Multivariate Verfahren

MSc Psychologie | MSc Klinische Psychologie und Psychotherapie

WiSe 2024/25

Prof. Dr. Dirk Ostwald

(13) Support Vector Machines

Anwendungsszenario

Geometrie linearer Diskriminanzfunktionen

Lernen

Anwendungsbeispiel

Selbstkontrollfragen

Anwendungsszenario

Geometrie linearer Diskriminanzfunktionen

Lernen

Anwendungsbeispiel

Selbstkontrollfragen

Anwendungsszenario

Psychotherapie Non-Response-Rate wird auf etwa 20 - 30% geschätzt

Vorhersage von Behandlungserfolg basierend auf klinischen Markern wäre hilfreich

- Therapieauswahloptimierung
- Lebensqualitätverbesserung
- Ressourcensensitivität

Digitale Datenbank von Psychotherapieverläufen als Trainingsdatensatz

Prädiktive Modellierung zur Etablierung eines prädiktiven klinischen Markerprofils

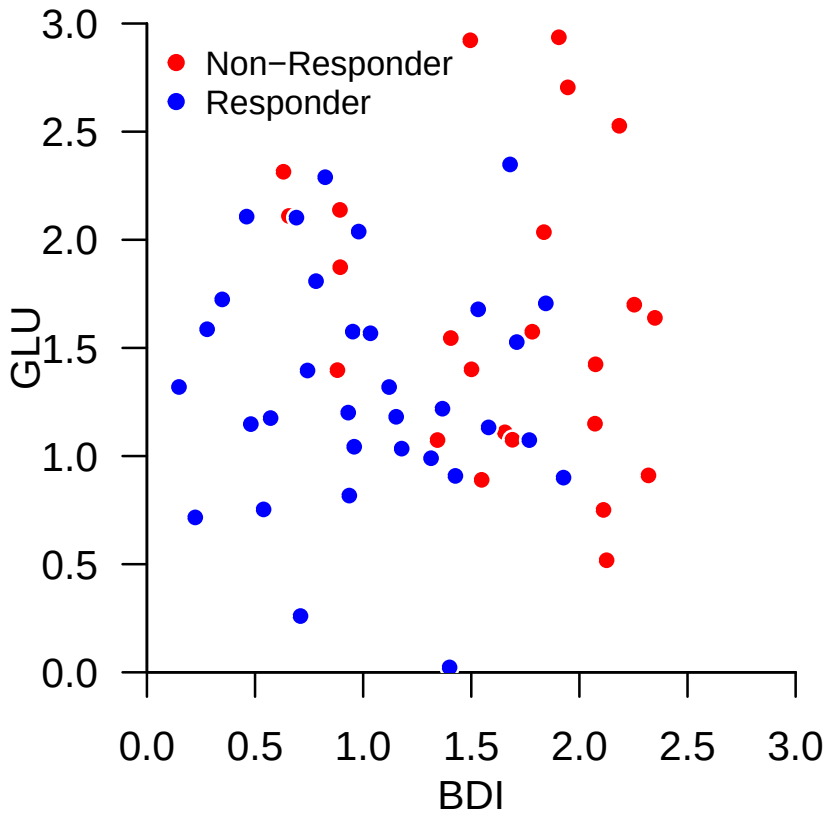
Treatmentsuccessvorhersage für neue Patient:innen

Anwendungsbeispiele

- BDI und GLU Werte bei Depressionsdiagnose als Prädiktoren von CBT Erfolg
- Lineare Diskriminanzanalyse, Logistische Regression, SVM, Neuronale Netze
- Schölkopf and Smola (2002) für eine SVM Einführung

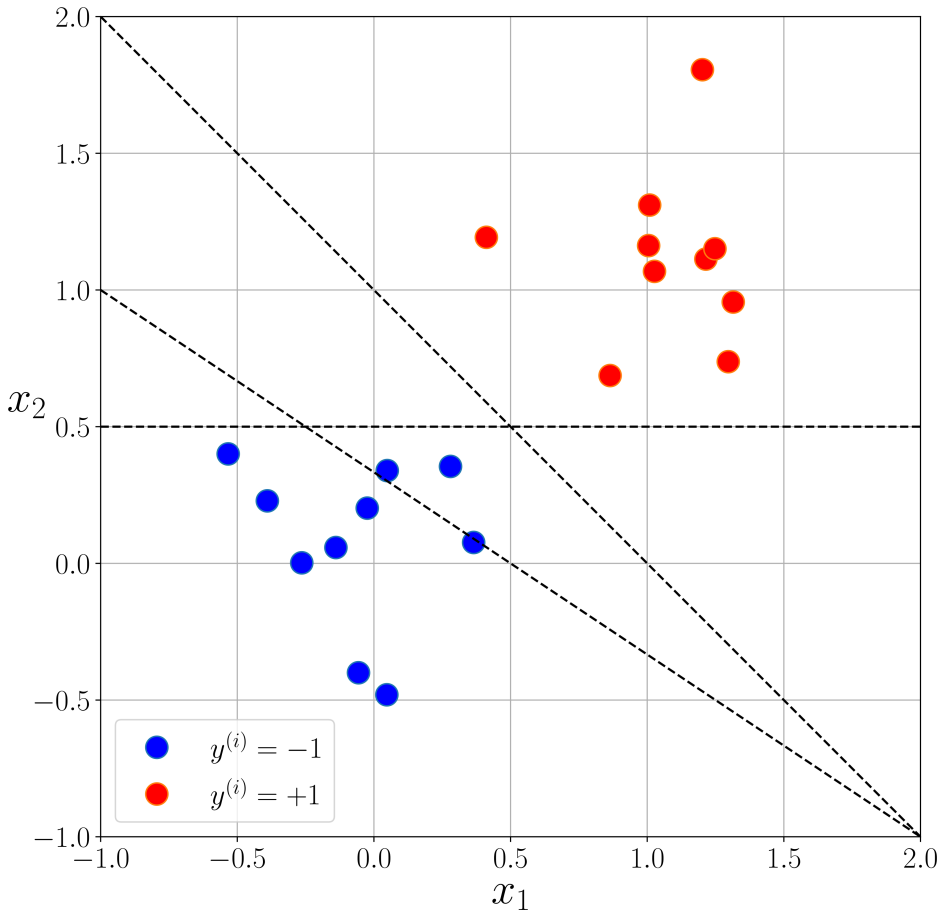
Anwendungsszenario

BDI und GLU Werte bei Depressionsdiagnose als Prädiktoren von CBT Erfolg



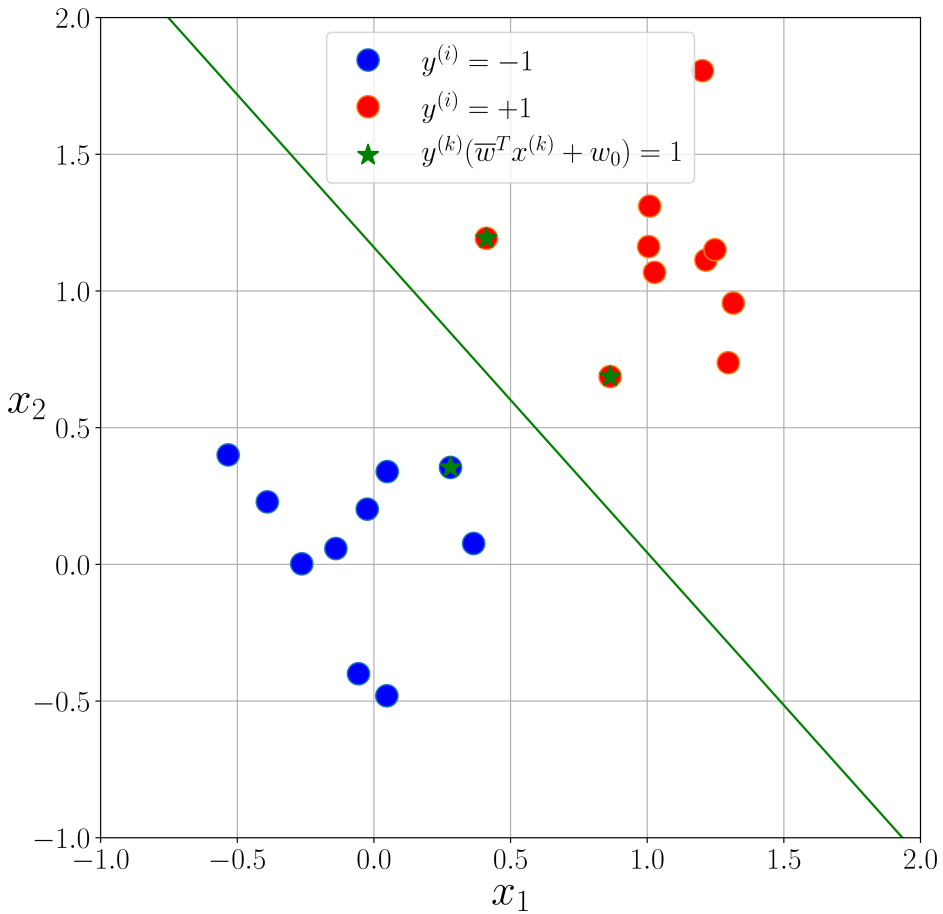
Grundlegende Fragestellung

- Was ist eine gute Klassifikationsgrenze für einen gegebenen Trainingsdatensatz?



Grundlegende Antwort der Support Vector Machine-Theorie

- Eine gute Klassifikationsgrenze hat einen großen Margin (Rand, Spielraum)



Vladimir N. Vapnik was born in Russia and received the Ph.D. degree in statistics from the Institute of Control Sciences, Academy of Science of the USSR, Moscow, Russia, in 1964.

Since 1991, he has been working for AT&T Bell Laboratories (since 1996, AT&T Labs Research), Red Bank, NJ. His research interests include statistical learning theory, theoretical and applied statistics, theory and methods for solving stochastic ill-posed problems, and methods of multidimensional function approximation. His main results in the last three years are related to the development of the support vector method. He is author of many publications, including seven monographs on various problems of statistical learning theory.

Anwendungsszenario

Vapniks Theorie des Statistischen Lernens

- Ein mathematischer Framework zur Identifikation guter daten-basierter prädiktiver Funktionen
- Fokus auf prädiktive Performanz, nicht Akkuratheit vom Modellapproximationen
- Support Vector Machines als Beispiele von Vapniks Theorie des statistischen Lernens

Historischer Abriss

- 1964: Vapnik and Chervonenkis (1964) entwickeln das Maximum Margin Klassifikation
- 1992: Boser, Guyon, and Vapnik (1992) schlagen das Prinzip von Kernelfunktionen vor
- 1995: Cortes and Vapnik (1995) schlagen Soft Margin Klassifikation vor
- Späte 1990er: SVM Toolboxen, Webseiten, und Summer Schools
- 2000er: SVM Hype in der Machine Learning Community
- 2010er: Neural networks Hype in der Machine Learning Community
- 2020er: Large Language Model und genereller KI Hype

Anwendungsszenario

Thema der Vorlesung

- Geometrische Grundlagen der Maximum- und Soft-Margin-Klassifikation

Hier nicht behandelte weiterführende SVM Themen

- Training als restringiertes quadratisches Optimierungsproblem
- Vapniks Theorie des Statistischen Lernens
- Optimalitätsfragen
- Multiklassenklassifikation
- Kernelmethoden
- Probabilistische Generalisierungen (vgl. Franc, Zien, and Schölkopf (2011))

Anwendungsszenario

Geometrie linearer Diskriminanzfunktionen

Lernen

Anwendungsbeispiel

Selbstkontrollfragen

Definition (Trainingsdatensatz)

Ein *Trainingsdatensatz*

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

ist eine Menge von n *Trainingsdatenpunkten*

$$\left(x^{(i)}, y^{(i)}\right) \text{ mit } x^{(i)} \in \mathbb{R}^m \text{ und } y^{(i)} \in \{-1, +1\} \text{ für } i = 1, ..., n, \tag{2}$$

wobei $x^{(i)}$ als m-*dimensionaler Featurevektor* und $y^{(i)}$ als *Tagetvariable* bezeichnet wird.

Bemerkungen

- $y^{(i)} \in \{-1, +1\}$ bezeichnet die Klassenzugehörigkeit von $x^{(i)} \in \mathbb{R}^m$.
- $x^{(i)} \in \mathbb{R}^m$ kann die Werte von m klinischen Markern des i-ten von n Patienten in einer von zwei Diagnosegruppen $y^{(i)}$ bezeichnen, die durch -1 bzw. $+1$ bezeichnet werden.

Definition (Lineare Diskriminanzfunktion)

Eine *lineare Diskriminanzfunktion* ist eine multivariate reellwertige Funktion der Form

$$h : \mathbb{R}^m \to \{-1, +1\}, x \mapsto h(x) := g(f(x)), \tag{3}$$

wobei

- f eine multivariate, reellwertige, parameterabhängige linear-affine Funktion der Form

  $$f : \mathbb{R}^m \to \mathbb{R}, x \mapsto f(x) := w^T x + w_0, \tag{4}$$

  mit *Parametervektor* $w \in \mathbb{R}^m$ und *Bias-Parameter* $w_0 \in \mathbb{R}$ ist und
- g eine univariate, reellwertige parameterunabhängige Klassifikationsfunktion

  $$g : \mathbb{R} \to \{-1, 1\}, f(x) \mapsto g(f(x)) := \begin{cases} -1, & \text{für } f(x) < 0 \\ +1, & \text{für } f(x) \geq 0 \end{cases} \tag{5}$$

  ist.

Im Featureraum induziert eine lineare Diskriminanzfunktion

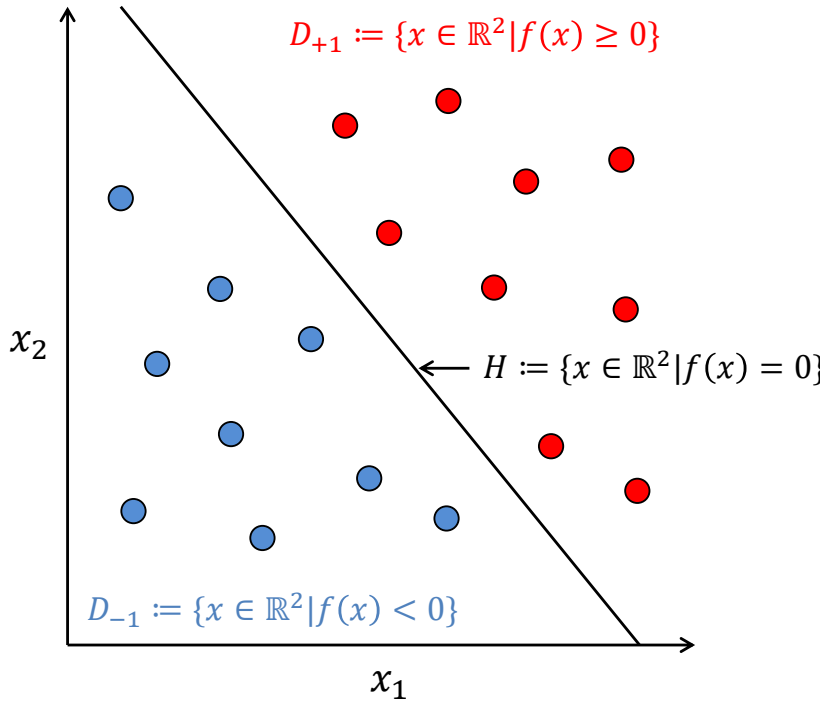
- eine Klassifikationsgrenze $H := \{x \in \mathbb{R}^m | f(x) = 0\}$, die als Hyperebene bezeichnet wird,
- eine *Klassifikationsregion* $D_{-1} := \{x \in \mathbb{R}^m | f(x) < 0\}$ und
- eine *Klassifikationssregion* $D_{+1} := \{x \in \mathbb{R}^m | f(x) \geq 0\}$.

Geometrie linearer Diskriminanzfunktionen

$$m \coloneqq 2$$

$$\Rightarrow x^{(i)} = \left(x_1^{(i)}, x_2^{(i)}\right)^T \in \mathbb{R}^2$$

- $x^{(i)}$ with $y^{(i)} = -1$
- $x^{(i)}$ with $y^{(i)} = +1$



Geradengleichung einer Hyperebene in $\mathbb{R}^2$

$$f(x) = 0 \Leftrightarrow w^T x + w_0 = 0 \Leftrightarrow w_1 x_1 + w_2 x_2 + w_0 = 0 \Leftrightarrow x_2 = -\frac{w_1}{w_2} x_1 - \frac{w_0}{w_2} \tag{6}$$

Theorem (Geometrie linearer Diskriminanzfunktionen)

Es sei

$$f : \mathbb{R}^m \to \mathbb{R}, x \mapsto f(x) := w^T x + w_0, \tag{7}$$

eine multivariate, reellwertige, parameterabhängige linear-affine Funktion und es sei

$$H := \{x \in \mathbb{R}^m | f(x) = 0\} \subset \mathbb{R}^m \tag{8}$$

ihre assoziierte *Hyperebene*. Dann gelten folgende geometrische Beziehungen

(1) w ist orthogonal zu jedem Vektor, der in Richtung von H zeigt.

(2) Die minimale Euklidische Distanz d zwischen einem $x \in \mathbb{R}^m$ und einem Punkt der Hyperebene ist

$$d = \frac{1}{||w||_2} f(x). \tag{9}$$

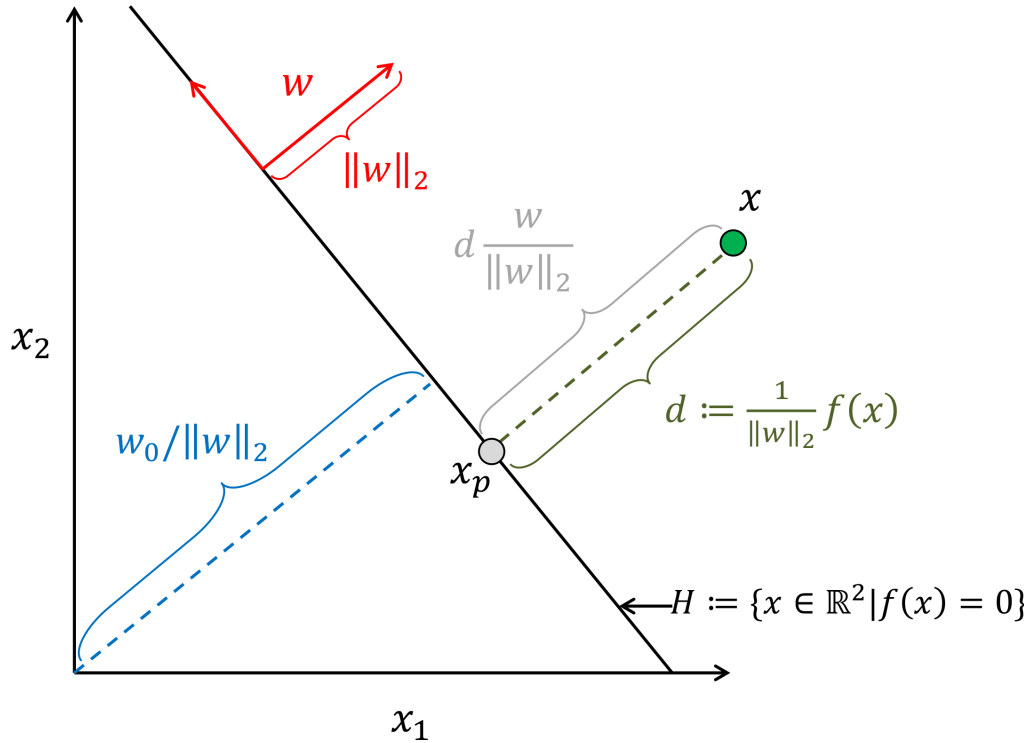
(3) Die minimale Euklidische Distanz d_0 zwischen dem Ursprung und einem Punkt der Hyperebene ist

$$d_0 = \frac{w_0}{||w||_2}. \tag{10}$$

Bemerkungen

- Die Wahl von w legt die Orientierung der Hyperebene fest.
- Die Wahl von w_0 legt die Lage der Hyperebene im Featureraum fest.

Geometrie linearer Diskriminanzfunktionen



Bemerkungen

- Die Wahl von w legt die Orientierung der Hyperebene fest.
- Die Wahl von w_0 legt die Lage der Hyperebene im Featureraum fest.

Beweis

Beweis von (1)

Es seien $x_a \in H_w$ und $x_b \in H_w$ zwei beliebige Punkte der Hyperebene, und es sei $y := (x_a - x_b)$ der sie verbindene Vektor in Richtung der Hyperebene. Dann gilt folgendes Gleichungssystem:

$$w^T x_a + w_0 = 0 \tag{11}$$

$$w^T x_b + w_0 = 0 \tag{12}$$

Subtraktion von (12) von (11) ergibt dann

$$w^T x_a - w^T x_b = 0 \Leftrightarrow w^T (x_a - x_b) = 0 \Leftrightarrow w^T y = 0. \tag{13}$$

Da $x_a \in H_w$ und $x_b \in H_w$ und damit auch y beliebig waren, ist w damit orthogonal zu jedem Vektor in Richtung der Hyperebene.

Beweis (fortgeführt)

Beweis von (2)

Wir betrachten die Zerlegung eines $x \in \mathbb{R}^m$ in seine orthogonale Projektion $x_p \in \mathbb{R}^m$ auf die Hyperebene und seinen Abstand d von der Hyperebene,

$$x = x_p + d\frac{w}{\|w\|_2} \tag{14}$$

Diese Zerlegung ist möglich, da nach (1) w orthogonal zu jedem Vektor in Richtung der Hyperebene ist und es gilt, dass (vgl. Abbildung oben)

$$\|\frac{w}{\|w\|_2}\|_2 = 1. \tag{15}$$

Wir betrachten nun die Anwendung der Funktion f der linearen Diskriminanzfunktion auf das so zerlegte x:

$$f(x) = w^T x + w_0 = w^T \left(x_p + d\frac{w}{\|w\|_2}\right) + w_0 = w^T x_p + w_0 + d\frac{w^T w}{\|w\|_2}. \tag{16}$$

Mit $x_p \in H_w$ und somit $w^T x_p + w_0 = 0$, ergibt sich

$$f(x) = d\frac{w^T w}{\|w\|_2} = d\frac{\|w\|_2^2}{\|w\|_2} = d\|w\|_2 \tag{17}$$

und damit

$$d = \frac{1}{\|w\|_2} f(x). \tag{18}$$

Beweis

Beweis von (3)

Für den minimalen Abstand des Ursprungs $x_0 = (0, \ldots, 0)^T \in \mathbb{R}^m$ zu Punkten der Hyperebene ergibt sich mit (2)

$$d_0 = \frac{1}{\|w\|_2} f(x_0) = \frac{1}{\|w\|_2} (w^T x_0 + w_0) = \frac{1}{\|w\|_2} w^T \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} + \frac{w_0}{\|w\|_2} = \frac{w_0}{\|w\|_2}. \tag{19}$$

□

Definition (Hyperebenenmargin und Supportvektor)

$\mathcal{D}$ sei ein Trainingsdatensatz, f sei eine multivariate reellwertige linear-affine Funktion und H sei die von f induzierte Hyperebene. Weiterhin seien für $i = 1, \ldots, n$

$$\left|d^{(i)}\right| := \left|\frac{1}{\|w\|_2} f\left(x^{(i)}\right)\right| = \frac{y^{(i)}}{\|w\|_2} f\left(x^{(i)}\right) = \frac{y^{(i)}(w^T x^{(i)} + w_0)}{\|w\|_2} \geq 0 \tag{20}$$

die Beträge der minimalen Distanzen der Featurevektoren $x^{(i)}$ von der Hyperebene. Der *Hyperebenenmargin* d^* von H in Bezug auf $\mathcal{D}$ ist dann definiert als das Minimum dieser Beträge,

$$d^* := \min_{i=1,\ldots,n} \left\{\left|d^{(i)}\right|\right\} = \min_{i=1,\ldots,n} \left\{ \frac{y^{(i)}(w^T x^{(i)} + w_0)}{\|w\|_2} \right\} \tag{21}$$

Weiterhin heißt der $i$te Featurevektor *Supportvektor*, wenn gilt, dass

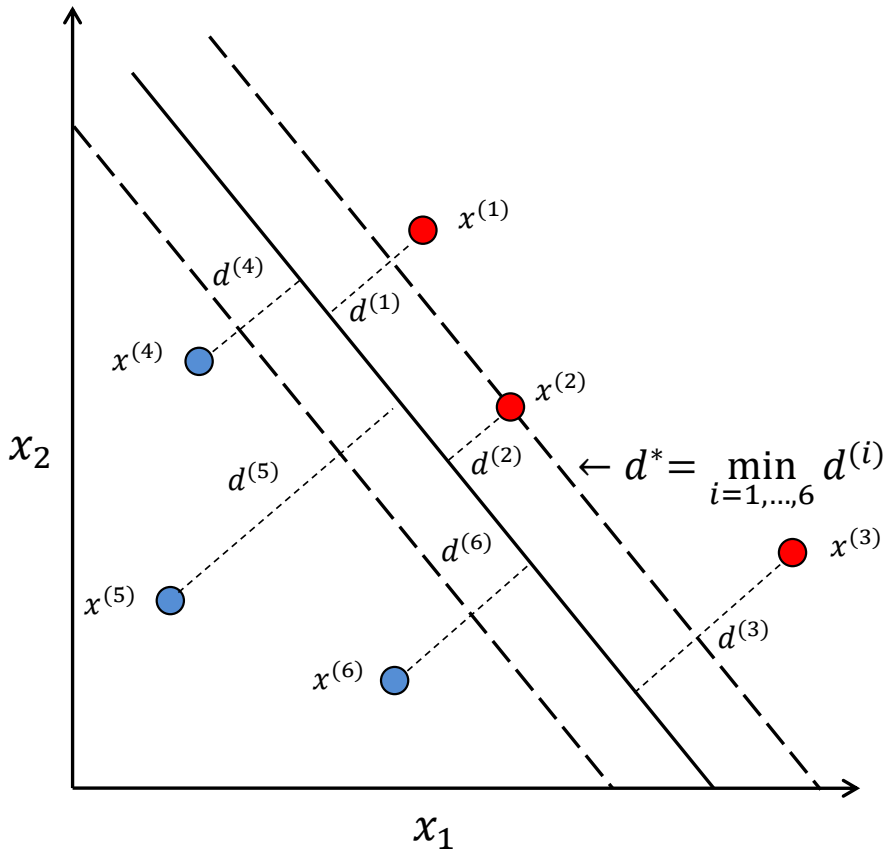
$$|d^{(i)}| = d^*, \tag{22}$$

wenn also $x^{(i)}$ auf dem Hyperebenenmargin liegt.

Bemerkungen

- Man beachte, dass mit $y^{(i)} = -1$ für $f(x^{(i)}) < 0$ und $y^{(i)} = +1$ für $f(x^{(i)}) \geq 0$ der Betrag von $f(x^{(i)})$ durch $y^{(i)} f(x^{(i)})$ gegeben ist.

Hyperebenenmargin und Supportvektoren



Definition (Äquivalente und kanonische Hyperebenen)

f sei eine multivariate reellwertige linear-affine Funktion und

$$H := \{x \in \mathbb{R}^m | f(x) = 0\} \tag{23}$$

sei die von f induzierte Hyperebene. Dann induzieren alle skalaren Vielfachen von f und damit von w und w_0 identische Hyperebenen, denn aus $f(x) = 0$ folgt, dass $af(x) = 0$ für alle $a \in \mathbb{R} \setminus \{0\}$. Entsprechend nennt man die Hyperebenen

$$H_a := \{x \in \mathbb{R}^m | af(x) = 0 \Leftrightarrow \tilde{w}^T x + \tilde{w}_0 = 0 \text{ mit } \tilde{w} = aw, \tilde{w}_0 = aw_0 \text{ für } a \in \mathbb{R} \setminus \{0\}\} \tag{24}$$

die Menge der zu H *äquivalenten Hyperebenen*. Für einen Supportvektor x^* und eine Menge äquivalenter Hyperebenen ist die *kanonische Hyperebene* definiert als die Hyperebene, für die gilt, dass

$$|f(x^*)| = y^* \left(w^T x^* + w_0\right) = 1. \tag{25}$$

Nach Definition der kanonischen Hyperebene ist der Margin der kanonischen Hyperebene

$$d^* = \frac{1}{\|w\|_2}. \tag{26}$$

Bemerkung

- Prinzipiell gibt es undendlich viele w und w_0, die die gleiche Hyperebene beschreiben.
- Der Begriff der kanonischen Hyperebene legt w und w_0 eindeutig fest.

Anwendungsszenario

Geometrie linearer Diskriminanzfunktionen

Lernen

Anwendungsbeispiel

Selbstkontrollfragen

Definition (Maximum-Margin-Klassifikation)

$\mathcal{D}$ sei ein *linear-separierbarer* Trainingsdatensatz, also ein Datensatz für den es eine lineare Diskriminanzfunktion gibt, die alle Trainingsdatenpunkte korrekt klassifiziert. Dann entspricht das Lernen der Parameter zum Zwecke der *Maximum-Margin-Klassifikation* dem restringierten Optimierungsproblem

$$\min_{w,w_0} \frac{1}{2}\|w\|_2^2 \text{ u.d.N. } y^{(i)}\left(w^T x^{(i)} + w_0\right) \geq 1 \text{ für } i = 1, ..., n \tag{27}$$

Dabei entspricht

- das Optimierungsproblem

  $$\min_{w,w_0} \frac{1}{2}\|w\|_2^2 \tag{28}$$

  dem Optimierungsproblem

  $$\max_{w,w_0} \frac{1}{\|w\|_2} \tag{29}$$

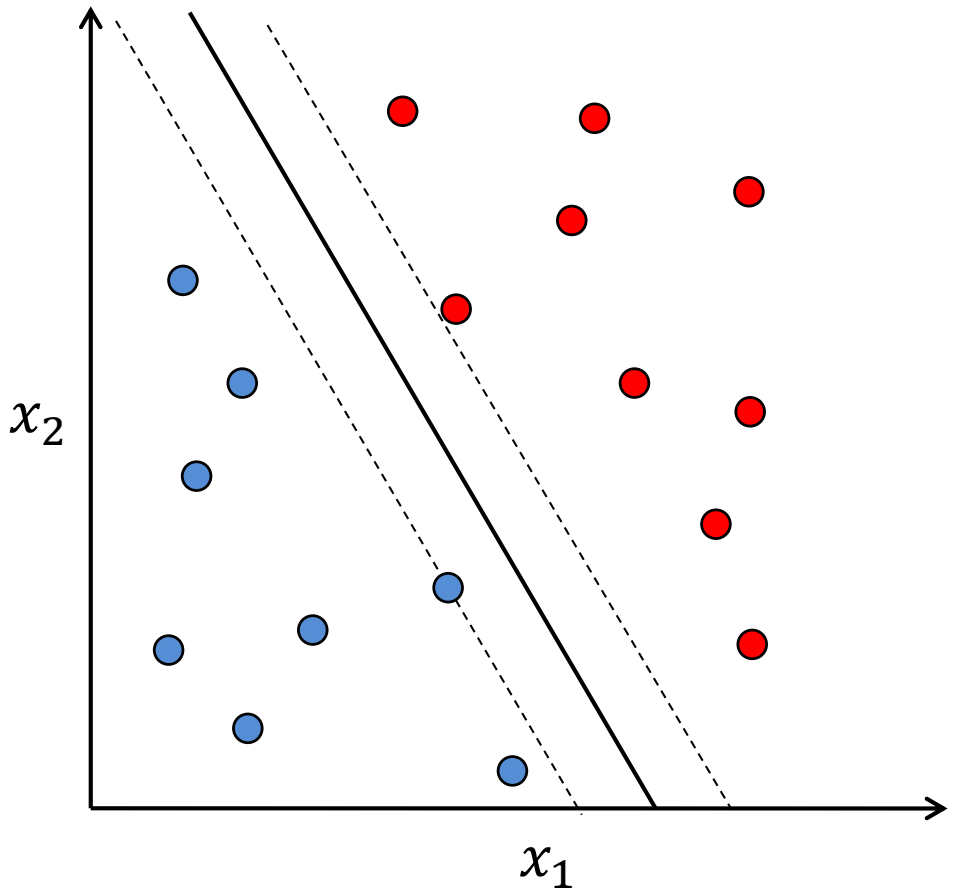
  und damit der Maximierung des Margins der Hyperebene und

- die Nebenbedingungen

  $$y^{(i)}\left(w^T x^{(i)} + w_0\right) \geq 1 \text{ für } i = 1, ..., n \tag{30}$$

  dem Ziel, dass alle Featurevektor im Falle der Ungleichheit ($>$) auf der korrekten Seite der Hyperebene liegen oder im Falle der Gleichheit ($=$) Supportvektoren bilden.

Maximum-Margin-Klassifikation bei linear-separierbarem Datensatz



$$\min_{w,w_0} \frac{1}{2}\|w\|_2^2 \text{ u. d. N. } y^{(i)}\left(w^T x^{(i)} + w_0\right) \geq 1$$

Definition (Soft-Margin-Klassifikation)

$\mathcal{D}$ sei ein nicht notwendigerweise *linear-separierbarer* Trainingsdatensatz. Dann entspricht das Lernen der Parameter zum Zwecke der *Soft-Margin-Klassifikation* dem restringierte Optimierungsproblem

$$\min_{w,w_0,\xi} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n}\xi_i^k \text{ u.d.N. } y^{(i)}\left(w^T x^{(i)} + w_0\right) \geq 1 - \xi_i, \xi_i \geq 0 \text{ für } i = 1, ..., n \tag{31}$$

Dabei ist $\xi := (\xi_1, ..., \xi_n)$ ein Vektor sogenannter *Schlupfvariablen* $\xi_i, i = 1, ..., n$, der Term $\sum_{i=1}^{n}\xi_i^k$ ein sogenannter *Verlust (Loss)*, k eine Verlustartkonstante (z.B. $k = 1$ für *hinge loss*, $k = 2$ für *quadratic loss*) und $C \in \mathbb{R}$ eine empirisch gewählte Konstante. Dabei entsprechen

- das Optimierungsproblem

$$\min_{w,w_0,\xi} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n}\xi_i^k \tag{32}$$

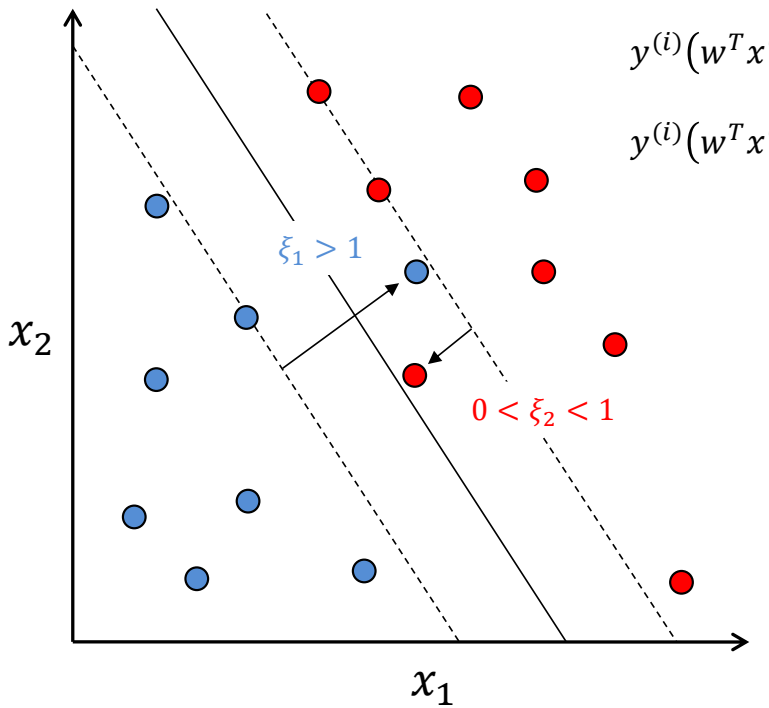
  dem Optimierungsproblem

$$\max_{w,w_0,\xi} \frac{1}{\|w\|_2} - C\sum_{i=1}^{n}\xi_i^k \tag{33}$$

  also der Maximierung des Hyperplanemargins bei gleichzeitiger Minimierung des Verlustes mit relativer Gewichtung C,

- die Nebenbedinungen
  (1) der korrekten Trainingsdatenpunktklassifikation und Margin-Maximierung für $\xi_i = 0$,
  (2) der korrekten Trainingsdatenpunktklassifikation für $0 < \xi < 1$, und
  (3) der inkorrekten Trainingsdatenpunktklassifikation für $\xi > 1$.

Soft-Margin-Klassifikation bei nicht linear-separierbarem Datensatz



$$y^{(i)}\left(w^T x^{(i)} + w_0\right) \geq 1$$
$$\downarrow$$
$$y^{(i)}\left(w^T x^{(i)} + w_0\right) \geq 1 - \xi_i$$

$$\min_{w,w_0,\xi} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n} \xi_i^k \text{ u.d.N. } y^{(i)}\left(w^T x^{(i)} + w_0\right) \geq 1 - \xi_i, \xi_i \geq 0$$

Kernelmethoden

SVM-Klassifikation kann als restringiertes quadratisches Optimierungsproblem formuliert werden

Für restringierte quadratische Optimierungsproblem gibt es viele Standardlösungsverfahren

Die duale Zielfunktion des Maximum-Margin-Klassifikationsproblem hat dabei die Form

$$q(\lambda) := \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y^{(i)} y^{(j)} {x^{(i)}}^T x^{(j)} \tag{34}$$

Verständnis dieser erfordert Kenntnisse der restringierten Optimierung (vgl. Appendix).

Die duale Zielfunktion hängt dabei nur von den Skalarprodukten

$${x^{(i)}}^T x^{(j)} \text{ für } i, j = 1, ..., n. \tag{35}$$

der Featurevektoren ab.

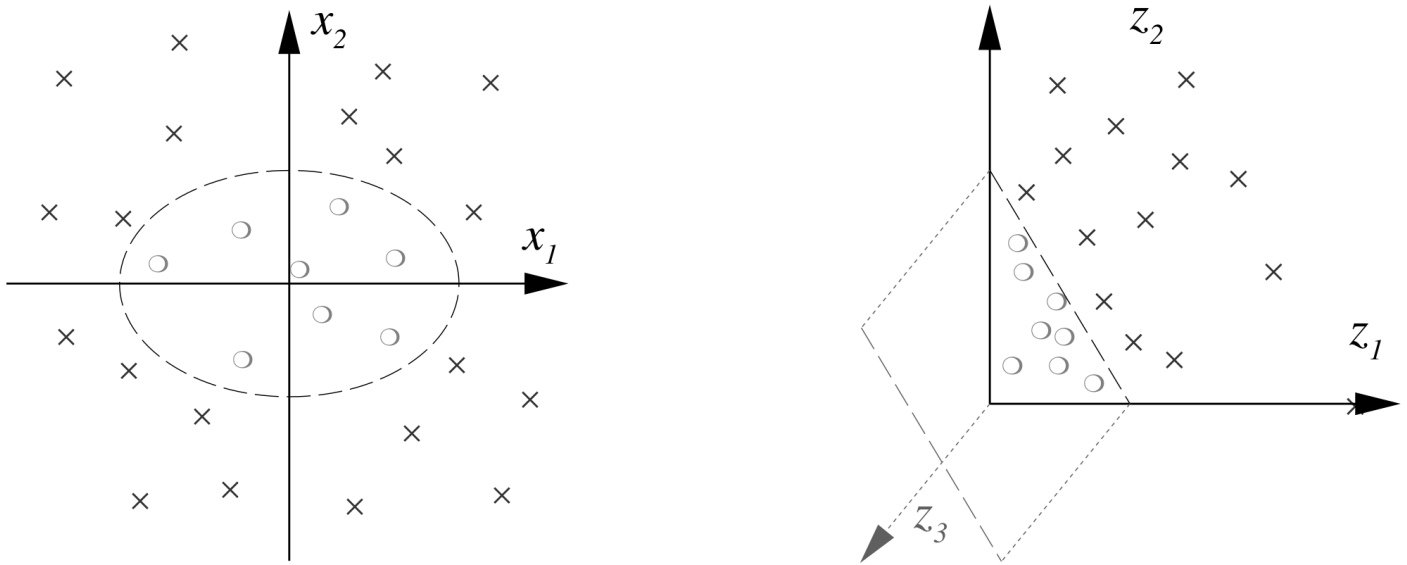
Projiziert man jeden Trainingsdatenfeaturevektor also in einen "hochdimensionalen" Featureraum, in dem man auf lineare Separierbarkeit hofft, so kann man auch mit hochdimensionalen Versionen der Featurevektoren mit überschaubaren Rechenaufwand SVM Klassifikation betreiben

Die Skalarprodukte in diesen "hochdimensionalen Featureräumen" werden *Kernel* genannt

Kernelmethoden

Idee der Kernelisierung der Maximum-Margin Klassifikation

$$\phi : \mathcal{X} \to \tilde{X}, x \mapsto \phi(x) \text{ mit } k(x, x') = \langle \phi(x), \phi(x') \rangle \tag{36}$$



Schölkopf and Smola (2002)

Anwendungsszenario

Geometrie linearer Diskriminanzfunktionen

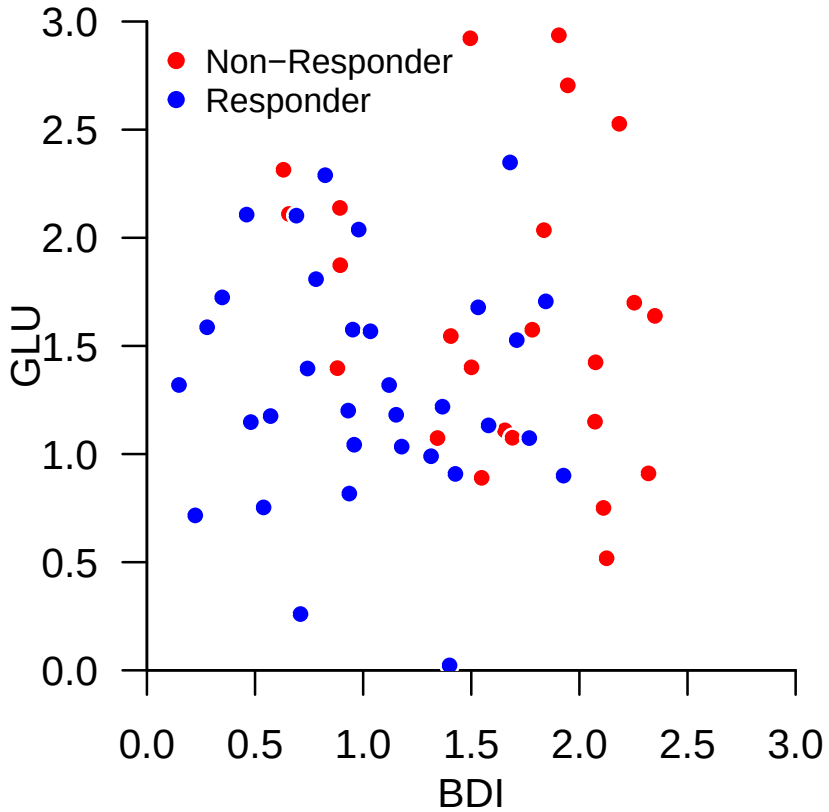
Lernen

Anwendungsbeispiel

Selbstkontrollfragen

Anwendungsbeispiel

BDI und GLU Werte bei Depressionsdiagnose als Prädiktoren von CBT Erfolg



Anwendungsbeispiel

`BDI` und `GLU` Werte bei Depressionsdiagnose als Prädiktoren von CBT Erfolg `RES`

BDI	GLU	RES
0.74	1.40	1
0.22	0.72	1
0.82	2.29	1
2.07	1.15	0
1.71	1.53	1
1.77	1.07	1
1.95	2.70	0
2.18	2.53	0
0.93	1.20	1
1.34	1.07	0
2.35	1.64	0
1.43	0.91	1
1.66	1.11	0
0.28	1.59	1
2.13	0.52	0
1.37	1.22	1
0.89	2.14	0
0.88	1.40	0
0.98	2.04	1
1.93	0.90	1

Anwendungsbeispiel

LOOCV mithilfe von `svm()` des Pakets `e1071`

Wrapper für C++ Implementation 'libsvm' (Chang and Lin (2011))

Soft-Margin-Klassifikation (`C-classification`) mit linearem Kernel und C-Parameter `cost = 1`.

```
library(e1071)                                                          # libsvm wrapper svm() u.v.a.m.
D       = read.csv("./13_Daten/13_Support_Vector_Machines.csv")         # Datensatz
K       = nrow(D)                                                       # Anzahl Cross Folds
kern    = "linear"                                                      # linearer Kernel
type    = "C-classification"                                            # Soft-margin Klassifikation
C       = 1                                                             # C-Parameter
y_pred  = matrix(rep(NaN, K*2), nrow = K)                               # Prädiktionsperformancearray
for(k in 1:K){                                                          # K-fold LOOCV
    x_train     = D[-k,1:2]                                             # Trainingsdatensatzfeatures
    y_train     = D[-k,3]                                               # Trainingsdatensatzlabels
    x_test      = D[ k,1:2]                                             # Testdatensatzfeaturevektor
    y_pred[k,1] = D[ k,3]                                               # Testdatensatztargetvariable
    svm_train   = svm(x_train, y_train, kernel = kern, type = type, cost = C)   # SVM Training
    y_pred[k,2] = as.numeric(predict(svm_train, x_test)) - 1            # SVM Prädiktion (svm() in {1,2})
}
```

```
Accuracy    :  0.72 , Sensitivity:  0.82 , Specificity:  0.58
```

Anwendungsbeispiel

`BDI` und `GLU` Werte bei Depressionsdiagnose als Prädiktoren von CBT Erfolg `RES`

BDI	GLU	RES
0.74	1.40	1
0.22	0.72	1
0.82	2.29	1
2.07	1.15	0
1.71	1.53	1
1.77	1.07	1
1.95	2.70	0
2.18	2.53	0
0.93	1.20	1
1.34	1.07	0
2.35	1.64	0
1.43	0.91	1
1.66	1.11	0
0.28	1.59	1
2.13	0.52	0
1.37	1.22	1
0.89	2.14	0
0.88	1.40	0
0.98	2.04	1
1.93	0.90	1

Anwendungsbeispiel

LOOCV mithilfe von `svm()` des Pakets `e1071`

Wrapper für C++ Implementation 'libsvm' (Chang and Lin (2011))

Soft-Margin-Klassifikation (`C-classification`) mit nichtlinearem Kernel und C-Parameter `cost = 1`.

```
library(e1071)                                                          # libsvm wrapper svm() u.v.a.m.
D       = read.csv("./13_Daten/13_Support_Vector_Machines.csv")         # Datensatz
K       = nrow(D)                                                       # Anzahl Cross Folds
kern    = "polynomial"                                                  # nichtlinearer Kernel
type    = "C-classification"                                            # Soft-margin Klassifikation
C       = 1                                                             # C-Parameter
y_pred  = matrix(rep(NaN, K*2), nrow = K)                               # Prädiktionsperformancearray
for(k in 1:K){                                                          # K-fold LOOCV
    x_train     = D[-k,1:2]                                             # Trainingsdatensatzfeatures
    y_train     = D[-k,3]                                               # Trainingsdatensatztargetvariablen
    x_test      = D[ k,1:2]                                             # Testdatensatzfeaturevektor
    y_pred[k,1] = D[ k,3]                                               # Testdatensatztargetvariable
    svm_train   = svm(x_train, y_train, kernel = kern, type = type, cost = C)   # SVM Training
    y_pred[k,2] = as.numeric(predict(svm_train, x_test)) - 1            # SVM Prädiktion (svm() in {1,2})
}
```

```
Accuracy    :  0.75 , Sensitivity:  0.97 , Specificity:  0.46
```

Anwendungsszenario

Geometrie linearer Diskriminanzfunktionen

Lernen

Anwendungsbeispiel

Selbstkontrollfragen

Selbstkontrollfragen

1. Geben Sie die Definition einer linearen Diskriminanzfunktion wieder.
2. Erläutern Sie die Begriffe der Hyperebene und der Klassifikationsregion.
3. Geben Sie das Theorem zur Geometrie linearer Diskriminanzfunktionen wieder.
4. Erläutern Sie die Bedeutung des Theorems zur Geometrie linearer Diskriminanzfunktionen.
5. Geben Sie die Definition des Hyperebenenmargin und eines Supportvektors wieder.
6. Geben Sie die Defintion der Maximum-Margin-Klassifikation wieder.
7. Geben Sie die Defintion der Soft-Margin-Klassifikation wieder.
8. Erläutern Sie den Unterschied zwischen Maximum-Margin-Klassifikation und Soft-Margin-Klassifikation.

Appendix

Theorem (Duales Problem des Maximum-Margin-Optimierungsproblems)

The duale Problem des Maximum-Margin-Optimierungsproblems

$$\min_{w,w_0} \frac{1}{2}\|w\|_2^2 \text{ u.d.N. } y^{(i)}(w^T x^{(i)} + w_0) \geq 1 \text{ für } i = 1, \ldots, n \tag{37}$$

ist durch

$$\max_{\lambda \in \mathbb{R}^n} q(\lambda) := \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y^{(i)} y^{(j)} {x^{(i)}}^T x^{(j)} \tag{38}$$

mit den Nebenbedingungen

$$\lambda \geq 0 \text{ und } \sum_{i=1}^{n} \lambda_i y^{(i)} = 0 \tag{39}$$

gegeben. Gegeben eine Lösung $\bar{\lambda}$ des dualen Problem sind alle $x^{(k)}$ mit $\lambda_k > 0, k = 1, \ldots, K$ Supportvektoren und die Lösungen für die Gewichts- und Biasparameter des primären Problems ergeben sich zu

$$\bar{w} = \sum_{i=1}^{n} \bar{\lambda_i} y^{(i)} x^{(i)} \text{ bzw. } \bar{w}_0 = \frac{1}{K} \sum_{k=1}^{K} \left(y^{(k)} - \bar{w}^T x^{(k)} \right). \tag{40}$$

Maximum-Margin-Klassifikation als quadratische Optimierungsproblem

Beweis

(1) Lagrangefunktion des primären Problems

Die Lagrangefunktion des primären Problems

$$\min_{w,w_0} \frac{1}{2}\|w\|_2^2 \text{ u.d.N. } y^{(i)}(w^T x^{(i)} + w_0) \geq 1 \text{ für } i = 1, ..., n \tag{41}$$

ist gegeben durch

$$L(w, w_0, \lambda) := \frac{1}{2} w^T w - \sum_{i=1}^{n} \lambda_i (y^{(i)}(w^T x^{(i)} + w_0) - 1) \tag{42}$$

(2) Duale Zielfunktion

Die duale Zielfunktion hat im vorliegenden Fall die Form

$$q : \mathbb{R}^n \to \mathbb{R}, \lambda \mapsto q(\lambda) := \min_{w,w_0} L(w, w_0, \lambda). \tag{43}$$

Die analytische Bestimmung der Lagrangefunktion hinsichtlich w und w_0 impliziert, die Ableitungen von L hinsichtlich w und w_0 zu bestimmen und gleich Null zu setzen. Dazu seien

$$\bar{w} := \arg\min_{w \in \mathbb{R}^m} L(w, w_0, \lambda) \text{ und } \bar{w_0} := \arg\min_{w_0 \in \mathbb{R}} L(w, w_0, \lambda) \tag{44}$$

Maximum-Margin-Klassifikation als quadratische Optimierungsproblem

Beweis (fortgeführt)

In Hinblick auf die Minimierung bezüglich w ergibt sich

$$\begin{aligned}\nabla_w L(w, w_0, \lambda) &= \nabla_w \left(\frac{1}{2} w^T w - \sum_{i=1}^{n} \lambda_i (y^{(i)} (w^T x^{(i)} + w_0) - 1) \right) \\ &= w - \nabla_w \left(\sum_{i=1}^{n} \lambda_i y^{(i)} w^T x^{(i)} + \lambda_i y^{(i)} w_0 - \lambda_i \right) \qquad (45) \\ &= w - \sum_{i=1}^{n} \lambda_i y^{(i)} x^{(i)}\end{aligned}$$

An der Minimalstelle von L bezüglich w ergibt sich also

$$\bar{w} = \sum_{i=1}^{n} \lambda_i y^{(i)} x^{(i)}. \qquad (46)$$

Beweis (fortgeführt)

In Hinblick auf die Minimierung bezüglich w_0 ergibt sich

$$\begin{aligned}\nabla_{w_0} L(w, w_0, \lambda) &= \nabla_{w_0} \left(\frac{1}{2} w^T w - \sum_{i=1}^{n} \lambda_i (y^{(i)} (w^T x^{(i)} + w_0) - 1) \right) \\ &= \nabla_{w_0} \left(\sum_{i=1}^{n} \lambda_i y^{(i)} w^T x^{(i)} + \lambda_i y^{(i)} w_0 - \lambda_i \right) \\ &= -\sum_{i=1}^{n} \lambda_i y^{(i)}\end{aligned} \tag{47}$$

An der Minimalstelle von L bezüglich w_0, ergibt sich also e

$$-\sum_{i=1}^{n} \lambda_i y^{(i)} = 0 \tag{48}$$

Wir erhalten also lediglich diese Bedingung, aber keinen Minimierer $\bar{w}_0$.

Maximum-Margin-Klassifikation als quadratische Optimierungsproblem

Beweis (fortgeführt)

Für die duale Zielfunktion ergibt sich also

$$
\begin{aligned}
q(\lambda) &= \min_{w,w_0} L(w, w_0, \lambda) \\
&= L(\bar{w}, \bar{w}_0, \lambda) \\
&= \frac{1}{2}\bar{w}^T\bar{w} - \sum_{i=1}^{n} \lambda_i \left(y^{(i)}(\bar{w}^T x^{(i)} + \bar{w}_0) - 1 \right) \\
&= \frac{1}{2}\left(\sum_{i=1}^{n} \lambda_i y^{(i)} x^{(i)}\right)^T \left(\sum_{j=1}^{n} \lambda_i y^{(j)} x^{(j)}\right) - \sum_{i=1}^{n} \lambda_i \left(y^{(i)} \left(\left(\sum_{j=1}^{n} \lambda_j y^{(j)} x^{(j)}\right)^T x^{(i)} + \bar{w}_0 \right) - 1 \right) \\
&= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j y^{(i)} y^{(j)} {x^{(i)}}^T x^{(j)} - \sum_{i=1}^{n} \lambda_i y^{(i)} \left(\left(\sum_{j=1}^{n} \lambda_j y^{(j)} x^{(j)}\right)^T x^{(i)} + \bar{w}_0 \right) + \sum_{i=1}^{n} \lambda_i \\
&= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j y^{(i)} y^{(j)} {x^{(i)}}^T x^{(j)} - \sum_{i=1}^{n} \lambda_i \lambda_j y^{(i)} y^{(j)} {x^{(i)}}^T x^{(j)} - \bar{w}_0 \sum_{i=1}^{n} \lambda_i y^{(i)} + \sum_{i=1}^{n} \lambda_i \\
&= \sum_{i=1}^{n} \lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j y^{(i)} y^{(j)} {x^{(i)}}^T x^{(j)} - \bar{w}_0 \sum_{i=1}^{n} \lambda_i y^{(i)} \\
&= \sum_{i=1}^{n} \lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j y^{(i)} y^{(j)} {x^{(i)}}^T x^{(j)}
\end{aligned}
$$

wobei sich die letzte Gleichung damit ergibt, dass an der Stelle $\bar{w}_0$ gilt, dass $\sum_{i=1}^{n} \lambda_i y^{(i)} = 0$.

Beweis (fortgeführt)

Wir haben damit gezeigt, dass sich die duale Zielfunktion des Maximum-Margin-Optimierungsproblems ergibt zu

$$q : \mathbb{R}^n \to \mathbb{R}, \lambda \to q(\lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y^{(i)} y^{(j)} {x^{(i)}}^T x^{(j)}. \tag{49}$$

(3) Das duale Problem

Das duale Problem zur Maximum-Margin-Klassifikation hat also die Form

$$\max_{\lambda \in \mathbb{R}} q(\lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y^{(i)} y^{(j)} {x^{(i)}}^T x^{(j)} \tag{50}$$

u.d.N.

$$\lambda_i \geq 0, i = 1, ..., n \text{ und } \sum_{i=1}^{n} \lambda_i y^{(i)} = 0 \tag{51}$$

wobei die letzte Nebenbedingung das Minimum der Lagrangefunktion bezüglich w_0 sicherstellt.

Maximum-Margin-Klassifikation als quadratische Optimierungsproblem

Beweis (fortgeführt)

(4) Formeln für die optimalen Gewichts- und Biasparameter

Eine Lösung des dualen Problems zur Maximum-Margin-Klassifikation entspricht dem optimalen Lagrangemultiplikatorenvektoren

$$\bar{\lambda} = \arg\max_{\lambda \in \mathbb{R}^n} q(\lambda) = \arg\max_{\lambda \in \mathbb{R}^n} L(\bar{w}, \bar{w}_0, \lambda). \tag{52}$$

Basierend auf der Minimierung von L bezüglich w ergibt sich dann für den optimalen Gewichtsparameter

$$\bar{w} = \sum_{i=1}^{n} \bar{\lambda}_i y^{(i)} x^{(i)} \tag{53}$$

Für den the optimal Biasparameter $\bar{w}_0$ halten wir zunächst fest, dass für alle $\lambda_k > 0, k = 1, \ldots, K$ mit den KKT Bedingungen gelten

$$\begin{aligned} & y^{(k)}(\bar{w}^T x^{(k)} + w_0) - 1 = 0 \\ \Leftrightarrow\ & y^{(k)}(\bar{w}^T x^{(k)} + w_0) = 1 \\ \Leftrightarrow\ & y^{(k)} y^{(k)}(\bar{w}^T x^{(k)} + w_0) = y^{(k)} \\ \Leftrightarrow\ & \bar{w}^T x^{(k)} + w_0 = y^{(k)} \end{aligned} \tag{54}$$

Dies impliziert zum einen, dass alle $x^{(k)}$ mit $\lambda_k > 0$ Supportvektoren sind, da ihr Abstant zur optimalen Hyperebene durch 1 gegeben ist und zum anderen, dass

$$\sum_{k=1}^{K} \bar{w}^T x^{(k)} + K w_0 = \sum_{k=1}^{K} y^{(k)} \Leftrightarrow w_0 = \frac{1}{K} \sum_{k=1}^{K} \left(y^{(k)} - \bar{w}^T x^{(k)} \right). \tag{55}$$

Theorem (Maximum-Margin-Klassifikation als quadratisches Programm)

Mit der Definition von

$$y := \left(y^{(i)}\right)_{i=1,\ldots,n} \in \mathbb{R}^n \text{ und } K := \left(x^{(i)^T} x^{(j)}\right)_{i,j=1,\ldots,n} \in \mathbb{R}^{n \times n}, \tag{56}$$

as well as

$$P := yy^T K \in \mathbb{R}^{n \times n}, q := -1_n, G := -I_n, h := 0_n, A := y^T, \text{ und } b := 0 \tag{57}$$

kann das duale Problem des Maximum-Margin-Optimierungsproblems als das quadratische Programm

$$\min_{\lambda \in \mathbb{R}^n} \frac{1}{2} \lambda^T P \lambda + q^T \lambda \text{ u.d.N. } G\lambda \leq h \text{ und } A\lambda = b \tag{58}$$

geschrieben werden und mit Standardimplementationen zur Lösung quadratischer Programme behandelt werden.

Maximum-Margin-Klassifikation als quadratische Optimierungsproblem

Beweis

Die Äquivalenzen

$$\begin{aligned}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y^{(i)}y^{(j)}{x^{(i)}}^T x^{(j)} &\Leftrightarrow \lambda^T yy^T K\lambda \Leftrightarrow \lambda^T P\lambda \\ \sum_{i=1}^{n}\lambda_i &\Leftrightarrow 1_n^T\lambda \Leftrightarrow q^T\lambda \\ \lambda \geq 0 &\Leftrightarrow -I_n \leq 0_n \Leftrightarrow G\lambda \leq h \\ \sum_{i=1}^{n}\lambda_i y^{(i)} = 0 &\Leftrightarrow y^T\lambda = 0 \Leftrightarrow A\lambda = b\end{aligned} \tag{59}$$

ergeben sich direkt mit den Regeln der Matrixmultiplikation.

□

Referenzen

Boser, Bernhard E, Isabelle M Guyon, and Vladimir N Vapnik. 1992. "A Training Algorithm for Optimal Margin Classifiers," 9.

Chang, Chih-Chung, and Chih-Jen Lin. 2011. "LIBSVM: A Library for Support Vector Machines." *ACM Transactions on Intelligent Systems and Technology* 2 (3): 1–27. https://doi.org/10.1145/1961189.1961199.

Cortes, Corinna, and Vladimir Vapnik. 1995. "Support-Vector Networks." *Machine Learning* 20 (3): 273–97. https://doi.org/10.1007/BF00994018.

Franc, Vojtech, Alex Zien, and Bernhard Schölkopf. 2011. "Support Vector Machines as Probabilistic Models." In *Proceedings of the 28 Th International Conference on Machine Learning*. Bellevue, WA.

Schölkopf, Bernhard, and Alexander J. Smola. 2002. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Adaptive Computation and Machine Learning. Cambridge, Mass: MIT Press.

Vapnik, Vladimir, and A Chervonenkis. 1964. "On a Class of Perceptrons." *Automation and Remote Control* 25 (1): 103–9.