AGML, Sommersemester 2020 — Vorlesung 11 (4 Seiten)

# Kernel-Funktionen

Thomas Kesselheim — Letzte Aktualisierung: 29. Mai 2020

In vielen Fällen kann man mittels linearer Klassifikation keine genügend guten Vorhersagen treffen. Wir werden uns heute komplexere Klassifikatoren anschauen. Die zugrundeliegenden Optimierungsprobleme können wir allerdings auf lineare Klassifikation zurückführen.

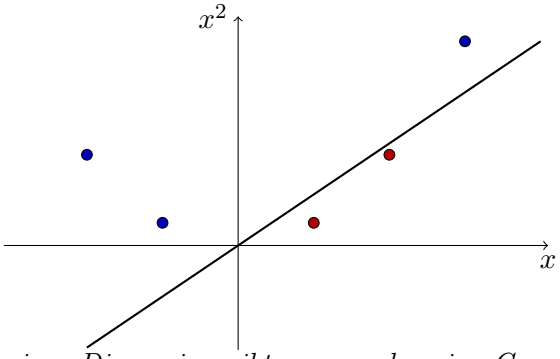
**Beispiel 11.1.** *Uns seien folgende Trainingsdaten gegeben:*

| $x_i$ | $y_i$ |
|---|---|
| $-2$ | $-1$ |
| $-1$ | $-1$ |
| $1$ | $+1$ |
| $2$ | $+1$ |
| $3$ | $-1$ |



*Hier ist lineare Klassifikation, also die Wahl einer Schwellenwertfunktion, offensichtlich keine sonderlich gute Idee. Es ist relativ offensichtlich, dass eigentlich ein Intervall gesucht wird. Interessant ist, dass ein Algorithmus dieses Intervall auch mittels linearer Klassikation finden kann, wenn wir als Merkmale $(x_i, x_i^2) \in \mathbb{R}^2$ ansehen.*

| $(x_i, x_i^2)$ | $y_i$ |
|---|---|
| $(-2, 4)$ | $-1$ |
| $(-1, 1)$ | $-1$ |
| $(1, 1)$ | $+1$ |
| $(2, 4)$ | $+1$ |
| $(3, 9)$ | $-1$ |




*Durch Hinzunahme einer Dimension gibt es nun also eine Gerade, die die Punkte separiert.*

## 1 Einbettungen und Feature Space

Anstatt lineare Klassifikation über dem Merkmalsraum $X$ betrachten wir diese nun über einem *Feature Space* $F$; zunächst ist $F = \mathbb{R}^n$, wobei $n \in \mathbb{N}$ unterschiedliche groß sein kann. Dazu ist uns eine Einbettung $\psi\colon X \to F$ gegeben.

**Beispiel 11.2.**

- *Im oben Beispiel ist $X = \mathbb{R}$, $F = \mathbb{R}^2$, $\psi(x) = (x, x^2)$.*
- *Eine Einbettung, über die wir schon implizit gesprochen haben, ist die folgende. Ist $X = \mathbb{R}^d$, können wir $F = \mathbb{R}^{d+1}$ und $\psi(\mathbf{x}) = (\mathbf{x}, 1)$ betrachten. Das heißt, wir fügen jedem $\mathbf{x}$-Vektor als letzte Komponente eine 1 an. Jetzt können wir uns auf lineare Klassifikation mittels Hyperebenen beschränken, die durch den Ursprung gehen.*
- *Allgemeiner können wir polynomielle Einbettungen betrachten. Sei dafür $X = \mathbb{R}^d$ und $k \in \mathbb{N}$ fest. Nun definieren wir $\psi(\mathbf{x})$ als den Vektor, dessen Komponenten alle möglichen Formen $\prod_{i=1}^d x_i^{j_i} = x_1^{j_1} \cdot x_2^{j_2} \cdot \ldots \cdot x_d^{j_d}$ mit $0 \leq j_i \leq k$ für alle $i$ hat. Die Dimension von $F$ ist $n = (k+1)^d$, kann also leicht sehr groß werden. Konkret können wir $d = 2$ und $k = 2$ anschauen, dann ist $\psi(x_1, x_2) = (1, x_1, x_1^2, x_2, x_1x_2, x_1^2x_2, x_2^2, x_1x_2^2, x_1^2x_2^2)$.*

- *Es könnte aber auch $X$ die Menge aller E-Mails sein und $F$ könnte ein Vektor irgendwelcher Eigenschaften sein, beispielsweise wie oft das gewisse Wörter vorkommen.*

Der Lernalgorithmus, der eine Einbettung $\psi$ benutzt, könnte also wie folgt aussehen:

1. Berechne die Einbettung der Trainingsdaten. Sei die eingebettete Trainingsmenge $\hat{S}$ entsprechend definiert als $(\psi(\mathbf{x}_1), y_1), \ldots, (\psi(\mathbf{x}_m), y_m)$.
2. Finde einen möglichst guten linearen Klassifikator $h_{\mathbf{w}} \colon F \to \{-1, +1\}$, mit Trainingsmenge $\hat{S}$.
3. Gib Hypothese $h \colon X \to \{-1, +1\}$ zurück mit

$$h(\mathbf{x}) = \begin{cases} +1 & \text{falls } \langle \mathbf{w}, \psi(x) \rangle \geq 0 \\ -1 & \text{sonst} \end{cases} .$$

Im zweiten Schritt könnten wir beispielsweise das Hard- oder das Soft-SVM-Problem auf $F$ mit Trainingsmenge $\hat{S}$ lösen.

Je nachdem, wie $\psi$ gewählt wird, also welche Features dem Algorithmus zur Verfügung stehen, werden die Ergebnisse besser oder schlechter. Deren Auswahl hängt von der Anwendung ab. Hier steckt ein bisschen die Kunst des Maschinellen Lernens.

## 2 Repräsentationssatz

Ob der Algorithmus, der die Einbettung nutzt, eine sinnvolle Laufzeit hat, hängt maßgeblich von der Dimension $n$ des Feature Space ab. Diese kann jedoch sehr hoch sein, wie beispielsweise bei der oben genannten polynomiellen Einbettung. Wir werden nun einen Satz zeigen, mit dessen Hilfe sich die Laufzeit jedoch drastisch reduzieren lässt.

Dafür nehmen wir an, dass wir im zweiten Schritt einen Vektor $\mathbf{w} \in \mathbb{R}^n$ suchen, der eine Funktion $f \colon \mathbb{R}^n \to \mathbb{R}$ minimiert, die die Form

$$f(\mathbf{w}) = f_1(\|\mathbf{w}\|) + f_2(\langle \mathbf{w}, \psi(\mathbf{x}_1) \rangle, \ldots, \langle \mathbf{w}, \psi(\mathbf{x}_m) \rangle) \tag{1}$$

hat, wobei $f_1 \colon \mathbb{R} \to \mathbb{R}$ monoton steigend und $f_2 \colon \mathbb{R}^m \to \mathbb{R}$ eine beliebige Funktion ist. Wichtig ist, dass beide Funktionen nur in einer sehr eingeschränkten Art von $\mathbf{w}$ abhängen. Die erste hängt lediglich von der Norm von $\mathbf{w}$ ab, die zweite lediglich von den Skalarprodukten von $\mathbf{w}$ mit $\mathbf{x}_1, \ldots, \mathbf{x}_m$.

Alle Arten zur linearen Klassifikation, die wir bislang kennengelernt haben, lassen sich so darstellen.

- Bei Soft-SVM ist dies relativ offensichtlich. Hier könnten wir

$$f_1(a) = \lambda a^2, \qquad f_2(a_1, \ldots, a_m) = \frac{1}{m} \sum_{i=1}^{m} \max\{0, 1 - y_i a_i\}$$

  wählen.

- Um Hard-SVM zu erfassen, nutzen wir

$$f_1(a) = a^2, \qquad f_2(a_1, \ldots, a_m) = \begin{cases} 0 & \text{falls } y_i a_i \geq 1 \text{ für alle } i \\ \infty & \text{sonst} \end{cases} .$$

  Die Funktion $f_2$ bringt also in diesem Fall die Nebenbedingungen zum Ausdruck.

- Auch die Zielfunktion, die Anzahl falsch klassifizierter Punkte lässt sich in dieser Form schreiben. Hier ist $f_1(a) = 0$ für alle $a$ und $f_2(a_1, \dots, a_m) = |\{i \mid y_i a_i \leq 0\}|$.

**Satz 11.3.** *Für jede Auswahl von Datenpunkten $\mathbf{x}_1, \dots, \mathbf{x}_m \in X$, Einbettungsfunktion $\psi\colon X \to F$, und jede Funktion $f$ der Form wie in Gleichung (1) gibt es $\alpha_1, \dots, \alpha_m$, sodass der Vektor $\mathbf{w}' = \sum_{i=1}^m \alpha_i \psi(\mathbf{x}_i)$ die Funktion $f$ minimiert.*

Das heißt, dass es um $f$ zu minimieren ausreicht, nur die Linearkombinationen von $\psi(\mathbf{x}_1), \dots \psi(\mathbf{x}_m)$ zu betrachten.

*Beweis von Satz 11.3.* Sei $\mathbf{w}^* \in F$ eine optimale Lösung des Optimierungsproblems. Die Vektoren $\psi(\mathbf{x}_1), \dots, \psi(\mathbf{x}_m)$ erzeugen eine Unterraum $U$ von $F$ von Dimension höchstens $m$. Wir betrachten nun eine Orthonormalbasis $\mathbf{b}_1, \dots, \mathbf{b}_k$ dieses Unterraums $U$. (Diese könnte man beispielsweise mit dem Gram-Schmidtschen Orthogonalisierungsverfahren bestimmen.) Das heißt $\langle \mathbf{b}_j, \mathbf{b}_j \rangle = 1$ und $\langle \mathbf{b}_j, \mathbf{b}_{j'} \rangle = 0$ für $j \neq j'$. Außerdem lässt sich jedes $\psi(\mathbf{x}_i)$ als Linearkombination von $\mathbf{b}_1, \dots, \mathbf{b}_k$ darstellen. Weil es sich um eine Orthonomalbasis handelt, ist dies besonders einfach. Es gilt

$$\psi(\mathbf{x}_i) = \sum_{j=1}^{k} \langle \psi(\mathbf{x}_i), \mathbf{b}_j \rangle \mathbf{b}_j \ .$$

Nun betrachten wir die Projektion von $\mathbf{w}^*$ auf $U$. Diese berechnet sich in ähnlicher Weise als

$$\mathbf{w}' = \sum_{j=1}^{k} \langle \mathbf{w}^*, \mathbf{b}_j \rangle \mathbf{b}_j \ .$$

Es gilt $\mathbf{w}' \in U$, denn $U$ umfasst ja genau alle Linearkombinationen von $\mathbf{b}_1, \dots, \mathbf{b}_k$. Wir können $\mathbf{w}'$ aber auch als Linearkombination von $\psi(\mathbf{x}_1), \dots, \psi(\mathbf{x}_m)$ schreiben, denn auch diese Vektoren erzeugen $U$. Das heißt, es gibt $\alpha_1, \dots, \alpha_m \in \mathbb{R}$ mit

$$\mathbf{w}' = \sum_{i=1}^{m} \alpha_i \psi(\mathbf{x}_i) \ .$$

Wir behaupten nun, dass $f(\mathbf{w}') \leq f(\mathbf{w}^*)$. Betrachten wir zunächst das Skalarprodukt von $\mathbf{w}'$ mit einem beliebigen $\mathbf{b}_{j'}$. Es gilt

$$\langle \mathbf{w}', \mathbf{b}_{j'} \rangle = \left\langle \sum_{j=1}^{k} \langle \mathbf{w}^*, \mathbf{b}_j \rangle \mathbf{b}_j, \mathbf{b}_{j'} \right\rangle = \sum_{j=1}^{k} \langle \mathbf{w}^*, \mathbf{b}_j \rangle \cdot \langle \mathbf{b}_j, \mathbf{b}_{j'} \rangle = \langle \mathbf{w}^*, \mathbf{b}_{j'} \rangle \ .$$

Somit gilt also auch

$$\langle \mathbf{w}', \psi(\mathbf{x}_i) \rangle = \left\langle \mathbf{w}', \sum_{j=1}^{k} \langle \psi(\mathbf{x}_i), \mathbf{b}_j \rangle \mathbf{b}_j \right\rangle = \sum_{j=1}^{k} \langle \psi(\mathbf{x}_i), \mathbf{b}_j \rangle \cdot \langle \mathbf{w}', \mathbf{b}_j \rangle = \sum_{j=1}^{k} \langle \psi(\mathbf{x}_i), \mathbf{b}_j \rangle \cdot \langle \mathbf{w}^*, \mathbf{b}_j \rangle = \langle \mathbf{w}^*, \psi(\mathbf{x}_i) \rangle \ .$$

Das heißt, dass $f_2(\langle \mathbf{w}', \psi(\mathbf{x}_1) \rangle, \dots, \langle \mathbf{w}', \psi(\mathbf{x}_m) \rangle) = f_2(\langle \mathbf{w}^*, \psi(\mathbf{x}_1) \rangle, \dots, \langle \mathbf{w}^*, \psi(\mathbf{x}_m) \rangle)$.

Eine analoge Rechnung liefert uns $\langle \mathbf{w}', \mathbf{w}' \rangle = \langle \mathbf{w}^*, \mathbf{w}' \rangle$. Definieren wir uns also $\mathbf{c} = \mathbf{w}^* - \mathbf{w}'$, stellen wir fest, dass $\langle \mathbf{w}', \mathbf{c} \rangle = \langle \mathbf{w}', \mathbf{w}^* \rangle - \langle \mathbf{w}', \mathbf{w}' \rangle = 0$. Somit gilt auch, dass

$$\|\mathbf{w}^*\|^2 = \langle \mathbf{w}' + \mathbf{c}, \mathbf{w}' + \mathbf{c} \rangle = \langle \mathbf{w}', \mathbf{w}' \rangle + \langle \mathbf{c}, \mathbf{c} \rangle = \|\mathbf{w}'\|^2 + \|\mathbf{c}\|^2 \ .$$

Dies bedeutet also auch, dass $\|\mathbf{w}'\| \leq \|\mathbf{w}^*\|$ und damit $f_1(\|\mathbf{w}'\|) \leq f_1(\|\mathbf{w}^*\|)$ aufgrund der Monotonie.

Insgesamt gilt also $f(\mathbf{w}') \leq f(\mathbf{w}^*)$. □

Aufgrund von Satz 11.3 können wir uns also darauf beschränken $\boldsymbol{\alpha} \in \mathbb{R}^m$ zu finden anstatt $\mathbf{w} \in \mathbb{R}^n$. Dies ist von enormem Nutzen, wenn $n \gg m$.

## 3 Effiziente Berechnung

Wie finden wir also einen Vektor $\boldsymbol{\alpha} \in \mathbb{R}^m$, so dass $f\left(\sum_{i=1}^m \alpha_i \psi(\mathbf{x}_i)\right)$ minimiert wird? Weiterhin hat $f$ die Form aus Gleichung (1). Das heißt, $f$ hängt nur von der Norm und den Skalarprodukten ab. Diese können wir auch direkt durch $\boldsymbol{\alpha}$ ausdrücken. Gilt nämlich $\mathbf{w} = \sum_{i=1}^m \alpha_i \psi(\mathbf{x}_i)$, dann auch

$$\langle \mathbf{w}, \psi(\mathbf{x}_j) \rangle = \left\langle \sum_{i=1}^m \alpha_i \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \right\rangle = \sum_{i=1}^m \alpha_i \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle$$

und

$$\|\mathbf{w}\| = \sqrt{\langle \mathbf{w}, \mathbf{w} \rangle} = \sqrt{\left\langle \sum_{i=1}^m \alpha_i \psi(\mathbf{x}_i), \sum_{j=1}^m \alpha_j \psi(\mathbf{x}_j) \right\rangle} = \sqrt{\sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle} \ .$$

Definieren wir uns also eine neue Funktion $K \colon X \times X \to \mathbb{R}$ über $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle$, dann lassen sich diese Ausdrücke schreiben als

$$\langle \mathbf{w}, \psi(\mathbf{x}_j) \rangle = \sum_{i=1}^m \alpha_i K(\mathbf{x}_i, \mathbf{x}_j)$$

und

$$\|\mathbf{w}\| = \sqrt{\sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)} \ .$$

Somit gilt also

$$f\left(\sum_{i=1}^m \alpha_i \psi(\mathbf{x}_i)\right) = f_1\left(\sqrt{\sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)}\right) + f_2\left(\sum_{i=1}^m \alpha_i K(\mathbf{x}_i, \mathbf{x}_1), \ldots, \sum_{i=1}^m \alpha_i K(\mathbf{x}_i, \mathbf{x}_m)\right) \ .$$

Insgesamt müssten wir also, um $f$ zu berechnen und auch zu minimieren, lediglich $K(\mathbf{x}_i, \mathbf{x}_j)$ für alle Paare $i$ und $j$ ausrechnen. Die einzelnen Werte von $\psi(\mathbf{x}_i)$ sind nicht gar nicht erforderlich. Das heißt, wir rechnen nicht einmal $m^2$ anstatt $m \cdot n$ Werte aus. Für große $n$ kann dies ein enormer Vorteil sein.

**Beispiel 11.4.** *Betrachten wir wieder die polynomielle Einbettung des $X = \mathbb{R}^d$. Relativ einfaches Nachrechnen ergibt, dass $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \langle \mathbf{x}_i, \mathbf{x}_j \rangle)^k$. Das heißt, diese Werte lassen sich relativ leicht ausrechnen. Eine Bestimmung der $m$ Vektoren $\psi(\mathbf{x}_1), \ldots, \psi(\mathbf{x}_m)$ mit je $(k+1)^d$ Einträgen ist nicht erforderlich.*

## 4 Kernels

Wie wir gesehen haben, ist es also nur nötig, die Funktion $K \colon X \times X \to \mathbb{R}$ auszurechnen. Eine solche Funktion nennt sich *Kernel*. Sie ersetzt gewissermaßen das Skalarprodukt auf $X$.

In der Tat ist es nicht einmal erforderlich, dass der Feature Space $F$ eine endliche Dimension hat, denn die Funktion $\psi \colon X \to F$ muss nicht explizit ausgewertet werden. Der Raum $F$ muss lediglich ein reeller Vektorraum sein, auf dem ein Skalarprodukt definiert ist, ein sogenannter *Hilbertraum*.

## Referenzen


- Understanding Machine Learning, Kapitel 16