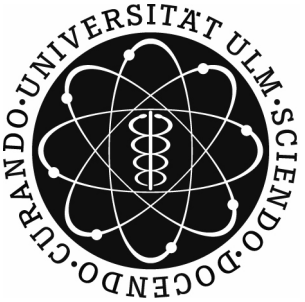
Universität Ulm
Fakultät für Mathematik und
Wirtschaftswissenschaften



Lineare Klassifikationsmethoden

Statistische Lerntheorie und ihre Anwendungen

Seminararbeit
in dem Institut für Stochastik

Prüfer: Prof. Dr. U. Stadtmüller, Prof. Dr. E. Spodarev
Betreuer: Msc Wolfgang Karcher

vorgelegt von:
Name, Vorname: Krieg, Verena

Abgabetermin: 20. Juli 2007

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

In meiner Seminararbeit über „Lineare Klassifikationsmethoden“ will ich einen Einblick in dieses Teilgebiet der statistischen Lerntheorie und deren Anwendungen geben. Dabei habe ich mich an dem Buch von T. Hastie, R. Tibshirani und J. Friedman mit dem Titel „The elements of statistical learning“ aus dem Jahre 2001 orientiert, wobei in diesem Buch mein Thema „Die linearen Klassifikationsmethoden“ im 4. Kapitel behandelt wird.

Diese Seminararbeit ist folgendermaßen aufgebaut: Nach der kleinen Einführung im zweiten Kapitel, das mit der Frage, „Was ist eigentlich „Klassifikation“?“ beginnt, diskutieren wir die einzelnen Verfahren, die für die linearen Klassifikationsmethoden in Betracht kommen.

Im dritten Kapitel sehen wir uns die lineare Regression an. Sie benutzt die Methode der kleinsten Fehlerquadrate, um eine Funktion aufzustellen, die die Daten in Klassen einteilt und mit der neue Beobachtungen klassifiziert werden können.

Im vierten Kapitel betrachten wir die lineare Diskriminanzanalyse, bei der die Annahme getroffen wird, dass die Daten in den einzelnen Klassen normalverteilt sind. Basierend auf dem Bayes-Theorem kann dann wie im dritten Kapitel eine Klassifikationsfunktion bestimmt werden.

Im fünften Kapitel dieser Seminarabeit wenden wir unsere Aufmerksamkeit auf die logistische Regression, bei der die Posterior-Wahrscheinlichkeiten der einzelnen Klassen als lineare Funktionen der Ausgangsdaten modelliert werden. Zur Schätzung der Parameter für die Klassifikationsfunktion verwendet man, wie bereits bei der linearen Diskriminanzanalyse, die Maximum-Likelihood-Methode. Da diese Bestimmung bzw. Schätzung der Parameter hier nicht trivial ist, gehen wir auf diese Berechnung im ersten Unterkapitel etwas genauer ein und stellen dann im zweiten Unterkapitel einen Vergleich zur linearen Diskriminanzanalyse mit den Vor- und Nachteilen der jeweiligen Verfahren an.

Im sechsten Kapitel liegt unsere Aufmerksamkeit auf der Hyperebenentrennung. Zunächst betrachten wir im ersten Unterkapitel den Perzeptron-Algorithmus von Rosenblatt. Dieser beruht auf dem Gradientenverfahren und sucht eine trennende Hyperebene zwischen den Klassen, die den Fehler zwischen den falsch-klassifizierten Punkten und der Klassifikationsfunktion

minimiert. Im zweiten Unterkapitel möchten wir, aufbauend auf Unterkapitel eins, zu einer optimalen Hyperebenentrennung gelangen. Hier wird durch das Lösen eines Optimierungsproblemes die Breite des Trennungsstreifens zwischen den Klassen maximiert. Wir werden am Ende sehen, dass die optimal trennende Hyperebene dann in der Mitte des Trennungsstreifens liegt.

Die Schlussfolgerungen mit Ergebnissen und kurzer Zusammenfassung sind dann im siebten Kapitel zu finden.

2 Einführung in die lineare Klassifikation

Beginnen wir nun mit der kleinen Einführung und mit der Frage: „Was ist eigentlich „Klassifikation“?“. Unsere zugrunde liegende Lerntheorie ist das „überwachte Lernen“. Wir haben Ausgangsdaten $X = (X_1, \ldots, X_p)$ der Dimension p gegeben und erhalten somit die Lerndaten (x_i, g_i) für $i = 1, \ldots, N$, wobei N die Anzahl der Beobachtungen ist und $X(\omega_i) = x_i$ gilt. Somit stellt $x_i = (x_{1i}, \ldots, x_{pi})$ einen p-dimensionalen Vektor dar und $G(x_i) = g_i$ eine Variable für die Klassenbezeichnung, die durch die Inputfunktion (Klassifikationsfunktion) G(x) auf der diskreten Menge $\mathcal{G}$ definiert ist. $G^{-1}(\cdot)$ teilt dann den Ausgangsraum in Regionen ein und jede Region $G^{-1}(g_i)$ wird nach der Klasse g_i benannt. Wir nehmen nun an, dass die Dimension von $\mathcal{G}$ gleich K ist, dass es also K Klassen gibt und dass wir Entscheidungsgrenzen zwischen diesen Klassen finden können. Diese Entscheidungsgrenzen sind abhängig von der Inputfunktion und sollten in unserem Fall linear sein.

Zur Motivation folgende Grafik:

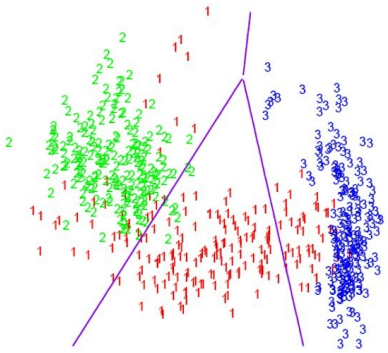


Abbildung 1: lineare Entscheidungsgrenzen durch LDA

Dieses Bild zeigt uns drei Klassen (rot (1), grün (2) und blau (3)) mit den linearen Entscheidungsgrenzen, die durch die lineare Diskriminanzanalyse gefunden wurden. Mit Hilfe dieser Entscheidungsgrenzen lassen sich dann neue Beobachtungen in die entsprechenden Klassen einteilen.

Nun stellt sich aber die Frage, wie man auf diese Entscheidungsgrenzen bzw. auf die zugehörige Klassifikationsfunktion kommt. Wie bereits oben erwähnt, gibt es dafür mehrere Methoden, die wir uns in den nächsten Kapiteln ansehen werden.

3 Lineare Regression

In diesem Kapitel betrachten wir nun die lineare Regression. Sie ist ein statistisches Analyseverfahren, um einen Datensatz der Form (x_i^*, y_i), $i = 1, \ldots, N$, durch ein lineare Funktion möglichst genau anzunähern. Hierbei stellen die x_i^* einen Vektor der Dimension p+1 dar mit $x_i^* = (1, x_i^T) = (1, x_{i1}, \ldots, x_{ip})$. y_i ist ein Vektor der Dimension K mit $y_i = (y_{i1}, \ldots, y_{iK})$.

Somit haben wir folgendes Modell gegeben:

$$Y = X\beta + \epsilon \qquad \text{bzw.}$$

$$\begin{pmatrix} y_{11} & \dots & y_{1K} \\ \vdots & & \vdots \\ y_{N1} & \dots & y_{NK} \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \dots & x_{1p} \\ \vdots & \vdots & & \vdots \\ 1 & x_{N1} & \dots & x_{Np} \end{pmatrix} \begin{pmatrix} \beta_{11} & \dots & \beta_{1K} \\ \vdots & & \vdots \\ \beta_{(p+1)1} & \dots & \beta_{(p+1)K} \end{pmatrix} + \begin{pmatrix} \epsilon_{11} & \dots & \epsilon_{1K} \\ \vdots & & \vdots \\ \epsilon_{N1} & \dots & \epsilon_{NK} \end{pmatrix}$$

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

Mit Y wird die Ergebnismatrix der N Beobachtungen und der K Klassen bezeichnet, die Designmatrix X besteht aus p Spalten für die Inputs der N Beobachtungen und einer zusätzlich Anfangsspalte für den Intercept, die aus Einsen besteht. ϵ stellt den Fehler dar, unter dem β geschätzt werden soll, so dass die Y-Werte möglichst genau angenommen werden. β ist also auch wieder eine Matrix mit p+1 Zeilen (Input und Intercept) und N Spalten für die Beobachtungen. Die Lösung für β erhält man mit Hilfe der Methode der kleinsten Quadrate.

3.1 Lineare Regression einer Indikatormatrix

Wir betrachten nun unsere diskrete Menge $\mathcal{G}$ und nehmen an, dass diese Menge $\mathcal{G}$ K Klassen besitzt. Für die Ergebnismatrix definieren wir uns K Klassenindikatoren Y_{ik}, die 1 sind, wenn G(x)=k ist, und ansonsten 0 sind (für $i = 1, \dots, N,\ k = 1, \dots, K$). Diese Indikatoren werden für jedes i zu einem Vektor $Y_i = (Y_{i1}, \dots, Y_{iK})$ zusammengefasst und dann wird die Ergebnismatrix $Y = (Y_1, \dots, Y_N)^T$ der N Beobachtungen aufgestellt. Y ist eine N×K Matrix, die aus Nullen und Einsen besteht, wobei in jeder Zeile nur eine Eins stehen darf. Somit erhalten wir folgende Schätzung[1]:

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

[1] vgl. [1], S.81

3.2 Klassifikation

Um eine neue Beobachtung einer Klasse zuordnen zu können, bestimmen wir zunächst den Schätzer $\widehat{\beta}$ mit

$$\widehat{\beta} = (X^T X)^{-1} X^T Y \text{ mit } X \in \mathbb{R}^{N \times (p+1)}$$

Mit Hilfe dieses Schätzers wird die lineare Entscheidungsfunktion aufgestellt: Eine neue Beobachtung mit Input x wird dann klassifiziert, indem man zuerst den geschätzten Output

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

bestimmt, dann die größte Komponente von $\widehat{Y}$ bestimmt und schließlich mit

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

klassifiziert.

Um dieses Verfahren zu veranschaulichen, folgt nun ein Beispiel.

3.3 Beispiel: Lineare Regression

Zuerst simulieren wir einen Datensatz mit zwei Klassen aus jeweils 1000 Zufallsvariablen unter folgenden Annahmen:

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

Wir treffen diese Annahmen, da sie später im Beispiel für die linearen Diskriminanzanalyse benötigen werden, um diese Klassifikationsmethode mit einem geringeren Aufwand durchführen zu können. Mit dem soeben erzeugten Datensatz wird dann die Designmatrix X aufgestellt, wobei die erste Spalte aus Einsen für den Intercept β_0 besteht, und die zweite und dritte Spalte aus den simulierten Daten. Anschließend wird für die 2000 Beobachtungen die Matrix Y aufgestellt. Da hier die Daten selbst simuliert wurden und wir die ersten 1000 Zufallsvariablen für die Klasse 1 simuliert

haben, stehen in den ersten 1000 Einträgen der ersten Spalte von Y Einsen und sonst Nullen und da die zweiten 1000 Zufallsvariablen der Klasse 2 angehören, stehen in den zweiten 1000 Einträgen der zweiten Spalte von Y Einsen und sonst Nullen. Mit diesen Informationen kann dann $\widehat{\beta}$ berechnet werden.

In Zahlen:

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

Die Entscheidungsgrenze zwischen den beiden Klassen erhalten wir nun durch das Gleichsetzen der Regressionsgeraden der beiden Klassen, die durch die $\widehat{\beta}$ beschrieben werden, da dann $\widehat{Y}_1 = \widehat{Y}_2$ gilt und somit eine Zuordnung sowohl zur ersten als auch zur zweiten Klasse möglich ist:

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

x_1 entspricht der zweiten Spalte von X. Nach dem Einsetzen der geschätzten Werte von $\widehat{\beta}$ lösen wir nach x_2 auf und erhalten die Entscheidungsgerade

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

Die nachfolgende Grafik zeigt uns den Plot der simulierten Daten. Mit blau wird die Klasse 1 gekennzeichnet, mit rot die Klasse 2, mit schwarz die Entscheidungsgrenze.

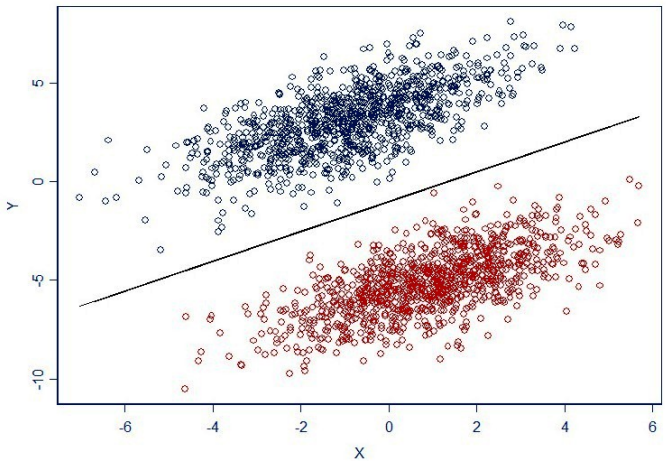


Abbildung 2: Beispiel: lineare Regression

Für die simulierten Ausgangsdaten erhalten wir hier keine Falsch-Klassifizierungen, was jedoch - auch für neue Beobachtungen - durchaus vorkommen kann. Eine neue Beobachtung wird dann aufgrund ihrer Lage - ober- oder unterhalb der Entscheidungsgrenze - klassifiziert. Liegt sie oberhalb, wird sie zur Klasse 1 zugeordnet, unterhalb zur Klasse 2.

3.4 Problem bei der linearen Regression

Bei dieser Methode kommt es jedoch zu einem Problem, wenn die Anzahl der Klassen gleich drei oder größer ist. Dies wird mit den beiden Grafiken auf der folgenden Seite veranschaulicht.

Wir sehen dort das Problem mit drei Klassen. Im linken Bild in Abbildung 3 wurde die lineare Regression verwendet und im rechten die lineare Diskriminanzanalyse. Beide Male wurden die zwei äußeren Klassen perfekt durch lineare Entscheidungsgrenzen getrennt, allerdings gibt es bei der linearen Regression ein Problem mit der mittleren Klasse. Diese wird komplett vergessen.

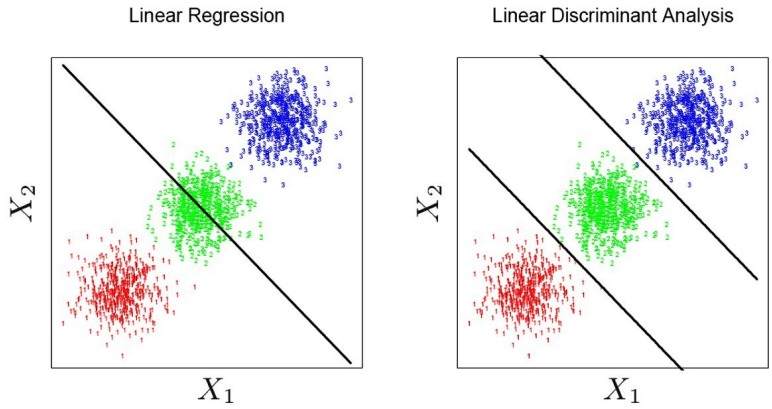


Abbildung 3: Lineare Regression vs. Lineare Diskriminanzanalyse

Wieso wird die mittlere Klasse vergessen? Man erstellt zunächst einen sogenannten Rugplot der Daten, indem man eine Gerade senkrecht zur Entscheidungsgrenze legt und die Daten auf diese Gerade projiziert (Abbildung 4, Strichmuster auf der x-Achse). Anhand diesem Muster kann dann abgelesen werden, in welchem Bereich welche Klasse anzutreffen sein sollte. Anschließend zeichnet man die zu den projizierten Daten gehörenden Regressionsfunktionen (Geraden/Kurven) in das Schaubild ein und kann anhand diesen dann feststellen, welche Klasse in welchem Bereich regressiert wird.

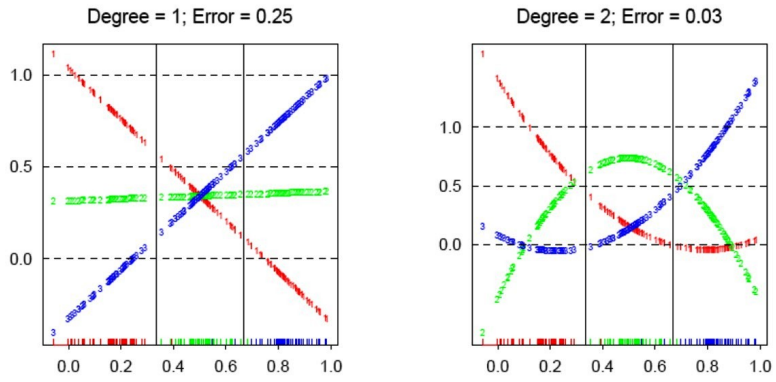


Abbildung 4: Ruplot und Regressionsfunktionen

Bei der linearen Regression (links) erhalten wir Geraden. Für die mittlere Klasse 2 erhalten wir die grüne Gerade, die nahezu horizontal verläuft und somit nie dominant ist, aufgrund dessen die Klasse 2 nicht als eigenständige

Klasse erkannt wird. Deswegen werden Beobachtungen der Klasse 2 entweder zur Klasse 1 oder zur Klasse 3 eingeteilt. Bei der linearen Diskriminanzanalyse (rechts) ergeben sich für die Regressionsfunktionen Polynome mit Rang größer gleich 2, und man sieht, dass nun auch die Kurve für die Klasse 2 dominant ist, also werden alle drei Klassen erkannt. Somit kann eine deutlich bessere Klassifikation neuer Beobachtungen erwartet werden.

4 Lineare Diskriminanzanalyse

Aufgrund des Problems, das sich im dritten Kapitel herauskristallisiert hat, wollen wir uns nun die lineare Diskriminanzanalyse ansehen und erhoffen uns dadurch eine bessere Performance und auch die Möglichkeit, die Datenmenge in mehr als zwei Klassen einteilen zu können.

Für die lineare Diskriminanzanalyse benötigen wir jedoch weitere Annahmen über unsere Lerndaten. Zum Einen benötigen wir die bedingte Dichte von X der jeweiligen Klasse k, die wir mit $f(x|k)$ bezeichnen, und die A-Prioriwahrscheinlichkeiten p_k der Klasse k, für die zusätzlich gelten muss, dass sie in Summe Eins ergeben, also $\sum_{k=1}^{K} p_k = 1$. Mit diesen Voraussetzungen können wir dann das Bayes-Theorem anwenden und wir erhalten folgenden Gleichung:

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

Somit können wir die bedingte Wahrscheinlichkeit, dass G=k unter der Bedingung, dass die Beobachtung mit Input X=x ist, durch die bedingte Dichte $f(x|k)$ und die A-Prioriwahrscheinlichkeiten p_k darstellen.

Bei der linearen Diskriminanzanalyse nimmt man für die Klassendichten die multivariate Normalverteilung an, mit dem Spezialfall, dass die Kovarianzmatrizen für alle Klassen gleich sind ($\Sigma_k = \Sigma \ \ \forall k$). Die bedingte Dichte sieht also wie folgt aus:

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

Mit μ_k sind die klassenbedingten Erwartungswerte bezeichnet. Die Kovarianzmatrizen und die klassenbedingten Erwartungswerte müssen aus der

Datenmenge geschätzt werden.

4.1 Klassifikation

Um eine neue Beobachtung einer Klasse zuordnen zu können, müssen wir hier allerdings noch einige Vorüberlegungen anstellen. Gehen wir nun zunächst von einem paarweisen Vergleich der Klassen aus und beginnen diesen mit zwei Klassen, die mit k und l bezeichnet werden und anschließend mit der Idee, dass eine neue Beobachtung mit Input x vorläufig der Klasse k zugeordnet wird, wenn die bedingte Wahrscheinlichkeit der Klasse k größer ist als die bedingte Wahrscheinlichkeit der Klasse l, also wenn

$$P(G = k|X = x) > P(G = l|X = x)$$

Andernfalls wird die Beobachtung vorläufig der Klasse l zugeordnet. Um diese Ungleichung in einer linearen Form darstellen zu können, wendet man den Trick mit dem Logarithmus an:

$$\frac{P(G = k|X = x)}{P(G = l|X = x)} > 1 \quad \Leftrightarrow \quad \log \frac{P(G = k|X = x)}{P(G = l|X = x)} > 0$$

$$\begin{aligned}
\log \frac{P(G = k|X = x)}{P(G = l|X = x)} &= \log \frac{f_k(x)}{f_l(x)} + \log \frac{p_k}{p_l} \\
&= \log \frac{p_k}{p_l} + \log \frac{\frac{1}{(2\pi)^{p/2}\ |\Sigma|^{1/2}}\ e^{-\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k)}}{\frac{1}{(2\pi)^{p/2}\ |\Sigma|^{1/2}}\ e^{-\frac{1}{2}(x-\mu_l)^T\Sigma^{-1}(x-\mu_l)}} \\
&= \log \frac{p_k}{p_l} + \log e^{-\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k)} - \log e^{-\frac{1}{2}(x-\mu_l)^T\Sigma^{-1}(x-\mu_l)} \\
&= \log \frac{p_k}{p_l} - \frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k) + \frac{1}{2}(x-\mu_l)^T\Sigma^{-1}(x-\mu_l) \\
&= \log \frac{p_k}{p_l} - \frac{1}{2}\ [x^T\Sigma^{-1}x - 2x^T\Sigma^{-1}\mu_k + \mu_k^T\Sigma^{-1}\mu_k] \\
&\quad + \frac{1}{2}\ [x^T\Sigma^{-1}x - 2x^T\Sigma^{-1}\mu_l + \mu_l^T\Sigma^{-1}\mu_l] \\
&= \log \frac{p_k}{p_l} + x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k - x^T\Sigma^{-1}\mu_l + \frac{1}{2}\mu_l^T\Sigma^{-1}\mu_l \\
&= \log \frac{p_k}{p_l} + x^T\Sigma^{-1}(\mu_k - \mu_l) - \frac{1}{2}\ [\mu_k^T\Sigma^{-1}\mu_k - \mu_l^T\Sigma^{-1}\mu_l] \\
&= \log \frac{p_k}{p_l} - \frac{1}{2}(\mu_k + \mu_l)^T\Sigma^{-1}(\mu_k - \mu_l) + x^T\Sigma^{-1}(\mu_k - \mu_l)
\end{aligned}$$

Der letzte Ausdruck ist linear in x. Diese Gleichung gilt nun für alle Klassenpaare und somit können die linearen Entscheidungsgrenzen gefunden werden, indem man diese =0 setzt:

$$\delta_k(x) = \delta_l(x)$$

wobei

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

die **Lineare Diskriminanzfunktion** genannt wird.

Beweis:

$$\log\frac{P(G=k|X=x)}{P(G=l|X=x)} > 0$$
$$\log\frac{p_k}{p_l} - \frac{1}{2}(\mu_k+\mu_l)^T\Sigma^{-1}(\mu_k-\mu_l) + x^T\Sigma^{-1}(\mu_k-\mu_l) > 0$$
$$\log p_k - \log p_l - \frac{1}{2}[\mu_k^T\Sigma^{-1}\mu_k - \mu_l^T\Sigma^{-1}\mu_l] + x^T\Sigma^{-1}\mu_k - x^T\Sigma^{-1}\mu_l > 0$$
$$\log p_k - \frac{1}{2}[\mu_k^T\Sigma^{-1}\mu_k - \mu_l^T\Sigma^{-1}\mu_l] + x^T\Sigma^{-1}\mu_k > \log p_l - \frac{1}{2}\mu_l^T\Sigma^{-1}\mu_l + x^T\Sigma^{-1}\mu_l$$
$$\Rightarrow \delta_k(x) > \delta_l(x)$$

Mit dieser linearen Diskriminanzfunktion $\delta_k(x)$ können wir dann die Klassifikationsregel aufstellen:

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

Mit anderen Worten: Man berechnet für die neue Beobachtung mit Input x die Diskriminanzfunktionen jeder einzelnen Klasse und teilt x der Klasse zu, bei der der Wert der Diskriminanzfunktion am Größten ist.

4.2 Beispiel: Lineare Diskriminanzanalyse

Wie im Beispiel für die Lineare Regression simulieren wir einen Datensatz mit zwei Klassen aus jeweils 1000 Zufallsvariablen unter den folgenden Annahmen für die Kovarianzmatrix und die klassenbedingten Erwartungswerte, da wir uns somit die Schätzung dieser Größen und den damit verbundenen

Aufwand sparen können.

Annahmen:

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

$$\log p_1 = \log p_2 = 0.5$$

Mit diesen Daten können wir die linearen Diskriminanzfunktionen der beiden Klassen aufstellen und erhalten die Entscheidungsgrenze durch das Gleichsetzen dieser Funktionen, wobei x_1 wieder die zweite Spalte von X (aus der Linearen Regression) ist und nach x_2 aufgelöst wird.

Berechnung der Entscheidungsgrenze durch:

$$\begin{aligned}
&\log \frac{P(G=1|X=x)}{P(G=2|X=x)} = 0 \qquad \Leftrightarrow \qquad \delta_1(x) = \delta_2(x) \\
&\Leftrightarrow x^T\Sigma^{-1}\mu_1 - \frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 + \log p_1 = x^T\Sigma^{-1}\mu_2 - \frac{1}{2}\mu_2^T\Sigma^{-1}\mu_2 + \log p_2 \\
&\Leftrightarrow \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}^T \begin{pmatrix} 3 & 2 \\ 2 & 3 \end{pmatrix}^{-1} \begin{pmatrix} -1 \\ 3 \end{pmatrix} - \frac{1}{2} \begin{pmatrix} -1 \\ 3 \end{pmatrix}^T \begin{pmatrix} 3 & 2 \\ 2 & 3 \end{pmatrix}^{-1} \begin{pmatrix} -1 \\ 3 \end{pmatrix} + 0.5 = \\
&\qquad = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}^T \begin{pmatrix} 3 & 2 \\ 2 & 3 \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ -5 \end{pmatrix} - \frac{1}{2} \begin{pmatrix} 1 \\ -5 \end{pmatrix}^T \begin{pmatrix} 3 & 2 \\ 2 & 3 \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ -5 \end{pmatrix} + 0.5 \\
&\Leftrightarrow \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}^T \begin{pmatrix} -1.8 \\ 2.2 \end{pmatrix} - \frac{1}{2}\ 8.4 = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}^T \begin{pmatrix} 2.6 \\ -3.4 \end{pmatrix} - \frac{1}{2}\ 19.6 \\
&\Leftrightarrow -1.8\ x_1 + 2.2\ x_2 - 4.2 = 2.6\ x_1 - 3.4\ x_2 - 9.8 \\
&\Leftrightarrow 5.6\ x_2 = -5.6 + 4.4\ x_1 \\
&\Leftrightarrow x_2 = -1 + 0.79
\end{aligned}$$

Die folgende Grafik auf der nächsten Seite zeigt uns den Plot der simulierten Daten. Mit blau wird die Klasse 1 bezeichnet, und mit rot die Klasse 2, mit schwarz die Entscheidungsgrenze.

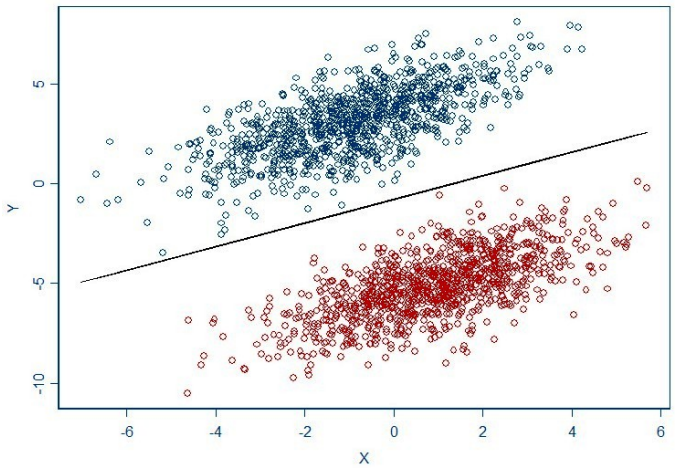


Abbildung 5: Beispiel: lineare Diskriminanzanalyse

Für die simulierten Daten erhalten wir auch hier wieder keine Falsch-Klassifizierungen und durch genaues Vergleichen der beiden Plotte (Abbildung 2 und 5) kann man eine Verbesserung der Entscheidungsgrenze bei der linearen Diskriminanzanalyse sehen. Diese fällt hier nun aber sehr gering aus und es liegt im Ermessen des einzelnen Beobachters, ob er die Veränderung der Geraden als gut oder schlecht bewertet. Dies hängt mit der Wahl der Datenmenge zusammen. Aufgrund des hier höheren Aufwandes, da eigentlich die Annahmen noch geschätzt werden müssten, würde man bei solcher Datenmengenstruktur die Lineare Regression vorziehen.

4.3 Quadratische Diskriminanzanalyse

Bisher haben wir nur den Spezialfall der multivariatnormalverteilten bedingten Dichte angesehen, bei der die Kovarianzmatrizen für jede Klasse gleich sind. Nun wollen wir uns ansehen, was passiert, wenn dies nicht der Fall ist. Die Σ_k sind nun nicht alle gleich und müssen somit für jede einzelne Klasse geschätzt werden. Durch entsprechende Überlegungen wie bei der linearen Diskriminanzanalyse erhalten wir eine nun quadratische Diskriminanzfunktionen

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

mit der Klassifikationsregel

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

Die jetzigen Entscheidungsgrenzen ($\delta_k(x) = \delta_l(x)$) sind nun aber nicht mehr linear in x, sondern quadratisch (siehe den zweiten Term von $\delta_k(x)$). Dies entspricht aber nicht einer linearen Lösung des Klassifikationsproblems und somit auch nicht der Lösungsmenge meines Seminarthemas. Aufgrund dessen werde ich nun hier nicht weiter auf dieses Verfahren eingehen.

5 Logistische Regression

In diesem Kapitel betrachten wir nun ein weiteres Verfahren, mit dem man lineare Entscheidungsgrenzen für die Klassifikation von Daten finden kann. Nachdem wir das Verfahren vorgestellt haben und uns auch hier ein Beispiel angesehen haben, werden wir uns mit den Vor- und Nachteilen dieses Verfahrens beschäftigen und einen Vergleich zur eben behandelten linearen Diskriminanzanalyse anstellen.

Bei diesem sehr verallgemeinerten Verfahren, der logistischen Regression, wollen wir die Posterior-Wahrscheinlichkeiten $P(G = k|X = x)$ der K Klassen mittels linearer Funktionen in x darstellen. Um das Modell ein bisschen vereinfacht darzustellen, werden wir im Folgenden die bedingte Wahrscheinlichkeit mit $P_k(x;\theta) = P(G = k|X = x)$ bezeichnen. Zudem benötigen wir wieder die A-Prioriwahrscheinlichkeit p_k der Klasse k, die in Summe eins ergeben muss ($\sum_{k=1}^{K} p_k = 1$).

5.1 Klassifikation

Um eine neue Beobachtung einer Klasse zuordnen zu können, benötigen wir wieder eine Entscheidungsgrenze, die wir erneut durch das Gleichsetzen der bedingten Wahrscheinlichkeiten der Klassen erhalten.

$$P(G = k|X = x) = P(G = l|X = x)$$

Das Modell der logistischen Regression erhält man dann durch den Vergleich der Klassen $1, \ldots, K-1$ mit der Klasse K. Dieses Modell sieht dann

folgendermaßen aus:

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

Um dieses Modell lösen zu können, muss die Parametermenge $\theta = \{\beta_{10}, \beta_1^T, \ldots, \beta_{(K-1)0}, \beta_{K-1}^T\}$ geschätzt werden. Durch diese geschätzten β's können dann die linearen Entscheidungsgrenzen aufgestellt werden.

5.2 Berechnung der Parameter der logistischen Regression

Dieses Unterkapitel befasst sich mit der konkreten Berechnung der Parameter, die für die logistische Regression benötigt werden. Wie wir sehen werden, ist diese Berechnung nicht ganz einfach.

Die zugrunde liegende Idee ist, die Parameter, die die bedingte Likelihoodfunktion von G bei gegebenem X maximiert, zu finden. Dazu benötigen wir zunächst die Likelihoodfunktion.

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

Aus dieser Funktion erhalten wir dann die log-Likelihoodfunktion der N Beobachtungen mit

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

Wir betrachten hier die log-Likelihoodfunktion, da wir später für die Maximierung die Ableitungen bestimmen müssen und dies bekanntlich für Summen einfacher als für Produkte ist. Zudem spielt beim Maximieren das Anwenden des Logarithmus keine Rolle, da der Logarithmus eine streng monoton wachsende Funktion auf $(0, \infty)$ ist.

Betrachten wir nun zur weiteren Vereinfachung den Spezialfall mit zwei Klassen, der dann analog auf mehrere Klassen erweitert werden kann. Die zwei Klassen werden durch den Indikator y_i bestimmt, der eins ist, falls der

Punkt zur Klasse 1 gehört und null, falls er zur Klasse 2 gehört.

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

Mit

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

erhalten wir die log-Likelihoodfunktion:

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

In dieser Funktion entspricht β einem Vektor der zu schätzenden Parameter mit $\beta = (\beta_{10}, \beta_1)$ und dem Vektor x_i mit einem konstantem Term 1 für β_{10}. Da nun diese β's geschätzt werden müssen, für die die log-Likelihoodfunktion maximal ist, setzen wir den Gradienten

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

gleich Null und erhalten dann p+1 Gleichungen, die nicht-linear in β sind. Dieses Gleichungssystem muss nun gelöst werden. Für die erste Komponente von x_i wissen wir, dass sie „$= 1$“ für β_{10} ist, also folgt daraus direkt, dass

$$\sum_{i=1}^{N} y_i = \sum_{i=1}^{N} P(x_i;\beta)$$

Aufgrund dieser Eigenschaft wird das restliche Gleichungssystem übersichtlicher und kann mit Hilfe des Newton-Verfahrens gelöst werden. Für dieses Verfahren benötigt man jedoch noch die zweite Ableitung der log-Likelihoodfunktion, die auch Hessematrix genannt wird.

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

Um die eigentliche Newton-Iteration durchführen zu können, muss zu Beginn

der Startwert β^{alt} gewählt werden und im ersten Schritt alle Ableitungen an diesem Startpunkt ausgewertet werden.

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

Die weiteren Schritte laufen analog zum ersten Schritt ab, mit dem Unterschied, dass das gerade gefundene β^{neu} zu β^{alt} wird und auch die Ableitungen an diesem β^{neu} ausgewertet werden müssen. Das Verfahren konvergiert, da die log-Likelihoodfunktion konkav ist, allerdings kann das Problem des sogenannten overshooting auftreten.

5.3 Beispiel: Logistische Regression

Wie beim Beispiel für die lineare Regression und die lineare Diskriminanzanalyse betrachten wir den simulierten Datensatz mit den zwei Klassen aus jeweils 1000 Zufallsvariablen. Die Matrix X entspricht hier exakt der Matrix für X aus der linearen Regression. Für den Startwert wählen wir $\beta^{alt} = 0$. Für die Klasse 1 sind die Einträge in Y mit dem Wert Eins versehen, für die Klasse 2 mit Null, also sieht unser Y hier so aus:

$$Y = (\underbrace{1, \ldots, 1}_{\text{1.Klasse}}, \underbrace{0, \ldots, 0}_{\text{2.Klasse}})^T$$

Mit diesen Daten können wir nun unsere log-Likelihoodfunktion aufstellen und die Optimierung in S-Plus durchführen lassen. Die Lösung dieser Optimierung liefert uns die Schätzer für die β's, die zur Aufstellung der Entscheidungsgrenze benötigt werden.

$$(\widehat{\beta}_1, \widehat{\beta}_2, \widehat{\beta}_3)^T = (43.2507, 39.11790, -27.07334)^T$$

Somit ergibt sich die Entscheidungsgrenze zu:

$$\begin{aligned}
&\log \frac{P(G=1|X=x)}{P(G=2|X=x)} = \widehat{\beta}_1 + \widehat{\beta}_2 x_1 + \widehat{\beta}_3 x_2 = 0 \\
&\Leftrightarrow x_2 = -\frac{\widehat{\beta}_1}{\widehat{\beta}_3} - \frac{\widehat{\beta}_2}{\widehat{\beta}_3}\, x_1 \\
&\Leftrightarrow x_2 = 1.597539 + 1.444887\, x_1
\end{aligned}$$

wobei mit x_1 wieder die zweite Spalte von X bezeichnet wird.

Die nachfolgende Grafik stellt nun den Plot der simulierten Daten dar. Klasse 1 (rote Kreise), Klasse 2 (schwarze Kreise) und die schwarze Entscheidungsgrenze wurden geplottet. Hier treten nun zum ersten Mal Falsch-Klassifizierungen auf, die mit der Wahl der simulierten Daten zusammenhängen. Denn wenn wie vorliegend die gemeinsame Dichte der Klassen tatsächlich die Dichte einer multivariaten Normalverteilung ist, dann kann es im schlimmsten Fall zu einem Effizienzverlust von 30% kommen[2].

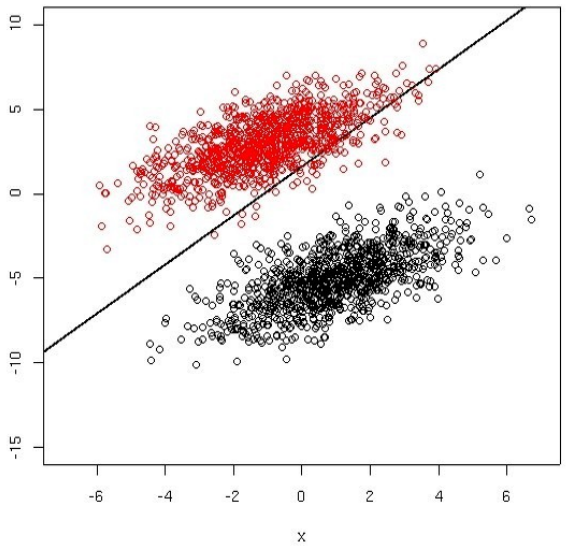


Abbildung 6: Beispiel: logistische Regression

[2] vgl. [1], S.105

5.4 Vergleich von Logistischer Regression und LDA

In diesem Unterkapitel stellen wir nun einen Vergleich zwischen der logistischen Regression und der Linearen Diskriminanzanalyse an.

Zur Erinnerung:

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:

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

Wie wir gesehen haben, lassen sich die beiden Verfahren durch sehr ähnliche lineare Funktionen ausdrücken, was auf einen Anschein der Gleichheit der Modelle schließen lässt. Dies ist aber nur ein Trugschluss, denn der wesentliche Unterschied dieser Verfahren besteht in der Schätzung der Parameter und der daraus resultierenden Koeffizienten der linearen Funktionen. Zudem wissen wir, dass das logistische Regressionsmodell allgemeiner sein muss, da dazu weniger Annahmen nötig sind als bei der linearen Diskriminanzanalyse.

Bei dem logistischen Regressionsmodell verwendet man eine beliebige Dichtefunktion und maximiert dann die bedingte Likelihoodfunktion. Ausreißer haben hier nur eine geringe Gewichtung und beeinflussen die Schätzung nur sehr wenig.

Bei der linearen Diskriminanzanalyse benötigt man die Dichtefunktion der multivariaten Normalverteilung und maximiert die log-Likelihoodfunktion. Zusätzlich müssen hier die Erwartungswerte μ_k, die A-Prioriwahrscheinlichkeiten p_k und die allgemeine Kovarianzmatrix für jede Klasse Σ geschätzt werden, was einen höheren Aufwand bedeutet, aber durch die zusätzlichen

Informationen kann die Schätzung natürlich auch genauer werden. Ausreißer spielen bei der Schätzung der allgemeinen Kovarianzmatrix eine Rolle, wodurch die lineare Diskriminanzanalyse nicht robust gegenüber großen Ausreißern ist.

Aufgrund diesen Tatsachen ist die logistische Regression robuster als die lineare Diskriminanzanalyse und sollte besonders dann verwendet werden, wenn man schon vor Anwendung der Verfahren weiß, dass Ausreißer in der Datenmenge vorhanden sind. Wenn die Höhe des Aufwandes keine Rolle spielt, und man von nur sehr wenigen und geringfügigen Ausreißern ausgehen kann, dann bietet sich verstärkt die lineare Diskriminanzanalyse an. In der Praxis zeigte sich zudem, dass beide Modelle oft sehr ähnliche Ergebnisse liefern.

6 Hyperebenentrennung

In diesem vorletzten Kapitel wird noch die Hyperebenentrennung vorgestellt. Bisher haben wir uns nur Datenmengen im zweidimensionalen Fall angesehen, weshalb uns nun dieses Verfahren als neu erscheint, doch dies ist nicht ganz der Fall. Die bisherigen Verfahren liefern im mehrdimensionalen Raum ebenfalls Hyperebenen statt Geraden.

Mit dem Verfahren der Hyperebenentrennung beschreibt man eine Prozedur, die lineare Entscheidungsgrenzen konstruiert, die explizit versuchen, die Daten so gut wie möglich in verschiedene Klassen aufzuteilen. Sie stellen eine Basis für die „Support vector machines“ dar, auf die in einem weiteren Seminar näher eingegangen wird. Wenn unsere Lerndaten durch Hyperebenen perfekt getrennt werden können, so muss das nicht gleichzeitig bedeuten, dass die lineare Diskriminanzanalyse und die anderen vorgestellten linearen Modelle auch eine perfekte Trennung für diesen Datensatz liefern.

Das nachfolgende Bild zeigt uns gerade diesen Fall:

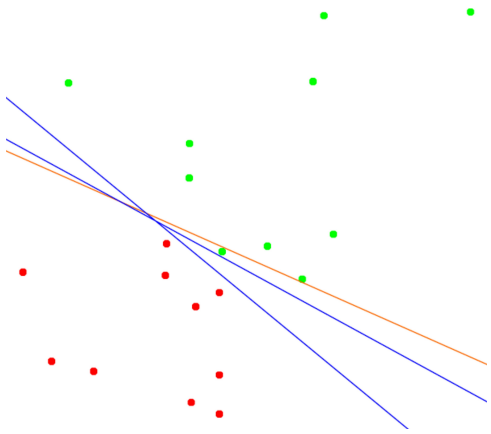


Abbildung 7: lineare Regression und Perzeptron-Algorithmus

Wir haben eine Datenmenge mit zwei Klassen, die grüne und die rote Klasse, gegeben. Die orange Linie zeigt uns die lineare Entscheidungsgrenze, die durch Anwendung der linearen Regression gefunden wurde. Hierbei sehen wir, dass ein Punkt falsch-klassifiziert wurde. Die zwei blauen Linien stellen zwei trennende Hyperebenen dar, die durch den Perzeptron-Algorithmus von Rosenblatt gefunden wurden, wobei der Algorithmus mit zwei verschiedenen Startpunkten durchgeführt wurde. Bei diesem Verfahren wurde jeweils kein Punkt falsch-klassifiziert.

6.1 Rosenblatt's Perzeptron-Algorithmus

In diesem Unterkapitel betrachten wir den gerade nebenbei erwähnten Perzeptron-Algorithmus von Rosenblatt.

Das Ziel dieses Algorithmus ist es, eine trennende Hyperebene zu finden, indem man die Abstände der falsch-klassifizierten Punkte zur Entscheidungsgrenze minimiert. Hierfür kodiert man zunächst einmal die beiden Klassen binär, man definiert also ein y_i mit

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

Zudem sagt man, ein Punkt x_i ist in der Klasse 1 falsch-klassifiziert,

wenn gilt, $x_i^T\beta + \beta_0 < 0$. Ist ein Punkt x_i in Klasse 2 falsch-klassifiziert, dann gilt entsprechend $x_i^T\beta + \beta_0 > 0$. Somit kann das Ziel dieses Verfahren definiert werden als Minimierung des Fehlers

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

, dass ein Punkt x_i, der nicht auf der Hyperebene liegt und aus der Menge $\mathcal{M}$, der Menge der falsch-klassifizierten Punkte, ist.

Aufgrund dieser Zielsetzung und diesen obigen Annahmen beruht nun der Algorithmus auf der Idee des Gradientenverfahrens mit der zusätzlichen Bedingung, dass nun die Menge der falsch-klassifizierten Punkte $\mathcal{M}$ fest ist. Zunächst müssen dann die partiellen Ableitungen

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

gebildet werden, um damit die Iteration

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

durchführen zu können, wobei mit ρ die Lernrate oder die sog. Schrittweite definiert ist.

Falls die Klassen linear trennbar sind, dann konvergiert der Algorithmus in endlichen Schritten gegen eine trennende Hyperebene. Doch falls die Klassen nicht linear trennbar sind, kann keine eindeutige Lösung gefunden werden.

6.1.1 Problem beim Rosenblatt's Perzeptron-Algorithmus

Es gibt jedoch auch Probleme mit diesem Algorithmus.

- Falls die Daten trennbar sind, kann es, wie wir in dem einfachen Beispiel schon gesehen haben, mehrere Lösungen geben, die vom Startwert abhängen.

- Zudem heißt es, der Algorithmus konvergiert in endlichen Schritten, doch wie groß ist „endlich“?

- Außerdem kann es ein Problem geben, wenn der Abstand zwischen den Klassen von vornherein sehr klein ist. Dann kann es sehr lange dauern, bis die perfekte Entscheidungsgrenze gefunden worden ist.

- Und was passiert, wenn die Daten gar nicht trennbar sind? Dann kann es dazu kommen, dass der Algorithmus gar nicht konvergiert und Zyklen entstehen, dass es also gar keine perfekte Hyperebene gibt, jedoch der Algorithmus nicht abbricht, sondern immer zwischen mehreren nicht perfekten Lösungen hin und her springt.

6.2 Optimale Hyperebenentrennung

Da wir mit dem Perzeptron-Algorithmus vom Startwert abhängige Entscheidungsgrenzen erhalten, betrachten wir in diesem Unterkapitel nun das Verfahren der optimalen Hyperebenentrennung und erhoffen uns, als Lösung eine Hyperebene zu erhalten, die die Klassen optimal trennt und eine perfekte Klassifikation neuer Beobachtungen ermöglicht.

Dieses Verfahren der optimalen Hyperebenentrennung teilt wieder den Datensatz in zwei Klassen auf und maximiert dabei die Breite des Trennungsstreifen (fat plane), der zwischen den beiden Klassen besteht. Dieses Verfahren ist natürlich nur anwendbar, wenn die Klassen auch wirklich trennbar sind, da sonst kein solcher Trennungsstreifen vorhanden wäre. Durch diese Maximierung der Breite des Trennungsstreifens wird die Performance bei der Klassifizierung verbessert.

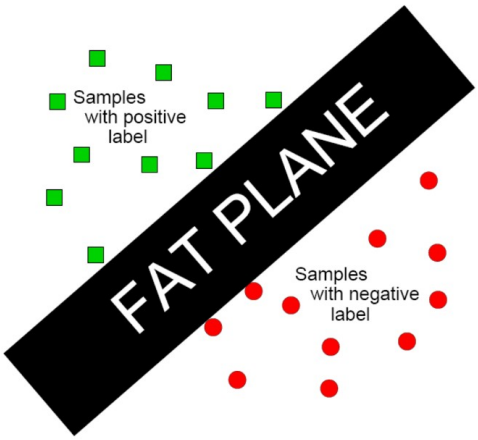


Abbildung 8: Breite des Trennungsstreifens = fat plane

Maximierung der Breite des Trennungsstreifens ist gleichzusetzen mit dem Lösen des verallgemeinerten Optimierungsproblems.

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

Wir maximieren den Abstand C von der Entscheidungsgrenze, die durch β und β_0 definiert ist, zum Punkt x_i, der nicht auf der Entscheidungsgrenze liegt, unter der Nebenbedingung, dass $y_i(x_i^T\beta+\beta_0) \geq C$. Von der Bedingung $\|\beta\| = 1$ können wir uns befreien, indem wir sie in die Nebenbedingung mit einfließen lassen. Wir können $\|\beta\| = \frac{1}{C}$ setzen, da die Nebenbedingungen dann weiterhin erfüllt sind, und da das Maximieren des Abstandes äquivalent ist zum Minimieren der Norm von $\|\beta\|^2$, können wir auch folgendes Minimierungsproblem lösen:

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

Man wählt dann β, β_0 so, dass die Breite des Trennungsstreifens maximal wird. Wegen der quadratischen Bedingung und den linearen Ungleichungen sprechen wir hier von einem konvexen Optimierungsproblem, das nun zu lösen ist.

Dazu stellen wir zunächst einmal die Lagrangefunktion auf und minimieren diese, indem wir die Ableitungen bilden und gleich Null setzen.

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

Daraus folgt dann, dass $\beta = \sum_{i=1}^N \lambda_i y_i x_i$ und $\beta_0 = \sum_{i=1}^N \lambda_i y_i$ ist. Setzt man diese Gleichungen in die Lagrangefunktion L_P ein, so erhält man das sog. Wolf-Dual, welches nun zu maximieren ist.

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

Für die Lösung müssen zusätzlich die Kuhn-Tucker-Bedingungen

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

gelten, anhand derer wir nun entscheiden können, ob x_i auf dem Rand des

Trennungsstreifen liegt ($\lambda_i > 0$) oder nicht ($\lambda_i = 0$).

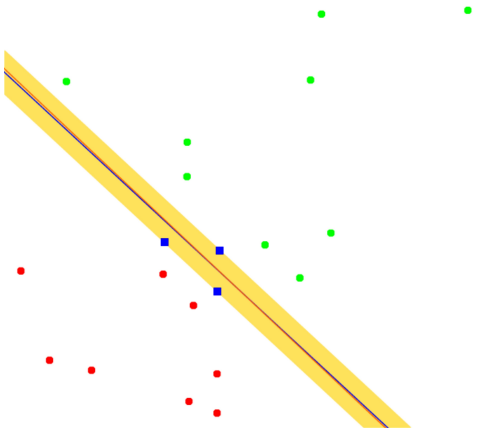
6.2.1 Klassifikation

Durch die oben beschriebene Maximierung erhält man die β's, mit denen die optimal trennende Hyperebene $\widehat{f}(x) = x^T\widehat{\beta} + \widehat{\beta}_0$ aufgespannt wird. Neue Beobachtungen mit Input x werden dann mit

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

klassifiziert.

6.2.2 Einfaches Beispiel mit 2 Klassen



Greifen wir noch einmal das einfache Beispiel mit zwei Klassen auf, die grüne und die rote Klasse. Die drei blauen Punkte liegen auf dem Rand des Trennungsstreifens. Der gelbe Bereich stellt den Trennungsstreifen mit maximaler Breite dar und die blaue Linie die optimal trennende Hyperebene, die den Trennungsstreifen halbiert.

7 Schlussfolgerungen

In dieser Seminararbeit wurden nun die einzelnen Verfahren der linearen Klassifikation vorgestellt und diskutiert. Es hat sich herausgestellt, dass jedes Verfahren Vor- und Nachteile mit sich bringt, die bei der Anwendung der Lerntheorie geschickt ausgenutzt werden sollen.

Die lineare Regression kann z.B. nur dann angewendet werden, wenn man die Datenmenge maximal in zwei Klassen unterscheiden will.

Die lineare Diskriminanzanalyse und die logistische Regression haben gezeigt, dass sie in der Praxis meist sehr ähnliche Ergebnisse liefern, sich aber im Aufwand und in den Annahmen, die über die Daten notwendig sind, unterscheiden. Zudem verschlechtert sich die lineare Diskriminanzanalyse, falls Ausreißer in den Daten vorhanden sind, aber auch die logistische Regression verschlechtert sich, wenn die Daten multivariat normalverteilt sind.

Am Ende der Seminararbeit wurde das Verfahren der Hyperebenentrennung mit dem Perzeptron-Algorithmus und der optimalen Hyperebenentrennung vorgestellt. Diese beiden Verfahren sind allerdings nur anwendbar, wenn man schon im voraus weiß, dass die Daten linear trennbar sind. Anderenfalls brechen die Verfahren ab bzw. liefern keine eindeutige Lösung. Falls die Daten jedoch linear trennbar sind, wurden in der Praxis sehr gute Ergebnisse mit diesen Verfahren gefunden, weshalb, aufbauend auf diesen Verfahren, die „Support vector machines“ entwickelt wurden.

Literatur

[1] T. Hastie, R. Tibshirani und J. Friedman, Februar 2001, The elements of statistical learning, Chapter 4 - Linear Methods for Classification

[2] http://www-stat.stanford.edu/~tibs/ElemStatLearn/

[3] http://de.wikipedia.org/