

Support-Vektor Maschinen

Feature Funktionen

Beispiele werden durch Zusatzinformationen, **Features**, aufbereitet.

Eine Funktion

$$\phi : X \rightarrow \mathbb{R}^N$$

heißt eine **Feature-Funktion**.

$$\phi(x) = (\phi_1(x), \dots, \phi_N(x))$$

ist der **Feature-Vektor** des Beispiels x und

$$\phi(X) = \{\phi(x) \mid x \in X\}$$

ist der **Feature-Raum**.

Der Lernalgorithmus erhält die klassifizierten Feature-Vektoren $\phi(x_1), \dots, \phi(x_s)$ anstatt der ursprünglichen Beispiele $x_1, \dots, x_s \in X$.

Feature-Funktionen: Der Bag-of-Words Ansatz

Atomarer Text (oder **Atome**) eines Textes T sind sinntragende Teilworte von T wie *Stammsilben* oder ganze *Worte*.

- Für ein Atom s und ein Dokument $x \in D$ aus einer Menge D von Dokumenten ist
 - ▶ $h_s(x)$ die Häufigkeit des Atoms s im Dokument x ,
 - ▶ $D(s)$ die Anzahl untersuchter Dokumente mit mindestens einem Auftreten des Atoms s und
 - ▶ $\kappa(x)$ eine Normalisierungskonstante.
- Für ein Dokument x ist $\phi(x) = (\phi_s(x) \mid s)$ ein Feature-Vektor mit

$$\phi_s(x) = \frac{h_s(x) \cdot \log_2\left(\frac{|D|}{|D(s)|}\right)}{\kappa(x)}$$

$\phi_s(x)$ gewichtet die Häufigkeit des Atoms s im Dokument x mit dem Informationsgehalt $\log_2\left(\frac{|D|}{|D(s)|}\right)$ des Atoms.

Feature-Funktionen: Interpolation

Wir erhalten Paare, die aus einem Punkt $x \in X \subseteq \mathbb{R}^3$ und einem Wert $y = p(x) \in \mathbb{R}$ bestehen. Gesucht ist ein Polynom p vom Grad ≤ 2 .

- Für $x = (x_1, x_2, x_3)$ wählen wir die Feature-Funktion

$$\phi(x) := (x_1^2, x_2^2, x_3^2, x_1 \cdot x_2, x_1 \cdot x_3, x_2 \cdot x_3, x_1, x_2, x_3, 1).$$

- Wir suchen also nach Koeffizienten c_i , so dass

$$(c_1 \cdot x_1^2 + c_2 \cdot x_2^2 + c_3 \cdot x_3^2) + (c_4 \cdot x_1 \cdot x_2 + c_5 \cdot x_1 \cdot x_3 + c_6 \cdot x_2 \cdot x_3) + (c_7 \cdot x_1 + c_8 \cdot x_2 + c_9 \cdot x_3) + c_0 = y$$

für alle Beispiele $x \in X$ mit Klassifikation y gilt.

Das unbekannte Polynom wird zu einer linearen Funktion, die durch ein lineares Gleichungssystem bestimmt werden kann.

Support-Vektor Maschinen: Die Grundidee

Eine binäre Klassifizierung $f : X \rightarrow \{0, 1\}$ ist zu lernen.

1. Der Entwickler konstruiert eine Feature Funktion $\phi : X \rightarrow \mathbb{R}^N$.
2. In $\{\phi(x_1), \dots, \phi(x_s)\}$ sind die positiven von den negativen Beispielen mit Hilfe einer Hyperebene im \mathbb{R}^N „bestmöglich“ zu trennen.

+ Halbräume sind die mächtigste, effizient lernbare Hypothesenklasse.

✓ Benutze lineare Programmierung.

? Für großes N ist der Feature-Raum $\phi(X)$ **hochdimensional**:

! Die Bestimmung einer trennenden Hyperebene ist aufwändig.

! Sind viele Beispiele notwendig? **Overfitting droht!**

Die Grundidee: Klassifikation mit großem Margin

Bestimme eine trennende Hyperebene, die die positiven von den negativen Beispielen mit **möglichst großem Margin** ρ trennt.

(1) Der Perzeptron-Algorithmus benötigt höchstens

$$\left(\frac{R}{\rho}\right)^2$$

Gegenbeispiele, wenn alle Beispiele die Norm $\leq R$ besitzen.

- ▶ Wenige Gegenbeispiele bei entsprechend großem Margin ρ .
- ▶ *Abhängigkeit von der Dimension des Feature Raums nur indirekt, nämlich über den Margin ρ und den Radius R !*

(2) Kein Overfitting für relativ kleines R und relativ großes ρ ?

Aber der Aufwand in der Bestimmung einer trennenden Hyperebene im hochdimensionalen Feature-Raum ist doch hoch, oder?!

Der Kernel-Trick

Was passiert *im Beispielraum* X bei der Trennung der positiven von den negativen Feature-Vektoren durch

$$F(x) = \langle w, \phi(x) \rangle + t?$$

Der Perzeptron Algorithmus bestimmt

$$w = \sum_{i=1}^s \alpha_i \cdot \phi(x_i)$$

als Linearkombination der Gegenbeispiele $\phi(x_1), \dots, \phi(x_s) \implies$

$$\begin{aligned} F(x) &= \langle w, \phi(x) \rangle + t = \left\langle \sum_{i=1}^s \alpha_i \cdot \phi(x_i), \phi(x) \right\rangle + t \\ &= \sum_{i=1}^s \alpha_i \cdot \underbrace{\langle \phi(x_i), \phi(x) \rangle}_{=: K(x_i, x)} + t. \end{aligned}$$

Definiere den **Kern** K durch $K(x, z) := \langle \phi(x), \phi(z) \rangle$.

Aus $F(x) = \sum_{i=1}^s \alpha_i \cdot \langle \phi(x_i), \phi(x) \rangle + t$ folgt

$$F(x) = \sum_{i=1}^s \alpha_i \cdot K(x_i, x) + t.$$

? Wann kann $K(x_i, x)$ schnell berechnet werden?

? Welche Funktionen K kommen in Frage?

- ▶ Offensichtlich ist $K(x, z) = \langle x, z \rangle$ eine Möglichkeit.
- ▶ Später: **polynomielle Kerne** $K(x, z) = (\langle x, z \rangle)^d$,
der **Gauß-Kern** $K(x, z) = e^{-\frac{\|x-z\|^2}{2\sigma^2}}$ und viele weitere Funktionen.

Im Beispielraum trennen wir also unter Umständen mit

komplexen Hyperflächen $\{x \mid \sum_{i=1}^s \alpha_i \cdot K(x_i, x) = -t\}$.

Wir möchten eine Signatur für einen Computer-Virus erstellen.

Für ein Alphabet Σ sei X eine Menge $X \subseteq \Sigma^*$ von Dateien für die wir wissen, ob sie befallen sind oder nicht.

Für eine Zahl d wähle die Eigenschaft

„ein Wort $u \in \Sigma^{\leq d}$ als Teilwort zu haben“

als Feature und definiere die Feature-Funktion

$$\phi(x) := (b_v \mid v \in \Sigma^{\leq d}),$$

wobei $b_v \in \{0, 1\}$ und $b_v = 1 \iff v$ ist ein Teilwort von x .

:-) Der Feature-Raum hat die viel zu große Dimension $\approx |\Sigma|^d$.

:-) $\langle \phi(x), \phi(x') \rangle =$ Anzahl gemeinsamer Teilstrings für x und $x' \implies$ der Kern lässt sich in Zeit $\mathcal{O}(d \cdot |x| \cdot |x'|)$ berechnen.

Lernen mit großem Margin

Wenn positive und negative Feature-Vektoren linear trennbar sind:

1. Bestimme eine Feature-Funktion $\phi : X \rightarrow \mathbb{R}$.
2. Fordere klassifizierte Beispiele $(\phi(x_1), f(x_1)), \dots, (\phi(x_s), f(x_s))$ an.
3. Trenne positive und negative Beispiele mit möglichst großem Margin:
 - ▶ Wenn die lineare Funktion $\langle w, z \rangle + t$ mit $\|w\| = 1$ den Margin $\rho > 0$ erreicht, dann ist $f(x_i) \cdot (\langle w, \phi(x_i) \rangle + t) \geq \rho$ für alle i .
 - ▶ Deshalb löse das Maximierungsproblem

maximiere w, t ρ so dass

$$\|w\| = 1 \text{ und}$$

$$\text{für jedes } i, (1 \leq i \leq s): f(x_i) \cdot (\langle w, \phi(x_i) \rangle + t) \geq \rho.$$

Das Optimierungsproblem hat nur lineare Nebenbedingungen bis auf die Bedingung $\|w\| = 1$.

- ? Wir können die Bedingung $\|w\| = 1$ durch $\|w\| \leq 1$ ersetzen: Die Bedingung ist zumindest konvex.
- ✓ Statt ρ unter der Nebenbedingung $\|w\| \leq 1$ zu maximieren, minimiere $\|w\|$ unter der Nebenbedingung $\rho \geq 1$:

$$\begin{aligned} &\text{minimiere}_{w,t} \|w\|^2 \quad \text{sodass} \\ &\quad \text{für jedes } i, (1 \leq i \leq s): f(x_i) \cdot (\langle w, \phi(x_i) \rangle + t) \geq 1. \end{aligned}$$

- + Das Optimierungsproblem ist gutartig: Eine konvexe quadratische Form ist unter linearen Nebenbedingungen zu minimieren,
- ? aber im hochdimensionalen Feature-Raum!

Bisherige Annahme: Vollständige Trennbarkeit (**Hard Margin**). Jetzt:

Nur **partielle** Trennbarkeit (**Soft Margin**)

Füge Slack Variablen ξ_j für jedes Beispiel hinzu und löse

$$\text{minimiere}_{w,t,\xi} \|w\|^2 + C \cdot \sum_{i=1}^m \xi_i \quad \text{sodass}$$

$$\begin{aligned} &\text{für jedes } i: f(x_i) \cdot (\langle w, \phi(x_i) \rangle + t) \geq 1 - \xi_i \\ &\text{und } \xi \geq 0. \end{aligned}$$

- Je größer ξ_j umso schlechter die Klassifizierung von $\phi(x_j)$.
- Der Parameter C definiert wie stark falsche Klassifizierungen bestraft werden: Bestimme C mit **Validierung**.

Beispielkomplexität

Wird der Margin ρ für ein Beispiel $\phi(x)$

- eingehalten.
- unterschritten,
- bzw. wird $\phi(x)$ sogar falsch klassifiziert?

Für eine positive reelle Zahl ist

$$\Phi_{\rho}(z) := \begin{cases} 0 & \rho < z \\ 1 - \frac{z}{\rho} & 0 \leq z \leq \rho \\ 1 & z < 0. \end{cases}$$

Im Folgenden: $h(x) = \langle w, x \rangle$ mit $\|w\| \leq 1$ sei eine lineare Funktion.

Der empirische Margin-Loss

- (a) Für eine Hypothese h mit $h : \mathbb{R}^N \rightarrow \mathbb{R}$, eine Verteilung D auf X und eine Beispielmenge $S = \{x_1, \dots, x_s\} \subseteq X$ ist

$$\text{Loss}^{S,\rho}(h) := \frac{1}{s} \cdot \sum_{i=1}^s \Phi_\rho \left(f(x_i) h(\phi(x_i)) \right)$$

der **empirische Margin-Loss**.

- (b) Des weiteren ist

$$\text{Loss}_D(h) = \mathbb{E}_{x \sim D} \left[\Phi_\rho \left(f(x) h(\phi(x)) \right) \right]$$

der **erwartete Hinge^a Loss**.

^a „hinge“ (engl.) steht für „Scharnier“ oder „Gelenk“.

Der erwartete 0-1 Loss ist nicht größer als der erwartete Hinge Loss.

Der Verallgemeinerungsfehler für SVMs

Es gelte $\|\phi(x)\| \leq R$ für alle $x \in X$ und es sei $\rho > 0$.

Bei s Beispielen gilt für jedes $\delta > 0$ und jede lineare Funktion $h(x) = \langle w, x \rangle$ mit $\|w\| \leq 1$,

$$\underbrace{\text{Loss}_D(h)}_{\text{erwarteter Hinge Loss}} \leq \underbrace{\text{Loss}^{S,\rho}(h)}_{\text{empirischer Margin-Loss}} + \frac{1}{\sqrt{s}} \cdot \left(2\sqrt{\frac{R^2}{\rho^2}} + \sqrt{\frac{\ln \frac{1}{\delta}}{2}} \right)$$

mit Wahrscheinlichkeit mindestens $1 - \delta$.

- $\frac{R}{\rho} \ll \sqrt{s}$, sonst ist Schranke bedeutungslos.
- 0-1 Loss \leq Hinge-Loss
 - ▶ Wir erhalten eine obere Schranke für den wahren Fehler.
- Neue Perspektive: Keine Forderung an die Beispielszahl, sondern eine Aussage über den Fehler.
 - ▶ Ist die Aussage gut oder schlecht?

Gute oder schlechte Fehlerschranke?

$$\underbrace{\text{Loss}_D(h)}_{\text{erwarteter Hinge Loss}} \leq \underbrace{\text{Loss}^{S,\rho}(h)}_{\text{empirischer Margin-Loss}} + \frac{1}{\sqrt{s}} \cdot \left(2\sqrt{\frac{R^2}{\rho^2}} + \sqrt{\frac{\ln \frac{1}{\delta}}{2}} \right)$$

Eine heuristische Rechnung führt auf $\varepsilon \leq \frac{1}{\sqrt{s}} \cdot \left(2\sqrt{\frac{R^2}{\rho^2}} + \sqrt{\frac{\ln \frac{1}{\delta}}{2}} \right) \implies$

$$\begin{aligned} \varepsilon^2 &\leq \frac{1}{s} \cdot \left(2\sqrt{\frac{R^2}{\rho^2}} + \sqrt{\frac{\ln \frac{1}{\delta}}{2}} \right)^2 \\ &\approx \frac{1}{s} \cdot \left(\frac{4R^2}{\rho^2} + \frac{\ln \frac{1}{\delta}}{2} \right) = \mathcal{O} \left(\frac{1}{s} \cdot \left(\frac{R^2}{\rho^2} + \ln \frac{1}{\delta} \right) \right) \end{aligned}$$

- Fehlerschranke wie im agnostischen Lernmodell.
- Schlechte Schranke, wenn R^2/ρ^2 sehr viel größer als die Dimension des Feature-Raums ist.

Fehlerschranke im Hard-Margin-Fall

- 1 $\text{LINEAR}_\rho(\text{Def})$ ist die Klasse aller Threshold-Funktionen mit Definitionsbereich Def und Margin ρ auf der Teilmenge $\text{Def} \subseteq \mathbb{R}^n$.
- 2 $\text{VC}(\text{LINEAR}_\rho(\text{Def})) \leq \frac{R^2}{\rho^2}$.

Im Idealfall gehört die Zielfunktion f zur Klasse $\text{LINEAR}_\rho(\text{Def}) \implies$

$$s \approx \frac{1}{\varepsilon} \cdot \left(\frac{R^2}{\rho^2} \ln\left(\frac{1}{\varepsilon}\right) + \ln\left(\frac{1}{\delta}\right) \right) \text{ bzw. } \varepsilon \approx \frac{1}{s} \cdot \left(\frac{R^2}{\rho^2} \ln\left(\frac{1}{\varepsilon}\right) + \ln\left(\frac{1}{\delta}\right) \right)$$

Bis auf logarithmische Faktoren gilt

$$\varepsilon \approx \frac{1}{s} \cdot \left(\frac{R^2}{\rho^2} + \ln\left(\frac{1}{\delta}\right) \right).$$

Was ist getan, was bleibt zu tun?

Ein vorläufiges Fazit:

- ✓ ERM-Hypothesen werden bestimmt.
- ✓ Hochwahrscheinlicher Lernerfolg bei kleinem empirischen Margin-Loss.
- ✓ Effizientes Training mit linearer Programmierung.
 - Aber wenn der Feature-Raum hoch-dimensional ist?????

1. Wie sehen typische Anwendungen aus?
2. Welche Funktionen können als Kern verwandt werden?
3. Wie findet man im hochdimensionalen Feature-Raum **schnell** eine lineare Trennung mit großem Margin?

Anwendungen

Text Klassifikation: Der Bag-of-Words Ansatz

Zwei Anwendungen:

- Klassifiziere Nachrichten einer Nachrichtenagentur in verschiedene Kategorien.
- Ordne medizinische Dokumente einer von 23 Krankheiten zu.

(1) Der Bag-of-Words Ansatz besitzt für jede Stammsilbe s und jedes Dokument x das Feature

$$\phi_s(x) = \frac{h_s(x) \cdot \log_2\left(\frac{|D|}{|D(s)|}\right)}{\kappa(x)}$$

(2) Das innere Produkt

$$\langle \phi(x), \phi(z) \rangle = \sum_s \phi_s(x) \cdot \phi_s(z)$$

wird benutzt. Auch polynomielle Kerne und der Gauß-Kern erreichen eine ähnliche Leistung: Die Wahl des Kerns ist nicht entscheidend.

(1) Nachrichtenklassifikation:

- ▶ Training mit 9603 Nachrichten der Agentur Reuters, Evaluierung mit 3299 Nachrichten.
- ▶ Durchschnittliche Nachrichtenlänge ungefähr 200 Worte.
- ▶ Die 10,000 Stammsilben mit größtem Informationsgehalt (auf der Trainingsmenge) werden ausgewählt.

(2) Krankheitszuordnung:

- ▶ 10,000 medizinische Dokumente werden zum Training und 10,000 Dokumente zur Evaluierung benutzt.
- ▶ 15,561 Stammsilben werden ausgewählt, wobei jede Stammsilbe in mindestens drei Dokumenten vorkommt.

In beiden Fällen ist die Lernleistung konventionellen Ansätzen (Bayes-Verfahren, Rocchio, C4.5 und k -nearest Neighbor) überlegen.

- 100 Objekte werden aus 72 verschiedenen Blickwinkeln aufgenommen.
 - Jedes Bild ist einem der Objekte zuzuordnen.
-
- Die Feature Funktion ϕ :
 - ▶ Die 7200 Bilder werden, nach Durchschnittsbildung auf 4×4 Gittern, von einer Auflösung von 128×128 Pixel auf 32×32 Pixel reduziert.
 - ▶ Die Feature-Funktion weist jedem Bild also einen Vektor von 1024 Graustufen zu.
 - Die Objekte sind aufgrund der großen Zahl der Features bereits mit dem linearen Kern trennbar.

- 1400 Photos der Corel Stock Photo Collection sind verschiedenen Kategorien zuzuweisen.
- 2/3 der Photos werden zum Training und 1/3 zum Test verwandt.

(1) Die Feature Funktion:

- ▶ Jedem Pixel wird sein HSV-Wert (Hue Saturation Value) zugewiesen. HSV-Werte sind „biologisch plausibler“ als RGB-Werte, da auch Helligkeitsinformation benutzt wird.
- ▶ Der 3-dimensionale Würfel der HSV-Werte wird in 4096 Farbbereiche zerlegt.
- ▶ Jedem Bild wird ein Histogramm als Feature-Vektor zugeordnet, das die Häufigkeit für jeden Farbbereich wiedergibt.
- ▶ Der Feature-Vektor ist translations-invariant.

(2) Der Kern $K(x, z) = \exp^{-||x-z||_1}$ mit $||x - z||_1 = \sum_i |x_i - z_i|$ wird erfolgreich eingesetzt.

- ▶ Die Wahl des Kerns ist kritisch.

- In einem Benchmark-Datensatz des US Postal Service für die Ziffererkennung treten 7291 Trainings- und 2007 Testbeispiele auf.
- Jedes Beispiel des Datensatzes wird als 16×16 Pixelmatrix mit 256 Graustufen repräsentiert.

- (1) Die Feature-Funktion: Es ist $\phi(x) = x$. Die Features entsprechen somit den Grauwerten der einzelnen Pixel.
- (2) Es werden polynomielle Kerne und der Gauß-Kern benutzt:
 - ▶ Für polynomielle Kerne ergibt sich lineare Separierbarkeit erst für den Grad $d \geq 3$.
- (3) Support-Vektor Maschinen werden somit ohne problem-spezifische Aufbereitung eingesetzt.

Die Lernleistung ist nur wenig schwächer als die der besten maßgeschneiderten Methoden.

Kerne

Sei V ein Vektorraum.

(a) $\langle -, - \rangle : V^2 \rightarrow \mathbb{R}$ ist ein **inneres Produkt** für V , falls

(*) $\langle -, - \rangle$ bilinear ist, falls also

$$\begin{aligned}\langle \alpha \cdot x + \beta \cdot y, u \rangle &= \alpha \cdot \langle x, u \rangle + \beta \cdot \langle y, u \rangle \quad \text{und} \\ \langle x, \gamma \cdot u + \delta \cdot v \rangle &= \gamma \cdot \langle x, u \rangle + \delta \cdot \langle x, v \rangle\end{aligned}$$

für alle $\alpha, \beta, \delta, \gamma \in \mathbb{R}$ und Vektoren $x, y, u, v \in V$ gilt,

(*) $\langle x, u \rangle = \langle u, x \rangle$ für alle Vektoren $x, u \in V$ gilt und falls

(*) für alle $x \in V$: $\langle x, x \rangle \geq 0$ und $\langle x, x \rangle = 0 \iff x = 0$.

(b) Ein **Kern** für eine Menge X ist eine Funktion $K : X^2 \rightarrow \mathbb{R}$, wobei

$$K(x, z) = \langle \phi(x), \phi(z) \rangle$$

für eine Funktion $\phi : X \rightarrow \mathbb{R}^N$ und ein inneres Produkt $\langle -, - \rangle$
(über einem Vektorraum V mit $X \subseteq V$).

$$\begin{aligned}\beta(x, z) &:= \frac{K(x, z)}{\sqrt{K(x, x)} \cdot \sqrt{K(z, z)}} = \frac{\langle \phi(x), \phi(z) \rangle}{\sqrt{\langle \phi(x), \phi(x) \rangle} \cdot \sqrt{\langle \phi(z), \phi(z) \rangle}} \\ &= \left\langle \frac{\phi(x)}{\|\phi(x)\|}, \frac{\phi(z)}{\|\phi(z)\|} \right\rangle = \text{Kosinus} \left(\phi(x), \phi(z) \right) \in [-1, 1].\end{aligned}$$

- Die Vektoren $\phi(x)$ und $\phi(z)$ – und damit die Beispiele x und z – sind „ähnlich“, wenn $\alpha(x, z)$ nahe bei Eins liegt.
- Positive und negative Beispiele sollten jeweils untereinander ähnlich sein, um eine lineare Trennung zu erlauben.

Wann ist K ein Kern? Der Satz von Mercer

Kerne: Auf welche Eigenschaften kommt es an?

Sei $S = \{x_1, \dots, x_s\} \subseteq X$ eine Beispielmenge und $K : X \times X \rightarrow \mathbb{R}$ eine Funktion.

Die $s \times s$ -**Gram-Matrix** G_K^S wird definiert durch

$$G_K^S[i, j] := K(x_i, x_j).$$

Wenn K ein Kern ist, dann ist K **symmetrisch** und die Gram-Matrix G_K^S ist **positiv semidefinit**, d.h. $u^T \cdot G_K^S \cdot u \geq 0$ gilt für alle Vektoren u :

$$\begin{aligned} u^T \cdot G_K^S \cdot u &= \sum_{i,j} u_i \cdot G_K^S[i, j] \cdot u_j = \sum_{i,j} u_i \cdot \langle \phi(x_i), \phi(x_j) \rangle \cdot u_j \\ &= \underbrace{\left\langle \sum_i u_i \cdot \phi(x_i), \sum_j u_j \cdot \phi(x_j) \right\rangle}_{=x} = \langle x, x \rangle \geq 0. \end{aligned}$$

Der Satz von Mercer

Die Funktion $K : X \times X \rightarrow \mathbb{R}$ sei symmetrisch. Dann gilt:

K ist ein Kern $\iff G_K^S$ ist für jede endliche Teilmenge $S \subseteq X$ positiv semidefinit.

\implies ✓

\longleftarrow Definiere $\mathbb{R}^X := \{ f : X \rightarrow \mathbb{R} \}$ und $\psi(x) := K(*, x)$.

1. Sei V der von den Funktionen $\psi(x) \in \mathbb{R}^X$ aufgespannte Unterraum von \mathbb{R}^X . (Interpretiere ψ als Feature-Funktion.) Dann ist

$$\left\langle \sum_i \alpha_i \cdot \psi(x_i), \sum_i \beta_i \cdot \psi(x'_i) \right\rangle^* := \sum_{i,j} \alpha_i \beta_j \cdot K(x_i, x'_j)$$

ein inneres Produkt auf V .

2. Es ist $K(x, x') = \langle \psi(x), \psi(x') \rangle^* \implies K$ ist ein Kern. □

Der Satz von Mercer: Erste Konsequenzen

α, β seien **nicht-negative** reelle Zahlen.

Wenn K_1 und K_2 Kerne sind, dann ist auch $K = \alpha \cdot K_1 + \beta \cdot K_2$ ein Kern.

Warum? Wir wenden den Satz von Mercer an. Sei S eine endliche Teilmenge von X . Es ist

$$\begin{aligned} u^T \cdot G_K^S \cdot u &= u^T \cdot \left(\alpha \cdot G_{K_1}^S + \beta \cdot G_{K_2}^S \right) \cdot u \\ &= \alpha \cdot \underbrace{\left(u^T \cdot G_{K_1}^S \cdot u \right)}_{\geq 0} + \beta \cdot \underbrace{\left(u^T \cdot G_{K_2}^S \cdot u \right)}_{\geq 0} \geq 0 \end{aligned}$$

und das war zu zeigen.

Symmetrische, positiv semidefinite Matrizen

1. Symmetrische Matrizen B sind diagonalisierbar, d.h. es gibt eine orthogonale Matrix U und eine Diagonalmatrix D mit

$$B = U^T \cdot D \cdot U.$$

2. Wenn B positiv semidefinit ist, dann sind alle Eigenwerte nichtnegativ.

$$B = (U^T \cdot \sqrt{D}) \cdot (\sqrt{D} \cdot U).$$

B ist genau dann symmetrisch und positiv semidefinit, wenn es eine Matrix M gibt mit

$$B = M^T \cdot M.$$

Weitere Konsequenzen

Wenn K_1 und K_2 Kerne sind, dann ist auch $K_1 \cdot K_2$ ein Kern.

$G_{K_1}^S$ ist positiv semidefinit $\implies G_{K_1}^S = M^T \cdot M$ für eine Matrix M .

$$\begin{aligned} & \sum_{i,j} u_i \left(G_{K_1}^S[i,j] \cdot G_{K_2}^S[i,j] \right) u_j = \sum_{i,j} u_i \left((M^T M)[i,j] \cdot G_{K_2}^S[i,j] \right) u_j \\ &= \sum_{i,j} u_i \left(\sum_k M^T[i,k] \cdot M[k,j] \cdot G_{K_2}^S[i,j] \right) u_j \\ &= \sum_k \left(\sum_{i,j} u_i M^T[i,k] \cdot M[k,j] \cdot G_{K_2}^S[i,j] u_j \right) \\ &= \sum_k \left(\underbrace{\sum_{i,j} u_i M^T[i,k] \cdot G_{K_2}^S[i,j] \cdot u_j M^T[j,k]}_{\geq 0} \right) \checkmark \end{aligned}$$

Und noch mehr Kerne

- (a) Sei K ein Kern und sei p ein Polynom mit nicht-negativen Koeffizienten. Dann ist auch $p(K)$ ein Kern.
- (b) $f : \mathbb{R} \rightarrow \mathbb{R}$ sei durch eine konvergierende Potenzreihe mit nicht-negativen Koeffizienten darstellbar. Dann ist auch $f(K)$ ein Kern.

(1) Kerne sind abgeschlossen unter Addition und Multiplikation. ✓

(2) $f = \lim_{n \rightarrow \infty} p_n$ für Polynome p_n .

- ▶ Für $S \subseteq X$ sind die Matrizen $G_{p_n(K)}^S$ positiv semidefinit.
- ▶ Es gilt $G_{f(K)}^S = \lim_{n \rightarrow \infty} G_{p_n(K)}^S$ und $G_{f(K)}^S$ ist positiv semidefinit, da

$$u^T \cdot G_{f(K)}^S \cdot u = u^T \cdot \left(\lim_{n \rightarrow \infty} G_{p_n(K)}^S \right) \cdot u = \lim_{n \rightarrow \infty} u^T \cdot G_{p_n(K)}^S \cdot u \geq 0.$$

Wir wissen jetzt, wie man aus bestehenden Kerne weitere Kerne baut.
Wir müssen diesen Prozess mit bestehenden Kernen starten!

- Sei $F : X \rightarrow \mathbb{R}$ eine beliebige Funktion. Dann ist

$$K(x, z) = F(x) \cdot F(z)$$

ein Kern: Benutze F als Feature-Funktion.

- Sei A eine symmetrische, positiv semidefinite Matrix. Dann ist

$$K(x, z) = x^T \cdot A \cdot z$$

ein Kern, denn

$$K(x, z) = x^T \cdot A \cdot z = x^T \cdot M^T \cdot M \cdot z = \langle M \cdot x, M \cdot z \rangle$$

ist ein Kern mit Feature-Funktion $\phi(x) := M \cdot x$.

(a) Sei K ein Kern. Dann ist auch $\exp^{K(x,z)}$ ein Kern.

(b) $\exp^{-\frac{\|x-z\|^2}{\sigma^2}}$ ist ein Kern und wird der Gauß-Kern genannt.

1. Die Exponentialfunktion hat die konv. Potenzreihe $\sum_{i=0}^{\infty} \frac{x^i}{i!}$. Alle Koeffizienten sind nicht-negativ und deshalb ist $\exp^{K(x,z)}$ ein Kern.

2. Es ist

$$\exp^{-\frac{\|x-z\|^2}{\sigma^2}} = \left(\exp^{-\frac{\|x\|^2}{\sigma^2}} \cdot \exp^{-\frac{\|z\|^2}{\sigma^2}} \right) \cdot \exp^{\frac{2 \cdot \langle x,z \rangle}{\sigma^2}}.$$

- ▶ Das erste Produkt ist ein Kern. Warum?
- ▶ Der dritte Faktor ist nach Teil (a) ein Kern und
- ▶ der Gauß-Kern ist als Produkt von Kernen ein Kern.

Was haben wir bisher erreicht?

- (1) Wir haben große Klassen von Kernfunktionen konstruiert, die in vielen Anwendungen Fast-Trennbarkeit garantieren.
- (2) Wir wissen, dass – *bis auf logarithmische Faktoren* – ungefähr

$$s \approx \frac{1}{\varepsilon^2} \cdot \left(\left(\frac{R}{\rho} \right)^2 + \ln\left(\frac{1}{\delta}\right) \right) \quad \text{bzw} \quad s \approx \frac{1}{\varepsilon} \cdot \left(\left(\frac{R}{\rho} \right)^2 + \ln\left(\frac{1}{\delta}\right) \right)$$

Beispiele im Soft-Margin-Fall bzw. Hard-Margin-Fall genügen \implies Erfolgreiches Lernen bei kleinem Margin-Loss.

Was ist noch zu tun? Die effiziente Berechnung einer linearen Trennung mit großem Margin in einem **hochdimensionalen** Feature-Raum!

- Der wichtige Gauß-Kern benötigt unendlich viele Features!

Konvexe Minimierung

- $f : \mathbb{R}^N \rightarrow \mathbb{R}$ ist **konvex**, falls für alle $x, y \in \mathbb{R}^N$ und alle $\lambda \in [0, 1]$
$$f(\lambda \cdot x + (1 - \lambda) \cdot y) \leq \lambda \cdot f(x) + (1 - \lambda) \cdot f(y),$$

falls also der Funktionsgraph stets unter oder auf der Sekante durch $(x, f(x))$ und $(y, f(y))$ liegt.

- Eine Menge $X \subseteq \mathbb{R}^N$ ist **konvex**, wenn für je zwei Punkte $x, y \in X$ die Strecke $\{\lambda \cdot x + (1 - \lambda) \cdot y \mid 0 \leq \lambda \leq 1\}$ von x nach y in X liegt.

(1) $\sum_i w_i^2$ ist konvex:

- ▶ Die Funktion $x \rightarrow x^2$ ist konvex
- ▶ und eine Summe konvexer Funktionen ist konvex.

(2) Der Durchschnitt konvexer Mengen ist konvex.

Das Ziel: minimiere_{w,t} $\frac{1}{2} \cdot \|w\|^2$
so dass für jedes i ($1 \leq i \leq s$): $f(x_i) \cdot (\langle w, \phi(x_i) \rangle + t) \geq 1$.

Welche speziellen Eigenschaften hat unser Minimierungsproblem?

- (1) Die Zielfunktion $f(w) := \frac{1}{2} \cdot \|w\|^2 = \sum_i w_i^2$ ist konvex.
- (2) Alle Restriktionen sind **lineare Ungleichungen**.
- (3) Eine konvexe Funktion ist über einer konvexen Menge, nämlich einem Durchschnitt von Halbräumen, zu minimieren.

Ein Minimierungsproblem heißt **konvex**, falls die Zielfunktion konvex ist und falls alle Nebenbedingungen konvex sind.

- (1) Wir müssen ein konvexes Minimierungsproblem lösen.
- (2) Jedes lokale Minimum x ist auch ein globales Minimum:
 - ▶ Ist x kein globales Minimum, dann gibt es eine Lösung y mit $f(y) < f(x)$.
 - ▶ Für die x und y verbindende Gerade gilt für $\lambda < 1$

$$f(\lambda \cdot x + (1 - \lambda) \cdot y) \leq \lambda \cdot f(x) + (1 - \lambda) \cdot f(y) < f(x)$$

und x ist kein lokales Minimum.

Das duale Problem

Das duale Problem

Unser Minimierungsproblem hat die allgemeine Form

$$\text{minimiere}_x f(x), \text{ so dass } h_j(x) \leq 0 \text{ f\"ur } j = 1, \dots, m, \quad (1)$$

wobei f, h_1, \dots, h_m konvex sind.

- (a) Hebe die Nebenbedingungen durch die nicht-negativen **Lagrange-Multiplikatoren** ξ_j in die Zielfunktion. Dann heit

$$L(x, \xi) = f(x) + \sum_j \xi_j \cdot h_j(x)$$

die **verallgemeinerte Lagrange-Funktion** von (1).

- (b) Das „duale“ Problem hat die Form

$$\text{maximiere}_{\xi \geq 0} \left(\text{minimiere}_x L(x, \xi) \right).$$

Was ist der Zusammenhang zum dualen Problem?

Für jedes $\xi \geq 0$ und jede Lösung y gilt

$$\min_x \left\{ f(x) + \sum_j \xi_j \cdot h_j(x) \right\} \leq f(y) + \sum_j \xi_j \cdot h_j(y) \leq f(y),$$

denn $h_1(y), \dots, h_m(y) \leq 0$ und $\xi \geq 0$.

Also gilt die **schwache Dualität**:

$$\begin{aligned} & \text{maximiere}_{\xi \geq 0} \left(\text{minimiere}_x L(x, \xi) \right) \\ & \leq \text{minimiere}_{x \text{ ist Lösung}} f(x). \end{aligned}$$

Wir zeigen, dass sogar Gleichheit gilt, d.h.

$$\begin{aligned} & \text{maximiere}_{\xi \geq 0} \left(\text{minimiere}_x L(x, \xi) \right) \\ & = \text{minimiere}_{x \text{ ist Lösung}} f(x). \end{aligned}$$

Die Karush-Kuhn-Tucker (KKT) Bedingungen

Die Funktionen f, h_1, \dots, h_m seien konvex und es gelte $h_1(x^*) \leq 0, \dots, h_m(x^*) \leq 0$. Dann sind äquivalent:

- (a) $f(x^*) = \min \{ f(x) : h_1(x) \leq 0, \dots, h_m(x) \leq 0 \}$.
- (b) Es gibt Lagrange-Multiplikatoren $\xi_1^*, \dots, \xi_m^* \geq 0$ mit
- (1) $\nabla f(x^*) + \sum_{j=1}^m \xi_j^* \cdot \nabla h_j(x^*) = \nabla L(x^*, \xi^*) = 0$ und
 - (2) $\xi_j^* \cdot h_j(x^*) = 0$ für alle j .

(a) \implies (b) **Wenn** das globale Minimum x^* ein innerer Punkt ist, d.h. wenn $h_j(x^*) < 0$ für alle j gilt:

- Der Gradient von f im Punkt x^* verschwindet: Es ist $\nabla f(x^*) = 0$.
- Setze $\xi_1^* = \dots = \xi_m^* = 0$ und die KKT-Bedingungen sind erfüllt.

Und **wenn nicht**?

(b) \implies (a) Siehe Skript.

Beweis der Dualität

f sei konvex mit globalem Minimum x^* und h_1, \dots, h_m seien konvex. Die Lagrange-Multiplikatoren seien $\xi_1^*, \dots, \xi_m^* \geq 0$.

(1) Dann ist $\nabla f(x^*) + \sum_{j=1}^m \xi_j^* \cdot \nabla h_j(x^*) = 0$.

(2) Es gilt

$$\min_y f(y) + \sum_j \xi_j^* \cdot h_j(y) = f(x^*) + \sum_j \xi_j^* \cdot h_j(x^*), \text{ da}$$

- ▶ $F(y) = f(y) + \sum_j \xi_j^* \cdot h_j(y)$ ist konvex, denn $\xi^* \geq 0$.
- ▶ Wegen (1) ist x^* ein lokales Minimum von F und damit auch ein globales Optimum.

(3) $\sum_j \xi_j^* \cdot h_j(x^*) = 0$, denn $\xi_j^* \cdot h_j^*(x) = 0$ für alle j (KKT). Also

$$\min_y f(y) + \sum_j \xi_j^* \cdot h_j(y) = f(x^*) + \underbrace{\sum_j \xi_j^* \cdot h_j(x^*)}_{=L(x^*, \xi^*)} = f(x^*).$$

Das primale und duale Problem

Aus den KKT-Bedingungen folgt

$$\begin{aligned} \max_{\xi \geq 0} \min_y L(y, \xi) &\geq \min_y L(y, \xi^*) = L(x^*, \xi^*) = f(x^*) \\ &\geq \max_{\xi \geq 0} \min_y L(y, \xi). \end{aligned}$$

schwache Dualität

Das **primale** Problem \min_x ist Lösung $f(x)$ werde von der Lösung x^* minimiert. Dann gilt der **Dualitätssatz**

$$\max_{\xi \geq 0} \left(\min_x L(x, \xi) \right) = L(x^*, \xi^*) = f(x^*).$$

Wir benötigen später „nur“ $\max_{\xi \geq 0} L(x^*, \xi) = f(x^*)$.

Eine äquivalente Form des Dualitätssatzes

Die Funktionen f und h_1, \dots, h_m seien konvex. Dann gilt

$$\max_{\xi \geq 0} \left(\min_x f(x) + \sum_j \xi_j \cdot h_j(x) \right) = \min_{x: h_j(x) \leq 0 \text{ für alle } j} f(x).$$

Es ist $\min_x \left(\max_{\xi \geq 0} f(x) + \sum_j \xi_j \cdot h_j(x) \right) = \min_{x: h_j(x) \leq 0 \text{ für alle } j} f(x) \implies$

Eine äquivalente Form des Dualitätssatzes:

$$\max_{\xi \geq 0} \left(\min_x f(x) + \sum_j \xi_j \cdot h_j(x) \right) = \min_x \left(\max_{\xi \geq 0} f(x) + \sum_j \xi_j \cdot h_j(x) \right)$$

Der Hard-Margin-Fall

Ist das duale Problem wirklich so kompliziert?

Es ist

$$\max_{\xi \geq 0} L(w^*, t^*, \xi) = \min_{w, t: h_1(w, t), \dots, h_m(w, t) \leq 0} \frac{1}{2} \cdot \langle w, w \rangle.$$

1. Wende die KKT-Bedingungen an, um w^* mit Hilfe von ξ^* zu eliminieren und
2. bestimme die Lagrange-Multiplikatoren ξ^* mit dem dualen Maximierungsproblem.

Die verallgemeinerte Lagrange Funktion für den Hard Margin Fall:

$$L(\mathbf{w}, t, \xi) = \frac{1}{2} \cdot \langle \mathbf{w}, \mathbf{w} \rangle - \sum_{i=1}^s \xi_i \cdot \left(f(x_i) \cdot (\langle \mathbf{w}, \phi(x_i) \rangle + t) - 1 \right).$$

Wende die KKT-Bedingungen an: Sei (\mathbf{w}^*, t^*) ein globales Minimum mit den Lagrange-Multiplikatoren $\xi_1^*, \dots, \xi_m^* \geq 0$.

$$\nabla_{\mathbf{w}} L(\mathbf{w}^*, t^*, \xi^*) = \mathbf{w}^* - \sum_{i=1}^s \xi_i^* \cdot f(x_i) \cdot \phi(x_i),$$

$$\nabla_t L(\mathbf{w}^*, t^*, \xi^*) = - \sum_{i=1}^s \xi_i^* \cdot f(x_i).$$

Es ist $\nabla_{\mathbf{w}} L(\mathbf{w}^*, t^*) = \mathbf{0}$ und $\nabla_t L(\mathbf{w}^*, t^*) = 0 \implies$

Bestimme \mathbf{w}^* nach Nullsetzung und t^* taucht nicht mehr auf.

Nach Rückeinsetzung ergibt sich

$$L(w^*, t^*, \xi^*) = -\frac{1}{2} \sum_{i,j=1}^s f(x_i) \xi_i^* \cdot \langle \phi(x_i), \phi(x_j) \rangle \cdot f(x_j) \xi_j^* + \sum_{i=1}^s \xi_i^*.$$

Das duale Problem $D := \max_{\xi \geq 0} (\min_{w,t} L(w, t, \xi))$ ist äquivalent zu

$$-\max_{\xi \geq 0} (\min_{w,t} L(w, t, \xi)) = -\max_{\xi \geq 0} L(w^*, t^*, \xi) = \min_{\xi \geq 0} -L(w^*, t^*, \xi).$$

$$D = \min_{\xi \geq 0} \frac{1}{2} \sum_{i,j=1}^s f(x_i) \xi_i \cdot \langle \phi(x_i), \phi(x_j) \rangle \cdot f(x_j) \xi_j - \sum_{i=1}^s \xi_i$$

so dass $\xi \geq 0$ und $\underbrace{\sum_{i=1}^s \xi_i \cdot f(x_i)}_{\text{KKT-Bedingung}} = 0.$

Es ist $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$. Das duale Problem hat also die Form

$$\text{minimiere}_{\xi} \quad \frac{1}{2} \sum_{i,j=1}^s f(x_i)\xi_i \cdot K(x_i, x_j) \cdot f(x_j)\xi_j - \sum_{i=1}^s \xi_i$$

$$\text{so dass } \xi \geq 0 \quad \text{und} \quad \sum_{i=1}^s \xi_i \cdot f(x_i) = 0.$$

Wir können im niedrig-dimensionalen Beispielraum minimieren.

- (1) Die Anzahl der Unbestimmten des Minimierungsproblems stimmt mit der Anzahl s der Beispiele überein.
- (2) Es gibt nur „wenige“, nämlich $s + 1$ Nebenbedingungen.

Support-Vektoren

Eine der KKT Bedingungen: $\xi_i^* \cdot (f(x_i) \cdot (\langle w^*, \phi(x_i) \rangle + t^*) - 1) = 0$.

(1) Setze $SV = \{ i \mid f(x_i) \cdot (\langle w^*, \phi(x_i) \rangle + t^*) = 1 \}$.

(2) Ein Beispiel x_i mit $i \in SV$ heißt ein **Support-Vektor**.

Es ist $\xi_i^* = 0$ für jedes $i \notin SV$ und deshalb

$$w^* = \sum_{i=1}^s f(x_i) \xi_i^* \cdot \phi(x_i) = \sum_{i \in SV} f(x_i) \xi_i^* \cdot \phi(x_i).$$

Nur die Support-Vektoren $\phi(x_i)$, also

*die am nächsten an der trennenden Hyperebene liegenden
Beispiele,*

bestimmen den optimalen Gewichtsvektor w^* .

Je weniger Support-Vektoren, umso wahrscheinlicher ist erfolgreiches Lernen.

(1) Warum?

- ▶ Die Hypothese kann durch wenige Beispiele beschrieben werden.
- ▶ Occam: Einfache Hypothesen machen erfolgreiches Lernen wahrscheinlicher.

(2) Eine Support-Vektor Hypothese erhält automatisch ein Gütesiegel bei wenigen Support-Vektoren.

Soft Margin

Im Feature-Raum zu lösen:

$$\min_{w, \beta, \eta} \|w\|^2 + C \cdot \sum_{i=1}^m \eta_i$$

so dass für jedes i : $f(x_i) \cdot (\langle w, \phi(x_i) \rangle + \beta) \geq 1 - \eta_i$ und $\eta_i \geq 0$.

Nach Analyse des dualen Programms löse im Beispielraum:

$$\min_{\xi \geq 0} \frac{1}{2} \sum_{i,j=1}^s f(x_i) \xi_i \cdot K(x_i, x_j) \cdot f(x_j) \xi_j - \sum_{i=1}^s \xi_i$$

sodass $0 \leq \xi_j \leq C$ für alle j und $\sum_{i=1}^s \xi_i \cdot f(x_i) = 0$.

Zusammenfassung

- (1) Der Anwender wählt eine Feature-Funktion ϕ und einen Kern ψ aus dem Werkzeugkasten (Polynome, Gauß-Funktion etc.)

$$K(\phi(x), \phi(z)) = \langle \psi(\phi(x)), \psi(\phi(z)) \rangle.$$

- (2) Die Bestimmung einer trennenden Hyperebene kann im „niedrig-dimensionalen“ Beispielraum durchgeführt werden.
- (3) Bei s Beispielen der Länge höchstens R und Margin ρ gilt

$$\underbrace{\text{Loss}_D(h)}_{\text{erwarteter Hinge Loss}} \leq \underbrace{\text{Loss}^{S,\rho}(h)}_{\text{empirischer Margin-Loss}} + \frac{1}{\sqrt{s}} \cdot \left(2\sqrt{\frac{R^2}{\rho^2}} + \sqrt{\frac{\ln \frac{1}{\delta}}{2}} \right)$$

mit Wahrscheinlichkeit mindestens $1 - \delta$.

- (4) Wenige Support-Vektoren machen erfolgreiches Lernen hochwahrscheinlich.