Lineare Klassifikationsmethoden

Verena Krieg

Universität Ulm
Fakultät für Mathematik und Wirtschaftswissenschaften

08. Mai 2007

Inhaltsverzeichnis

Inhaltsverzeichnis

Was ist „Klassifikation“?

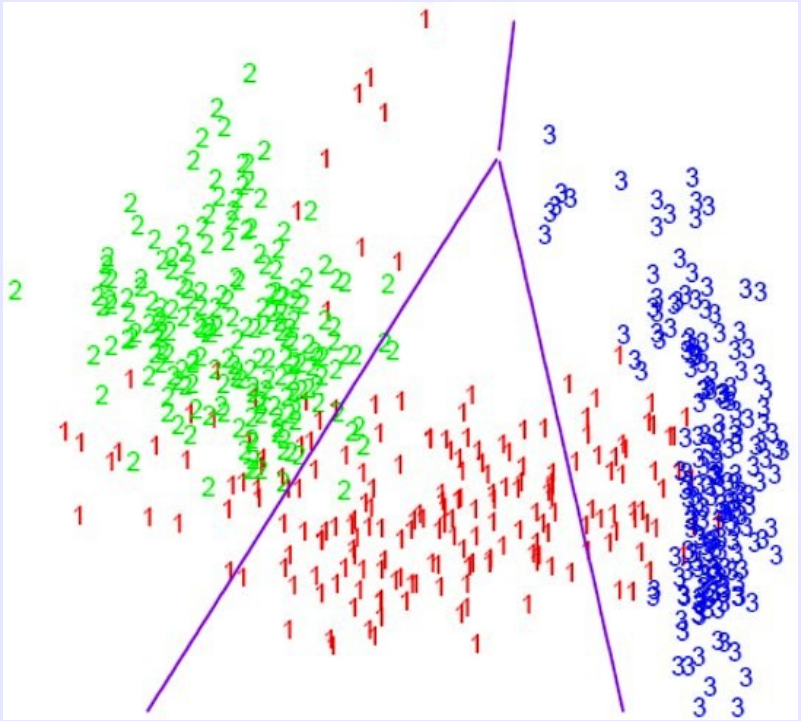
- überwachtes Lernen (supervised learning)
  - Ausgangsdaten: $X = (X_1, \ldots, X_p)$
  - Lerndaten: $(x_i, g_i),\ x_i = (x_{1i}, \ldots, x_{pi})$
    mit $X(\omega_i) = x_i,\ G(x_i) = g_i,\ i = 1, \ldots, N$
  - Inputfunktion: $G(x) \in \mathcal{G}$, $\mathcal{G}$ diskret
- $G^{-1}(\cdot)$ teilt Ausgangsraum in Regionen ein
- Jede Region $G^{-1}(g_i)$ wird nach der Klasse g_i benannt
- $|\mathcal{G}| = K \Rightarrow \exists$ K Klassen
- $\Rightarrow$ Entscheidungsgrenzen

Lineare Modelle

- Entscheidungsgrenzen
  - linear
  - abhängig von der Inputfunktion G(x)

- **lineare Klassifikationsmethoden**
  - lineare Regression
  - lineare Diskriminanzanalyse
  - logistische Regression
  - Hyperebenentrennung

Lineare Entscheidungsgrenzen



$\Rightarrow$ Daten mit linearen Entscheidungsgrenzen (lineare Diskriminanzanalyse)

Inhaltsverzeichnis

Lineare Regression

- statistisches Analyseverfahren
- Daten der Form $(x_i, y_i),\ x_i \in \mathbb{R}^{p+1},\ y_i \in \mathbb{R}^K$
- Modell:

$$Y = X\beta + \epsilon$$

$Y \in \mathbb{R}^{N\times K}$, $\beta \in \mathbb{R}^{(p+1)\times K}$, $X \in \mathbb{R}^{N\times(p+1)}$, $\epsilon \in \mathbb{R}^{N\times K}$

- Ziel: Schätzen der β's
- $\Rightarrow$ Lösung durch Methode der kleinsten Quadrate

Lineare Regression einer Indikatormatrix

- diskrete Menge $\mathcal{G}$, $i = 1, \ldots, N$ Beobachtungen
- Annahme: $\mathcal{G}$ hat K Klassen
  $\Rightarrow \exists$ K Klassenindikatoren Y_{ik}, $k = 1, \ldots, K$

$$Y_{ik} = \begin{cases} 1 & \text{, G(x)=k} \\ 0 & \text{, sonst} \end{cases}$$

- $Y_i = (Y_{i1}, \ldots, Y_{iK})$; bestimme die N×K Übergangsmatrix Y
- Y = Matrix aus 0en und 1en, eine 1 pro Zeile
- $\Rightarrow$ Schätzung:

$$\widehat{Y} = X(X^T X)^{-1} X^T Y = X\widehat{\beta}$$

Klassifikationsalgorithmus

- Bestimme $\widehat{\beta} = (X^T X)^{-1} X^T Y$ mit $X \in \mathbb{R}^{N \times (p+1)}$
- Klassifikation einer neuen Beobachtung mit Input x:
  - Bestimme den geschätzten Output:

$$\widehat{Y} = [(1, x)\widehat{\beta}]^T = [(1, x_1, \ldots, x_p)\widehat{\beta}]^T = \begin{pmatrix} \widehat{Y}_1 \\ \vdots \\ \widehat{Y}_K \end{pmatrix}$$

  - Bestimme größte Komponente von $\widehat{Y}$ und klassifiziere mit:

$$\widehat{G}(x) = \arg\max_{k \in \mathcal{G}} \widehat{Y}_k$$

Beispiel: Lineare Regression

- Simulation zweier Klassen (hierzu später mehr)
-

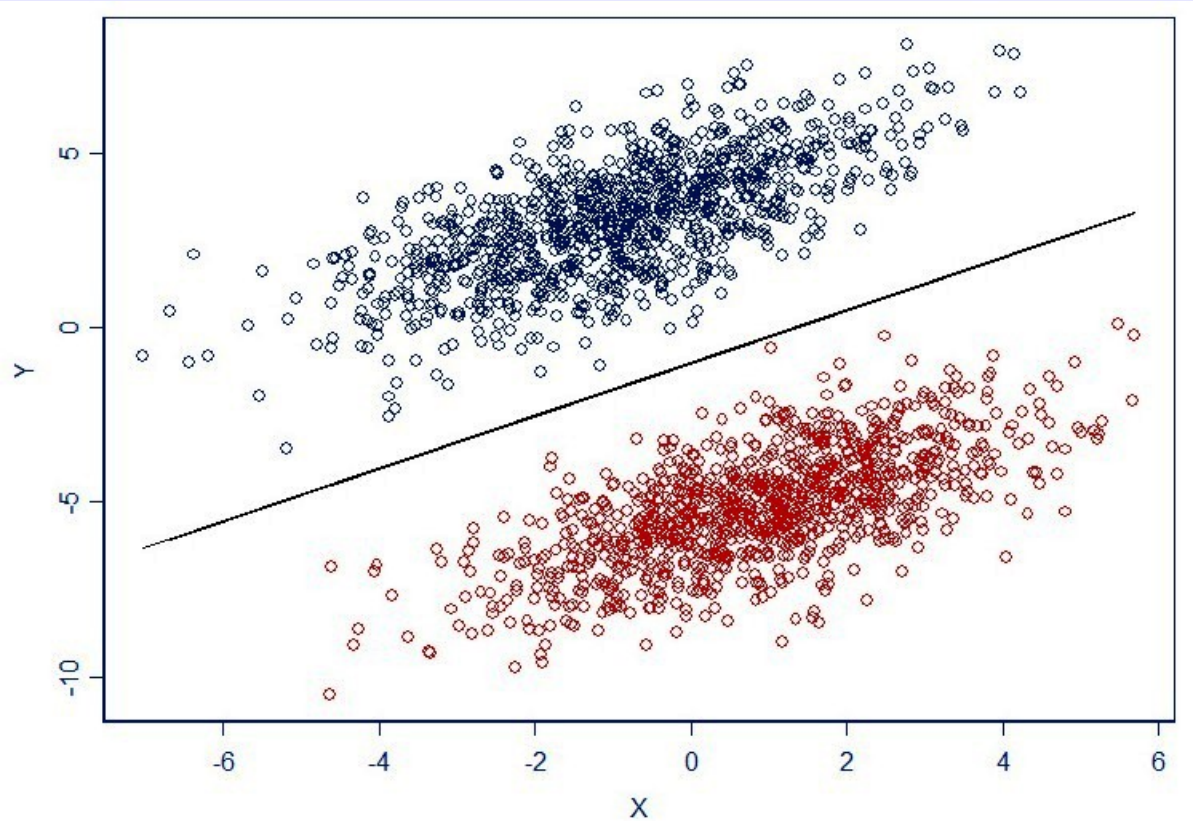
$$X = \begin{pmatrix} 1 & 2.71 & 5.24 \\ \vdots & \vdots & \\ 1 & 0.04 & 4.71 \\ 1 & -0.04 & 3.44 \\ \vdots & \vdots & \vdots \\ 1 & 0.31 & -8.03 \end{pmatrix} \quad Y = \begin{pmatrix} 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & 1 \end{pmatrix} \quad \Rightarrow \widehat{\beta} = \begin{pmatrix} 0.60 & 0.40 \\ -0.07 & 0.07 \\ 0.10 & -0.10 \end{pmatrix}$$

- Berechnung der Entscheidungsgrenze durch:

$$\widehat{\beta}_{11} + \widehat{\beta}_{21} x_1 + \widehat{\beta}_{31} x_2 = \widehat{\beta}_{12} + \widehat{\beta}_{22} x_1 + \widehat{\beta}_{23} x_2$$

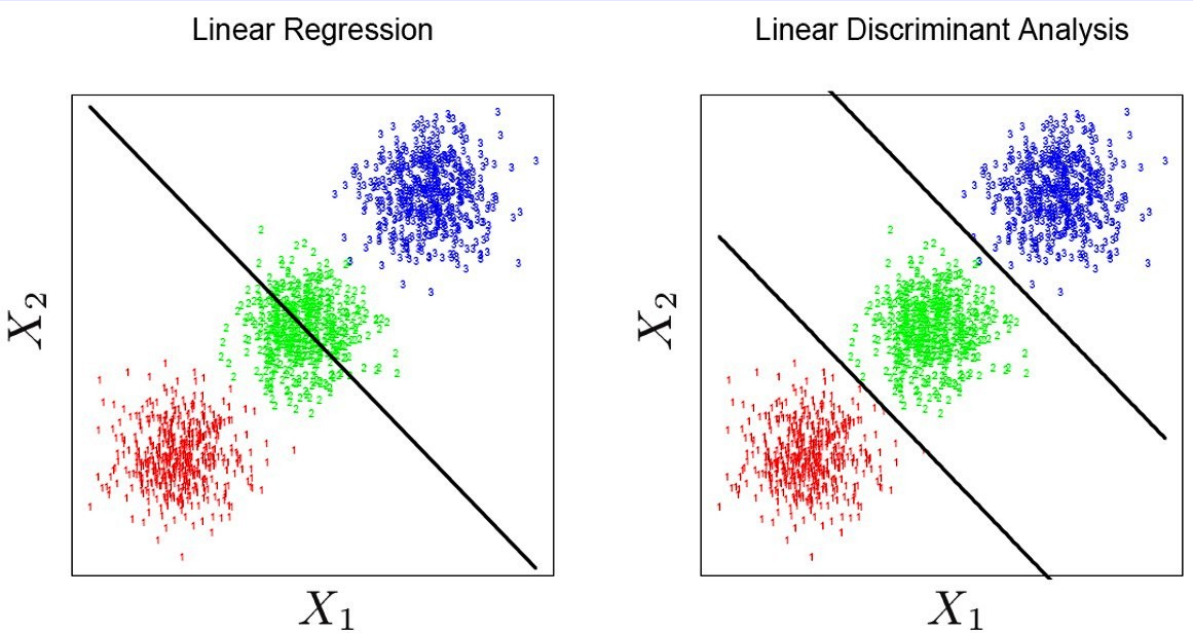
$x_1 =$ 2. Spalte von X

Beispiel: Lineare Regression



$$x_2 = \frac{\widehat{\beta}_{12} - \widehat{\beta}_{11}}{\widehat{\beta}_{31} - \widehat{\beta}_{32}} + \frac{\widehat{\beta}_{22} - \widehat{\beta}_{21}}{\widehat{\beta}_{31} - \widehat{\beta}_{32}} x_1 = -1 + 0.7\ x_1$$

Problem: Anzahl Klassen: $K \geq 3$



$\Rightarrow$ Klassen durch lineare Entscheidungsgrenzen perfekt getrennt
$\Rightarrow$ Problem: mittlere Klasse
$\Rightarrow$ Klassen können durch andere überdeckt werden

Inhaltsverzeichnis

Lineare Diskriminanzanalyse (LDA)

- $f(x|k)$: bedingte Dichte von X in der Klasse k
- p_k: A-Prioriwahrscheinlichkeit der Klasse k, $\sum_{k=1}^{K} p_k = 1$

$\Rightarrow$ Anwendung vom Bayes-Theorem liefert:

$$P(G = k|X = x) = \frac{f(x|k)p_k}{\sum_{l=1}^{K} f(x|l)p_l}$$

Lineare Diskriminanzanalyse (LDA)

Klassendichte mit multivariater Normalverteilung

$$f(x|k) = \frac{1}{(2\pi)^{\frac{p}{2}}(\det \Sigma_k)^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)}$$

- Spezialfall: $\Sigma_k = \Sigma \quad \forall k$

Klassifikation

- Idee: paarweiser Vergleich der Klassen
- $P(G = k|X = x) > P(G = l|X = x)$
  $\Rightarrow$ x zur Klasse k zuordnen
- log-Raten, in x linear:

$$\log \frac{P(G = k|X = x)}{P(G = l|X = x)} = \log \frac{p_k}{p_l} - \frac{1}{2}(\mu_k + \mu_l)^T \Sigma^{-1}(\mu_k - \mu_l) + x^T \Sigma^{-1}(\mu_k - \mu_l)$$

$$\log \frac{P(G = k|X = x)}{P(G = l|X = x)} > 0$$

- Ungleichung gilt für alle Klassenpaare

Klassifikation

- **Lineare Diskriminanzfunktion**:

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log p_k$$

- Klassifikationsregel:

$$G(x) = \arg\max_k \delta_k(x)$$

Beispiel: Lineare Diskriminanzanalyse

- Simulation zweier Klassen mit je 1000 multivariat normalverteilten ZV mit

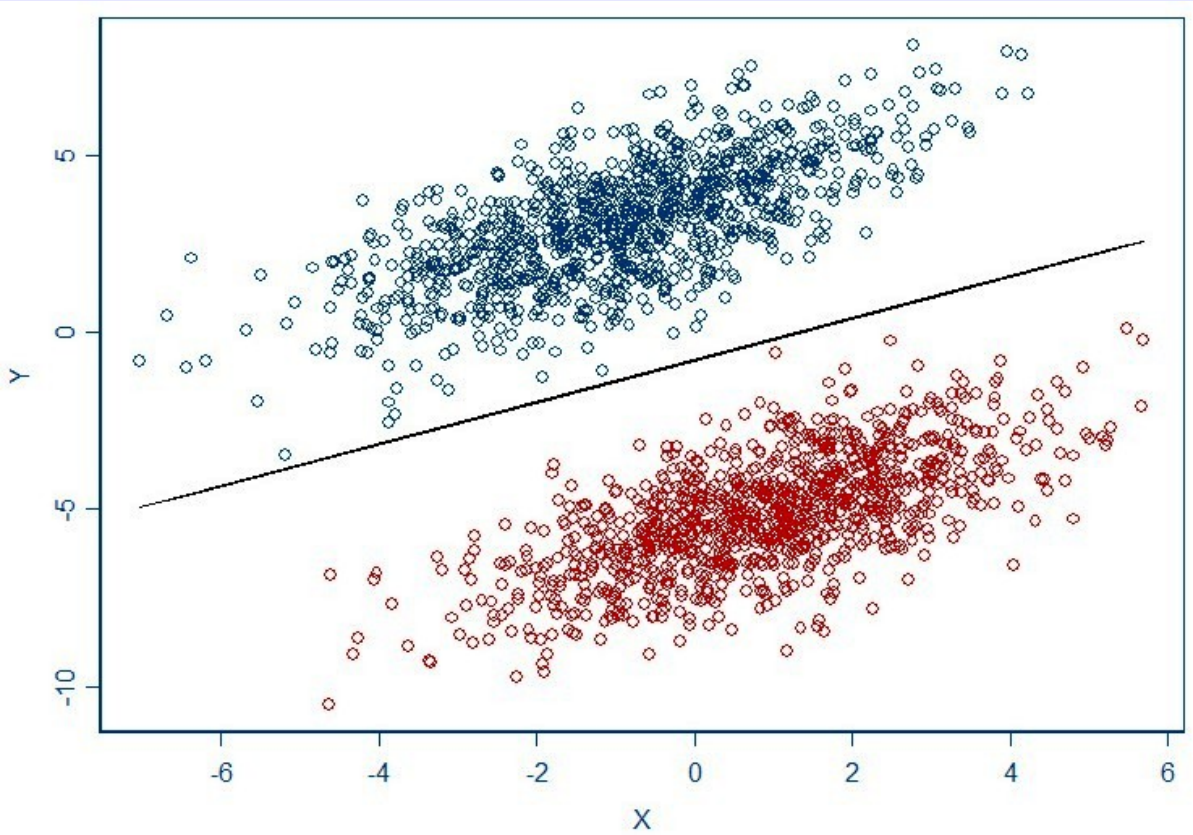
$$\mu_1 = \begin{pmatrix} -1 \\ 3 \end{pmatrix} \quad \mu_2 = \begin{pmatrix} 1 \\ -5 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 3 & 2 \\ 2 & 3 \end{pmatrix}$$

- Berechnung der Entscheidungsgrenze durch:

$$\log \frac{P(G = 1|X = x)}{P(G = 2|X = x)} = 0$$

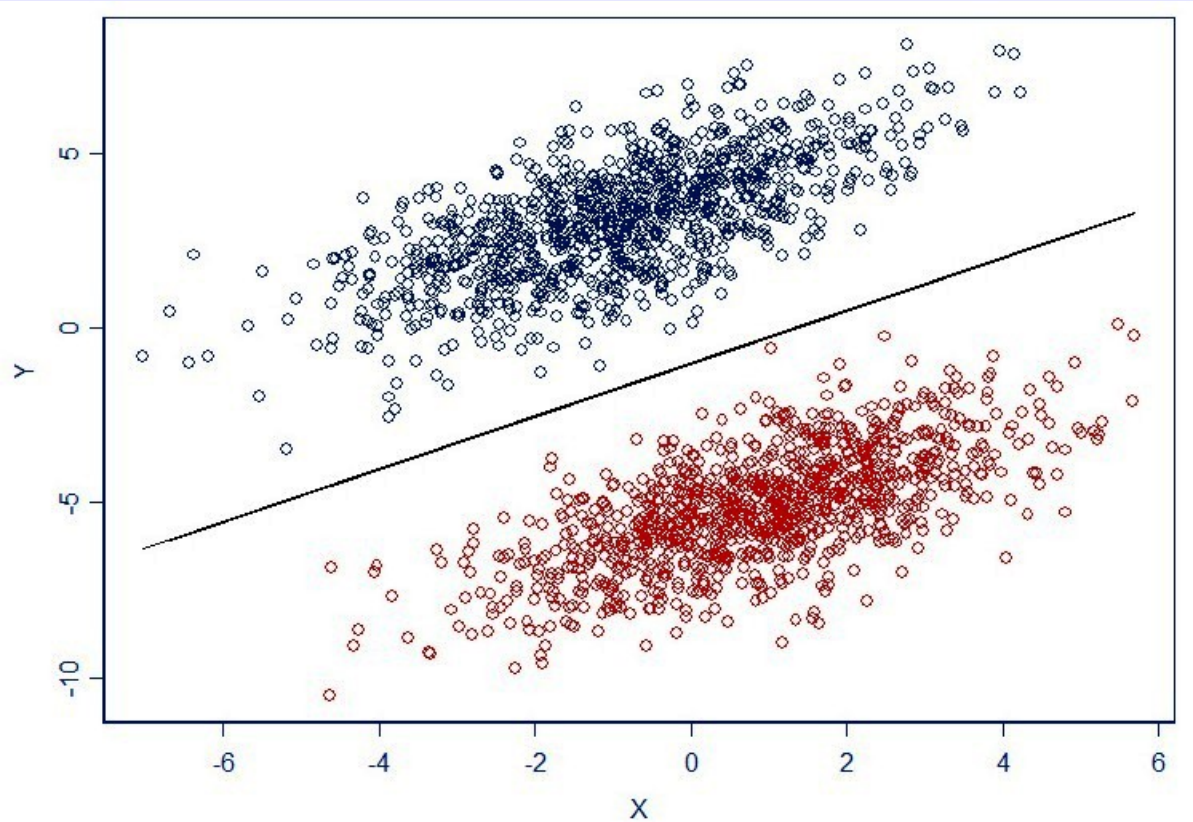
$$x^T \Sigma^{-1} \mu_1 - \frac{1}{2} \mu_1^T \Sigma^{-1} \mu_1 + \log p_1 = x^T \Sigma^{-1} \mu_2 - \frac{1}{2} \mu_2^T \Sigma^{-1} \mu_2 + \log p_2$$

Beispiel: Lineare Diskriminanzanalyse



$$x_2 = -0.79 + 0.59\ x_1$$

Beispiel: Lineare Regression



$$x_2 = -1 + 0.7\ x_1$$

Quadratische Diskriminanzanalyse (QDA)

- Σ_k sind nicht alle gleich
- Schätzen der Kovarianzmatrizen für jede einzelne Klasse

- **Quadratische Diskriminanzfunktion**

$$\delta_k(x) = -\frac{1}{2}\log\det\Sigma_k - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) + \log p_k$$

- Klassifikationsregel:

$$G(x) = \arg\max_k \delta_k(x)$$

- Entscheidungsgrenzen: quadratisch in x

Inhaltsverzeichnis

Logistische Regression

- $X\beta = g^T(x)$ mit $g(x) = \log \frac{x}{1-x} \quad \forall x \in (0,1)$
- Bezeichnung: $P_k(x;\theta) = P(G = k|X = x)$
- p_k: A-Prioriwahrscheinlichkeit der Klasse k, $\sum_{k=1}^{K} p_k = 1$
- Parametermenge: $\theta = \{\beta_{10}, \beta_1^T, \ldots, \beta_{(K-1)0}, \beta_{K-1}^T\}$
- Entscheidungsgrenze: $P(G = k|X = x) = P(G = l|X = x)$

- **Modell** (Vergleich der Klassen $1, \ldots, K-1$ mit der Klasse K):

$$\log \frac{P_k(x;\theta)}{P_K(x;\theta)} = \beta_{k0} + \beta_k^T x \quad \forall k = 1, \ldots, K-1$$

$$\Rightarrow P_k(x;\theta) = \frac{\exp(\beta_{k0} + \beta_k^T x)}{1 + \sum_{l=1}^{K-1} \exp(\beta_{l0} + \beta_l^T x)} \quad \forall k = 1, \ldots, K-1$$

$$P_K(x;\theta) = \frac{1}{1 + \sum_{l=1}^{K-1} \exp(\beta_{l0} + \beta_l^T x)}$$

Inhaltsverzeichnis

Berechnung der Parameter der logistischen Regression

- Finde Parameter, die die bedingte Likelihoodfunktion von G bei gegebenem X maximieren
- Likelihoodfunktion (unabhängige Lerndaten):

$$L(\theta) = \prod_{i=1}^{N} P_{g_i}(x_i; \theta)$$

- log-Likelihoodfunktion der N Beobachtungen:

$$\log L(\theta) = \sum_{i=1}^{N} \log P_{g_i}(x_i; \theta)$$

mit $P_k(x_i; \theta) = Pr(G = k | X = x_i; \theta)$

Spezialfall: 2 Klassen

- Bestimme 2 Klassen mit Hilfe von

$$y_i = \begin{cases} 1 & \text{, falls } g_i = 1 \\ 0 & \text{, falls } g_i = 2 \end{cases}$$

- Setze $P_1(x;\theta) = P(x;\theta), \quad P_2(x;\theta) = 1 - P(x;\theta)$

- **log-Likelihoodfunktion**:

$$\begin{aligned} \log L(\beta) &= \sum_{i=1}^{N} \{y_i \log P(x_i;\beta) + (1-y_i)\log(1 - P(x_i;\beta))\} \\ &= \sum_{i=1}^{N} \{y_i \beta^T x_i - \log(1 + e^{\beta^T x_i})\} \end{aligned}$$

- $\beta = (\beta_{10}, \beta_1)$, x_i mit konstantem Term 1 für β_{10}

Spezialfall: 2 Klassen

- Maximierung der log-Likelihoodfunktion:

$$\frac{\partial \log L(\beta)}{\partial \beta} = \sum_{i=1}^{N} x_i(y_i - P(x_i;\beta)) = 0$$

  $\Rightarrow$ p+1 Gleichungen nicht-linear in β
- 1. Komponente von $x_i = 1 \Rightarrow \sum_{i=1}^{N} y_i = \sum_{i=1}^{N} P(x_i;\beta)$
- Lösung des GLS mit **Newton-Verfahren**

Newton-Verfahren

- 2. Ableitung oder Hessematrix

$$\frac{\partial^2 \log L(\beta)}{\partial \beta \partial \beta^T} = -\sum_{i=1}^{N} x_i x_i^T P(x_i;\beta)(1 - P(x_i;\beta))$$

- Startwert: β^{alt}
- Newton-Iteration:

$$\beta^{neu} = \beta^{alt} - \left(\frac{\partial^2 \log L(\beta)}{\partial \beta \partial \beta^T}\right)^{-1} \frac{\partial \log L(\beta)}{\partial \beta}$$

  wobei Ableitungen an β^{alt} ausgewertet wurden
- Konvergenz, falls log-Likelihoodfunktion konkav, aber Problem: „overshooting“

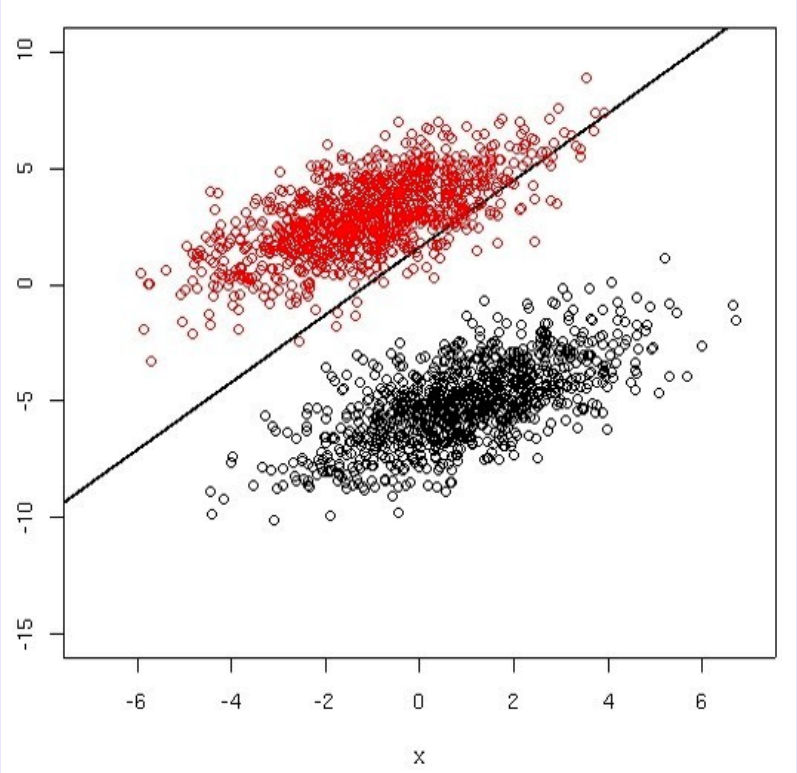
Beispiel

- zwei Klassen (siehe LDA)
- X wie im Beispiel zur linearen Regression
- Startwert beta.start=0 wählen
- $Y = (1, \ldots, 1, 0, \ldots, 0)^T$
- log-Likelihoodfunktion aufstellen
- Optimierung in S-Plus durchführen
- $\Rightarrow (\widehat{\beta}_1, \widehat{\beta}_2, \widehat{\beta}_3)^T = (43.2507, 39.11790, -27.07334)^T$
- Berechnung der Entscheidungsgrenze durch:

$$\log \frac{P(G = 1|X = x)}{P(G = 2|X = x)} = \widehat{\beta}_1 + \widehat{\beta}_2 x_1 + \widehat{\beta}_3 x_2 = 0$$

$x_1 = 2$. Spalte von X

Beispiel



$$x_2 = -\frac{\widehat{\beta}_1}{\widehat{\beta}_3} - \frac{\widehat{\beta}_2}{\widehat{\beta}_3}\, x_1 = 1.597539 + 1.444887\, x_1$$

Inhaltsverzeichnis

Lineare Diskriminanzanalyse:

$$\begin{aligned}\log \frac{P(G=k|X=x)}{P(G=K|X=x)} &= \log \frac{p_k}{p_K} - \frac{1}{2}(\mu_k+\mu_K)^T\Sigma^{-1}(\mu_k-\mu_K) + x^T\Sigma^{-1}(\mu_k-\mu_K) \\ &= \alpha_{k0} + \alpha_k^T x\end{aligned}$$

Logistische Regression:

$$\log \frac{P(G=k|X=x)}{P(G=K|X=x)} = \beta_{k0} + \beta_k^T x$$

$\Rightarrow$ Anschein der Gleichheit der Modelle
$\Rightarrow$ Unterschied: Schätzung der Koeffizienten
$\Rightarrow$ Logistische: allgemeiner, da weniger Annahmen nötig

- **Logistisches Regressionsmodell**:
  - Beliebige Dichtefunktion
  - Maximierung der bedingten Likelihoodfunktion
  - Ausreißer $\Rightarrow$ geringe Gewichtung

- **Lineare Diskriminanzanalyse**:
  - Dichtefunktion der Normalverteilung
  - Maximierung der log-Likelihoodfunktion
  - Schätzung von $\widehat{\mu_k}$, $\widehat{\Sigma}$ und $\widehat{p_k}$
    $\Rightarrow$ mehr Informationen $\Rightarrow$ effizientere Schätzung
  - Ausreißer $\Rightarrow$ Schätzung der allgemeinen Kovarianzmatrix
    $\Rightarrow$ nicht robust gegenüber großen Ausreißern

$\Rightarrow$ Logistische Regression ist robuster als die LDA

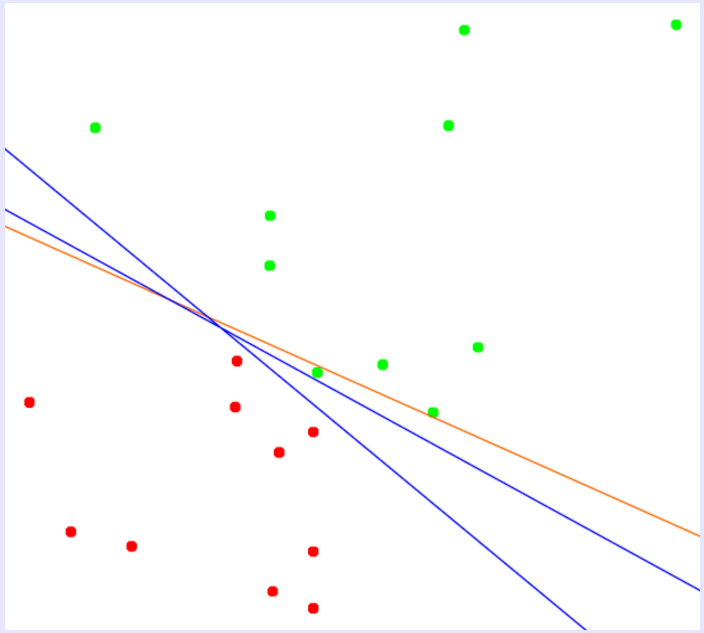
$\Rightarrow$ Praxis: beide Modelle liefern oft sehr ähnliche Ergebnisse

Inhaltsverzeichnis

Hyperebenentrennung

- Prozedur konstruiert lineare Entscheidungsgrenzen, die explizit versuchen die Daten so gut es geht in verschiedene Klassen aufzuteilen
- Basis für „Support vector machines“
- Lerndaten durch Hyperebenen perfekt getrennt
  $\Rightarrow$ LDA und andere lineare Modelle liefern nicht immer eine perfekte Trennung

Einfaches Beispiel mit 2 Klassen



orange Linie: Lösung der kleinsten Quadrate (ein Lernpunkt falsch klassifiziert)

2 blaue trennende Hyperebenen: durch Perzeptron-Algorithmus mit verschiedenen Startpunkten

Inhaltsverzeichnis

Rosenblatt's Perzeptron-Algorithmus (1958)

- Ziel: Finde trennende Hyperebene
- Kodierung der 2 Klassen:

$$y_i = \begin{cases} 1 & \text{, Klasse 1} \\ -1 & \text{, Klasse 2} \end{cases}$$

- Punkt x_i in Klasse 1 falsch-klassifiziert $\Rightarrow x_i^T \beta + \beta_0 < 0$
  Punkt x_i in Klasse 2 falsch-klassifiziert $\Rightarrow x_i^T \beta + \beta_0 > 0$
- Ziel: minimiere den Fehler mit $x_i \notin L = \{x : \beta_0 + \beta^T x = 0\}$

$$D(\beta, \beta_0) = -\sum_{i \in \mathcal{M}} y_i (x_i^T \beta + \beta_0)$$

$\mathcal{M}$: Menge der falschklassifizierten Punkte

Rosenblatt's Perzeptron-Algorithmus (1958)

- **Idee**: Gradientenverfahren (Annahme: $\mathcal{M}$ fest)

$$\partial \frac{D(\beta, \beta_0)}{\partial \beta} = -\sum_{i \in \mathcal{M}} y_i x_i \qquad \partial \frac{D(\beta, \beta_0)}{\partial \beta_0} = -\sum_{i \in \mathcal{M}} y_i$$

- **Iteration**:

$$\begin{pmatrix} \beta \\ \beta_0 \end{pmatrix} \leftarrow \begin{pmatrix} \beta \\ \beta_0 \end{pmatrix} + \rho \begin{pmatrix} y_i x_i \\ y_i \end{pmatrix}$$

ρ: Lernrate (Schrittweite)

Rosenblatt's Perzeptron-Algorithmus (1958)

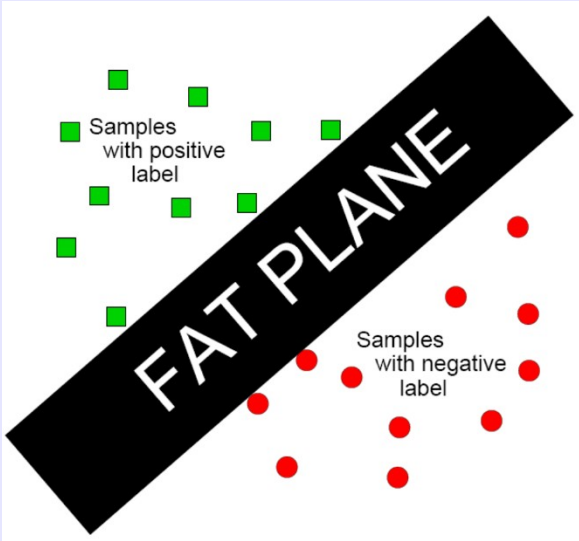
- Klassen linear trennbar
  $\Rightarrow$ Algorithmus konvergiert in endlichen Schritten gegen eine trennende Hyperebene

- Probleme mit diesem Algorithmus: (Ripley (1996))
  - Daten trennbar
    $\Rightarrow$ viele Lösungen, jeweils abhängig vom Startwert
  - „endliche“ Anzahl an Schritten kann sehr groß sein
  - Je kleiner der Abstand zwischen den Klassen, desto länger dauert es die Entscheidungsgrenze zu finden.
  - Daten nicht trennbar
    $\Rightarrow$ Algorithmus konvergiert nicht; Zyklen entstehen

Inhaltsverzeichnis

Optimale Hyperebenentrennung

- Teilt zwei Klassen und maximiert die Breite des Trennungsstreifens zwischen den Klassen (Vapnik, 1996)
- $\Rightarrow$ bessere Performance bei der Klassifizierung
- bisher: Lösung nicht immer eindeutig



Optimale Hyperebenentrennung

- **verallgemeinertes Optimierungsproblem**

$$\max_{\beta,\beta_0,\|\beta\|=1} C \quad \text{mit NB: } y_i(x_i^T\beta+\beta_0) \geq C, \quad i=1,\ldots,N$$

- äquivalent: $(\|\beta\| = \frac{1}{C})$

$$\min_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2 \quad \text{mit NB: } y_i(x_i^T\beta+\beta_0) \geq 1, \quad i=1,\ldots,N$$

- Wähle β, β_0 so, dass Breite des Trennungsstreifen maximal
  $\Rightarrow$ konvexes Optimierungsproblem

Lösung des Optimierungsproblems

- Lagrangefunktion:

$$\min_{\beta,\beta_0} L_P = \min_{\beta,\beta_0} \left\{ \frac{1}{2} \|\beta\|^2 - \sum_{i=1}^{N} \lambda_i [y_i(x_i^T \beta + \beta_0) - 1] \right\}$$

- Ableitungen $= 0$:

$$\Rightarrow \beta = \sum_{i=1}^{N} \lambda_i y_i x_i \qquad 0 = \sum_{i=1}^{N} \lambda_i y_i$$

- Einsetzen in L_P $\Rightarrow$ sog. Wolfe-Dual

$$\max L_D = \max \left\{ \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} \lambda_i \lambda_k y_i y_k x_i^T x_k \right\} \quad \text{mit: } \lambda_i \geq 0$$

Lösung des Optimierungsproblems

- zusätzlich: Kuhn-Tucker-Bedingung:

$$\lambda_i[y_i(x_i^T\beta + \beta_0) - 1] = 0 \quad \forall i$$

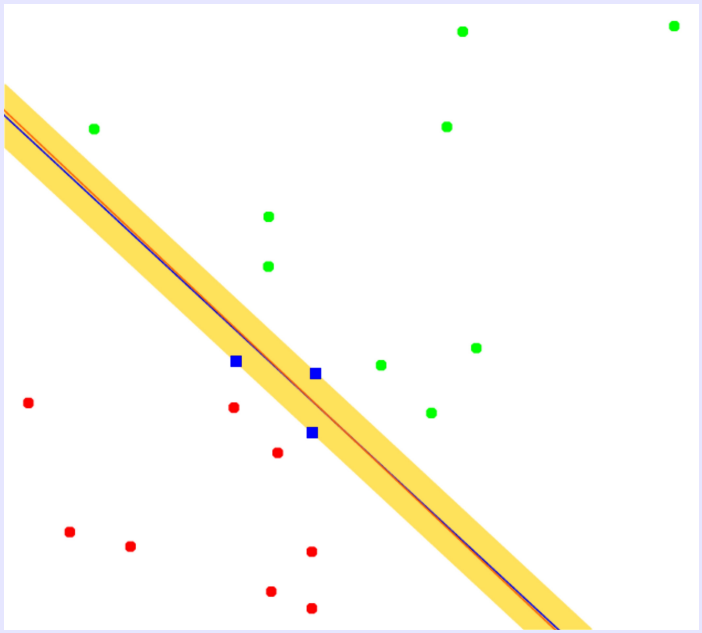
$\Rightarrow \lambda_i > 0 \Rightarrow x_i$ liegt auf dem Rand des Trennungsstreifen
$\Rightarrow \lambda_i = 0 \Rightarrow x_i$ liegt nicht auf dem Rand

- **Klassifizierung neuer Beobachtungen**:

$$\widehat{G}(x) = \text{sign } \widehat{f}(x)$$

mit $\widehat{f}(x) = x^T\widehat{\beta} + \widehat{\beta}_0$

Einfaches Beispiel mit 2 Klassen



gelber Bereich: maximaler Trennungsstreifen

blaue Linie: optimal trennende Hyperebene (halbiert den Trennungsstreifen)

Fragen

Vielen Dank für eure Aufmerksamkeit