10 Bayes-Klassifikation

10.1 Mathematische Grundlagen

Im *Modell der Linearen Diskriminanzanalyse* (LDA) haben ein Zufallsvektor $x \in \mathbb{R}^m$ und eine Labelvariable $y \in \{0, 1\}$ die gemeinsame Verteilung

$$\mathbb{P}(x, y) = \mathbb{P}(x|y)\,\mathbb{P}(y)\ , \tag{1}$$

wobei die Verteilung der Labelvariable eine *Bernoulli-Verteilung* ist

$$p(y) = \mathrm{Bern}(y; \mu) \tag{2}$$

und die bedingte Verteilung des Zufallsvektors eine *multivariate Normalverteilung* ist

$$p(x|y) = \begin{cases} N(x; \mu_0, \Sigma)\ , & \text{wenn } y = 0 \\ N(x; \mu_1, \Sigma)\ , & \text{wenn } y = 1\ . \end{cases} \tag{3}$$

Hierbei repräsentiert $\mu \in [0, 1]$ die Wahrscheinlichkeit bzw. Häufigkeit der "positiven Klasse" $y = 1$; $\mu_0 \in \mathbb{R}^m$ und $\mu_1 \in \mathbb{R}^m$ repräsentieren die klassenspezifischen Erwartungswertparameter; und $\Sigma \in \mathbb{R}^{m\times m}$ p.d. repräsentiert den klassenübergreifenden Kovarianzmatrixparameter.

Wie in der Vorlesung gezeigt wurde, sind die *Maximum-Likelihood-Schätzer* dieser Parameter für einen gegebenen Datensatz $\mathcal{D} = \{(x_1, y_1), ..., (x_n, y_n)\}$

$$\begin{aligned} \hat{\mu} &= \frac{1}{n}\sum_{i=1}^{n} y_i \\ \hat{\mu}_0 &= \frac{1}{\sum_{i=1}^{n}(1-y_i)}\sum_{i=1}^{n}(1-y_i)\,x_i \\ \hat{\mu}_1 &= \frac{1}{\sum_{i=1}^{n} y_i}\sum_{i=1}^{n} y_i\,x_i \\ \hat{\Sigma} &= \frac{1}{n}\sum_{i=1}^{n}\left(x_i - \hat{\mu}_{y_i}\right)\left(x_i - \hat{\mu}_{y_i}\right)^{\mathrm{T}}\ , \end{aligned} \tag{4}$$

wobei $y_i \in \{0, 1\}$ die Labelvariable des i-ten Datenpunkts und $x_i \in \mathbb{R}^m$ den Featurevektor des i-ten Datenpunkts darstellt.

Wenn die Parameter eines LDA-Modells einmal geschätzt sind, kann aus ihnen die *Posterior-Wahrscheinlichkeit*, d.h. die bedingte Wahrscheinlichkeit $p(y = 1|x)$ für einen potentiell neuen Datenpunkt $x \in \mathbb{R}^m$ berechnet werden

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

wobei $\tilde{x}$ der *erweiterte Featurevektor* ist

$$\tilde{x} := \begin{pmatrix} 1 \\ x \end{pmatrix} \in \mathbb{R}^{m+1} \tag{6}$$

und β den *Inferenzparametervektor* darstellt:

$$\beta := \begin{pmatrix} \frac{1}{2} \left(\mu_0^T \Sigma^{-1} \mu_0 - \mu_1^T \Sigma^{-1} \mu_1 \right) + \ln \left(\frac{\mu}{1-\mu} \right) \\ -\Sigma^{-1} (\mu_0 - \mu_1) \end{pmatrix} \in \mathbb{R}^{m+1} \, . \tag{7}$$

Auf Grundlage der Posterior-Wahrscheinlichkeit kann ein *Bayes-Klassifikator* für die LDA definiert werden, der einen potentiell neuen Datenpunkt x der Klasse 1 zuschreibt, wenn $p(y = 1|x) > 0.5$, und der Klasse 0 zuschreibt, wenn $p(y = 1|x) \leq 0.5$.

In der vorliegenden Übung wollen wir LDA-Modellschätzung via *leave-one-out cross-validation* sowie Bayes-Klassifikation auf Grundlage der geschätzten Modellparameter für einen realen Datensatz nachvollziehen.

10.2 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$subject, 'subA')            # Personen aus Studie A
cols  = c('novelty.SAME', 'memory.SAME')         # Definition Variablen
X     = t(as.matrix(D[rows,cols]))               # Datenmatrix
y     = t(as.matrix(D$memory[rows]))             # gruppendefinierende Variable
y_med = median(y)                                # Median-Gedächtnisleistung
y     = 0*(y <= y_med) + 1*(y > y_med)           # Labelvariable
print(dim(X))                                    # Überprüfung Datenmatrix
print(dim(y))                                    # Überprüfung Labelvariable
print(y_med)                                     # Ausgabe Median
```

10.3 Erste Programmieraufgabe

Führen Sie eine Bayes-Klassifikation auf Grundlage der Datenmatrix X und der Labelvariable y durch, indem Sie die LDA-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`.
- Definieren Sie stattdessen die Variable `p_y` durch n-fache Replikation von `NaN`.
- Berechnen Sie im Abschnitt "Training" innerhalb der Schleife `n_train` als die Anzahl der Spalten von `x_train` und berechnen Sie `mu_hat` als Stichprobenmittel über `y_train`.
- Berechnen Sie mittels `rowMeans` die Schätzer der Erwartungswertparameter μ_0 und μ_1, indem Sie die Spalten der Trainingsdatenfeatures mit `y_train == 0` bzw. `y_train == 1` indizieren. Speichern Sie die Ergebnisse als `mu_0_hat` und `mu_1_hat`.
- Initialisieren Sie `Sigma_hat` als eine $m \times m$ Nullmatrix.
- Fügen Sie dann folgenden Code innerhalb der Schleife ein:

```
for (j in 1:n_train) {
    Sigma_hat = Sigma_hat + (1/n_train) *
                ((y_train[j] == 0)*(x_train[,j]-mu_0_hat) %*% t((x_train[,j]-mu_0_hat))
                +(y_train[j] == 1)*(x_train[,j]-mu_1_hat) %*% t((x_train[,j]-mu_1_hat)))
}
beta_hat      = matrix(c((1/2)*( t(mu_0_hat) %*% solve(Sigma_hat) %*% mu_0_hat
                               - 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)
```

- 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 in den i-ten Eintrag der 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] 109
```

10.4 Abbildung in R

Die in der Datenmatrix enthaltenen Variablen sollen nun unter Berücksichtigung der berechneten Posterior-Wahrscheinlichkeit sowie ihrer tatsächlichen Klassenzugehörigkeit visualisiert werden. Erklären Sie dazu den folgenden R-Code und die Abbildung, die er erzeugt:

```
# probabilistische Prädiktion
num_cols  = 100                                    # Anzahl Stufen Farbgradient
col_fct   = colorRampPalette(c("red", "purple", "blue"))    # Farbpalette
cols      = col_fct(num_cols)                      # Farbgradient
col_ind   = as.integer(p_y * 99) + 1               # Gradientenindizes
p_y_cols  = cols[col_ind]                          # Datenpunktfarben

# multivariate Isokonturen
library(ellipse)
iso_0_hat = ellipse(Sigma_hat, level = 0.5, centre = mu_0_hat)
iso_1_hat = ellipse(Sigma_hat, level = 0.5, centre = mu_1_hat)

# Abbildungsparameter
library(latex2exp)
par(
    family    = "sans",
    pty       = "m",
    bty       = "o",
    lwd       = 1,
    las       = 1,
    mgp       = c(2,1,0),
    xaxs      = "i",
    yaxs      = "i",
    cex       = 1.2)

# Datenpunkte Klasse 0
plot(X[1,y==0], X[2,y==0],
    pch       = 15,
    col       = p_y_cols[y==0],
    xlab      = "Novelty-SAME-Score",
    ylab      = "Memory-SAME-Score",
    xlim      = c(-3, +3),
    ylim      = c(-3, +3))

# Datenpunkte Klasse 1
points(X[1,y==1], X[2,y==1],
    pch       = 16,
    col       = p_y_cols[y==1])

# geschätze Verteilung Klasse 0
lines(iso_0_hat[,1], iso_0_hat[,2],
    col       = "Red",
    lty       = 2,
    lwd       = 2)
points(mu_0_hat[1], mu_0_hat[2],
```

```
    col       = "Red",
    pch       = 3,
    lwd       = 2,
    cex       = 1)

# geschätze Verteilung Klasse 1
lines(iso_1_hat[,1], iso_1_hat[,2],
    col       = "Blue",
    lty       = 2,
    lwd       = 2)
points(mu_1_hat[1], mu_1_hat[2],
    col       = "Blue",
    pch       = 3,
    lwd       = 2,
    cex       = 1)

# Legende
legend("topleft", c("Gedächtnisleistung < Median (y = 0)",
                    "Gedächtnisleistung > Median (y = 1)",
                    "geschätze Verteilung negative Fälle (y = 0)",
                    "geschätze Verteilung positive Fälle (y = 1)",
                    "Posterior-Wahrscheinlichkeit p(y = 1|x)"),
    lty        = 0,
    lwd        = c( 1,  1, 2, 2,  1),
    pch        = c(15, 16, 3, 3, 17),
    col        = c("gray50", "gray50", "red", "blue", "purple"),
    bty        = "n")

# Speichern
dev.copy2pdf(
    file      = "Abbildungen/Bayes-Klassifikation_1.pdf",
    width     = 9,
    height    = 9)
```

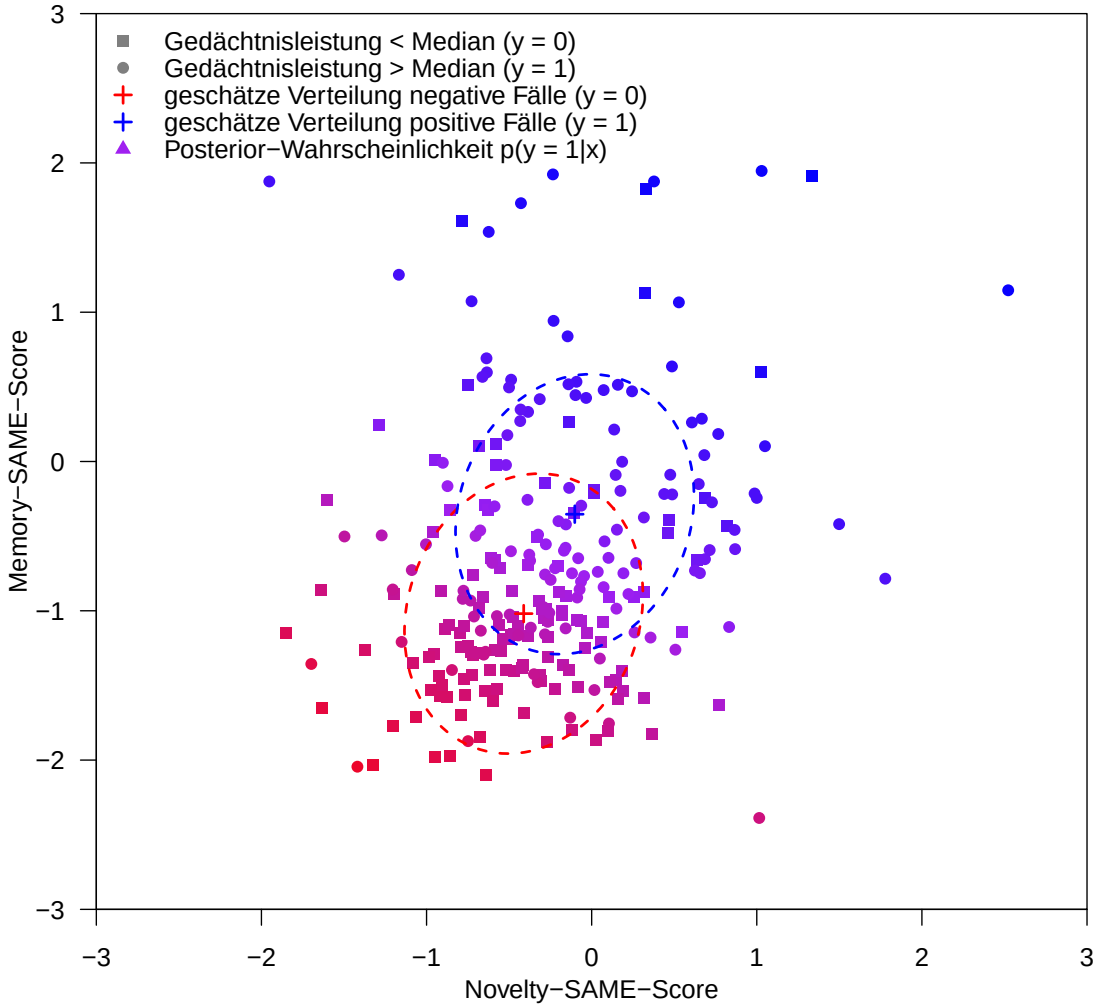

Abbildung 1. Novelty-SAME-Score und Memory-SAME-Score, getrennt nach Gedächtnisleistung (Vierecke: $y = 0$; Kreise: $y = 1$), eingefärbt nach Posterior-Wahrscheinlichkeit $p(y = 1|x)$ (rot: 0; blau: 1; violett: 0.5), mit geschätzten Erwartungswerten (Kreuze) sowie geschätzten Kovarianzmatrizen (Ellipsen) für beide Klassen.

10.5 Zweite Programmieraufgabe

Evaluieren Sie die Klassifikationsperformanz, indem Sie die Sensitivität, Spezifizität und 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) und die Klassifikationsgenauigkeit (ACC).
- Geben Sie die Resultate ihrer Analyse aus. Sie sollten folgende Ergebnisse erhalten:

```
true positive rate (sensitivity)      :  0.6202
true negative rate (specificity)      :  0.7769
Klassifikationsgenauigkeit (accuracy) :  0.6988
```

10.6 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 sind die vier Modellparameter der Linearen Diskriminanzanalyse (LDA) ein _____ (μ), zwei _____ (μ_0, μ_1) und eine _____ (Σ). Im vorliegenden Datensatz ergeben sich die Labels durch *median-split* der Variable _____, wobei "negative Fälle" _____ und "positive Fälle" _____. Die LDA-basierte Bayes-Klassifikation der so erzeugten Labelvariable aus den Variablen _____ und _____ erfolgt mittels *leave-one-out cross-validation*. Die sich ergebende true positive rate (TPR, "Sensitivität") in Höhe von _____ bedeutet, dass _____. Die sich ergebende true negative rate (TNR, "Spezifizität") in Höhe von _____ bedeutet, dass _____.

10.7 Mögliche Klausurfrage

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

Frage: Gegeben sei das Modell der linearen Diskriminanzanalyse für die Label-Zufallsvariable y mit Ergebnisraum $\{0, 1\}$ und den Feature-Zufallsvektor x mit Ergebnisraum $\mathbb{R}^m$. Welcher Wahrscheinlichkeitsverteilung folgt in diesem Modell der Feature-Zufallsvektor x in Abhängigkeit von der Label-Zufallsvariable y?

a) Bernoulli-Verteilung

b) Gamma-Verteilung

c) univariate Normalverteilung

d) multivariate Normalverteilung

10.8 Kinderwitz

Wie nennt man einen Keks unter einem Baum?

Antwort: Ein schattiges Plätzchen.