12 Logistische Regression

12.1 Mathematische Grundlagen

Im *Modell der Logistischen Regression* (LR) folgt eine Labelvariable $y \in \{0, 1\}$ mit dem assoziierten Zufallsvektor $x \in \mathbb{R}^m$ einer *Bernoulli-Verteilung*

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

mit dem *Parametervektor* $\beta \in \mathbb{R}^{m+1}$ und dem *erweiterten Featurevektor*

$$\tilde{x} := \begin{pmatrix} 1 \\ x \end{pmatrix} \in \mathbb{R}^{m+1} \ . \tag{2}$$

Wie in der Vorlesung gezeigt wurde, muss für einen gegebenen Datensatz

$$\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\} \tag{3}$$

der *Maximum-Likelihood-Schätzer* des Parametervektors

$$\begin{aligned} \hat{\beta} &= \arg\max_{\beta} \ln \prod_{i=1}^{n} p(y_i) = \arg\max_{\beta} \sum_{i=1}^{n} \ln p(y_i) \\ &= \arg\max_{\beta} \sum_{i=1}^{n} \left[y_i \ln(f(\tilde{x}_i^{\mathrm{T}} \beta)) + (1 - y_i) \ln(1 - f(\tilde{x}_i^{\mathrm{T}} \beta)) \right] \ , \end{aligned} \tag{4}$$

wobei $f(x) = 1/\left(1 + \exp\left(-x\right)\right)$, mithilfe eines Gradientenverfahrens für die *Log-Likelihood-Funktion* des Modells geschätzt werden.

Wenn die Parameter eines LR-Modells einmal geschätzt sind, können diese zur *Inferenz*, d.h. zur Berechnung der bedingten Wahrscheinlichkeit $p(y = 1|x)$ für einen potentiell neuen Datenpunkt $x \in \mathbb{R}^m$ verwendet werden:

$$p(y = 1|x) = \frac{1}{1 + \exp(-\tilde{x}^T \beta)} \ . \tag{5}$$

Als *Klassifikationsregel* für die LR ergibt sich, dass der potentiell neue Datenpunkt x der Klasse 1 zugeschrieben wird, wenn $p(y = 1|x) > 0.5$, und der Klasse 0 zugeschrieben wird, wenn $p(y = 1|x) \leq 0.5$.

In der vorliegenden Übung wollen wir LR-Modellschätzung via *leave-one-out cross-validation* sowie die Prädiktion von Labels auf Grundlage der geschätzten Modellparameter für einen realen Datensatz nachvollziehen.

12.2 Analyse in R

Die Datei `FADE_SAME.csv` enthält den in der ersten Seminarsitzung vorgestellten Datensatz. Erklären Sie die Funktion des folgenden R-Codes:

```
# Daten einlesen
fname = 'FADE_SAME.csv'                          # Dateiname
D     = read.csv(fname)                          # Dataframe

# Datenmatrix extrahieren
rows  = startsWith(D$subject, 'subA')            # Personen aus Studie A
cols  = c('novelty.FADE','novelty.SAME',         # Definition Variablen
          'memory.FADE', 'memory.SAME')
X     = t(as.matrix(D[rows,cols]))               # Datenmatrix
y     = t(as.matrix(D$sex[rows]))                # gruppendefinierende Variable
y     = 0*(y == "male") + 1*(y == "female")      # Labelvariable
# y   = t(as.matrix(D$Abitur[rows]))             # gruppendefinierende Variable
# y   = 0*(y == "no") + 1*(y == "yes")           # Labelvariable
print(dim(X))                                    # Überprüfung Datenmatrix
print(dim(y))                                    # Überprüfung Labelvariable
print(sum(y))                                    # Anzahl positiver Fälle
```

12.3 Erste Programmieraufgabe

Führen Sie eine Klassifikationsanalyse auf Grundlage der Datenmatrix X und der Labelvariable y durch, indem Sie die LR-Modellparameter mit dem Prinzip der *leave-one-out cross-validation* schätzen und zur Vorhersage des jeweils ausgelassenen Datenpunkts verwenden. Gehen Sie dazu wie folgt vor:

- Kopieren Sie den R-Code aus Abschnitt 8.4 des Arbeitsblatts (8) *Prädiktive Modellierung*.
- Entfernen Sie im Abschnitt "Vorbereitung Datenanalyse" die Definition der Variable `L`.
- Berechnen Sie im Abschnitt "Training" innerhalb der Schleife `n_train` als die Anzahl der Spalten von `x_train`.
- Trainieren Sie mit `glm()` ein logistisches Regressionsmodell auf den Trainingsdaten (`x_train`, `y_train`) und extrahieren Sie dessen Parameter aus dem Feld `$coefficients` als eine Matrix mit $m+1$ Zeilen und 1 Spalte (d.h. einen $(m+1)$-dimensionalen Vektor).
- Erzeugen Sie im Abschnitt "Test" innerhalb der Schleife mittels `rbind` den erweiteren Featurevektor $\tilde{x}$ anhand der oben angegebenen Formel.
- Berechnen Sie die bedingte Wahrscheinlichkeit $p(y = 1|x)$ gemäß der oben angegebenen Formel. Speichern das Ergebnis als die Variable `p_y`.
- Modifizieren Sie die Klassifikationsregel derart, dass das prädizierte Label des i-ten Datenpunkts 0 ist, wenn $p(y = 1|x) \leq 0.5$, und 1 anderenfalls.

- Geben Sie nach der Schleife zur Überprüfung die Anzahl positiver Klassifikationen (`y_pred[,2]==1`) aus. Sie sollten folgende Ergebnisse erhalten:

```
[1] 235
```

12.4 Abbildung in R

Wir wollen nun die geschätzten Modellparameter $\hat{\beta}$ sowie den Bernoulli-Parameter μ als Funktion des linearen Prädiktors $\eta = \tilde{x}^{\mathrm{T}}\beta$ im Vergleich mit den tatsächlich beobachteten Labels y visualisieren. Erklären Sie dazu den folgenden R-Code und die Abbildung, die er erzeugt:

```
# logistische Regressionsparameter
lr            = glm(t(y) ~ t(X), family = 'binomial')       # IWLS-Parameterlernen
beta_hat      = as.matrix(lr$coefficients, nrow = m+1)      # Parameterschätzer
X_tilde       = rbind(rep(1,n), X)                  # erweiterte Featurematrix
eta_hat       = t(beta_hat) %*% X_tilde             # linearer Prädiktor (geschätzt)
eta           = seq(-5, +5, len = 21)               # linearer Prädiktor (w.a.u.)
mu            = 1/(1+exp(-eta))                     # Bernoulliparameter (w.a.u.)

# Abbildungsparameter
library(latex2exp)
par(
    family    = "sans",
    mfcol     = c(1,2),
    pty       = "m",
    bty       = "l",
    lwd       = 1,
    las       = 1,
    mgp       = c(2,1,0),
    xaxs      = "i",
    yaxs      = "i",
    font.main = 1,
    cex       = 1,
    cex.main  = 1)

# geschätzter Parametervektor \beta_hat
barplot(as.vector(beta_hat),
    col       = "gray60",
    border    = "black",
    xlim      = c(0, length(beta_hat)+1),
    ylim      = c(-1, +1),
    ylab      = TeX("$\\hat{\\beta}$"),
    cex.names = 0.6,
    names.arg = c('intercept', cols),
    main      = TeX("$\\hat{\\eta} = \\tilde{x}^T \\hat{\\beta}$"))

# Bernoulli-Parameter \mu
plot(eta, mu,
    type      = "b",
```

```
    pch       = 16,
    col       = "gray60",
    xlim      = c(min(eta)+1, max(eta)-1),
    ylim      = c(-0.1, +1.1),
    xlab      = TeX("$\\hat{\\eta}$"),
    ylab      = TeX("$y$"),
    main      = TeX("$\\hat{\\mu} = 1/(1 + exp(-\\hat{\\eta}))$"))

# geschätzter linearer Prädiktor \eta_hat
# und tatsächliche Labels y
points(eta_hat[y==0], y[y==0],
    pch       = 16,
    col       = "red")
points(eta_hat[y==1], y[y==1],
    pch       = 16,
    col       = "blue")
grid()

# Legende
legend("topleft", c(TeX("$y=0$"), TeX("$y=1$"), TeX("$\\mu$")),
    pch       = c(16,16,16),
    col       = c("red","blue","gray60"),
    bty       = "n",
    cex       = 1,
    x.intersp = 1,
    y.intersp = 2)

# Speichern
dev.copy2pdf(
    file      = "Abbildungen/Logistische_Regression_1.pdf",
    width     = 10,
    height    = 5)
```

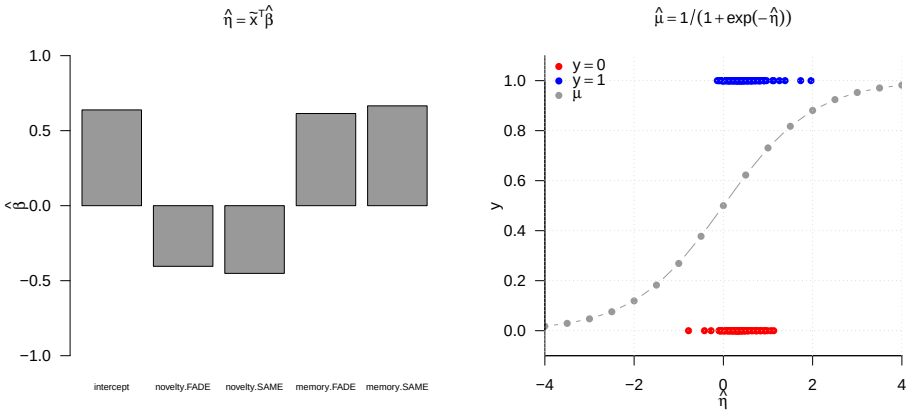

Abbildung 1. Parameterschätzer $\hat{\beta}$ eines logistischen Regressionsmodells mit vier Features (links) und bedingte Wahrscheinlichkeit $p(y = 1|x)$ als Funktion des linearen Prädiktors $\hat{\eta} = \tilde{x}^T\hat{\beta}$ (rechts).

12.5 Zweite Programmieraufgabe

Evaluieren Sie die Klassifikationsperformanz, indem Sie die Sensitivität, Spezifizität und (balancierte) Genauigkeit der Klassifikation berechnen. Gehen Sie dazu wie folgt vor:

- Orientieren Sie sich für diese Aufgabe an Abschnitt 8.5 des Arbeitsblatts (8) *Prädiktive Modellierung*.
- Berechnen Sie die Einträge der 2×2 Konfusionsmatrix und speichern Sie die Ergebnisse als `TN`, `FP`, `FN` und `TP`.
- Berechnen Sie mithilfe der in Vorlesung (8) *Prädiktive Modellierung* angegebenen Formeln die true positive rate (TPR), true negative rate (TNR) sowie accuracy (ACC) und balanced accuracy (BAC).
- Geben Sie die Resultate ihrer Analyse aus. Kommentieren Sie die Klassifikationsperformanz im Hinblick auf den Unterschied zwischen TPR und TNR. Sie sollten folgende Ergebnisse erhalten:

```
true positive rate (sensitivity) :  0.889
true negative rate (specificity) :  0.066
accuracy                         :  0.552
balanced accuracy                :  0.477
```

- Kommentieren Sie im ersten Code-Segment in Abschnitt 12.1 diejenigen Zeilen, die die Labelvariable "Geschlecht" extrahieren, und entkommentieren Sie diejenigen Zeilen, die die Labelvariable "Abitur" extrahieren. Führen Sie die Analyse erneut durch. Sie sollten folgende Ergebnisse erhalten:

```
true positive rate (sensitivity) :  0.837
true negative rate (specificity) :  0.471
accuracy                         :  0.714
balanced accuracy                :  0.654
```

12.6 Lückentext

Füllen Sie mit den in der Übung gewonnenen Erkenntnissen den folgenden Lückentext aus und präsentieren Sie die Ergebnisse im Seminar:

Lückentext: Für die Klassifikation einer binären Labelvariable mit den Werten 0 und 1 aus einer $m \times n$ Datenmatrix benutzt die Logistische Regression (LR) einen $(m+1)$-dimensionalen Parametervektor β. Aus dem Parametervektor ergibt sich der skalare _____ $\eta = \tilde{x}^{\mathrm{T}}\beta$. Aus dem linearen Prädiktor ergibt sich der _____ $\mu = 1/\left(1 + \exp\left(-\eta\right)\right)$. Die Labelvariable wird schließlich als _____-verteilt mit dem Parameter μ angenommen. Im vorliegenden Datensatz wurden die Labels aus der Variable _____ gewonnen, wobei "negative Fälle" _____ und "positive Fälle" _____. Die Klassifikation der so erzeugten Labelvariable aus _____ Features erfolgt mittels *leave-one-out cross-validation*. Es ergeben sich eine true positive rate (TPR, "Sensitivität") von _____ und eine true negative rate (TNR, "Spezifizität") von _____. Die balanced accuracy der Klassifikation ist mit _____ kleiner als die accuracy mit _____, weil _____.

12.7 Mögliche Klausurfrage

Präsentieren Sie im Seminar folgende Klausurfrage und erklären Sie die richtige Antwort:

Frage: Welche Funktion ist die Mean-Funktion im Modell der Logistischen Regression?

a) $f(\eta) = \eta$

b) $f(\eta) = \ln\left(\frac{\eta}{1-\eta}\right)$

c) $f(\eta) = \frac{1}{1+\exp(-\eta)}$

d) $f(\eta) = (x - \eta)^2$

12.8 Kinderwitz

Wohin geht ein Zyklop mit Sehproblemen?

Antwort: Zum Augearzt.