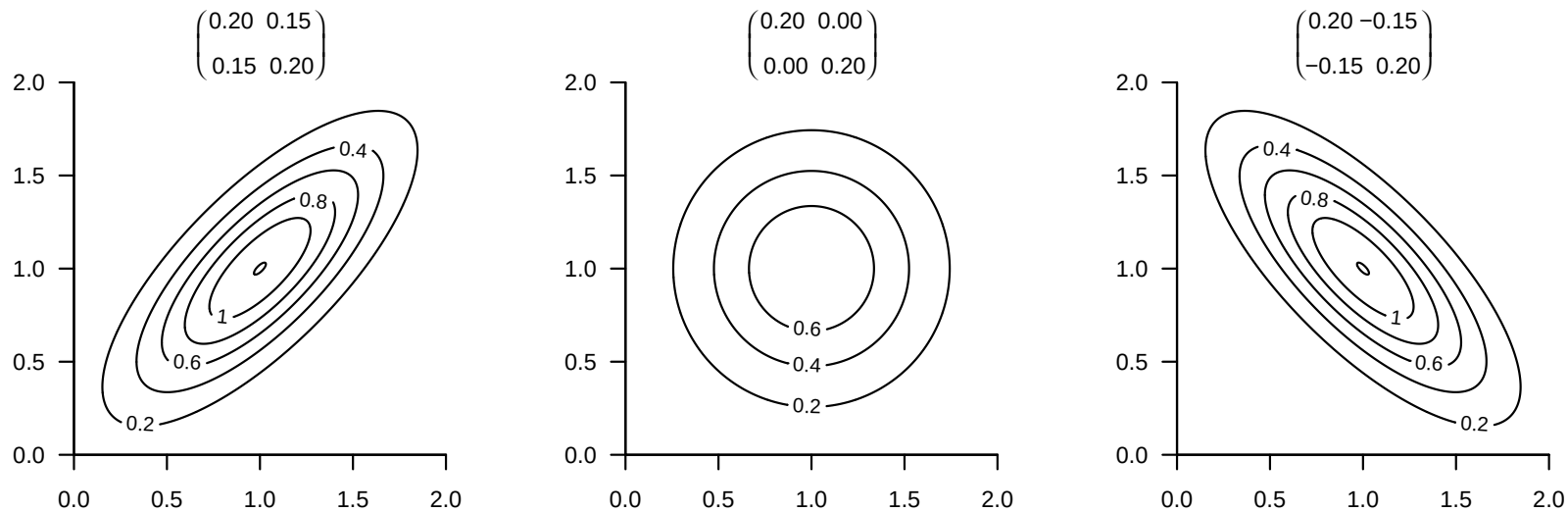
Dann sagen wird, dass X einer *multivariaten (oder m-dimensionalen) Normalverteilung* mit *Erwartungswertparameter* $\mu \in \mathbb{R}^m$ und positive-definitem *Kovarianzmatrixparameter* $\Sigma \in \mathbb{R}^{m \times m}$ unterliegt und nennen X einen *(multivariat) normalverteilten Zufallsvektor*. Wir kürzen dies mit $X \sim N(\mu, \Sigma)$ ab. Die WDF eines multivariat normalverteilten Zufallsvektors bezeichnen wir mit

$$N(x; \mu, \Sigma) := (2\pi)^{-\frac{m}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right). \tag{4}$$

Bemerkungen

- Der Parameter $\mu \in \mathbb{R}^m$ entspricht dem Wert höchster Wahrscheinlichkeitsdichte
- Die Diagonalelemente von Σ spezifizieren die Breite der WDF bezüglich $X_1, \ldots, X_m$.
- Das i,jte Element von Σ spezifiziert die Kovarianz on X_i und X_j.
- Der Term $(2\pi)^{-\frac{m}{2}} |\Sigma|^{-\frac{1}{2}}$ ist die Normalsieriungskonstante für den Exponentialfunktionsterm.

Zweidimensionale Normalverteilungen



Definition (Bernoulli Verteilung)

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

$$p : \mathcal{X} \to [0, 1], x \mapsto p(x) := \mu^x (1 - \mu)^{1-x} \text{ mit } \mu \in [0, 1]. \tag{5}$$

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

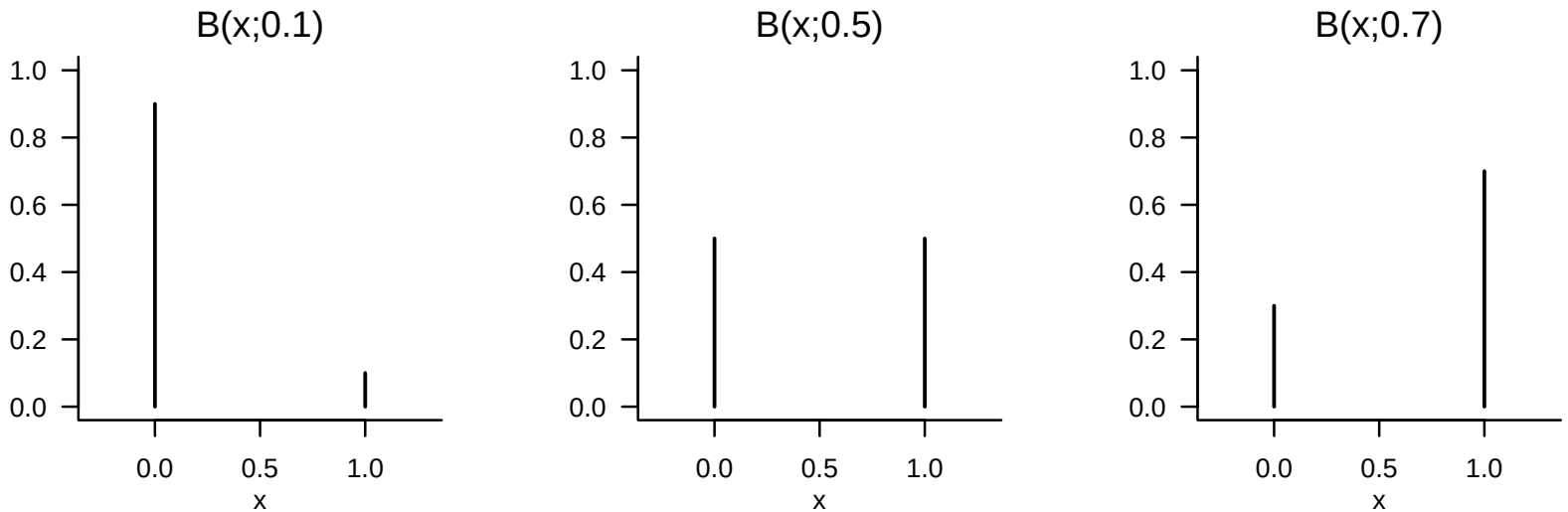
$$B(x; \mu) := \mu^x (1 - \mu)^{1-x}. \tag{6}$$

Bemerkungen

- Eine Bernoulli-Zufallsvariable kann als Modell eines Münzwurfs dienen.
- $\mu \in [0, 1]$ ist die Wahrscheinlichkeit dafür, dass X den Wert 1 annimmt,

$$\mathbb{P}(X = 1) = \mu^1 (1 - \mu)^{1-1} = \mu. \tag{7}$$

Bernoulli Verteilungen



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

$$\mathbb{P}(X, Y) = \mathbb{P}(Y)\mathbb{P}(X|Y) \tag{8}$$

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

$$p(y) = B(y; \mu) \tag{9}$$

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

$$p(x|y) = N(x; \mu_0, \Sigma)^{1-y} N(x; \mu_1, \Sigma)^y \tag{10}$$

mit $\mu_0, \mu_1 \in \mathbb{R}^m$ und $\Sigma \in \mathbb{R}^{m \times m}$ p.d. definiert ist. Wir bezeichnen die gemischte WDF/WMF des LDA Modells mit

$$p(x, y) := p(y)p(x|y) = B(y; \mu)N(x; \mu_0, \Sigma)^{1-y} N(x; \mu_1, \Sigma)^y \tag{11}$$

Bemerkung

Aus generativer Sicht wird ein Trainingsdatensatz

$$\left\{ \left(x^{(i)}, y^{(i)} \right) \right\}_{i=1}^{n} \text{ mit } x^{(i)} \in \mathbb{R}^m \text{ und } y^{(i)} \in \{0, 1\} \tag{12}$$

eines LDA Modells wie folgt erzeugt:

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

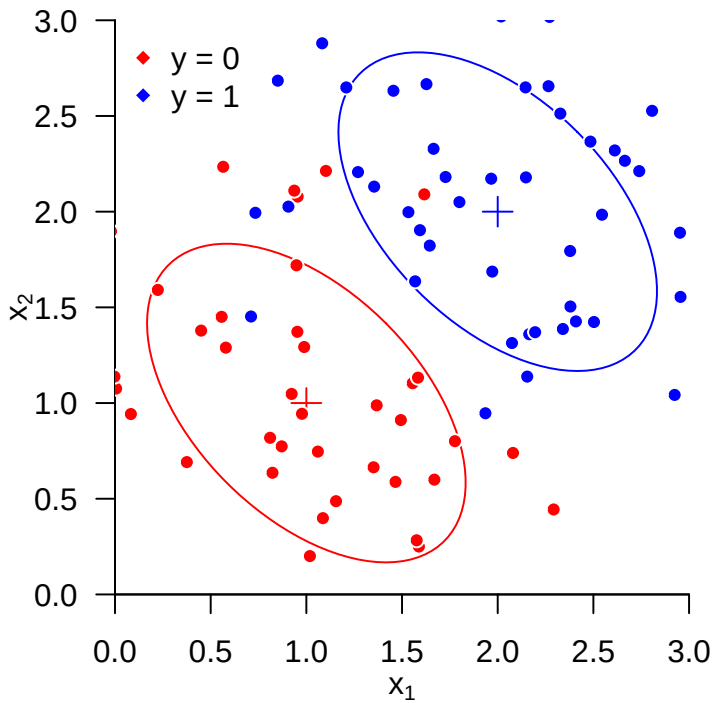
(2) In Abhängigkeit vom Wert von $y^{(i)}$ wird $x^{(i)}$ dann durch Ziehen aus einer multivariaten Normalverteilung mit Kovarianzmatrixparameter Σ und Erwartungswertparameter μ_0 für $y^{(i)} = 0$ oder μ_1 für $y^{(i)} = 1$ erzeugt.

Datengeneration

```
# R Paket für multivariate Normalverteilung
library(mvtnorm)
set.seed(0)

# Modellparameter
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( 0.50, -0.25,                     # Kovarianzmatrixparameter
                  -0.25,  0.50),
                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){
  y[i]   = rbinom(1,1,mu)                           # y^{(i)} \sim B(\mu)
  x[,i] = ((y[i] == 0)*rmvnorm(1, mu_0, Sigma)      # x^{i} \sim N(\mu_0,\Sigma)^{1-y}N(\mu_1,\Sigma)^y
          +(y[i] == 1)*rmvnorm(1, mu_1, Sigma))
}

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

Datengeneration

$$\mu = 0.5, \mu_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mu_1 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \Sigma = \begin{pmatrix} 0.50 & -0.10 \\ -0.10 & 0.50 \end{pmatrix}$$



Theorem (LDA Inferenz)

$p(x, y)$ sei die WMF/WDF eines LDA Model. Dann gilt

$$p(y = 1|x) = \frac{1}{1 + \exp(-\tilde{x}^T \beta)} \text{ und } p(y = 0|x) = 1 - p(y = 1|x), \tag{13}$$

wobei

$$\tilde{x} := \begin{pmatrix} 1 \\ x \end{pmatrix} \in \mathbb{R}^{m+1} \tag{14}$$

der *erweiterten Featurevektor* und

$$\beta := \begin{pmatrix} \frac{1}{2}\left(\mu_0^T \Sigma^{-1} \mu_0 - \mu_1^T \Sigma^{-1} \mu_1\right) + \ln\left(\frac{\mu}{1-\mu}\right) \\ -\Sigma^{-1}(\mu_0 - \mu_1) \end{pmatrix} \in \mathbb{R}^{m+1}. \tag{15}$$

der *Inferenzparametervektor* sind.

Bemerkungen

- Für einen Beweis verweisen wir auf den Appendix.
- $p(y|x)$ kann zur Prädiktion der Klasse eines $x \in \mathbb{R}^m$ genutzt werden.
- Diese Prädiktion hängt von den LDA Modellparametern $\mu, \mu_0, \mu_1, \Sigma$ ab.

Definition (Klassifikationsregel der linearen Diskriminanzanalyse)

$p(x, y)$ sei die WMF/WDF eines LDA Modells. Dann ist die *Klassifikationregel* definiert als

$$\delta : \mathbb{R}^m \to \{0, 1\}, x \mapsto \delta(x) := \begin{cases} 0 & \text{für } p(y = 0|x) \geq p(y = 1|x) \\ 1 & \text{für } p(y = 0|x) < p(y = 1|x) \end{cases} \tag{16}$$

Bemerkung

- Es gilt

$$\delta(x) = 1 \Leftrightarrow p(y = 1|x) > p(y = 0|x) \Leftrightarrow p(y = 1|x) > 0.5. \tag{17}$$

Inferenz und Klassifikation bei bekannten Modellparametern

$$\mu = 0.5, \mu_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mu_1 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \Sigma = \begin{pmatrix} 0.10 & -0.05 \\ -0.05 & 0.10 \end{pmatrix}$$

```
# Inferenz und Klassifikation für die ersten k Datenpunkte
library(matlib)                                          # Matrixtools
k           =  10                                        # Anzahl Datenpunkte
x_tilde     = rbind(rep(1,k), x[,1:k])                   # erweiterte Featurevektoren
beta        = matrix(                                    # Inferenzparametervektor
              c((1/2)*(  t(mu_0) %*% inv(Sigma) %*% mu_0
                       - t(mu_1) %*% inv(Sigma) %*% mu_1)
                       + log(mu/(1-mu)),
                -inv(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	1.137	1.800	2.380	0.871	2.485	2.145	0.083	0.823	2.61	1.466
x_2	3.243	2.050	1.504	0.774	2.366	2.649	0.943	0.636	2.32	0.588
p(y = 1|x)	0.996	0.968	0.972	0.004	0.999	0.999	0.000	0.002	1.00	0.022
delta(x)	1.000	1.000	1.000	0.000	1.000	1.000	0.000	0.000	1.00	0.000
y	1.000	1.000	1.000	0.000	1.000	1.000	0.000	0.000	1.00	0.000

Theorem (LDA Diskriminanzfunktion)

$p(x, y)$ sei die WMF/WDF eines LDA Mmodells und $\beta \in \mathbb{R}^{m+1}$ sei der Inferenzparametervektor. Dann kann die LDA Klassifikationsregel δ als eine lineare Diskriminanzfunktion der Form

$$h : \mathbb{R}^m \to \{0, 1\}, x \mapsto h(x) := g(f(x)), \tag{18}$$

mit

$$f : \mathbb{R}^m \to \mathbb{R}, x \mapsto f(x) := w^T x + w_0, \tag{19}$$

und

$$g : \mathbb{R} \to \{0, 1\}, f(x) \mapsto g(f(x)) := \begin{cases} 0, & f(x) \geq 0 \\ 1, & f(x) < 0 \end{cases} \tag{20}$$

geschrieben werden, d.h. es gilt $\delta(x) = h(x)$ für alle $x \in \mathbb{R}^m$. Inbesondere gilt dabei

$$w_0 = \beta_1 \text{ und } w = (\beta_2, \ldots, \beta_{m+1})^T \tag{21}$$

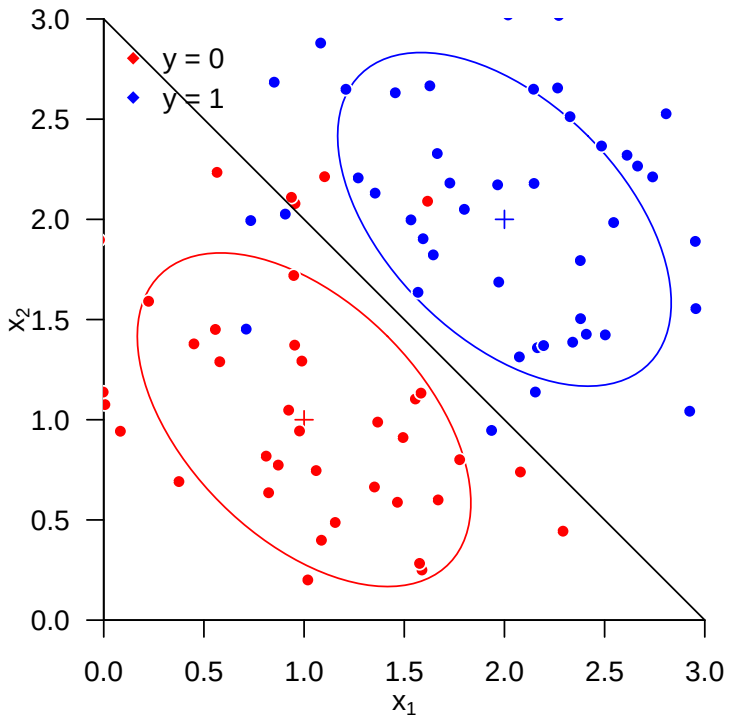
Bemerkungen

- Für einen Beweis verweisen wir auf den Appendix.
- Zur Visualisierung in zweidimensionalen Featureräumen dient die Graphgleichungen für Hyperebenen

$$f(x) = 0 \Leftrightarrow w^T x + w_0 = 0 \Leftrightarrow w_1 x_1 + w_2 x_2 + w_0 = 0 \Leftrightarrow x_2 = -\frac{w_1}{w_2} x_1 - \frac{w_0}{w_2} \tag{22}$$

Implementation der Diskriminanfunktion bei bekannten Modellparametern

```
# Diskriminanzfunktion
x_1  = seq(0,3,len = 1e2)                    # x_1
w_0  = beta[1]                               # w_0
w    = beta[2:(m+1)]                         # w
x_2  = -(w[1]/w[2])*x_1 - (w_0/w[2])         # x_2
```



Theorem (LDA Maximum Likelihood Schätzer)

$p(x, y)$ sei die WMF/WDF eines LDA Modells 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

$$
\begin{aligned}
\hat{\mu} &= \frac{1}{n} \sum_{i=1}^{n} 1_{\{y^{(i)}=1\}}, \\
\hat{\mu}_0 &= \frac{1}{\sum_{i=1}^{n} 1_{\{y^{(i)}=0\}}} \sum_{i=1}^{n} 1_{\{y^{(i)}=0\}} x^{(i)}, \\
\hat{\mu}_1 &= \frac{1}{\sum_{i=1}^{n} 1_{\{y^{(i)}=1\}}} \sum_{i=1}^{n} 1_{\{y^{(i)}=1\}} x^{(i)}, \\
\hat{\Sigma} &= \frac{1}{n} \sum_{i=1}^{n} \left(x^{(i)} - \hat{\mu}_{y^{(i)}} \right) \left(x^{(i)} - \hat{\mu}_{y^{(i)}} \right)^T .
\end{aligned}
\tag{23}
$$

Bemerkungen

- Für einen Beweis verweisen wir auf den Appendix

Bemerkungen (fortgeführt)

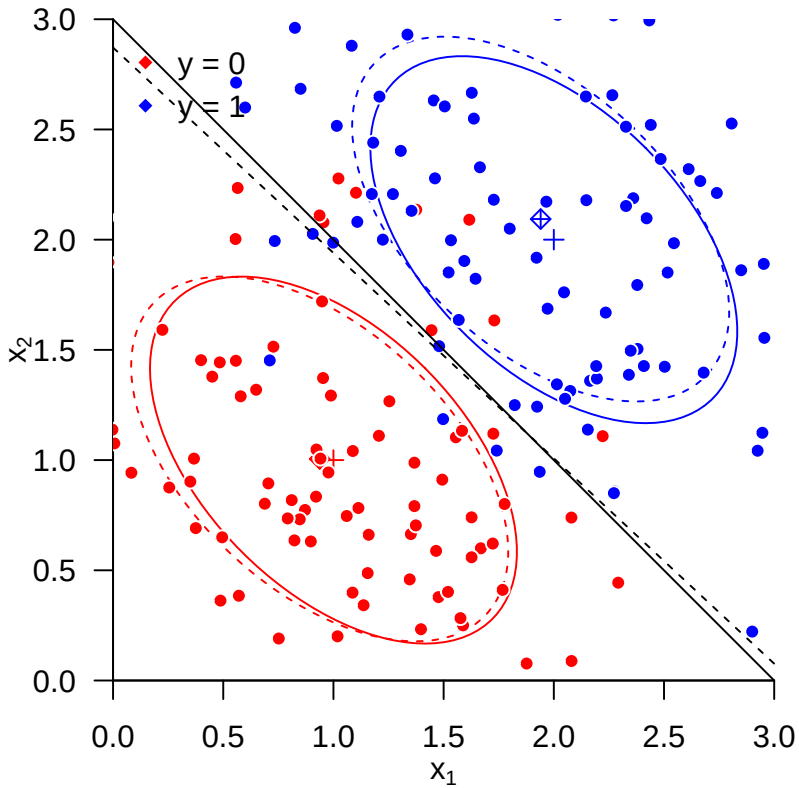
- μ wird als die relative Häufigkeit der 1en im Trainingsdatensatz geschätzt.
- μ_0 und μ_1 werden als Stichprobenmittel aller $x^{(i)}$ mit $y^{(i)} = 0$ bzw. $y^{(i)} = 1$ geschätzt.
- Σ wird durch die empirische Kovarianzmatrix aller $x^{(i)}, i = 1, \ldots, n$ geschätzt.
- Substitution ergibt die Schätzer $\hat{\beta}$, $\hat{w}$, $\hat{w}_0$ und $\hat{h}$

Implementation

```
# Parameterlernen bei gegebenen Featurevektorset x \in \mathbb{R}^{m \times n} und Labelset y \in {0,1}^n
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) %*% inv(Sigma_hat) %*% mu_0_hat  # \hat{beta}
                               - t(mu_1_hat) %*% inv(Sigma_hat) %*% mu_1_hat)
                               + log(mu_hat/(1-mu_hat)),
                         -inv(Sigma_hat) %*% (mu_0_hat-mu_1_hat)),
                       nrow = m+1)
w_0_hat       = beta_hat[1]                                                # \hat{w}
w_hat         = beta_hat[2:(m+1)]                                          # \hat{w_0}
x_2_hat       = -(w_hat[1]/w_hat[2])*x_1 - (w_0_hat/w_hat[2])              # \hat{h}
```

Implementation

$$\hat{\mu} = 0.52, \hat{\mu}_0 = \begin{pmatrix} 0.94 \\ 1.01 \end{pmatrix}, \hat{\mu}_1 = \begin{pmatrix} 1.94 \\ 2.09 \end{pmatrix}, \hat{\Sigma} = \begin{pmatrix} 0.53 & -0.26 \\ -0.26 & 0.49 \end{pmatrix}$$



Prädiktion des Studienerfolgs

Nach Rudolf and Buse (2020) Kapitel 4

Datensatz zum Verhältnis psychologischer Diagnostik und Studienerfolg

$n = 30$ Studierende naturwissenschaftlicher Studiengänge

Featurevektor $x \in \mathbb{R}^4$

- x_1 Intelligenztestscore
- x_2 Mathematiktestscore
- x_3 Gewissenhaftigkeitscore
- x_4 Verträglichkeitscore

Label $y \in \{0, 1\}$

- 0: ungenügend (Studienabbruch aufgrund nicht bestandener Prüfungen)
- 1: gut (Abschlussnote besser als 2.5)

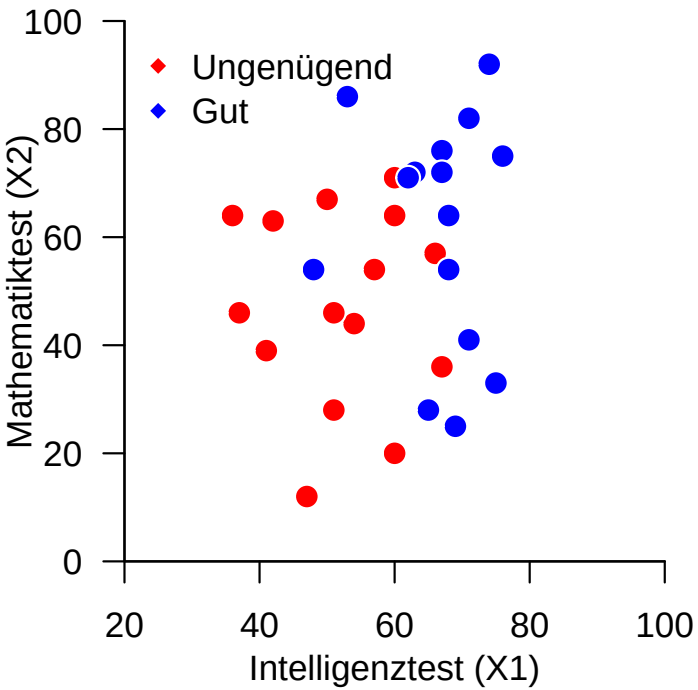
Datensatz $i = 1, ..., 15$

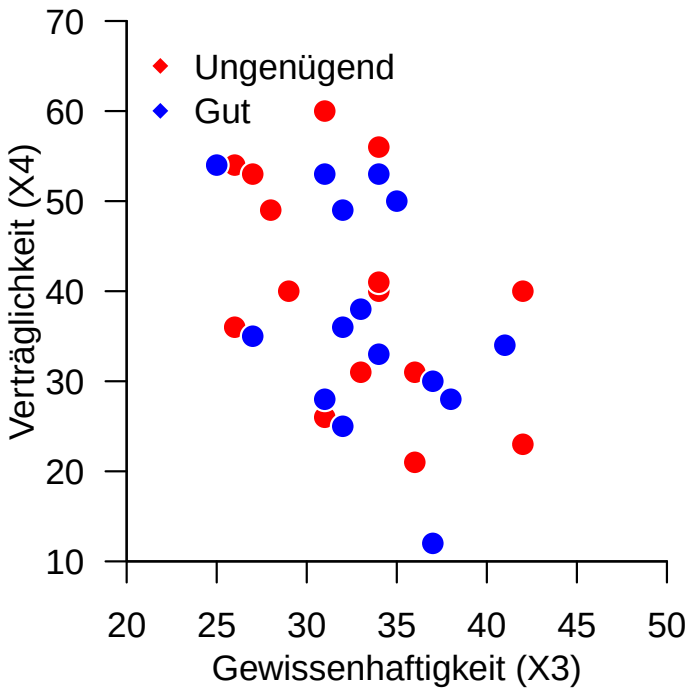
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
X1	54	60	67	41	66	51	51	37	57	47	50	42	60	36	60
X2	44	20	36	39	57	28	46	46	54	12	67	63	64	64	71
X3	31	33	26	31	34	42	34	36	28	34	27	26	29	36	42
X4	60	31	54	26	56	23	40	31	49	41	53	36	40	21	40
y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Datensatz $i = 16, ..., 30$

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
X1	71	65	67	68	75	71	68	63	48	53	62	69	67	74	76
X2	41	28	76	54	33	82	64	72	54	86	71	25	72	92	75
X3	37	33	38	34	25	32	34	32	41	27	31	32	31	35	37
X4	30	38	28	53	54	49	33	36	34	35	53	25	28	50	12
y	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Datensatz


Ungenügend
Gut
Mathematiktest (X2)
Intelligenztest (X1)
0
20
40
60
80
100
20
40
60
80
100

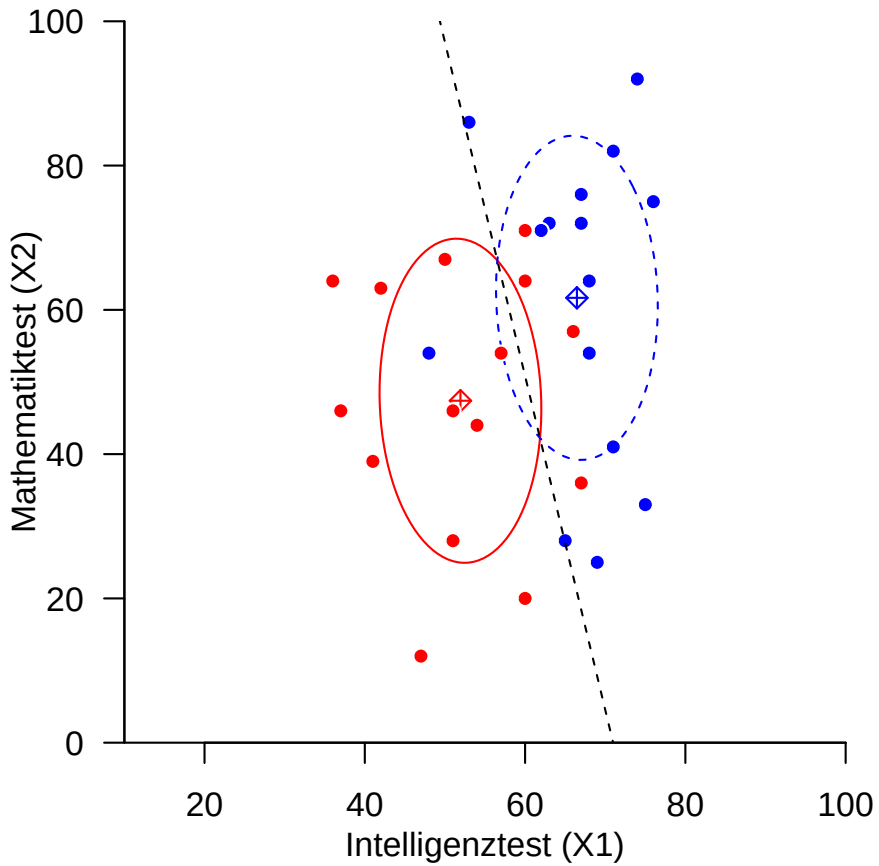

Ungenügend
Gut
Verträglichkeit (X4)
Gewissenhaftigkeit (X3)
10
20
30
40
50
60
70
20
25
30
35
40
45
50

Parameterlernen und lineare Diskriminanzfunktion

```
library(matlib)                                                          # Matrixtools
D             = read.table(file.path(getwd(), "7_Daten", "studienerfolg.csv")) # Datensatz
x             = as.matrix(D[1:2,])                                       # Featureselektion
y             = as.matrix(D[5,])                                         # Label
D             = rbind(x,y)                                               # Featureselected Datensatz
n             = ncol(x)                                                  # Datensatzgröße
m             = nrow(x)                                                  # Featurevektordimensionalität
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(0,4), nrow = 2)                               # \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) %*% inv(Sigma_hat) %*% mu_0_hat  # \hat{beta}
                               - t(mu_1_hat) %*% inv(Sigma_hat) %*% mu_1_hat)
                               + log(mu_hat/(1-mu_hat)),
                       -inv(Sigma_hat) %*% (mu_0_hat-mu_1_hat)), nrow = 3)
w_0_hat       = beta_hat[1]                                              # \hat{w}
w_hat         = beta_hat[2:(m+1)]                                        # \hat{w_0}
x_1           = seq(min(x[1,]), max(x[1,]), len = 1e2)                   # x_1
x_2_hat       = -(w_hat[1]/w_hat[2])*x_1 - (w_0_hat/w_hat[2])            # \hat{h}
x_tilde       = rbind(1, x)
p_y_giv_x     = 1/(1+exp(-t(x_tilde) %*% beta_hat))
delta         = as.numeric(p_y_giv_x >= 0.5)
cat("Accuracy: ", mean(delta == y))
```

```
> Accuracy:  0.833
```

Parameterlernen und lineare Diskriminanzfunktion



Lineare Diskriminanzanalyse | Leave-One-Out Cross-Validation, $m = 2$

```
# Datensatz
D                = read.table(file.path(getwd(), "7_Daten", "studienerfolg.csv"))
x                = as.matrix(D[1:2,])
y                = as.matrix(D[5,])
D                = rbind(x,y)
K                = ncol(D)
p_y_giv_x        = matrix(rep(NaN, ncol(y)), nrow = 1)
delta            = matrix(rep(NaN, ncol(y)), nrow = 1)

# K-fache Leave-One-Out Cross-Validation
for(k in 1:K){

    # Datensatzpartition
    x_train      = as.matrix(x[,-k])
    y_train      = as.matrix(y[,-k])
    x_test       = as.matrix(x[, k])
    y_test       = as.matrix(y[, k])

    # Trainingsdatensatz-basiertes Parameterlernen
    n            = ncol(x_train)
    m            = nrow(x_train)
    mu_hat       = mean(y_train)
    mu_0_hat     = rowMeans(x_train[, y_train == 0])
    mu_1_hat     = rowMeans(x_train[, y_train == 1])
    Sigma_hat    = matrix(rep(0,m^2), nrow = m)
    for(i in 1:n){
      Sigma_hat  = (Sigma_hat + (1/n)*
                   ((y_train[i] == 0)*(x_train[,i]-mu_0_hat) %*% t((x[,i]-mu_0_hat))
                   +(y_train[i] == 1)*(x_train[,i]-mu_1_hat) %*% t((x[,i]-mu_1_hat))))
    }
    beta_hat     = matrix(c((1/2)*( t(mu_0_hat) %*% inv(Sigma_hat) %*% mu_0_hat  # \hat{beta}
                                  - t(mu_1_hat) %*% inv(Sigma_hat) %*% mu_1_hat)
                                  + log(mu_hat/(1-mu_hat)),
                         -inv(Sigma_hat) %*% (mu_0_hat-mu_1_hat)), nrow = m+1)

    # Prädiktion
    x_test_tilde = rbind(1, x_test)
    p_y_giv_x[k] = 1/(1+exp(-t(x_test_tilde) %*% beta_hat))
    delta[k]     = as.numeric(p_y_giv_x[k] >= 0.5)
}
cat("Accuracy: ", mean(delta == y))
```

```
> Accuracy:  0.667
```

Lineare Diskriminanzanalyse | Leave-One-Out Cross-Validation, $m = 2$

k	x_1	x_2	p(y = 1|x)	delta(x)	y
1	54	44	1.0	1	0
2	60	20	1.0	1	0
3	67	36	1.0	1	0
4	41	39	1.0	1	0
5	66	57	0.0	0	0
6	51	28	0.0	0	0
7	51	46	0.0	0	0
8	37	46	0.0	0	0
9	57	54	0.1	0	0
10	47	12	0.0	0	0
11	50	67	0.2	0	0
12	42	63	0.0	0	0
13	60	64	0.9	1	0
14	36	64	0.0	0	0
15	60	71	1.0	1	0
16	71	41	0.7	1	1
17	65	28	0.1	0	1
18	67	76	1.0	1	1
19	68	54	0.9	1	1
20	75	33	0.8	1	1
21	71	82	1.0	1	1
22	68	64	0.9	1	1
23	63	72	0.9	1	1
24	48	54	0.0	0	1
25	53	86	0.5	0	1
26	62	71	0.8	1	1
27	69	25	0.5	0	1
28	67	72	0.9	1	1
29	74	92	1.0	1	1
30	76	75	1.0	1	1

Prediction Accuracy = 0.67.

Lineare Diskriminanzanalyse | Leave-One-Out Cross-Validation, $m = 3$

k	x_1	x_2	x_3	p(y = 1|x)	delta(x)	y
1	54	44	31	1.0	1	0
2	60	20	33	1.0	1	0
3	67	36	26	1.0	1	0
4	41	39	31	0.0	0	0
5	66	57	34	0.0	0	0
6	51	28	42	0.0	0	0
7	51	46	34	0.0	0	0
8	37	46	36	0.0	0	0
9	57	54	28	0.0	0	0
10	47	12	34	0.0	0	0
11	50	67	27	0.0	0	0
12	42	63	26	0.0	0	0
13	60	64	29	0.0	0	0
14	36	64	36	0.0	0	0
15	60	71	42	0.4	0	0
16	71	41	37	0.0	0	1
17	65	28	33	0.0	0	1
18	67	76	38	0.4	0	1
19	68	54	34	0.2	0	1
20	75	33	25	0.2	0	1
21	71	82	32	0.9	1	1
22	68	64	34	0.6	1	1
23	63	72	32	0.5	0	1
24	48	54	41	0.0	0	1
25	53	86	27	0.2	0	1
26	62	71	31	0.7	1	1
27	69	25	32	0.3	0	1
28	67	72	31	0.9	1	1
29	74	92	35	1.0	1	1
30	76	75	37	1.0	1	1

Prediction Accuracy = 0.60.

Lineare Diskriminanzanalyse | Leave-One-Out Cross-Validation, $m = 4$

k	x_1	x_2	x_3	x_4	p(y = 1|x)	delta(x)	y
1	54	44	31	60	0.0	0	0
2	60	20	33	31	0.0	0	0
3	67	36	26	54	0.0	0	0
4	41	39	31	26	0.0	0	0
5	66	57	34	56	0.0	0	0
6	51	28	42	23	0.0	0	0
7	51	46	34	40	0.0	0	0
8	37	46	36	31	0.0	0	0
9	57	54	28	49	0.0	0	0
10	47	12	34	41	0.0	0	0
11	50	67	27	53	0.0	0	0
12	42	63	26	36	0.0	0	0
13	60	64	29	40	0.0	0	0
14	36	64	36	21	0.0	0	0
15	60	71	42	40	0.0	0	0
16	71	41	37	30	0.0	0	1
17	65	28	33	38	0.0	0	1
18	67	76	38	28	0.1	0	1
19	68	54	34	53	0.0	0	1
20	75	33	25	54	1.0	1	1
21	71	82	32	49	1.0	1	1
22	68	64	34	33	1.0	1	1
23	63	72	32	36	1.0	1	1
24	48	54	41	34	0.0	0	1
25	53	86	27	35	0.9	1	1
26	62	71	31	53	0.3	0	1
27	69	25	32	25	1.0	1	1
28	67	72	31	28	1.0	1	1
29	74	92	35	50	1.0	1	1
30	76	75	37	12	1.0	1	1

Prediction Accuracy = 0.80.

Vorbemerkungen

Lineare Diskriminanzanalyse

Logistische Regression

Selbstkontrollfragen

Appendix

Definition (Generalisiertes Lineares Modell)

$x \in \mathbb{R}^m$ sei ein erweiterter Featurevektor und y das assoziierte Label. Weiterhin sei für einen *Parametervektor* $\beta \in \mathbb{R}^m$

$$\eta := x^T \beta \tag{24}$$

ein *linearer Prädiktor*. Dann ist ein generalisierte lineares Modell definiert mithilfe einer zweimal differenzierbaren und invertierbaren *link function*

$$g : \mathbb{R} \to \mathbb{R}, \mathbb{E}(y) \mapsto g(\mathbb{E}(y)) =: \eta. \tag{25}$$

definiert. Die Inverse der link function,

$$g^{-1} : \mathbb{R} \to \mathbb{R}, \eta \mapsto g^{-1}(\eta) = \mathbb{E}(y) \tag{26}$$

heißt *mean function* und wird mit f bezeichnet, so dass

$$f : \mathbb{R} \to \mathbb{R}, \eta \mapsto f(\eta) = \mathbb{E}(y). \tag{27}$$

Definition (Allgemeines Lineares Modell als Generalisiertes Lineares Modell)

Das Allgemeine Lineare Modell mit u.i.v. Störvariablen ist das Generalisierte Lineare Model, bei dem

1. die Labelvariable eine univariat normalverteilte Zufallsvariable

$$y \sim N(\mu, \sigma^2), \tag{28}$$

   ist und

2. die link function durch die Identität

$$g : \mathbb{R} \to \mathbb{R}, \mu \mapsto g(\mu) := \mu =: \eta. \tag{29}$$

   gegeben ist.

Weil die Inverse der Identität wiederrum die Identität ist, folgt, dass die mean function des ALM durch

$$f : \mathbb{R} \to \mathbb{R}, \eta \mapsto f(\eta) = \eta = \mu. \tag{30}$$

gegeben ist. Die Parameter des Allgemeinen Linearen Modells sind die Komponenten des Vektors $\beta \in \mathbb{R}^m$ des linearen Prädiktors $\eta = x^T \beta$ und der Parameter $\sigma^2 > 0$.

Definition (Logistische Regression als Generalisiertes Lineares Modell)

Das Modell der Logistischen Regression (LR) ist das Generalisierte Lineare Modell, bei dem

1. die Labelvariable eine Bernoulli-Zufallsvariable

$$y \sim B(\mu) \tag{31}$$

   ist und

2. die link function durch die *standard logit function*

$$g : [0,1] \to \mathbb{R}, \mu \mapsto g(\mu) := \ln\left(\frac{\mu}{1-\mu}\right) =: \eta \tag{32}$$

   gegeben ist.

Die Parameter des Logistischen Regressionsmodels sind die Komponenten des Vektors $\beta \in \mathbb{R}^m$ des linearen Prädiktors $\eta = x^T\beta$.

Theorem (Mean function der Logistischen Regression)

Die Inverse der link function des Modells der Logistischen Regression und somit seine mean function ist die *standard logistic function*

$$f : \mathbb{R} \to [0, 1], \eta \mapsto f(\eta) = \frac{1}{1 + \exp(-\eta)}. \tag{33}$$

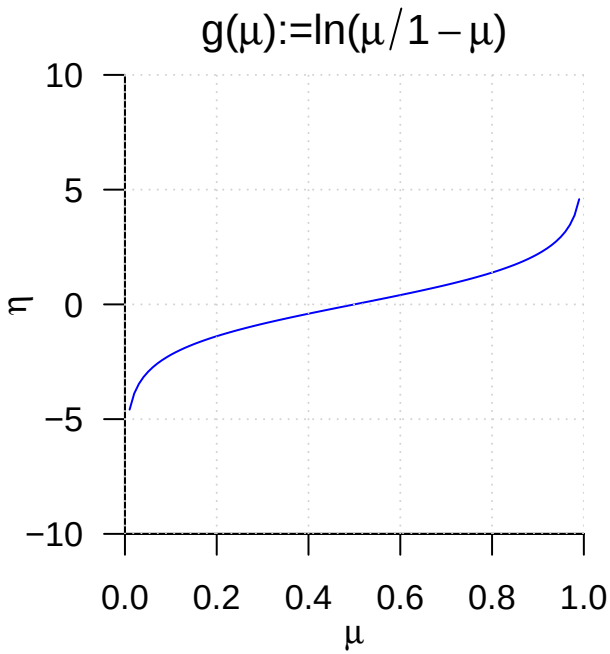
Beweis

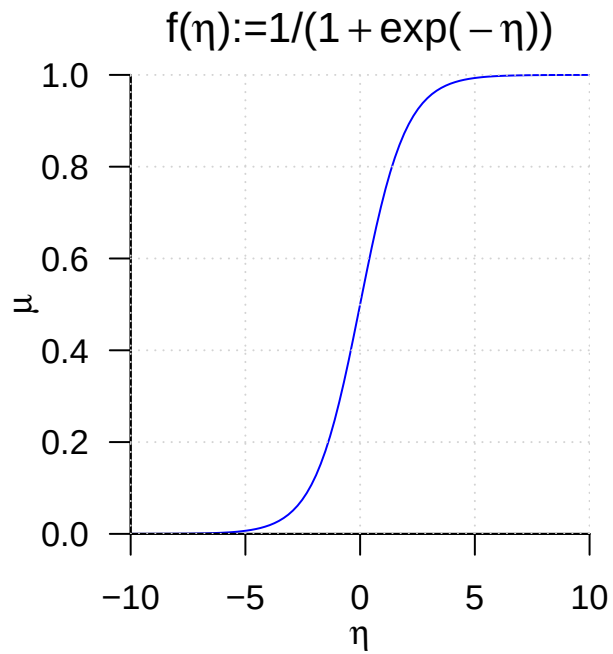
Umformen der logit function ergibt

$$\begin{aligned}
\eta &= \ln(\mu/(1-\mu)) \\
\Leftrightarrow -\eta &= -\ln(\mu/(1-\mu)) \\
\Leftrightarrow -\eta &= \ln((1-\mu)/\mu) \\
\Leftrightarrow \exp(-\eta) &= (1-\mu)/\mu \\
\Leftrightarrow \mu \exp(-\eta) &= 1-\mu \\
\Leftrightarrow \exp(-\eta) &= \mu^{-1} - 1 \\
\mu &= 1/(\exp(-\eta) + 1)
\end{aligned}$$

□

Link und Mean Funktionen





Definition (Modell der Logistischen Regression)

y sei eine Zufallsvariable mit Ergebnisraum $\{0, 1\}$. Dann ist das *Modell der Logistischen Regressionsmodell* definiert als die WMF

$$p(y) = B\left(y; \frac{1}{1 + \exp(-x^T \beta)}\right), \tag{34}$$

wobei $x \in \mathbb{R}^{m+1}$ einen erweiterten Featurevektor und $\beta \in \mathbb{R}^{m+1}$ den *Parametervektor* bezeichnen

Bemerkung

- Aus generativer Sicht wird ein Trainingsdatensatz

$$\left\{\left(x^{(i)}, y^{(i)}\right)\right\}_{i=1}^{n} \text{ mit } x^{(i)} \in \mathbb{R}^{m+1} \text{ und } y^{(i)} \in \{0, 1\} \tag{35}$$

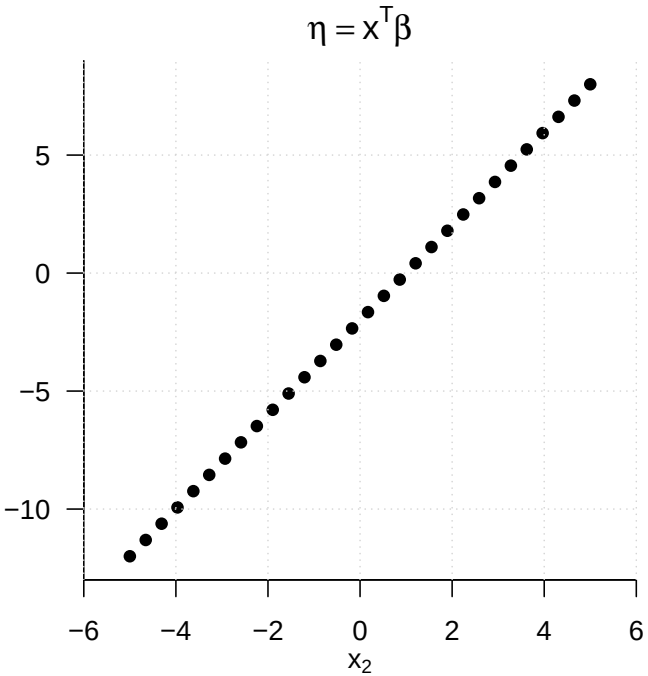
  eines LR Modells wie folgt erzeugt:

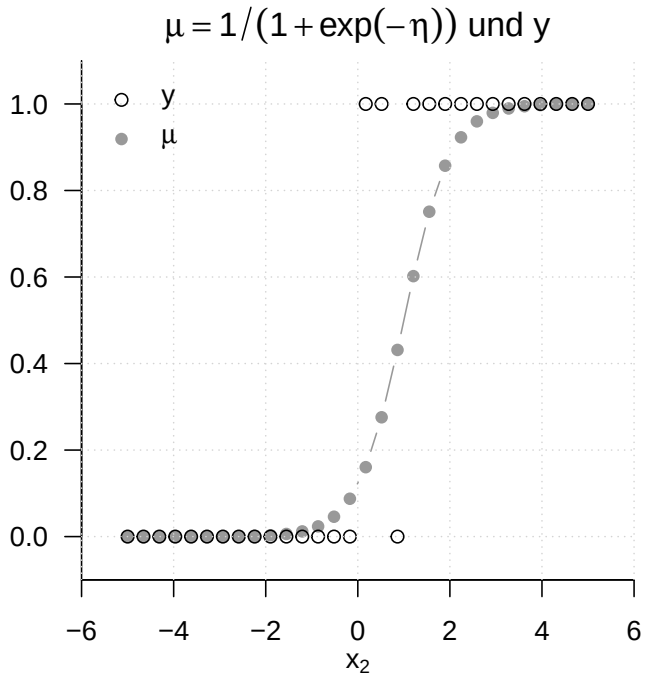
(1) Definition von $x^{(i)}$,
(2) Ziehen von $y^{(i)}$ aus $p(y) = B(y; \mu)$ mit Erwartungswertparameter $\mu = \frac{1}{1+\exp(-x^{(i)T}\beta)}$.

Datengeneration bei einfacher Logistischer Regression ($m = 1$)

```
# Modellparameter
m    = 1                                # Featurevektoredimensionalität
n    = 30                               # Anzahl Datenpunkte
x    = matrix(c(rep(1,n),               # Definition des erweiterten Featurevektors
                seq(-5,5, len = n)),
              nrow  = 2,
              byrow = TRUE)
beta = matrix(c(-2,2), nrow = 2)        # wahrer, aber unbekannter, Parametervektor
eta  = t(x) %*% beta                    # wahrer, aber unbekannter linearer Prädiktor
mu   = 1/(1+exp(-eta))                  # wahrer, aber unbekannter, Bernoulliparametervektor

# Dategeneration
set.seed(2)                             # Zufallsgeneratorzustand
y    = rep(NaN,2)                       # Datenarray
for(i in 1:n){
  y[i] = rbinom(1,1,mu[i])              # Bernoullivariablenrealisierung
}
```

Datengeneration bei einfacher Logistischer Regression ($m = 1, \beta = (-2, 2)^T, n = 30$)





Definition (Klassifikationsregel der Logistischen Regression)

$p(y)$ sei die WMF eines Logistischen Regressionsmodells. Dann ist die *Klassifikationregel* definiert als

$$\delta : \mathbb{R}^m \to \{0, 1\}, x \mapsto \delta(x) := \begin{cases} 0 & \text{für } p(y = 0) \geq p(y = 1) \\ 1 & \text{für } p(y = 0) < p(y = 1) \end{cases} \tag{36}$$

Bemerkung

- Es gilt

$$\delta(x) = 1 \Leftrightarrow p(y = 1) > p(y = 0) \Leftrightarrow p(y = 1) > 0.5. \tag{37}$$

Theorem (Lineare Diskriminanzfunktion der Logistischen Regression)

$p(y)$ sei die WMF eines Logistischen Regressionsmodells und $\beta \in \mathbb{R}^{m+1}$ sei der Parametervektor. Dann kann die Klassifikationsregel δ der Logistischen Regression als eine lineare Diskriminanzfunktion der Form

$$h : \mathbb{R}^m \to \{0, 1\}, x \mapsto h(x) := g(f(x)), \tag{38}$$

mit

$$f : \mathbb{R}^m \to \mathbb{R}, x \mapsto f(x) := w^T x + w_0, \tag{39}$$

und

$$g : \mathbb{R} \to \{0, 1\}, f(x) \mapsto g(f(x)) := \begin{cases} 0, & f(x) \geq 0 \\ 1, & f(x) < 0 \end{cases} \tag{40}$$

geschrieben werden, d.h. es gilt $\delta(x) = h(x)$ für alle $x \in \mathbb{R}^m$. Inbesondere gilt dabei

$$w_0 = \beta_1 \text{ und } w = (\beta_2, \ldots, \beta_{m+1})^T \tag{41}$$

Bemerkung

- CAVE: f bezeichnet hier eine linear-affine Funktion, nicht die standard logistic function.
- Ein Beweis ergibt sich in Analogie zum Fall der Linearen Diskriminanzanalyse

Theorem (Log Likelihood Funktion der Logistischen Regression)

$\{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$ sei ein Trainingsdatensatz aus erweiterten Featurevektoren und assoziierten Labelvariablenrealisierungen und f sei die standard logistic function. Dann hat die Log Likelihood Funktion der Logistischen Regression die Form

$$\ell : \mathbb{R}^m \to \mathbb{R}, \beta \mapsto \ell(\beta) := \sum_{i=1}^{n} y^{(i)} \ln(f({x^{(i)}}^T \beta)) + (1 - y^{(i)}) \ln(1 - f({x^{(i)}}^T \beta)).$$

Beweis

Wir halten zunächst fest, dass für u.i.v. Labelvariablen gilt, dass

$$\ell(\beta) := \ln p(y^{(1)}, \ldots, y^{(n)}) = \ln \prod_{i=1}^{n} p(y^{(i)}) = \sum_{i=1}^{n} \ln p(y^{(i)})$$

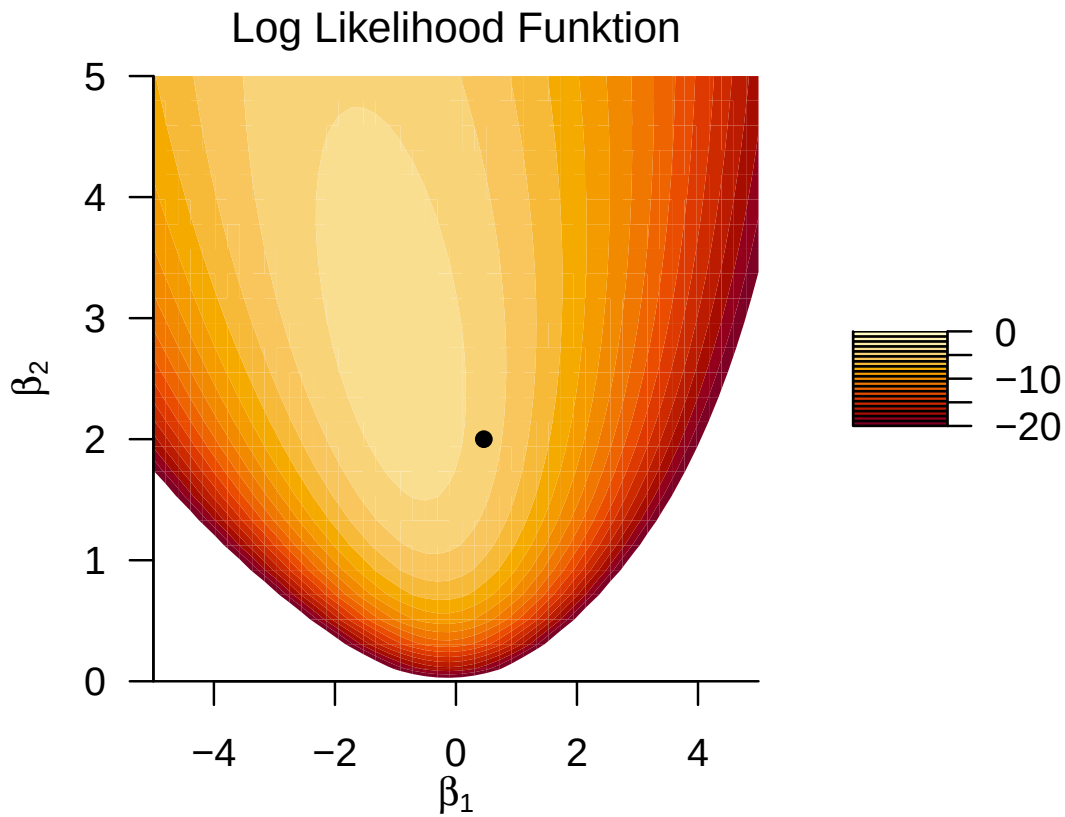
Mit der WMF der Bernoulliverteilung folgt dann

$$\begin{aligned} \ell(\beta) &= \sum_{i=1}^{n} \ln(f({x^{(i)}}^T \beta)^{y^{(i)}} (1 - f({x^{(i)}}^T \beta))^{1-y^{(i)}}) \\ &= \sum_{i=1}^{n} y^{(i)} \ln(f({x^{(i)}}^T \beta)) + (1 - y^{(i)}) \ln(1 - f({x^{(i)}}^T \beta)) \end{aligned}$$

Implementation der Log Likelihood Funktion

```
# Funktionsdefinitionen
# ------------------------------------------------------------------------------------------
# Standard Logistic Function
f = function(eta){
    return(1/(1 + exp(-eta)))
}

# Log Likelihood Function
llh = function(x,y,beta){
    n   = ncol(x)
    ell = 0
    for(i in 1:n){
        ell = ell + y[i]*log(f(t(x[,i]) %*% beta)) + (1-y[i])*log(1-f(t(x[,i]) %*% beta))
    }
    return(ell)
}

# Log Likelihood Funktion Auswertung
# ------------------------------------------------------------------------------------------
beta_min  = -5                                                    # beta Minimum
beta_max  =  5                                                    # beta Maximum
beta_res  = 5e1                                                   # beta Auflösung
beta_1    = seq(beta_min, beta_max, length.out = beta_res)        # beta_1 Raum
beta_2    = seq(beta_min, beta_max, length.out = beta_res)        # beta_2 Raum
ell       = matrix(rep(NaN, beta_res*beta_res), nrow = beta_res) # Log Likelihood Funktion Array
for(i in 1:beta_res){
    for(j in 1:beta_res){
        beta12   = matrix(c(beta_1[i], beta_2[j]), nrow = 2)
        ell[i,j] = llh(x,y,beta12)
    }
}
```

Visualisierung der Log Likelihood Funktion



Theorem (Gradientverfahren der Logistischen Regression)

$p(y)$ sei das Modell einer Logistischen Regression und $\{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$ sei ein entsprechender Trainingsdatensatz. Dann kann eine Maximum Likelihood Schätzer $\hat{\beta}$ für den Parametervektor β des LRM durch folgendes Gradientenverfahren gewonnen werden:

(0) Wähle $\beta^0 \in \mathbb{R}, \alpha > 0, \delta > 0$

(1) Für $k = 0, 1, 2, \ldots$ bis zur Konvergenz setze

$$\beta^{(k+1)} := \beta^{(k)} + \alpha \nabla \ell(\beta^{(k)}). \tag{42}$$

wobei $\nabla \ell \left(\beta^k\right)$ den Gradienten der Log Likelihood Funktion der Logistischen Regression bezeichnet und die Form

$$\nabla \ell(\beta) = \begin{pmatrix} \frac{\partial}{\partial \beta_1} \ell(\beta) \\ \frac{\partial}{\partial \beta_2} \ell(\beta) \\ \vdots \\ \frac{\partial}{\partial \beta_m} \ell(\beta) \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} (y^{(i)} - f({x^{(i)}}^T \beta)) x_1^{(i)} \\ \sum_{i=1}^{n} (y^{(i)} - f({x^{(i)}}^T \beta)) x_2^{(i)} \\ \vdots \\ \sum_{i=1}^{n} (y^{(i)} - f({x^{(i)}}^T \beta)) x_m^{(i)} \end{pmatrix} \tag{43}$$

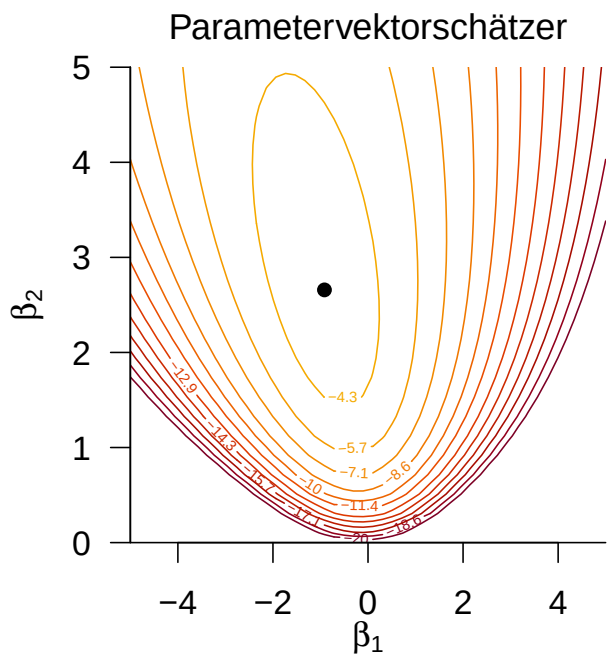
Bemerkungen

- Für einen Beweis verweisen wir auf den Appendix.

Bemerkungen (fortgeführt)

- Das reine Gradientenverfahren zum Lernen der Parameter eines LR Modells ist recht instabil.
- Iteratively Weighted Least Squares Verfahren werden zur ML Schätzung in GLMs bevorzugt (Green 1984).
- IWLS Verfahren nutzen Gradienten und Hesse-Matrix ähnlich wie Gauss-Newton Verfahren.
- R implementiert in der `glm()` ein IWLS Verfahren.

```
lr       = glm(y ~ x[2,], family = 'binomial')        # generalized linear model fit
beta_hat = lr$coefficients                            # Parametervektorschätzer
```



Prädiktion des Studienerfolgs

Nach Rudolf and Buse (2020) Kapitel 4

Datensatz zum Verhältnis psychologischer Diagnostik und Studienerfolg

$n = 30$ Studierende naturwissenschaftlicher Studiengänge

Featurevektor $x \in \mathbb{R}^4$

- x_1 Intelligenztestscore
- x_2 Mathematiktestscore
- x_3 Gewissenhaftigkeitscore
- x_4 Verträglichkeitscore

Label $y \in \{0, 1\}$

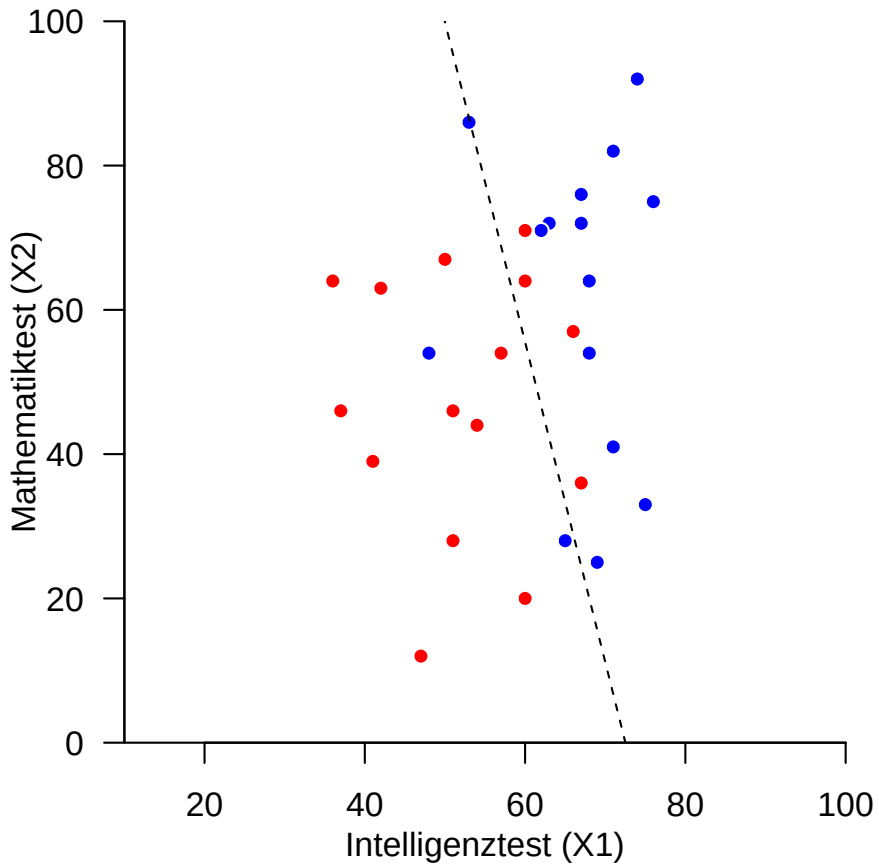
- 0: ungenügend (Studienabbruch aufgrund nicht bestandener Prüfungen)
- 1: gut (Abschlussnote besser als 2.5)

Parameterlernen, Prädiktion und lineare Diskriminanzfunktion

```
library(matlib)                                                          # Matrixtools
D            = read.table(file.path(getwd(), "7_Daten", "studienerfolg.csv")) # Datensatz
x            = as.matrix(D[1:2,])                                        # Featureselektion
y            = as.matrix(D[5,])                                          # Label
D            = rbind(x,y)                                                # Featureselected Datensatz
n            = ncol(x)                                                   # Datensatzgröße
m            = nrow(x)                                                   # Featurevektordimensionalität
lr           = glm(t(y) ~ t(x), family = 'binomial')                       # generalized linear model fit
beta_hat     = as.matrix(lr$coefficients, nrow = m + 1)                  # Parameterschätzung
w_0_hat      = beta_hat[1]                                               # \hat{w}
w_hat        = beta_hat[2:(m+1)]                                         # \hat{w_0}
x_1          = seq(min(x[1,]), max(x[1,]), len = 1e2)                    # x_1
x_2_hat      = -(w_hat[1]/w_hat[2])*x_1 - (w_0_hat/w_hat[2])             # \hat{h}
x_tilde      = rbind(1, x)
p_y          = 1/(1+exp(-t(x_tilde) %*% beta_hat))
delta        = as.numeric(p_y >= 0.5)
cat("Accuracy: ", mean(delta == y))
```

```
> Accuracy:  0.767
```

Parameterlernen und lineare Diskriminanzfunktion



Logistische Regression | Leave-One-Out Cross-Validation, $m = 2$

```
# Datensatz
D                = read.table(file.path(getwd(), "7_Daten", "studienerfolg.csv"))
x                = as.matrix(D[1:2,])
y                = as.matrix(D[5,])
D                = rbind(x,y)
K                = ncol(D)
p_y              = matrix(rep(NaN, ncol(y)), nrow = 1)
delta            = matrix(rep(NaN, ncol(y)), nrow = 1)

# K-fache Leave-One-Out Cross-Validation
for(k in 1:K){

    # Datensatzpartition
    x_train      = as.matrix(x[,-k])
    y_train      = as.matrix(y[,-k])
    x_test       = as.matrix(x[, k])
    y_test       = as.matrix(y[, k])

    # Trainingsdatensatz-basiertes Parameterlernen
    n            = ncol(x_train)
    m            = nrow(x_train)
    lr           = glm(y_train ~ t(x_train), family = 'binomial')
    beta_hat     = as.matrix(lr$coefficients, nrow = m + 1)

    # Prädiktion
    x_test_tilde = rbind(1, x_test)
    p_y[k]       = 1/(1+exp(-t(x_test_tilde) %*% beta_hat))
    delta[k]     = as.numeric(p_y[k] >= 0.5)
}
cat("Accuracy: ", mean(delta == y))
```

```
> Accuracy:  0.733
```

Logistische Regression | Leave-One-Out Cross-Validation, $m = 2$

k	x_1	x_2	p(y = 1)	delta(x)	y
1	54	44	0.2	0	0
2	60	20	0.2	0	0
3	67	36	0.7	1	0
4	41	39	0.0	0	0
5	66	57	0.8	1	0
6	51	28	0.1	0	0
7	51	46	0.1	0	0
8	37	46	0.0	0	0
9	57	54	0.4	0	0
10	47	12	0.0	0	0
11	50	67	0.2	0	0
12	42	63	0.0	0	0
13	60	64	0.6	1	0
14	36	64	0.0	0	0
15	60	71	0.7	1	0
16	71	41	0.8	1	1
17	65	28	0.3	0	1
18	67	76	0.9	1	1
19	68	54	0.8	1	1
20	75	33	0.8	1	1
21	71	82	1.0	1	1
22	68	64	0.9	1	1
23	63	72	0.8	1	1
24	48	54	0.0	0	1
25	53	86	0.3	0	1
26	62	71	0.7	1	1
27	69	25	0.5	0	1
28	67	72	0.9	1	1
29	74	92	1.0	1	1
30	76	75	1.0	1	1

Prediction Accuracy = 0.73.

Logistische Regression | Leave-One-Out Cross-Validation, $m = 3$

k	x_1	x_2	x_3	p(y = 1)	delta(x)	y
1	54	44	31	0.2	0	0
2	60	20	33	0.2	0	0
3	67	36	26	0.7	1	0
4	41	39	31	0.0	0	0
5	66	57	34	0.9	1	0
6	51	28	42	0.1	0	0
7	51	46	34	0.1	0	0
8	37	46	36	0.0	0	0
9	57	54	28	0.3	0	0
10	47	12	34	0.0	0	0
11	50	67	27	0.2	0	0
12	42	63	26	0.0	0	0
13	60	64	29	0.6	1	0
14	36	64	36	0.0	0	0
15	60	71	42	1.0	1	0
16	71	41	37	0.9	1	1
17	65	28	33	0.3	0	1
18	67	76	38	0.9	1	1
19	68	54	34	0.8	1	1
20	75	33	25	0.7	1	1
21	71	82	32	1.0	1	1
22	68	64	34	0.9	1	1
23	63	72	32	0.8	1	1
24	48	54	41	0.0	0	1
25	53	86	27	0.1	0	1
26	62	71	31	0.7	1	1
27	69	25	32	0.5	0	1
28	67	72	31	0.9	1	1
29	74	92	35	1.0	1	1
30	76	75	37	1.0	1	1

Prediction Accuracy = 0.73.

Logistische Regression | Leave-One-Out Cross-Validation, $m = 4$

k	x_1	x_2	x_3	x_4	p(y = 1)	delta(x)	y
1	54	44	31	60	0.0	0	0
2	60	20	33	31	0.5	0	0
3	67	36	26	54	0.4	0	0
4	41	39	31	26	0.0	0	0
5	66	57	34	56	0.6	1	0
6	51	28	42	23	0.3	0	0
7	51	46	34	40	0.1	0	0
8	37	46	36	31	0.0	0	0
9	57	54	28	49	0.2	0	0
10	47	12	34	41	0.0	0	0
11	50	67	27	53	0.0	0	0
12	42	63	26	36	0.0	0	0
13	60	64	29	40	0.9	1	0
14	36	64	36	21	0.1	0	0
15	60	71	42	40	1.0	1	0
16	71	41	37	30	1.0	1	1
17	65	28	33	38	0.4	0	1
18	67	76	38	28	1.0	1	1
19	68	54	34	53	0.5	1	1
20	75	33	25	54	0.6	1	1
21	71	82	32	49	1.0	1	1
22	68	64	34	33	1.0	1	1
23	63	72	32	36	0.9	1	1
24	48	54	41	34	0.0	0	1
25	53	86	27	35	0.4	0	1
26	62	71	31	53	0.4	0	1
27	69	25	32	25	0.9	1	1
28	67	72	31	28	1.0	1	1
29	74	92	35	50	1.0	1	1
30	76	75	37	12	1.0	1	1

Prediction Accuracy = 0.77.

Vorbemerkungen

Lineare Diskriminanzanalyse

Logistische Regression

Selbstkontrollfragen

Appendix

Selbstkontrollfragen

1. Definieren Sie den Begriff des Binären Klassifikationsdatensatzes.
2. Definieren Sie die Bernoulli Verteilung.
3. Definieren Sie das Modell der Linearen Diskriminanzanalyse.
4. Erläutern Sie die Erzeugung von Daten unter dem Modell der Linearen Diskriminanzanalyse.
5. Erläutern Sie den Begriff der Inferenz im Modell der Linearen Diskriminanzanalyse.
6. Definieren Sie die Klassifikationsregel der Linearen Diskriminanzanalyse.
7. Wie werden die Parameter eines Linearen Diskriminanzanalysemodells gelernt?
8. Erläutern Sie den Ablauf einer Leave-One-Out Cross-Validation mithilfe der Linearen Diskriminanzanalyse.
9. Definieren Sie die standard logistic function.
10. Definieren Sie das Modell der Logistischen Regression.
11. Erläutern Sie die Erzeugung von Daten unter dem Modell der Logistischen Regression.
12. Warum gibt es im Modell der Logistischen Regression keine Inferenz?
13. Definieren Sie die Klassifikationsregel der Logistischen Regression.
14. Wie werden die Parameter eines Logistischen Regressionsmodells gelernt?
15. Erläutern Sie den Ablauf einer Leave-One-Out Cross-Validation mithilfe der Logistischen Regression.

Beweis des LDA Inferenz Theorems

Wir halten zunächst fest, dass

$$\begin{aligned} p(y=1|x) &= \frac{p(x,y=1)}{p(x,y=0)+p(x,y=1)} \\ &= \frac{\frac{p(x,y=1)}{p(x,y=1)}}{\frac{p(x,y=0)}{p(x,y=1)}+\frac{p(x,y=1)}{p(x,y=1)}} \\ &= \frac{1}{1+\frac{p(x,y=0)}{p(x,y=1)}} \\ &= \frac{1}{1+\exp\left(\ln\left(\frac{p(x,y=0)}{p(x,y=1)}\right)\right)} \\ &= \frac{1}{1+\exp\left(-\ln\left(\frac{p(x,y=1)}{p(x,y=0)}\right)\right)} \end{aligned} \tag{44}$$

Mit der Definition des LDA Modells gilt dann

$$p(x,y=1) = p(x|y=1)p(y=1) = N(x;\mu_1,\Sigma)\mu \tag{45}$$

und

$$p(x,y=0) = p(x|y=0)p(y=0) = N(x;\mu_0,\Sigma)(1-\mu) \tag{46}$$

Beweis des LDA Inferenz Theorems (fortgeführt)

Wir erhalten also

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

Beweis des LDA Diskriminanzfunktion Theorems

Wir zeigen die Äquivalenz von $\delta(x) = 0$ und $f(x) \geq 0$ womit der Rest des Theorems sofort folgt. Es ergibt sich

$$
\begin{aligned}
\delta(x) &= 0 \\
\Leftrightarrow p(y=0|x) &\geq p(y=1|x) \\
\Leftrightarrow \frac{p(y=0|x)}{p(y=1|x)} &\geq 1 \\
\Leftrightarrow \ln\left(\frac{p(y=0|x)}{p(y=1|x)}\right) &\geq \ln 1 \\
\Leftrightarrow \ln\left(\frac{1 - \frac{1}{1+\exp(-\tilde{x}^T\beta)}}{\frac{1}{1+\exp(-\tilde{x}^T\beta)}}\right) &\geq 0 \\
\Leftrightarrow \ln\left(\frac{\frac{1+\exp(-\tilde{x}^T\beta)}{1+\exp(-\tilde{x}^T\beta)} - \frac{1}{1+\exp(-\tilde{x}^T\beta)}}{\frac{1}{1+\exp(-\tilde{x}^T\beta)}}\right) &\geq 0 \\
\Leftrightarrow \ln\left(\frac{\frac{\exp(-\tilde{x}^T\beta)}{1+\exp(-\tilde{x}^T\beta)}}{\frac{1}{1+\exp(-\tilde{x}^T\beta)}}\right) &\geq 0 \\
\Leftrightarrow \ln\left(\exp(-\tilde{x}^T\beta)\right) &\geq 0
\end{aligned}
$$

Beweis des LDA Diskriminanzfunktion Theorems (fortgeführt)

Es ergibt sich also

$$
\begin{aligned}
\delta(x) = 0 \\
\Leftrightarrow -\tilde{x}^T \beta \geq 0 \\
\Leftrightarrow -\begin{pmatrix} 1 & x^T \end{pmatrix} \begin{pmatrix} \frac{1}{2}\mu_0 \Sigma^{-1} \mu_0 - \ln(1-\mu) - \frac{1}{2}\mu_1 \Sigma \mu_1 + \ln \mu \\ -\Sigma^{-1}(\mu_0 - \mu_1) \end{pmatrix} \geq 0 \\
\Leftrightarrow x^T \Sigma^{-1}(\mu_0 - \mu_1) + \frac{1}{2}\mu_1 \Sigma \mu_1 + \ln(1-\mu) - \frac{1}{2}\mu_0 \Sigma^{-1} \mu_0 - \ln \mu \geq 0 \\
\Leftrightarrow \left(x^T \Sigma^{-1}(\mu_0 - \mu_1)\right)^T + \frac{1}{2}\mu_1 \Sigma \mu_1 + \ln(1-\mu) - \frac{1}{2}\mu_0 \Sigma^{-1} \mu_0 - \ln \mu \geq 0 \\
\Leftrightarrow (\mu_0 - \mu_1)^T \Sigma^{-1} x + \frac{1}{2}(\mu_1 - \mu_0)^T \Sigma^{-1}(\mu_1 - \mu_0) + \ln\left(\frac{1-\mu}{\mu}\right) \geq 0 \\
\Leftrightarrow: w^T x + w_0 \geq 0
\end{aligned}
$$

Beweis des LDA Maximum Likelihood Schätzer Theorems

(1) Formulierung der Log Likelihood Funktion

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

Appendix

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

Beweis des LDA Maximum Likelihood Schätzer Theorems (fortgeführt)

(2) Gradient der Log Likelihood Funktion

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

$$\nabla \ell(\mu, \mu_0, \mu_1, \Sigma) = \begin{pmatrix} \frac{\partial}{\partial \mu} \ell(\mu, \mu_0, \mu_1, \Sigma) \\ \frac{\partial}{\partial \mu_0} \ell(\mu, \mu_0, \mu_1, \Sigma) \\ \frac{\partial}{\partial \mu_1} \ell(\mu, \mu_0, \mu_1, \Sigma) \\ \frac{\partial}{\partial \Sigma} \ell(\mu, \mu_0, \mu_1, \Sigma) \end{pmatrix}$$

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

Beweis des LDA Maximum Likelihood Schätzer Theorems (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 des LDA Maximum Likelihood Schätzer Theorems (fortgeführt)

Für die partielle Ableitung hinsichtlich Σ ergibt sich

$$
\begin{aligned}
&\frac{\partial}{\partial \Sigma} \ell(\mu, \mu_1, \mu_0, \Sigma) \\
&= \frac{\partial}{\partial \Sigma} \sum_{i=1}^{n} \left(1 - y^{(i)}\right) \left(-\frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma| - \frac{1}{2} \left(x^{(i)} - \mu_0\right)^T \Sigma^{-1} \left(x^{(i)} - \mu_0\right)\right) \\
&\quad + \frac{\partial}{\partial \Sigma} \sum_{i=1}^{n} y^{(i)} \left(-\frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma| - \frac{1}{2} \left(x^{(i)} - \mu_1\right)^T \Sigma^{-1} \left(x^{(i)} - \mu_1\right)\right) \\
&= \sum_{i=1}^{n} \left(1 - y^{(i)}\right) \left(-\frac{1}{2} \frac{\partial}{\partial \Sigma} \ln |\Sigma| - \frac{1}{2} \frac{\partial}{\partial \Sigma} \left(x^{(i)} - \mu_0\right)^T \Sigma^{-1} \left(x^{(i)} - \mu_0\right)\right) \\
&\quad + \sum_{i=1}^{n} y^{(i)} \left(-\frac{1}{2} \frac{\partial}{\partial \Sigma} \ln |\Sigma| - \frac{1}{2} \frac{\partial}{\partial \Sigma} \left(x^{(i)} - \mu_1\right)^T \Sigma^{-1} \left(x^{(i)} - \mu_1\right)\right)
\end{aligned}
\tag{47}
$$

Beweis des LDA Maximum Likelihood Schätzer Theorems (fortgeführt)

... und damit

$$\begin{aligned}\frac{\partial}{\partial \Sigma}\ell(\mu, \mu_1, \mu_0, \Sigma) = & \sum_{i=1}^{n}\left(1 - y^{(i)}\right)\left(-\frac{1}{2}\Sigma - \frac{1}{2}\left(x^{(i)} - \mu_0\right)\left(x^{(i)} - \mu_0\right)^T\right) \\ & + \sum_{i=1}^{n} y^{(i)}\left(-\frac{1}{2}\Sigma - \frac{1}{2}\left(x^{(i)} - \mu_1\right)\left(x^{(i)} - \mu_1\right)^T\right) \\ = & \frac{n}{2}\Sigma - \frac{1}{2}\sum_{i=1}^{n}\left(x^{(i)} - \mu_{1\{y^{(i)}=1\}}\right)\left(x^{(i)} - \mu_{1\{y^{(i)}=1\}}\right)^T.\end{aligned} \tag{48}$$

Beweis des LDA Maximum Likelihood Schätzer Theorems (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 LDA Modells.

Beweis des LDA Maximum Likelihood Schätzer Theorems (fortgeführt)

Nullsetzen der ersten Gradientenkomponente ergibt

$$\frac{1}{\hat{\mu}} \sum_{i=1}^{n} 1_{\{y^{(i)}=1\}} - \frac{1}{1-\hat{\mu}} \sum_{i=1}^{n} 1_{\{y^{(i)}=0\}} = 0$$

$$\Leftrightarrow \frac{1}{\hat{\mu}} \sum_{i=1}^{n} y^{(i)} - \frac{1}{1-\hat{\mu}} \sum_{i=1}^{n} \left(1 - y^{(i)}\right) = 0$$

$$\Leftrightarrow \frac{1-\hat{\mu}}{\hat{\mu}} \sum_{i=1}^{n} y^{(i)} - \sum_{i=1}^{n} \left(1 - y^{(i)}\right) = 0$$

$$\Leftrightarrow \frac{1-\hat{\mu}}{\hat{\mu}} \sum_{i=1}^{n} y^{(i)} - n + \sum_{i=1}^{n} y^{(i)} = 0$$

$$\Leftrightarrow (1-\hat{\mu}) \sum_{i=1}^{n} y^{(i)} - \hat{\mu} n + \hat{\mu} \sum_{i=1}^{n} y^{(i)} = 0$$

Beweis des LDA Maximum Likelihood Schätzer Theorems (fortgeführt)

... und weiter

$$\Leftrightarrow (1-\hat{\mu})\sum_{i=1}^{n} y^{(i)} - \hat{\mu}n + \hat{\mu}\sum_{i=1}^{n} y^{(i)} = 0$$

$$\Leftrightarrow (1-\hat{\mu}+\hat{\mu})\sum_{i=1}^{n} y^{(i)} = \hat{\mu}n$$

$$\Leftrightarrow \hat{\mu}n = \sum_{i=1}^{n} y^{(i)}$$

$$\Leftrightarrow \hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} 1_{\{y^{(i)}=1\}}.$$

Beweis des LDA Maximum Likelihood Schätzer Theorems (fortgeführt)

Nullsetzen der zweiten Gradientenkomponente ergibt

$$\sum_{i=1}^{n} \left(1 - y^{(i)}\right) \left(\left(x^{(i)} - \hat{\mu}_0\right)^T \Sigma^{-1}\right) = 0$$

$$\Leftrightarrow \sum_{i=1}^{n} 1_{\{y^{(i)}=0\}} \left(x^{(i)} - \hat{\mu}_0\right)^T = 0$$

$$\Leftrightarrow \sum_{i=1}^{n} 1_{\{y^{(i)}\}} x^{(i)} - \sum_{i=1}^{n} 1_{\{y^{(i)}=0\}} \hat{\mu}_0 = 0$$

$$\Leftrightarrow \sum_{i=1}^{n} 1_{\{y^{(i)}=0\}} \hat{\mu}_0 = \sum_{i=1}^{n} 1_{\{y^{(i)}\}} x^{(i)}$$

$$\Leftrightarrow \hat{\mu}_0 = \frac{1}{\sum_{i=1}^{n} 1_{\{y^{(i)}=0\}}} \sum_{i=1}^{n} 1_{\{y^{(i)}\}} x^{(i)}.$$

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

Beweis des LDA Maximum Likelihood Schätzer Theorems (fortgeführt)

Nullsetzen der vierten Gradientenkomponente ergibt dann schließlich

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

Beweis des LR Gradientenverfahrens

Um die $j$te partielle Ableitung der Log Likelihood Funktion zu bestimmen, halten wir zunächst fest, dass sich die Ableitung der logistic function f hinsichtlich η zu

$$\frac{d}{d\eta} f : \mathbb{R} \to \mathbb{R}, \eta \mapsto \frac{d}{d\eta} f(\eta) = f(\eta)(1 - f(\eta)) \tag{49}$$

ergibt. Dies kann wie folgt eingesehen werden:

$$\begin{aligned}
\frac{d}{d\eta} f(\eta) &= \frac{d}{d\eta} \left(1 + \exp(-\eta)\right)^{-1} \\
&= -\left(1 + \exp(-\eta)\right)^{-2} \cdot \exp(-\eta) \cdot (-1) \\
&= \frac{\exp(-\eta)}{(1 + \exp(-\eta))^2} \\
&= \frac{1 + \exp(-\eta) - 1}{(1 + \exp(-\eta))^2} \\
&= \frac{1 + \exp(-\eta)}{(1 + \exp(-\eta))^2} - \frac{1}{(1 + \exp(-\eta))^2} \\
&= \frac{1}{1 + \exp(-\eta)} - \frac{1}{(1 + \exp(-\eta))^2} \\
&= \frac{1}{1 + \exp(-\eta)} \left(1 - \frac{1}{1 + \exp(-\eta)}\right) \\
&= f(\eta)(1 - f(\eta))
\end{aligned}$$

Beweis des LR Gradientenverfahrens (fortgeführt)

Damit ergibt sich dann für $\frac{\partial}{\partial \beta_j} \ell, j = 1, \ldots, m$:

$$
\begin{aligned}
&\frac{\partial}{\partial \beta_j} \ell(\beta) \\
&= \frac{\partial}{\partial \beta_j} \left(\sum_{i=1}^{n} y^{(i)} \ln \left(f \left({x^{(i)}}^T \beta \right) \right) + \left(1 - y^{(i)} \right) \ln \left(1 - f \left({x^{(i)}}^T \beta \right) \right) \right) \\
&= \sum_{i=1}^{n} y^{(i)} \frac{\partial}{\partial \beta_j} \left(\ln \left(f \left({x^{(i)}}^T \beta \right) \right) \right) + \left(1 - y^{(i)} \right) \frac{\partial}{\partial \beta_j} \left(\ln \left(1 - f \left({x^{(i)}}^T \beta \right) \right) \right) \\
&= \sum_{i=1}^{n} y^{(i)} \frac{1}{f \left({x^{(i)}}^T \beta \right)} \left(\frac{\partial}{\partial \beta_j} \left(f \left({x^{(i)}}^T \beta \right) \right) \right) + \left(1 - y^{(i)} \right) \frac{1}{1 - f \left({x^{(i)}}^T \beta \right)} \frac{\partial}{\partial \beta_j} \left(1 - f \left({x^{(i)}}^T \beta \right) \right) \\
&= \sum_{i=1}^{n} \left(y^{(i)} \frac{1}{f \left({x^{(i)}}^T \beta \right)} - \left(1 - y^{(i)} \right) \frac{1}{1 - f \left({x^{(i)}}^T \beta \right)} \right) \frac{\partial}{\partial \beta_j} \left(f \left({x^{(i)}}^T \beta \right) \right)
\end{aligned}
$$

Appendix

Beweis des LR Gradientenverfahrens (fortgeführt)

$$
\begin{aligned}
\frac{\partial}{\partial \beta_j} \ell(\beta) &= \sum_{i=1}^{n} \left(y^{(i)} \frac{1}{f\left(x^{(i)^T} \beta\right)} - \left(1 - y^{(i)}\right) \frac{1}{1 - f\left(x^{(i)^T} \beta\right)} \right) \\
&\quad \times f\left(x^{(i)^T} \beta\right) \left(1 - f\left(x^{(i)^T} \beta\right)\right) \frac{\partial}{\partial \beta_j} \left(x^{(i)^T} \beta\right) \\
&= \sum_{i=1}^{n} \left(y^{(i)} \frac{1}{f\left(x^{(i)^T} \beta\right)} - \left(1 - y^{(i)}\right) \frac{1}{1 - f\left(x^{(i)^T} \beta\right)} \right) f\left(x^{(i)^T} \beta\right) \left(1 - f\left(x^{(i)^T} \beta\right)\right) x_j^{(i)} \\
&= \sum_{i=1}^{n} \left(y^{(i)} \frac{f\left(x^{(i)^T} \beta\right) \left(1 - f\left(x^{(i)^T} \beta\right)\right)}{f\left(x^{(i)^T} \beta\right)} - \left(1 - y^{(i)}\right) \frac{f\left(x^{(i)^T} \beta\right) \left(1 - f\left(x^{(i)^T} \beta\right)\right)}{1 - f\left(x^{(i)^T} \beta\right)} \right) x_j^{(i)} \\
&= \sum_{i=1}^{n} \left(y^{(i)} \left(1 - f\left(x^{(i)^T} \beta\right)\right) - \left(1 - y^{(i)}\right) f\left(x^{(i)^T} \beta\right) \right) x_j^{(i)} \\
&= \sum_{i=1}^{n} \left(y^{(i)} - y^{(i)} f\left(x^{(i)^T} \beta\right) - f\left(x^{(i)^T} \beta\right) + y^{(i)} f\left(x^{(i)^T} \beta\right) \right) x_j^{(i)} \\
&= \sum_{i=1}^{n} \left(y^{(i)} - f\left(x^{(i)^T} \beta\right) \right) x_j^{(i)}.
\end{aligned}
$$

References

Dwyer, Dominic B., Peter Falkai, and Nikolaos Koutsouleris. 2018. "Machine Learning Approaches for Clinical Psychology and Psychiatry." *Annual Review of Clinical Psychology* 14 (1): 91–118. https://doi.org/10.1146/annurev-clinpsy-032816-045037.

Green, P. J. 1984. "Iteratively Reweighted Least Squares for Maximum Likelihood Estimation, and Some Robust and Resistant Alternatives." *Journal of the Royal Statistical Society: Series B (Methodological)* 46 (2): 149–70. https://doi.org/10.1111/j.2517-6161.1984.tb01288.x.

Rudolf, Matthias, and Johannes Buse. 2020. *Multivariate Verfahren*. Göttingen: Hogrefe.