Data Science

Vorlesung 7 - Lineare Modelle

Prof. Dr. Christian Bockermann

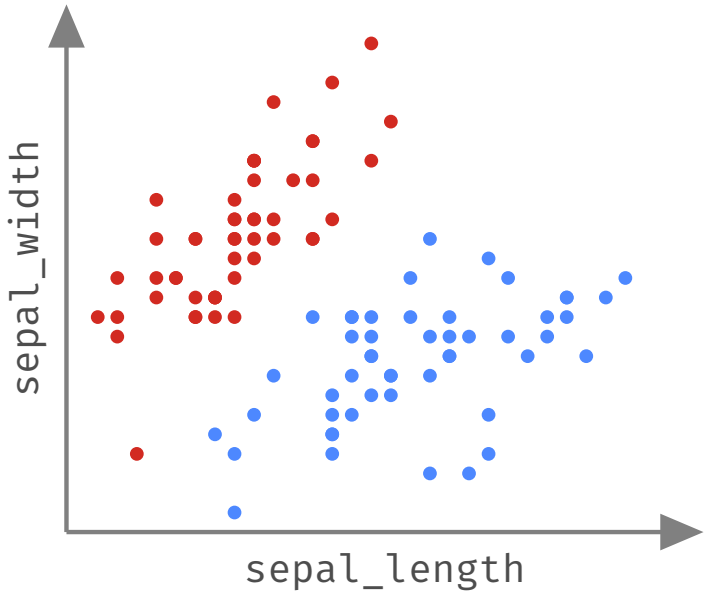
Hochschule Bochum

Sommersemester 2021

Inhalt

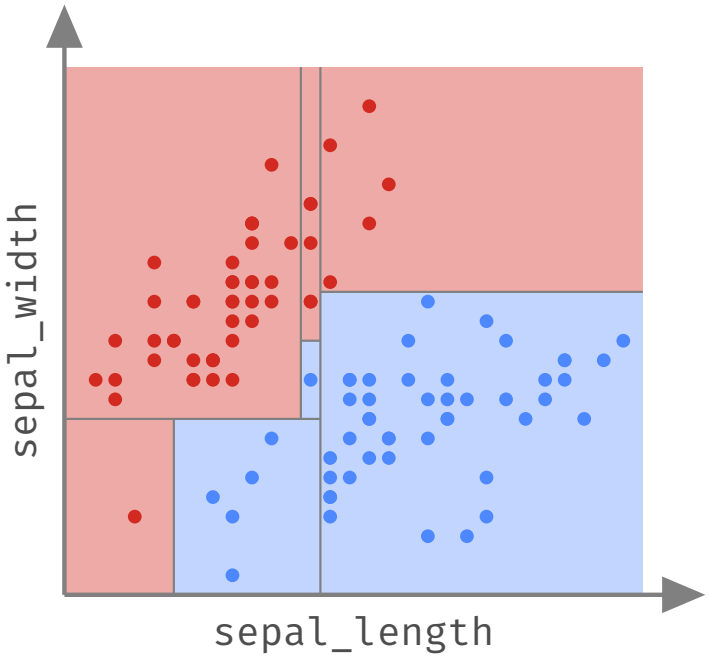
Wir betrachten im Folgenden lineare Modelle zur Klassifikation

Iris Daten mit Klassen *setosa* und *versicolor* als Beispiel:



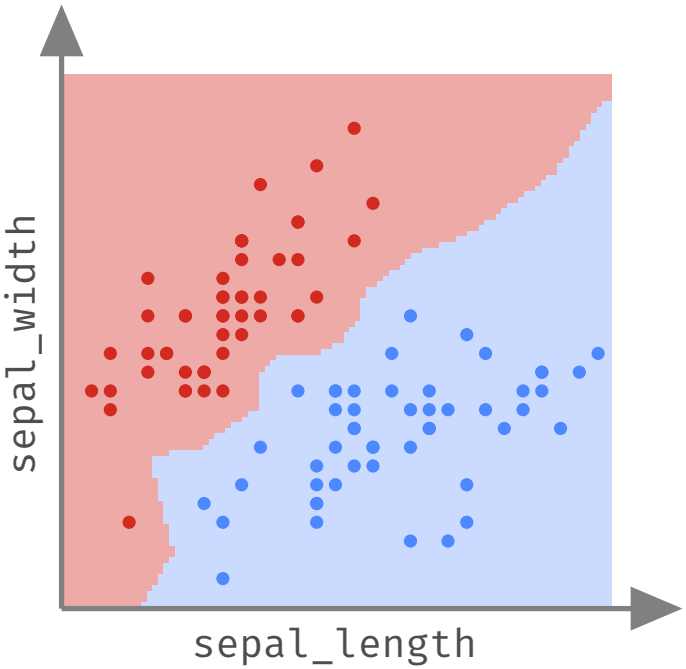
Aufgabe: Wir suchen eine Funktion f, die für unbekannten Datensatz **x** die zugehörige Klasse vorhersagt!

Bisher betrachtet: Entscheidungsbäume und nächste Nachbarn



Entscheidungsbaum

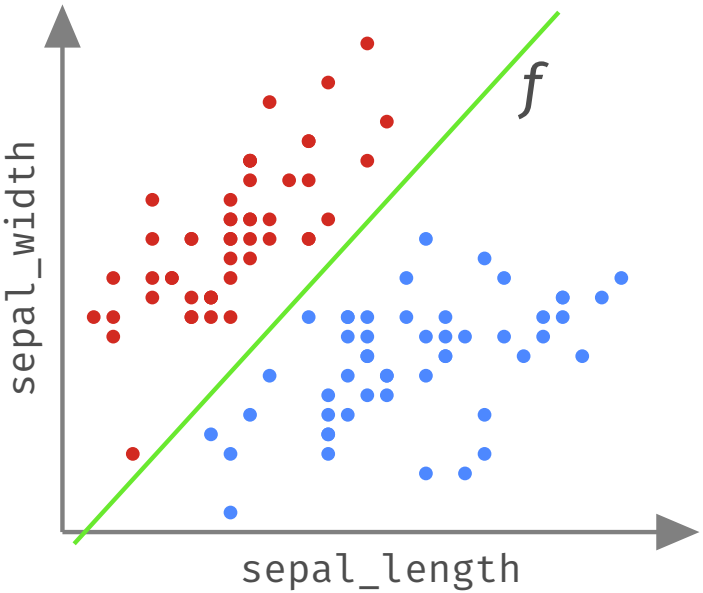
Trennung nach einzelnen Attributen, achsenparallel



k-nächste Nachbarn

Trennung in Regionen, nach Distanz (Berechnung über alle Attribute)

Idee: Daten mit einer Geraden trennen



Geradengleichung (Schule) im 2-dimensionalen Raum ($\mathbb{R}^2$):

$$f(x) = b \cdot x + c$$

Frage 1:

- Was machen wir, wenn unsere Daten nicht im $\mathbb{R}^2$ sind, sondern im $\mathbb{R}^d$ ($d > 2$)?

Frage 2:

- Wie finden wir die richtige "Gerade" / das richtige f?

Frage 1:

- Was machen wir, wenn unsere Daten nicht im $\mathbb{R}^2$ sind, sondern im $\mathbb{R}^d$ ($d > 2$)?

Frage 2:

- Wie finden wir die richtige "Gerade" / das richtige f?

Vorgehen:

1. Wir brauchen ein Konzept für höherdimensionale "Geraden"
2. Wir müssen f in Parameter zerlegen und dann die richtigen Parameter suchen (= Training)

Bisher betrachten wir rein numerische Daten

- Daten hatten d Attribute
- Jedes Attribut numerische Werte
- Jedes Beispiel ist Element des d-dimensionalen Raums $\mathbb{R}^d$

Beispiel: Iris-Daten enthalten 4 numerische Attribute

sepal_length	sepal_width	petal_length	petal_width
4.700	3.200	1.300	0.200
6	2.200	4	1
4.600	3.100	1.500	0.200
7.600	3	6.600	2.100
6.300	2.900	5.600	1.800
5.400	3.900	1.700	0.400

$$\mathbf{x}_3 = \begin{pmatrix} 4.6 \\ 3.1 \\ 1.5 \\ 0.2 \end{pmatrix}$$

Vektor-Darstellung der
Zeile 3 aus dem Datensatz

Daten im Vektorraum

Im Folgenden betrachten wir unsere Daten als Teilmenge des Vektorraums $\mathbb{R}^d$.

Hinweis:

Die folgenden Folien enthalten einige Grundlagen zum Begriff des Vektorraums.
Das sieht zunächst nach viel Mathematik aus, die meisten Dinge davon benutzen wir unbewusst aber täglich bzw. seit Beginn des Kurses.

Was ist ein Vektorraum?

Ein d-dimensionaler Vektorraum V ist eine Menge von Vektoren über einem Körper (z.B. $\mathbb{R}$).

- Vektoren aus V sind d-Tupel mit Werten aus $\mathbb{R}$
- Vektoren können addiert werden, z.B. im $\mathbb{R}^2$

$$\begin{pmatrix} -3 \\ 2 \end{pmatrix} + \begin{pmatrix} 7 \\ 11 \end{pmatrix} = \begin{pmatrix} 4 \\ 13 \end{pmatrix}$$

- Multiplikation mit Skalaren aus dem Körper, z.B.

$$2 \cdot \begin{pmatrix} -3 \\ 2 \end{pmatrix} = \begin{pmatrix} -6 \\ 4 \end{pmatrix}$$

Vektorraum - Skalarprodukt

Für einen Vektorraum V ist das Skalarprodukt zweier Vektoren $\mathbf{v}, \mathbf{w}$ definiert als

$$\langle \mathbf{v}, \mathbf{w} \rangle = \sum_{i=1}^{d} v_i \cdot w_i$$

Durch das Skalarprodukt ist eine Norm definiert als

$$||v|| = \sqrt{\langle v, v \rangle}$$

Die Norm definiert quasi die Länge eines Vektors.

Vektorraum - Metrik

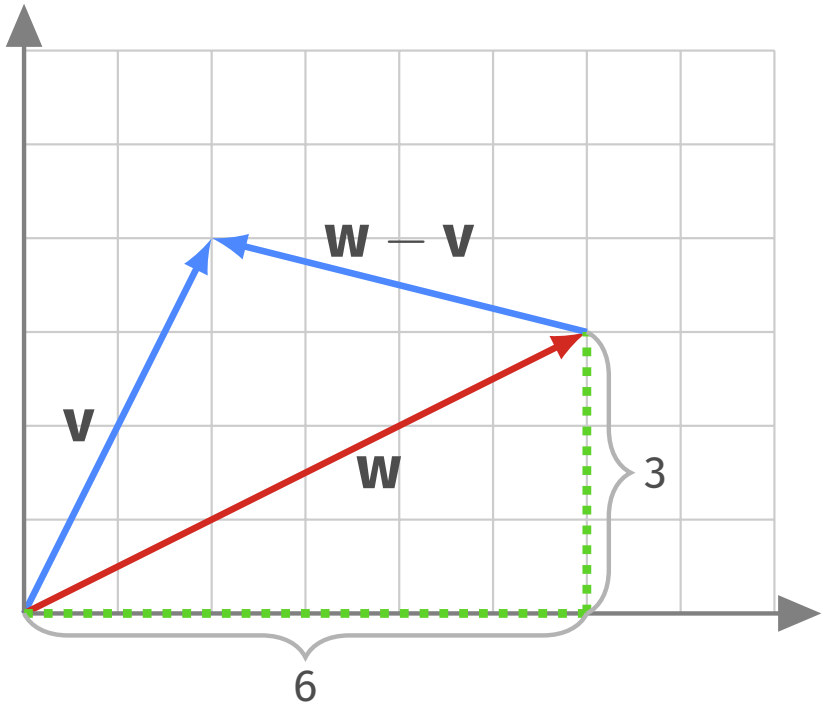
Über die Norm ist eine Metrik (Distanz) zwischen Vektoren definiert als

$$d(v, w) = ||v - w||$$

Mit dem normalen Skalarprodukt ergibt sich für den Vektorraum $\mathbb{R}^d$ daraus die euklidische Distanz.

Das haben wir schon bei k-NN implizit benutzt!

Graphische Anschauung im $\mathbb{R}^2$



$$\mathbf{v} = \begin{pmatrix} 2 \\ 4 \end{pmatrix}, \mathbf{w} = \begin{pmatrix} 6 \\ 3 \end{pmatrix}$$

$$\mathbf{w} - \mathbf{v} = \begin{pmatrix} 4 \\ -1 \end{pmatrix}$$

Die Norm (=Länge) von $\mathbf{w}$ ergibt sich über Satz von Pythagoras:

$$||\mathbf{w}|| = \sqrt{\langle \mathbf{w}, \mathbf{w} \rangle} = \sqrt{6 \cdot 6 + 3 \cdot 3} \approx 6.708$$

Notationen für die folgenden Folien:

- Ein fetter Kleinbuchstabe $\mathbf{x}$ bezeichnet einen Vektor
- x_i bezeichnet die i-te Komponente von $\mathbf{x}$
- Fetter Großbuchstabe $\mathbf{X}$ bezeichnet Menge von Vektoren
- $\mathbf{x}^T$ bezeichnet den transponierten Vektor von $\mathbf{x}$

Beispiele:

$$\mathbf{w} = \begin{pmatrix} 4 \\ 8 \\ 3 \end{pmatrix}, \quad \mathbf{w}^T = \begin{pmatrix} 4 & 8 & 3 \end{pmatrix}, \quad w_1 = 4, \quad w_2 = 8$$

Vektoren in Pandas

Ein Pandas `Series` Objekt stellt einen Vektor dar:

```
# Erzeuge Vektoren v und w

v = pd.Series([5, 3, 7])
w = pd.Series([4, 8, 3])

# Multiplikation mit Zahl:
w2 = 2 * w

# Skalarproduct <v, w> zweier Vektoren:
sum( v * w )
```

Fragen 1: Wie trennen wir Daten im $\mathbb{R}^d$?

d=2: Im $\mathbb{R}^2$ funktioniert die Gerade:

$$y = f(x) = w \cdot x + b$$

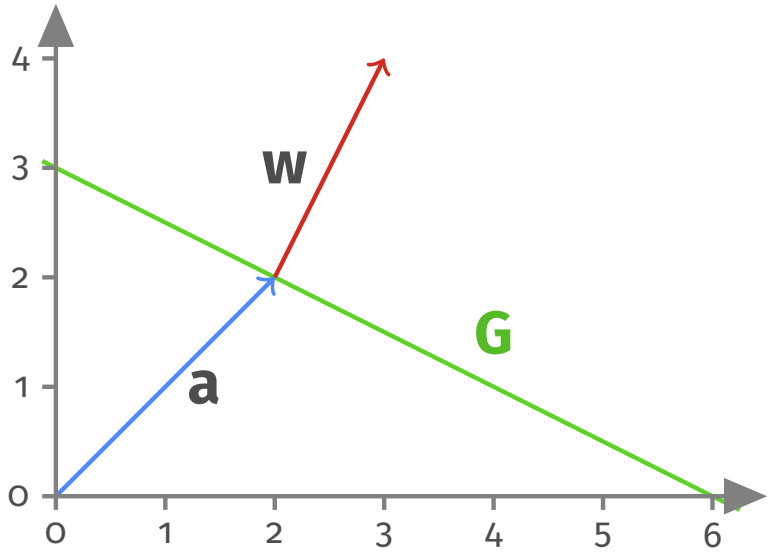
Eine Gerade ist ein Untervektorraum der Dimension 1.

d=3: Im $\mathbb{R}^3$ können wir Ebenen zu Trennung benutzen:

$$z = f(x, y) = w_1 \cdot x + w_2 \cdot y + b$$

Eine Ebene ist ein Untervektorraum der Dimension 2.

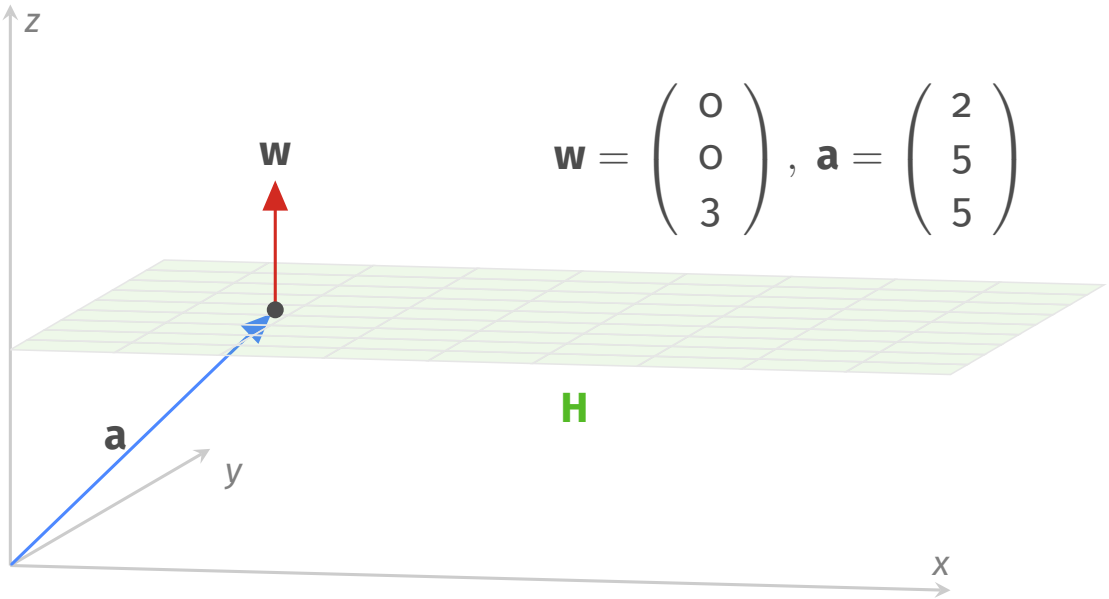
Beispiel: Gerade im $\mathbb{R}^2$ ist ein 1-dimensionaler Unterraum



Gerade ist definiert durch Stützvektor $\mathbf{a} = \binom{2}{2}$ und Normalenvektor $\mathbf{w} = \binom{1}{2}$

$$\mathbf{G} = \left\{(x, y) \in \mathbb{R}^2 \mid 1 \cdot x + 2 \cdot y = 6\right\}$$

Beispiel: Ebene im $\mathbb{R}^3$ ist 2-dimensionaler Unterraum



Ebene ist definiert durch Stützvektor **a** und Normalenvektor **w**

$$\mathbf{H} = \left\{(x, y, z) \in \mathbb{R}^3 \mid 0 \cdot x + 0 \cdot y + 3 \cdot z = 15\right\}$$

Trennebenen in höheren Dimensionen

Allgemein ist eine Hyperebene in einem Vektorraum $\mathbb{R}^d$ ein Untervektorraum der Dimension $d-1$.

Eine Hyperebene lässt sich beschreiben als

$$\begin{aligned} f(\mathbf{x}) &= w_d \cdot x_d + w_{d-1} \cdot x_{d-1} + \ldots + w_1 \cdot x_1 + b \\ &= \sum_{i=1}^{d} w_i x_i + b \quad \text{mit } b \in \mathbb{R}, \mathbf{x}, \mathbf{w} \in \mathbb{R}^d. \end{aligned} \tag{1}$$

Die Funktion/Hyperebene f wird also durch die Wahl von $\mathbf{w}$ und b bestimmt.

Die Summe $\sum_{i=1}^{d} w_i x_i$ in Formel (1) beschreibt das Skalarprodukt zweier Vektoren $\mathbf{x}$ und $\mathbf{w}$ und läßt sich schreiben als

$$\sum_{i=1}^{d} w_i x_i = \langle \mathbf{w}, \mathbf{x} \rangle = \mathbf{w}^T \mathbf{x} = \mathbf{x}^T \mathbf{w}.$$

Dabei ist $\langle \mathbf{x}, \mathbf{w} \rangle$ die Schreibweise als Skalarprodukt und $\mathbf{w}^T \mathbf{x}$, bzw. $\mathbf{x}^T \mathbf{w}$ das Gleiche als Matrizenmultiplikation.

Wir können unsere Funktion f somit schreiben als

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$

Parametrisierung unseres Modells

Wir haben (bei der Beschränkung auf lineare Modelle) nun eine Darstellung für das, was wir *lernen* wollen:

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$$

w wird auch Gewichtsvektor (daher "w" für *weights*), Koeffizienten oder Parameter genannt.

Der Parameter b ist die konstante Achsenverschiebung und wird auch *intercept* genannt.

Vorbereitung auf's Lernen

Durch die Darstellung mit $\mathbf{w}$ und b können wir mit d Parametern jede lineare Hyperebene wählen.

Das Training unseres Modells besteht jetzt in der Auswahl der richtigen Parameter

$$\mathbf{w} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{pmatrix} \quad \text{und} \quad b$$

Ein einfaches lineares Modell

Wie kommen wir nun zu einer trennenden Ebene?

Beispiel: Einfacher Algorithmus über Klassenmittelpunkte

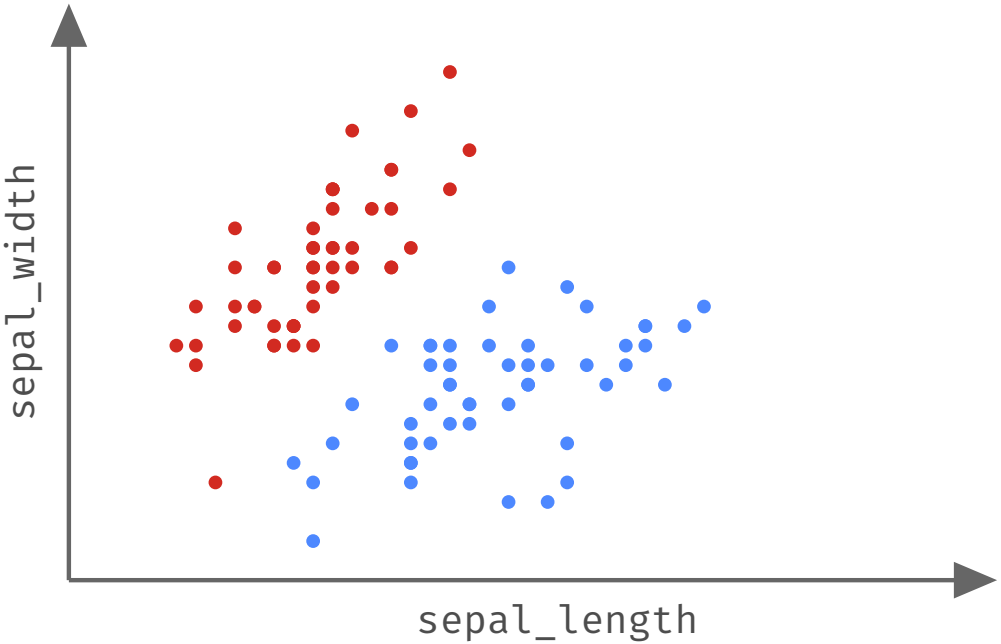
Es sei hier $\mathbf{X}_{[K]}$ die Menge der Daten der Klasse K und $|\mathbf{X}_{[K]}|$ die Anzahl der Punkte in der Menge

Die Mittelpunkte der Klassen setosa und versicolor

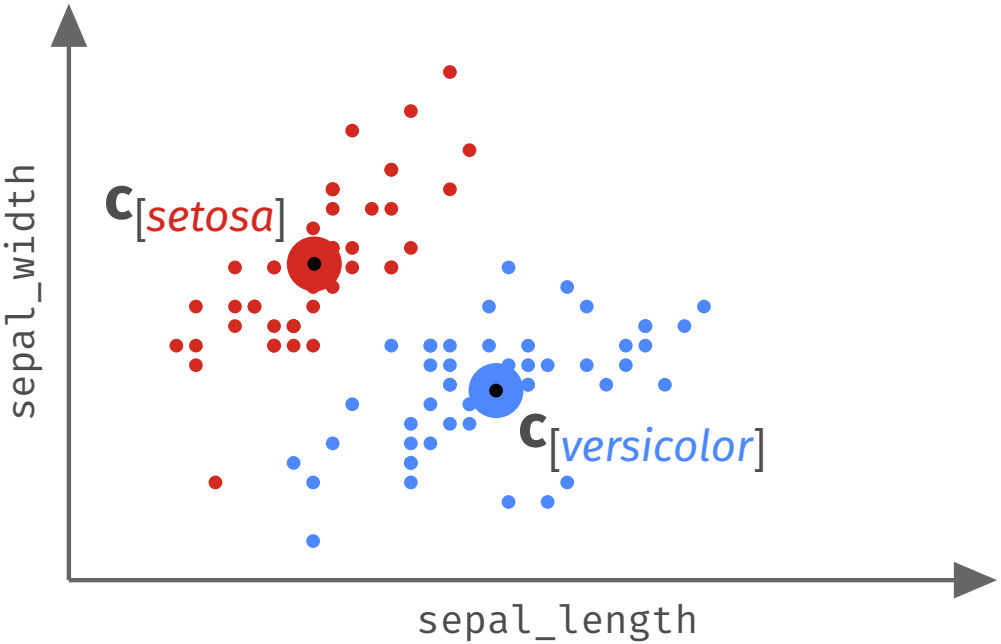
$$\mathbf{c}_{setosa} = \frac{1}{|\mathbf{X}_{[setosa]}|} \sum_{\mathbf{x} \in \mathbf{X}_{[setosa]}} \mathbf{x} \tag{2}$$

$$\mathbf{c}_{versicolor} = \frac{1}{|\mathbf{X}_{[versicolor]}|} \sum_{\mathbf{x} \in \mathbf{X}_{[versicolor]}} \mathbf{x} \tag{3}$$

Wie kommen wir nun zu einer trennenden Ebene?

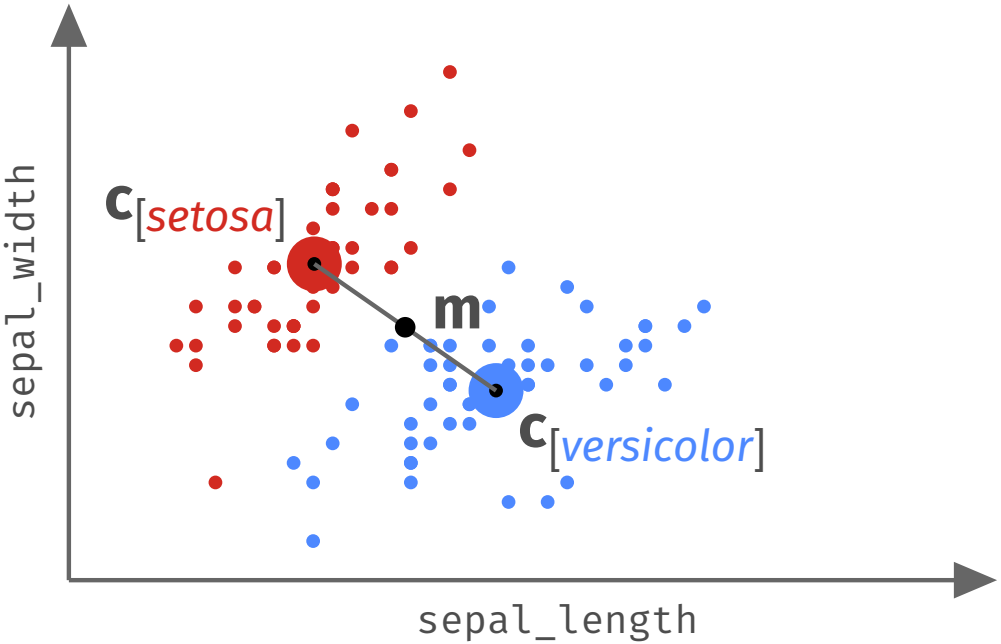


Wie kommen wir nun zu einer trennenden Ebene?



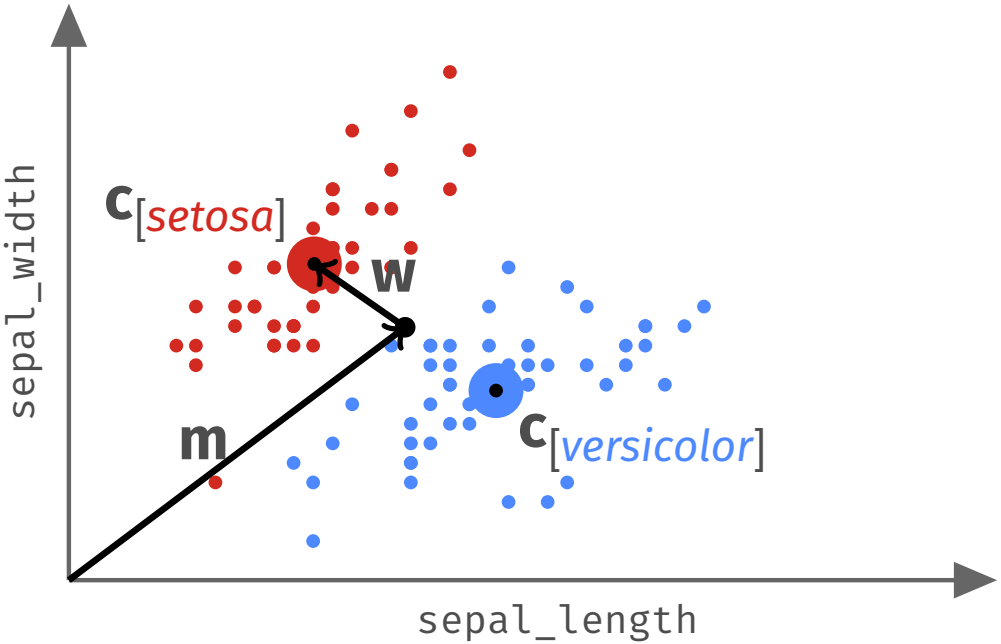
(1.) Berechne die Klassen-Mittelpunkte $\mathbf{c}_{[setosa]}$ und $\mathbf{c}_{[versicolor]}$.

Wie kommen wir nun zu einer trennenden Ebene?



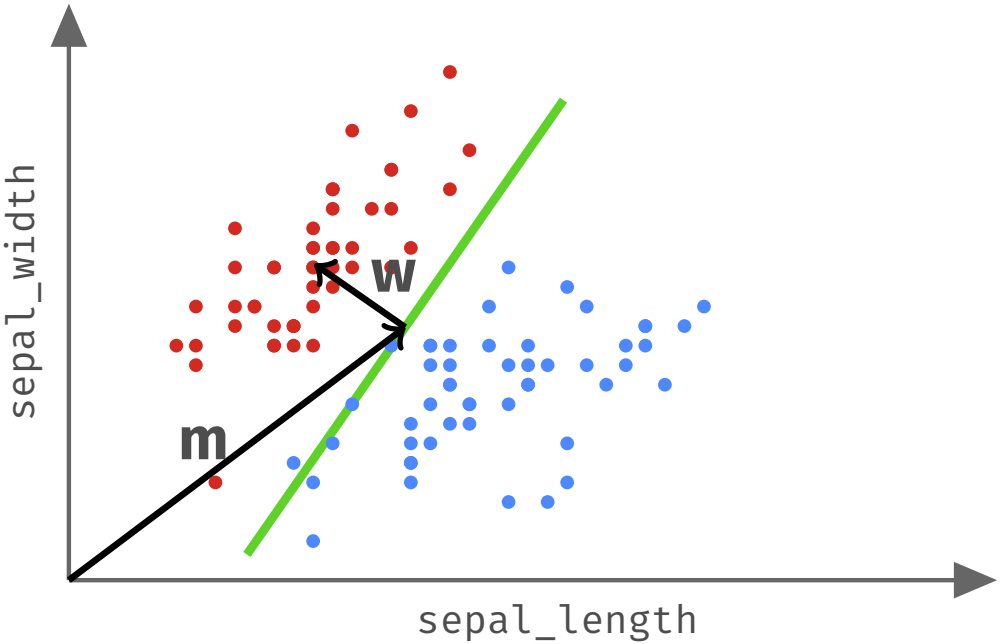
(2.) Bestimme Mittelpunkt $\mathbf{m} = \frac{1}{2}(\mathbf{c}_{[versicolor]} + \mathbf{c}_{[setosa]})$

Wie kommen wir nun zu einer trennenden Ebene?



(3.) Stützvektor **m**, Normalenvektor $\mathbf{w} = \mathbf{c}_{[setosa]} - \mathbf{m}$

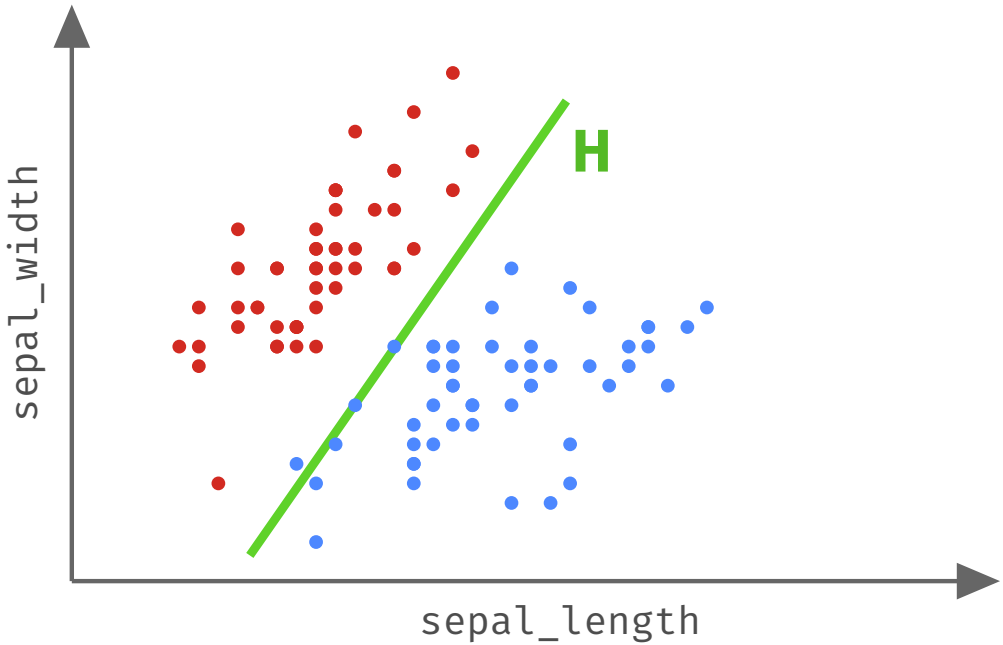
Wie kommen wir nun zu einer trennenden Ebene?



(3.) Ebene mit Stützvektor $\mathbf{m}$, Normalvektor $\mathbf{w}$:

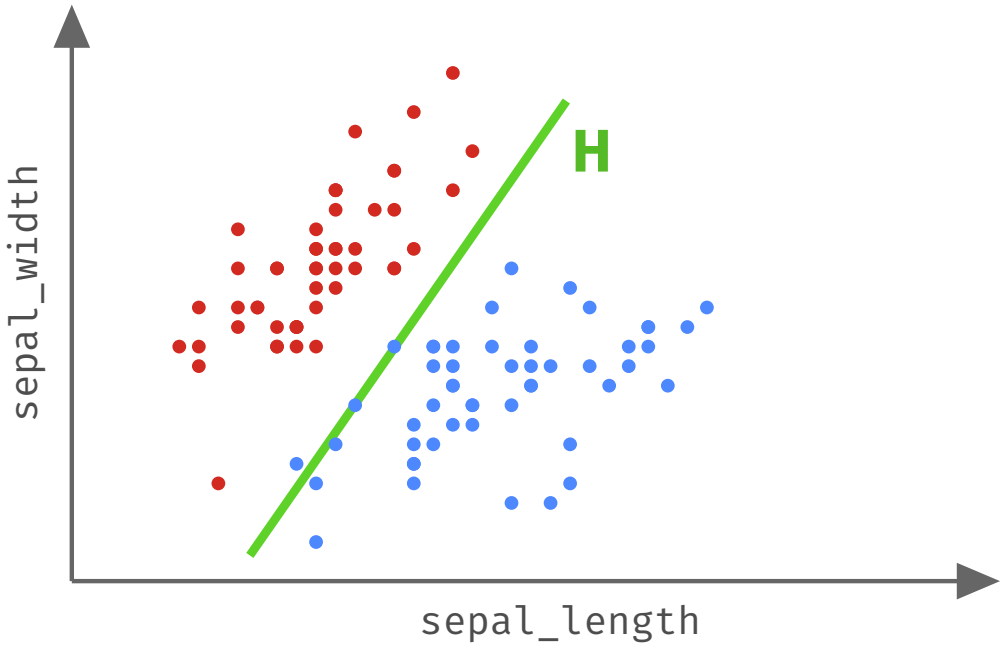
$$\mathbf{w}^T(\mathbf{x} - \mathbf{m}) = 0 \Leftrightarrow \mathbf{w}^T\mathbf{x} - \underbrace{\mathbf{w}^T\mathbf{m}}_{\approx -1.54} = 0$$

Ergebnis des einfachen Verfahrens:



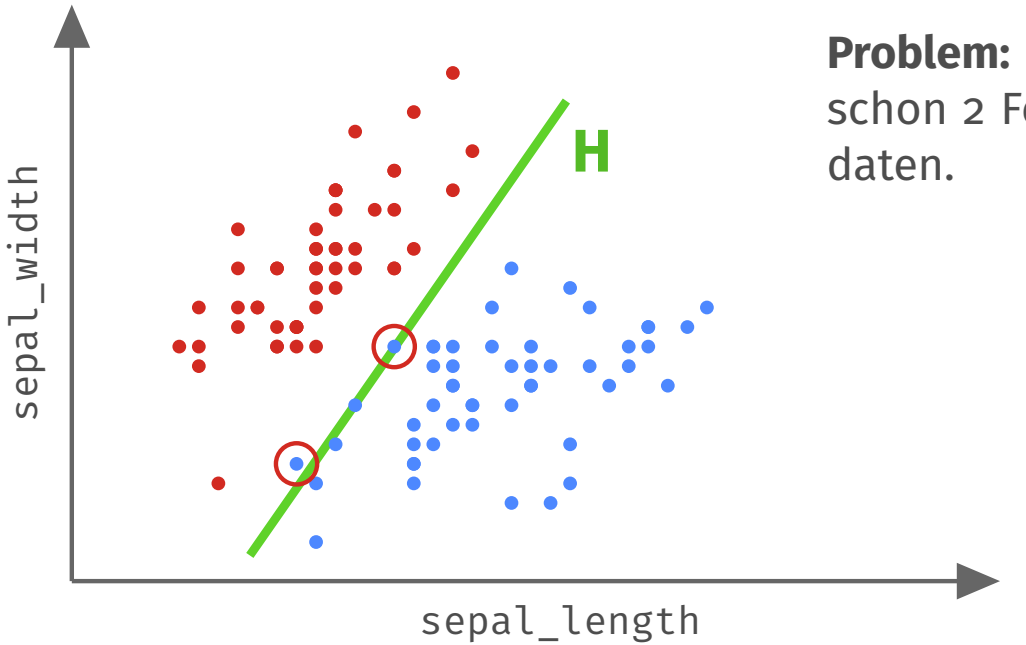
$$\mathbf{H} = \left\{ \mathbf{x} \in \mathbb{R}^2 \mid \mathbf{w}^T \mathbf{x} = -1.54 \right\}, \ \mathbf{w} = \begin{pmatrix} -0.465 \\ 0.324 \end{pmatrix}$$

Ergebnis des einfachen Verfahrens:



$$\mathbf{H} = \left\{ \mathbf{x} \in \mathbb{R}^2 \mid \mathbf{w}^T \mathbf{x} = -1.54 \right\}, \ \mathbf{w} = \begin{pmatrix} -0.465 \\ 0.324 \end{pmatrix}$$

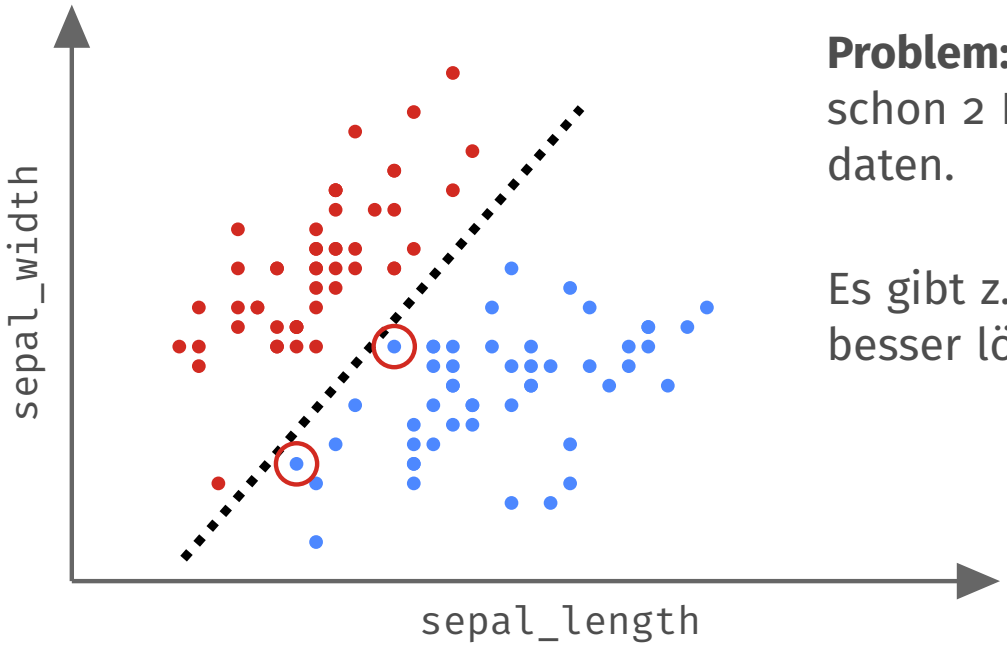
Ergebnis des einfachen Verfahrens:



Problem: Ebene **H** erzeugt schon 2 Fehler auf Trainingsdaten.

$$\mathbf{H} = \left\{ \mathbf{x} \in \mathbb{R}^2 \mid \mathbf{w}^T\mathbf{x} = -1.54 \right\}, \ \mathbf{w} = \begin{pmatrix} -0.465 \\ 0.324 \end{pmatrix}$$

Ergebnis des einfachen Verfahrens:



Problem: Ebene **H** erzeugt schon 2 Fehler auf Trainingsdaten.

Es gibt z.B. Ebene **H***, die das besser löst.

$$\mathbf{H} = \left\{ \mathbf{x} \in \mathbb{R}^2 \mid \mathbf{w}^T\mathbf{x} = -1.54 \right\}, \ \mathbf{w} = \begin{pmatrix} -0.465 \\ 0.324 \end{pmatrix}$$

Die Maximum-Margin Idee (SVM)

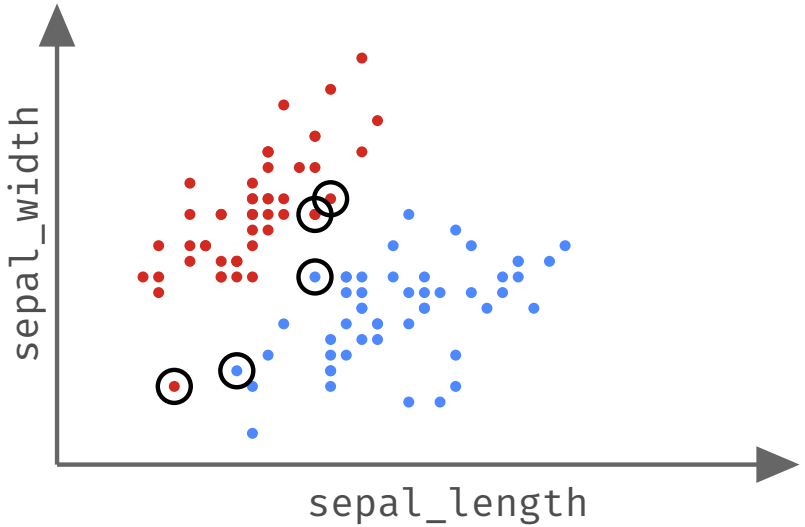
Das einfache Verfahren klappt nicht gut

- Berechnet $\mathbf{w}, b$ ohne Berücksichtigung des Trainingsfehlers
- Bezieht alle Punkte gleich-wichtig in Berechnung von $\mathbf{w}, b$ ein
- Ist nur Heuristik um lineare Modelle zu erklären :-)

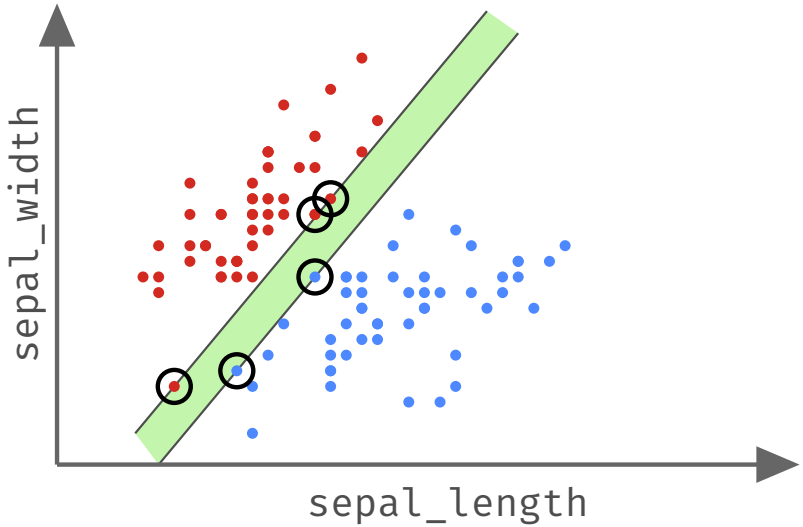
Frage: Wie kommen wir zu guten $\mathbf{w}$ und b?

- Sind alle Datenpunkte gleich wichtig?
-

Betrachten wir die Grenzpunkte der beiden Klassen:

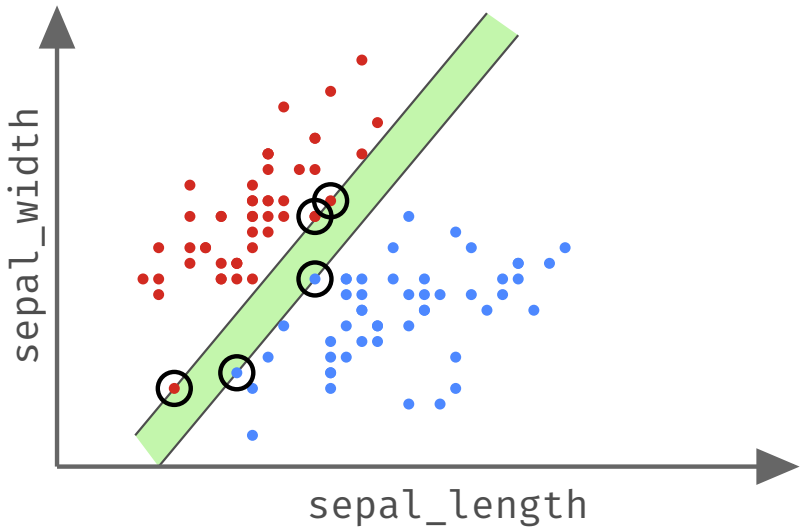


Betrachten wir die Grenzpunkte der beiden Klassen:



- Geraden durch Grenzpunkte erzeugt eine Art *Korridor*
- Grüner Bereich enthält die Ebenen, die fehlerfrei klassifizieren
- Es gibt **unendlich viele** fehlerfreie Ebenen in diesem Bereich

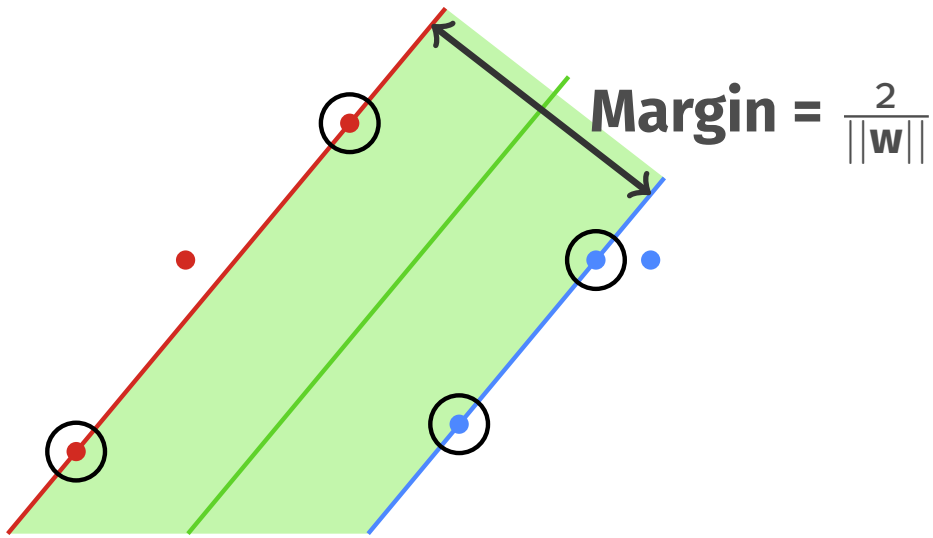
Betrachten wir die Grenzpunkte der beiden Klassen:



- Geraden durch Grenzpunkte erzeugt eine Art *Korridor*
- Grüner Bereich enthält die Ebenen, die fehlerfrei klassifizieren
- Es gibt **unendlich viele** fehlerfreie Ebenen in diesem Bereich

Wir wollen eine Ebene auswählen - welche ist die Beste?

Idee: Wähle Ebene mit größtem Abstand zu beiden Klassen

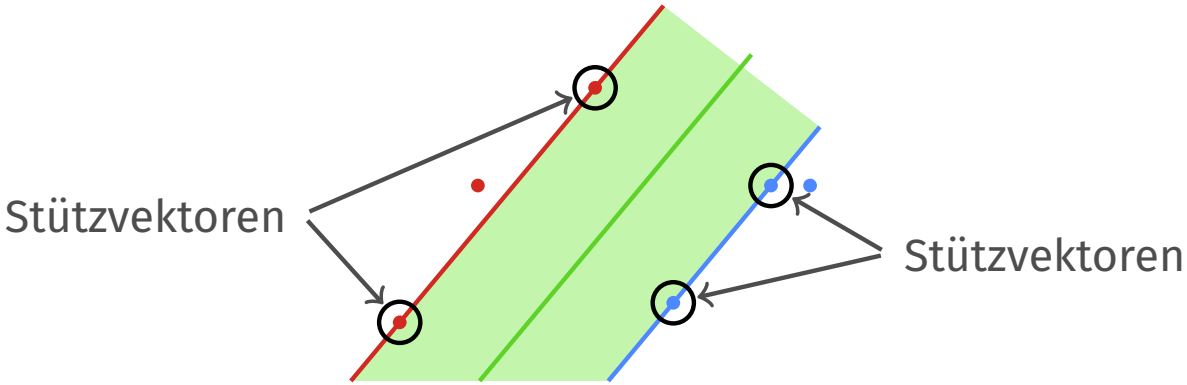


Führt zu Optimierungsproblem (Maximierung des Margin)

Statt $\frac{1}{||\mathbf{w}||}$ zu maximieren, können wir auch $\frac{1}{2}||\mathbf{w}||^2$ minimieren.
(Trick, um ein konvexes Optimierungsproblem zu erhalten)

Klassifikation mit der Support Vector Machine (SVM)

- Support Vector Machine = Stützvektor Methode
- Stützvektoren sind die Punkte, die den Margin definieren



- SVM findet optimale Hyperebene

SciKit Learn enthält SVM Classifier

```
# SVC = Linear Support Vector Classifier
from sklearn.svm import SVC

# Daten Laden, X, y erzeugen

m = SVC(kernel='linear')  # lineares Modell
m.fit(X, y)
```

◀ Probieren Sie es im Notebook aus!

Notebook: Vorlesung/V7-Lineare-SVM.ipynb