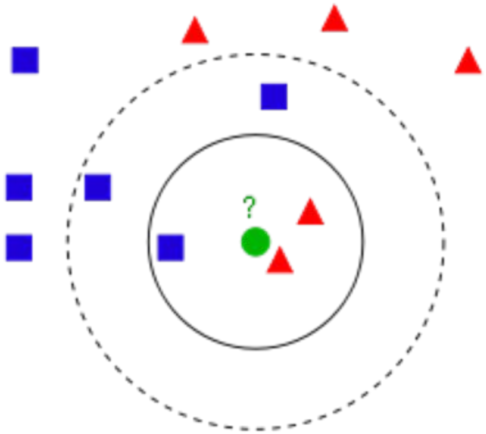
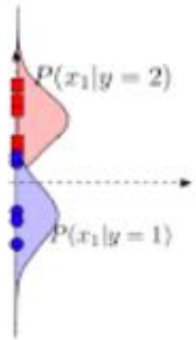
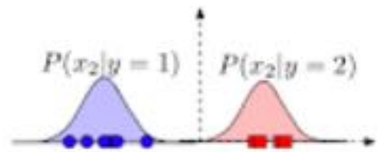
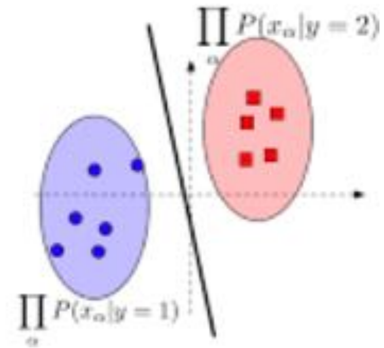
Machine Learning Applications

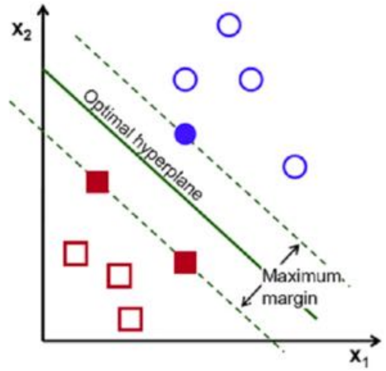
Weitere Verfahren des Maschinellen Lernen:
kNN, Bayes-Klassifikator, Stützvektormethode und Clustering

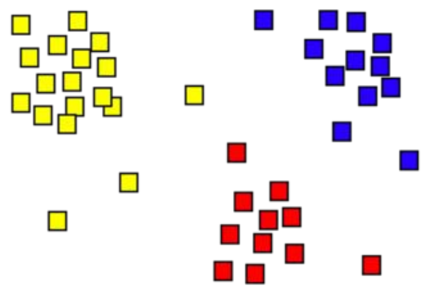








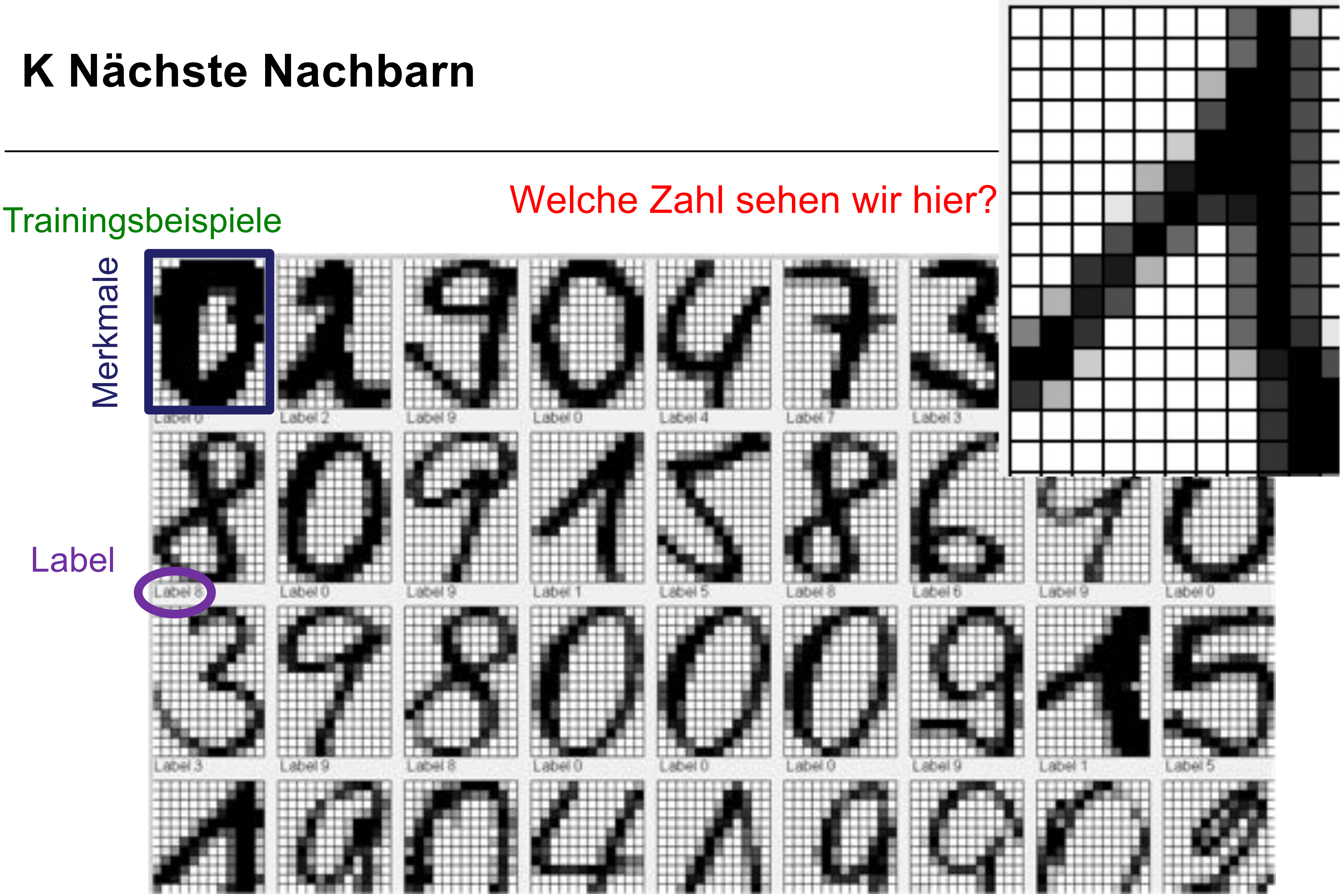




Basierende auf Folien von Katharina Morik, Uwe Ligges, Claus Weihs, Lutz Plümer und viele anderen.
Danke fürs Offenlegen ihrer Folien

K Nächste Nachbarn

Welche Zahl sehen wir hier?



TECHNISCHE UNIVERSITÄT DARMSTADT

Grundidee

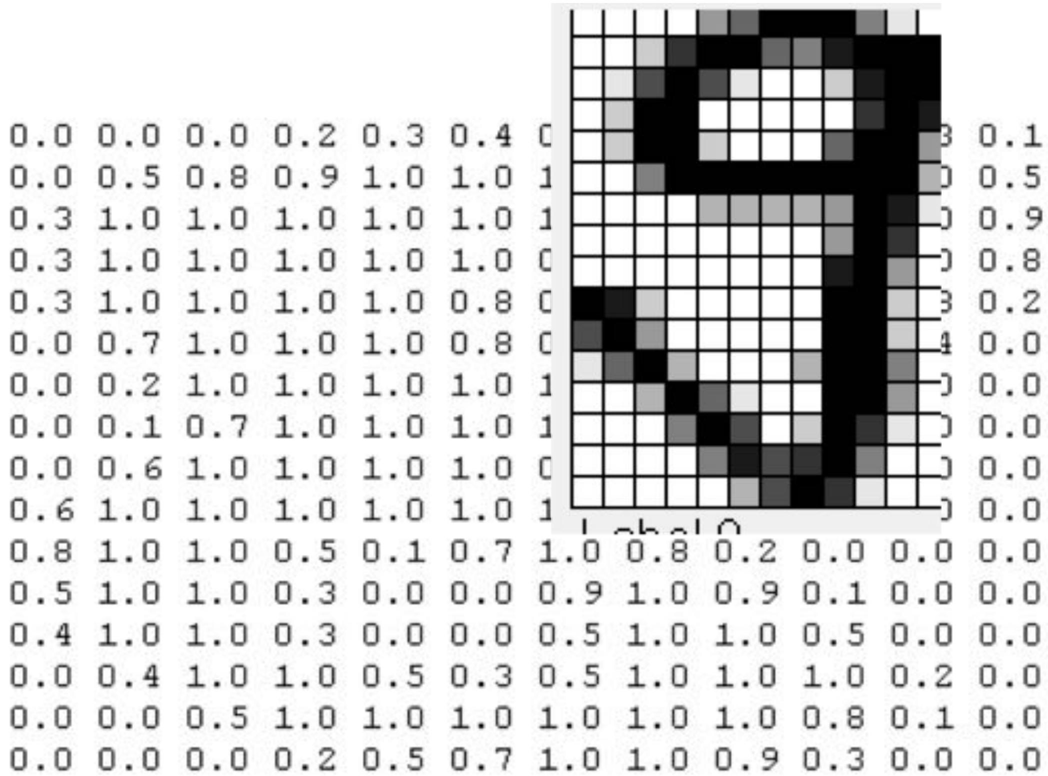
Zwei Bilder repräsentieren die gleiche Ziffer, wenn die Bilder ähnlich sind

Ähnlich = ähnliche Grauwertverteilung

Wir stellen Bilder als eine Matrix von Grauwerten dar

Ziffer = 12 x 16 Matrix von Grauwerten in [0,1]

Vektor von Grauwerten der Länge 192



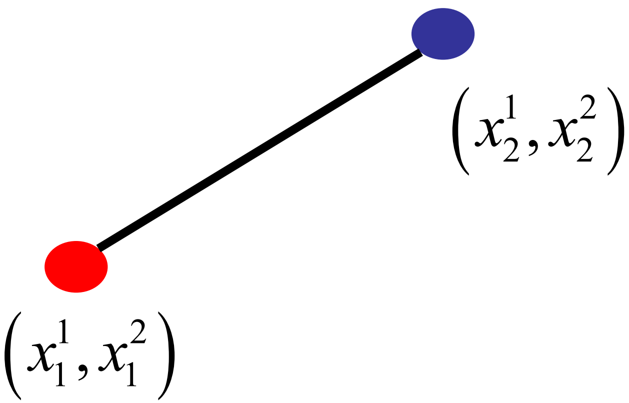
Ähnlichkeit, Abstand

Es stellt sich also die Frage, wie ähnlich zwei Bilder/Datenpunkte sind

Die Ähnlichkeit wird durch den Abstand beschrieben

- Je ähnlicher, desto kleiner der Abstand
- Sie sind identisch, wenn ihr Abstand 0 ist

Das wichtigste Abstandsmaß für reelle Merkmale/Vektoren ist der "euklidische Abstand"



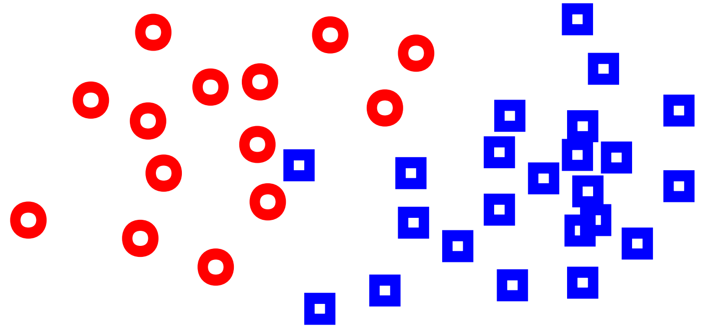
$$dist(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

$$\cos(\vec{x}, \vec{y}) = \frac{\vec{x}.\vec{y}}{|\vec{x}||\vec{y}|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

Handschriftenerkennung

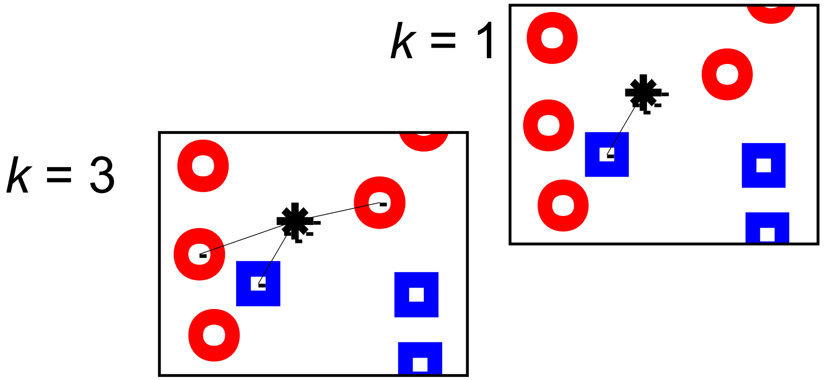
(1) Um die Bedeutung des Bildes p zu finden, finde das Trainingsbild x mit dist(p,x) minimal (durch lineare Suche über alle Trainingsdaten)

(2) Gib das Label von x aus



Erkennungsrate für k=1: 0.934

Mehrheitslabel für k=3: 0.945



Allgemein: K nächste Nachbarn

Idee:

- Finde in den Trainingsdaten den/die zur aktuellen Instanz ähnlichste(n) Instanz(en)
- Die aktuelle Instanz bekommt denselben Wert wie diese(r) Nachbar(n)

Schwierigkeiten:

- Welches Ähnlichkeitsmaß?
- Wieviele Nachbarn?
- Was, wenn die Werte der Nachbarn nicht übereinstimmen?
- Wie können wir die Suche effizient gestallten?

Allgemein: K näschte Nachbarn

Idee:

- Finde in den Trainingsdaten den/die zur aktuellen Instanz ähnlichste(n) Instanz(en)
- Die aktuelle Instanz bekommt denselben Wert wie diese(r) Nachbar(n)

Jede Instanz hat die Form $<\mathbf{x}_i, f(\mathbf{x}_i)>$

1. Berechne Abstand zwischen Testinstanz $\mathbf{x}_i$ und jeder Trainingsinstanz
2. Wähle die k-nächsten Nachbarn $\mathbf{n}_1, \ldots, \mathbf{n}_k$ aus
3. Der Wert/das Label für x ergibt sich durch:
   - $f(\mathbf{x}) = A(f(\mathbf{n}_1), \ldots, f(\mathbf{n}_k))$
   - A ist eine Auswahlfunktion

Ähnlichkeitsmaße für 0/1 Features

X seien die positiven Features von Instanz A und Y die positiven Features von Instanz B, die Ähnlichkeit von A und B ist dann:

Matching Koeffizient:	$\lvert X \cap Y \rvert$
Dice Koeffizient :	$2\lvert X \cap Y \rvert / (\lvert X \rvert + \lvert Y \rvert)$
Jaccard Koeffizient :	$\lvert X \cap Y \rvert / \lvert X \cup Y \rvert$
Overlap Koeffizient :	$\lvert X \cap Y \rvert / \min(\lvert X \rvert, \lvert Y \rvert)$
Kosinus:	$\lvert X \cap Y \rvert / (\lvert X \rvert \mathrm{x} \lvert Y \rvert)^{1/2}$

Was ist das richtige Ähnlichkeitsmaß?

Im Allgemeinen ist das schwer zu sagen:

„We know it when we see it“

Die Bedeutung von „Ähnlichkeit“ scheint eher ein philosophisches Problem zu sein. Maschinelles Lernen benutzt den Begriff und seine Bedeutung eher pragmatisch

Auswahlfunktion

Klassifikation: Mehrheitsentscheidung

Regression:

$$f(x) = \frac{\sum_{i=1}^{k} f(n_i)}{k}$$

Oder auch gewichtete Varianten:

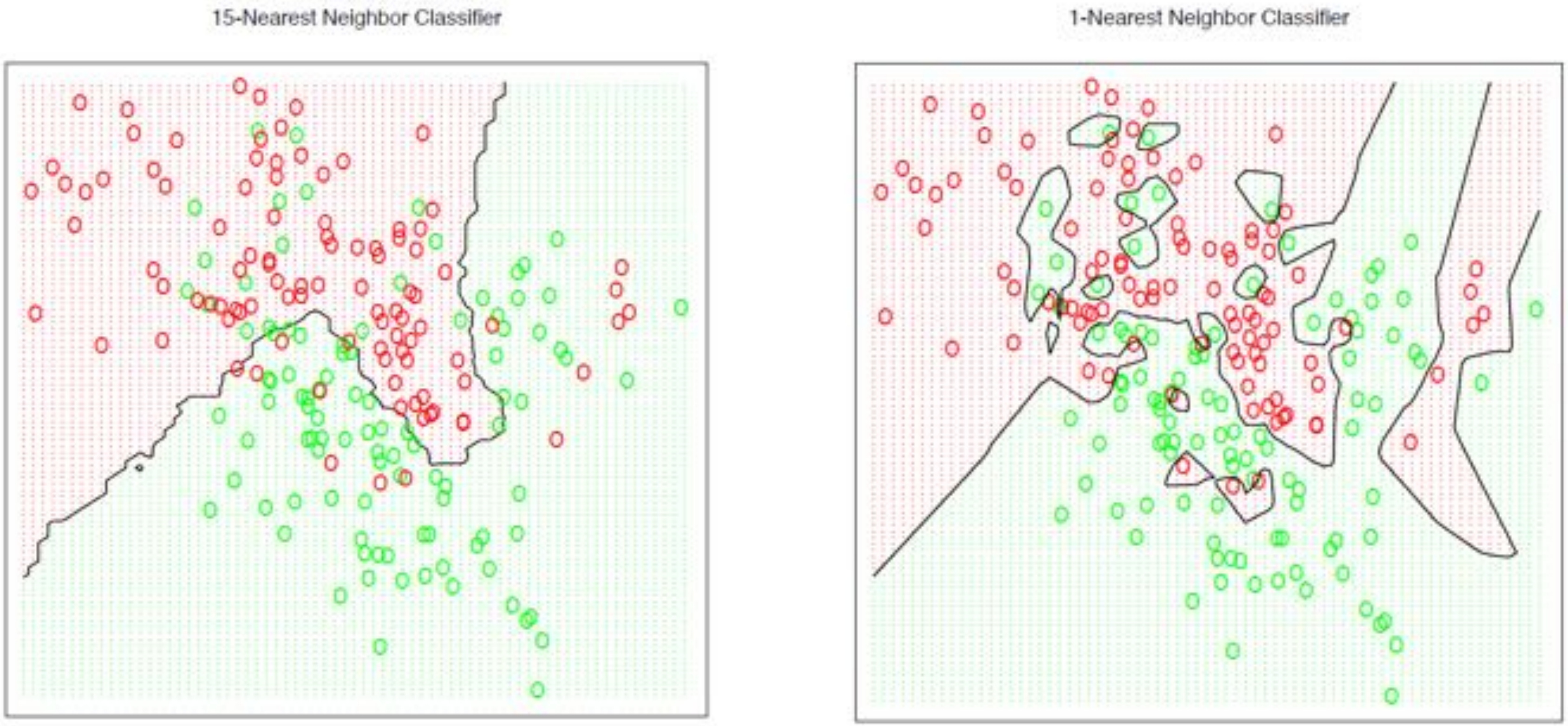
$$f(x) = \sum_{i=1}^{k} sim(n_i, x) f(n_i)$$

oder

$$f(x) = \frac{\sum_{i=1}^{k} w_i \times f(n_i)}{\sum_{i=1}^{k} w_i} \quad \text{mit} \quad w_i = \frac{1}{d(n_i, x)^2}$$

Overfitting

Die Wahl eines guten „k“ ist schwierig aber auch wichtig

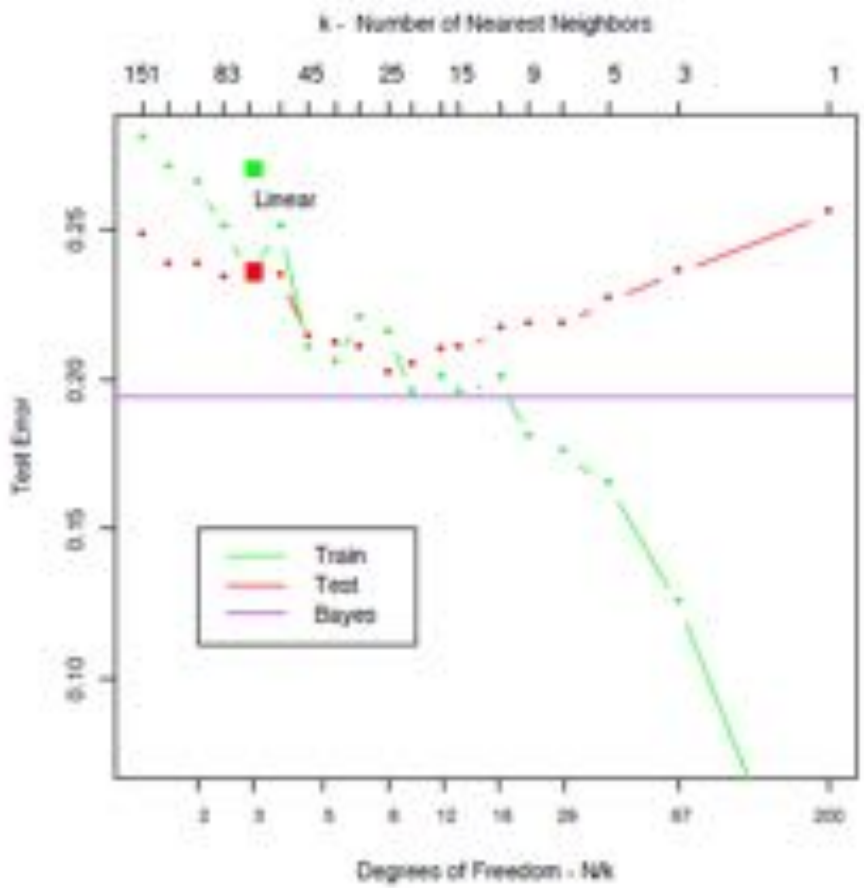


Overfitting

Falls aus **x** = **x*** gerade y = y* folgt, gibt es bei k = 1 keinen Trainingsfehler

Wenn allein der Trainingsfehler das Optimierungskriterium ist, würden wir stets k = 1 nehmen und nur auswendig lernen

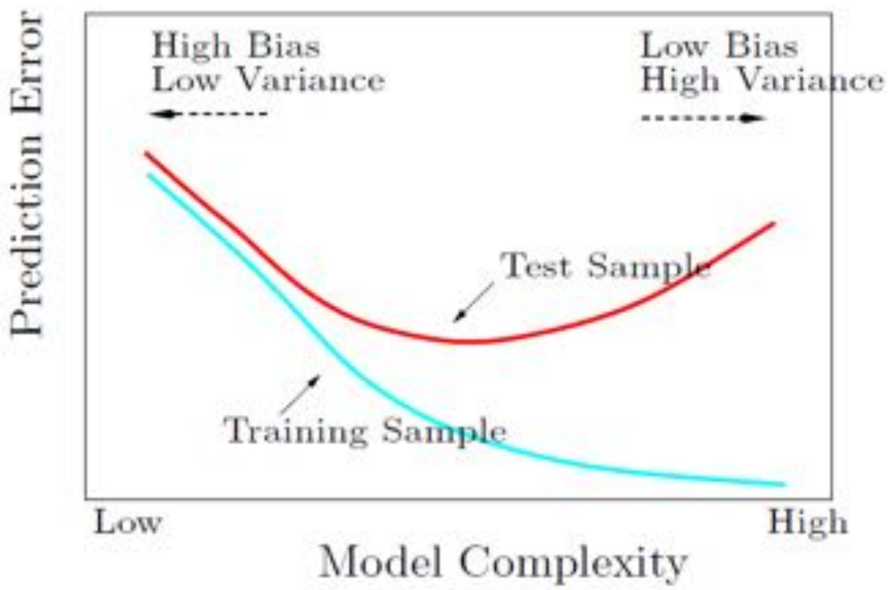
Wie wir gesehen haben, ergibt das vermutlich auf den Testdaten einen großen Fehler!



Bias-Varianz Tradeoff kNN

Wenn man die richtige, dicht besetzte Nachbarschaft hat, verzerrt kNN die Vorhersage nicht (kleiner Bias).

Wenn — wie bei hohen Dimensionen — die Nachbarschaft wild variiert, schwankt auch die Güte der Vorhersage (große Varianz).



Was wissen wir jetzt?

kNN ist ein einfacher Lernalgorithmus

Seine Performanz hängt von k und der gewählten Metrik ab

Fluch der hohen Dimensionen!

Naive Bayes Klassifikator

Eigentlich ist das Ziel der Klassifikation die Bestimmung einer insofern **optimalen Klassifikationsregel**, dass die Fehlerwahrscheinlichkeit dieser Regel minimal ist

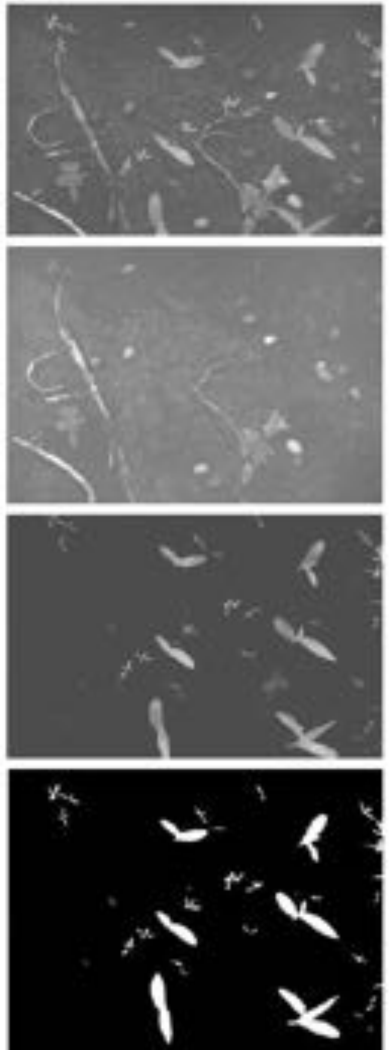
$$P(y_{\mathrm{Regel}}(x) \neq y_{\mathrm{wahr}}(x)) \;\rightarrow\; \mathrm{min!}$$

Erkennung von „Schrott“ in der Produktion: Klettenlabkraut in Winterweizen

Hohe Ertragsverluste, niedrige Schadschwelle, Behinderung der Erntearbeiten, hohe Lebensdauer des Samens

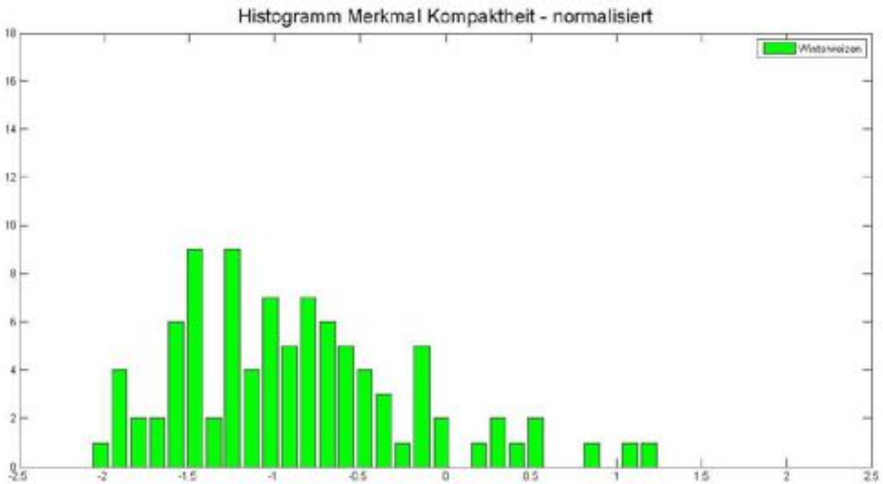
Datenvorverarbeitung / Segmentierung

1. Datenvorverarbeitung
   - Erfassung im Rot- und Infrarot-Bereich
2. Durch Ausnutzung der Temperaturdifferenz gelingt die Trennung von Pflanze und Untergrund/Boden
3. Resultiert in einem Differenzbild
   - Binärbild isoliert die Pflanzen
4. Bildverarbeitung liefert Formparameter
   - **Kompaktheit**
   - **Schlankheit**

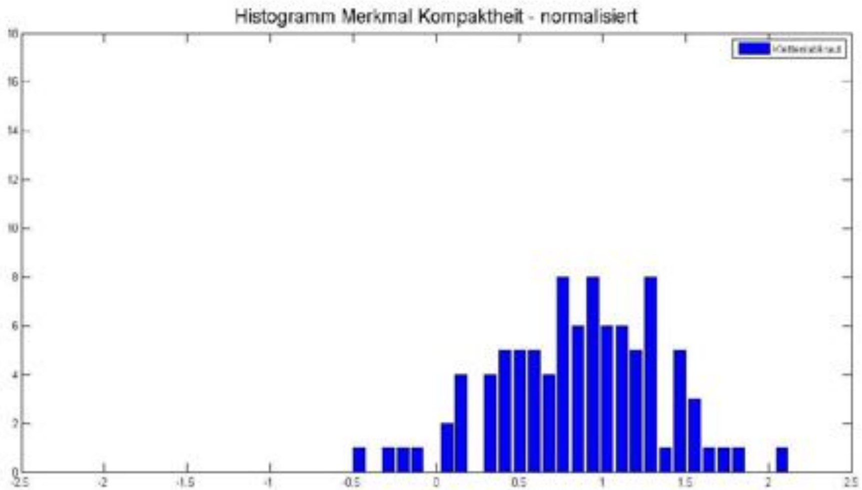


[Diss. OEBEL]

Histogramme für Kompaktheit



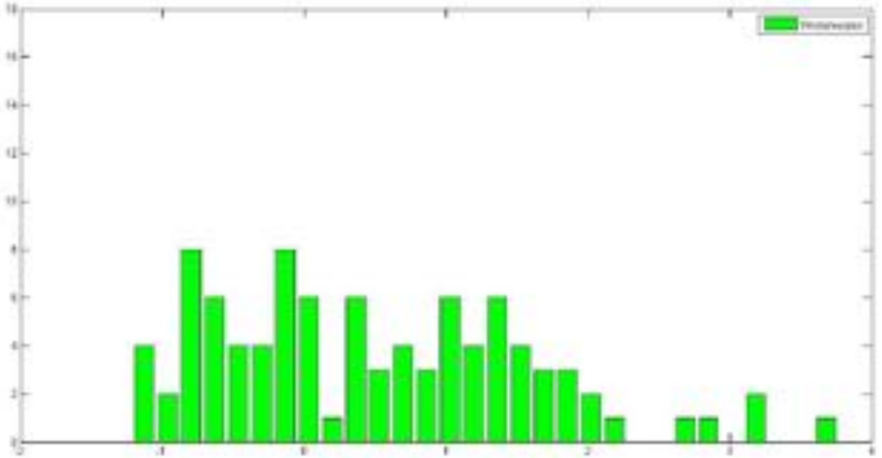
Winterweizen



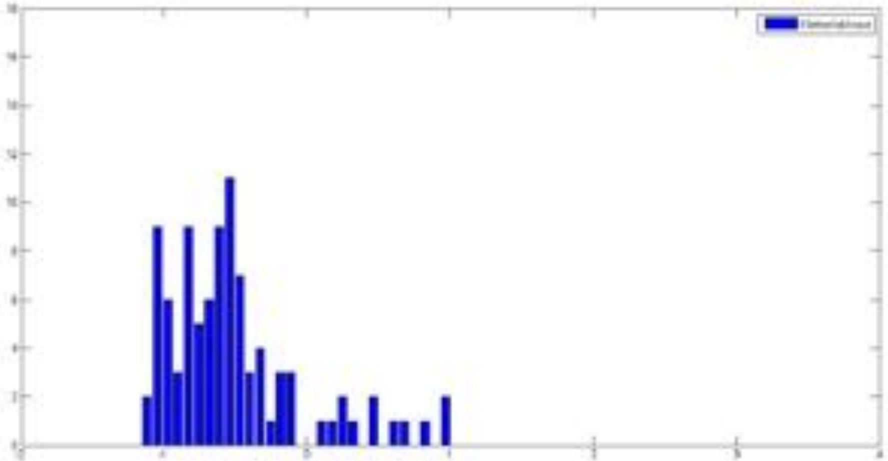
Klettenkleblaub

[Diss. OEBEL]

Histogramme für Schlankheit



Winterweizen

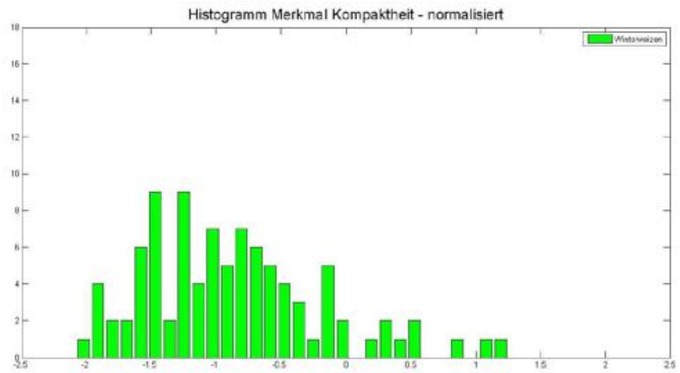


Klettenkleblaub

[Diss. OEBEL]

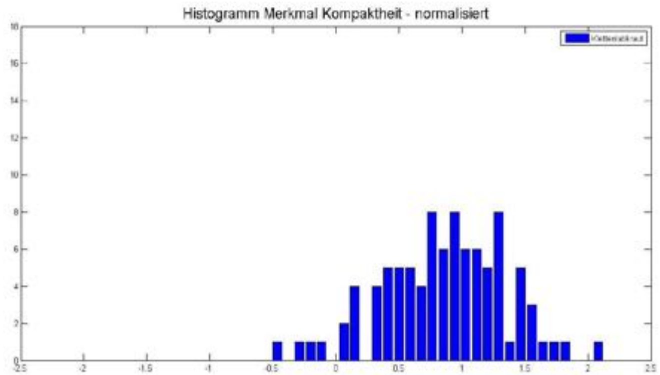
Bedingte Wahrscheinlichkeit (für das Merkmal „Kompaktheit“)

Das Merkmal „Kompaktheit“ hat offensichtlich unterschiedliche Verteilungen für Winterweizen und Klettenkleblaub



Winterweizen

P(A|B) ist die Wahrscheinlichkeit von A unter der Bedingung B



Klettenkleblaub

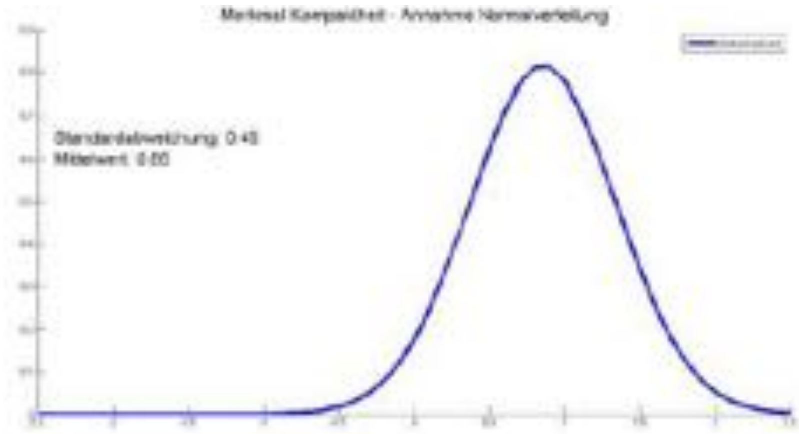
hier: die Wahrscheinlichkeit der Beobachtung eines bestimmten Werts der Kompaktheit unter der Bedingung z.B. „Klettenlabkraut“

Annahme: Kompaktheit „normalverteilt“

Machen wir die vereinfachende Annahme, das Merkmal Kompaktheit sei **„normalverteilt“**

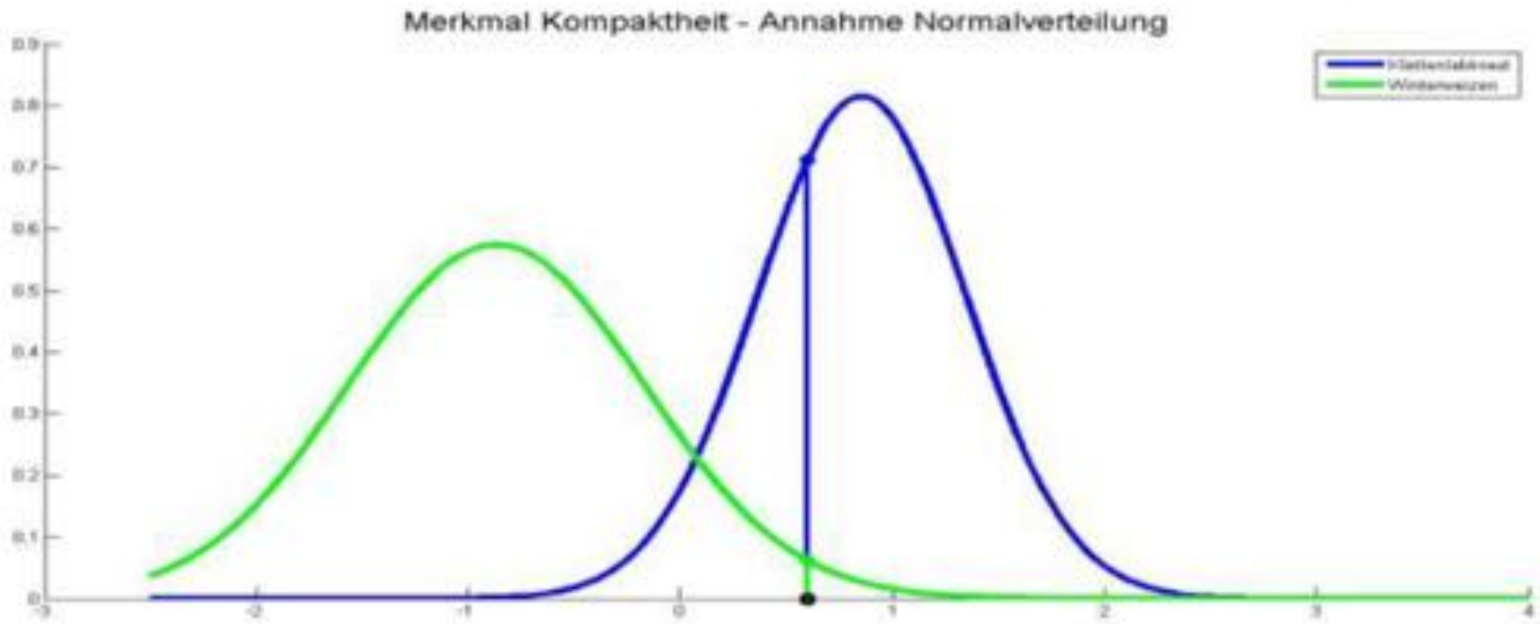
Dann wird jede der beiden Verteilungen durch die beiden Parameter **Mittelwert** und **Standardabweichung** eindeutig beschrieben

Die Wahrscheinlichkeit der Beobachtung einer bestimmten Ausprägung des Merkmals Komapktheit beschreibt die Wahrscheinlichkeitsdichte-Funktion pdf („Probability Density Function) – hier auch „Likelihood“ genannt



Klettenkleblaub

Maximum-Likelihood-Klassifikator

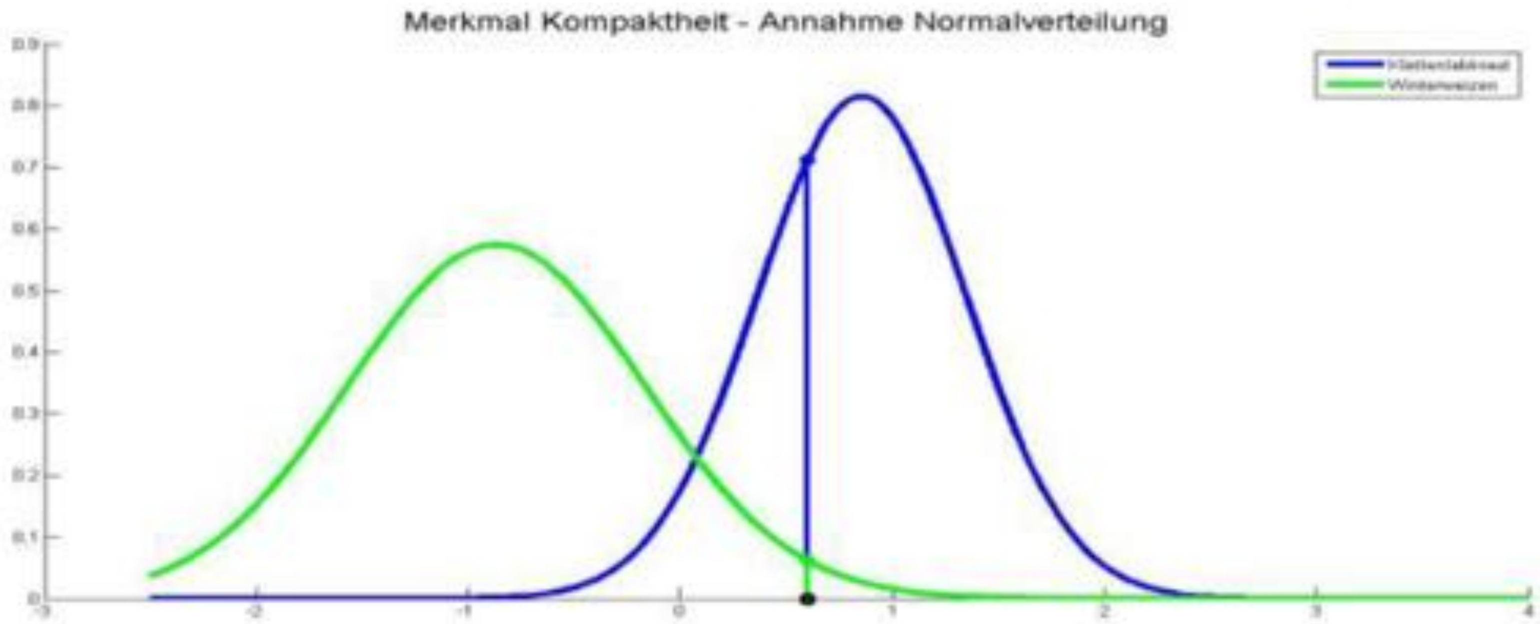


p_Klettenlabkraut = p(kompaktheit | Klettenlabkraut)

p_Winterweizen   = p(kompaktheit | Winterweizen)

Der Likelihood-Klassifikator wählt die Klasse mit der höchsten Likelihood

Zahlenbeispiel



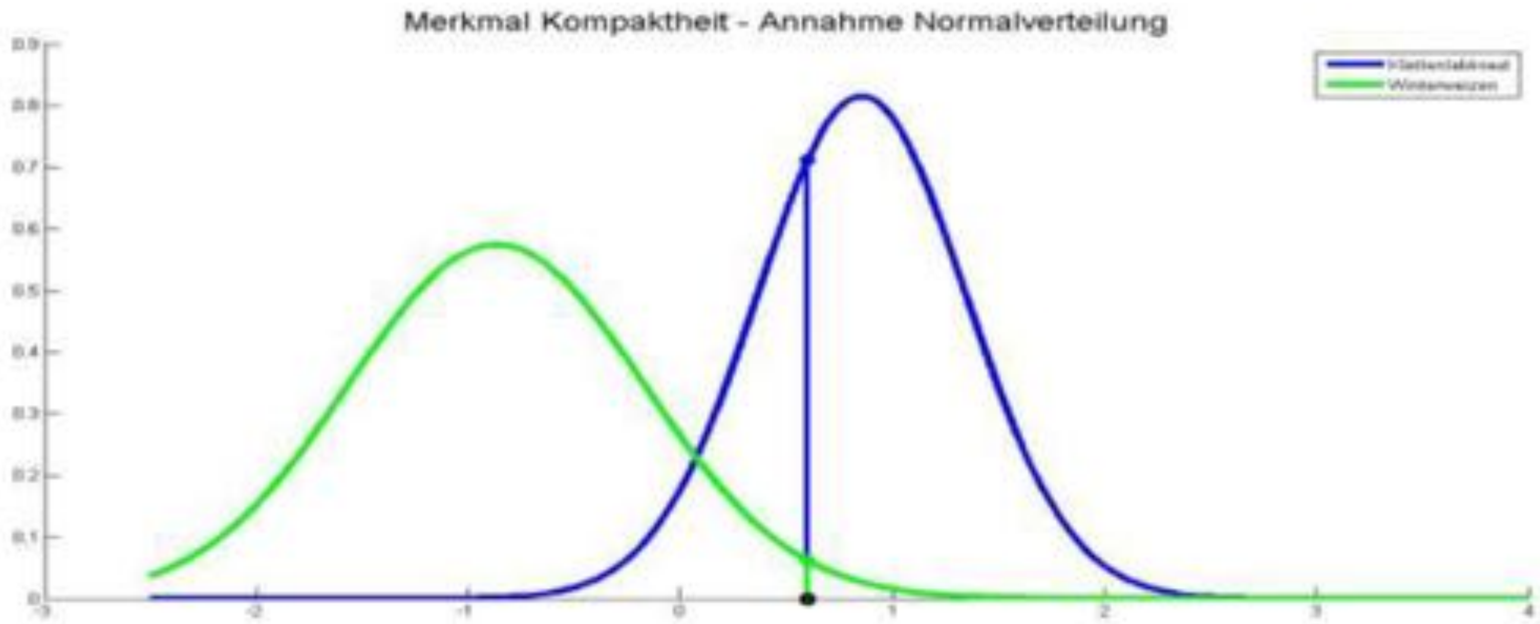
Sei die Kompaktheit = 0.6

p(0.6 | Klettenlabkraut) = 0.7119

p(0.6 | Winterweizen) = 0.0614

Vorhersage/Klassifikation: **Klettenlabkraut**

„A Priori“ Gleichverteilt



Achtung! Wir haben eine (vielleicht unrealistische) Annahme getroffen:

die „a priori“ Wahrscheinlichkeiten beider Klassen seien gleich

p(Klettenlabkraut) = p(Winterweizen) = 0.5

Dann muss die Summe beider Wahrscheinlichkeiten 1 ergeben. Durch „Normierung“ erhalten wir dann aus den beiden Likelihoods die Wahrscheinlichkeit der Klassifikation: p(Klettenlabkraut) = 0.7119 / (0.0614 + 0.7119) = 0.92

Im Allgemeinen: Satz von Bayes

$$P(K|M) = \frac{P(M|K) * P(K)}{P(M)}$$

$$\text{wenn } P(M) > 0$$

$$P(Klasse|Merkmal) = \frac{Likelihood * Prior}{P(M)}$$

$$\text{wenn } P(M) > 0$$

Bayes´scher Klassifiktor (I)

Likelihood: bedingte Wahrscheinlichkeit des Merkmals

Prior: Vorwisssen über die Häufigkeit des Auftretens der Klasse (des Labels)

- p(Winterweizen) = 0.95 und p(Klettenlabkraut) = 0.05
- Der „Prior“ für **W** ist wesentlich höher als für **K**

$$P(Klasse|Merkmal) = \frac{Likelihood * Prior}{P(M)} \text{ wenn } P(M) > 0$$

Aber wie geht das an? P(Merkmal) also die Verteilung der Daten ist doch in den meisten Fällen nicht bekannt ?! Da wir maximieren, und P(Merkmal) für alle Klassen gleich ist, ist das kein Problem.

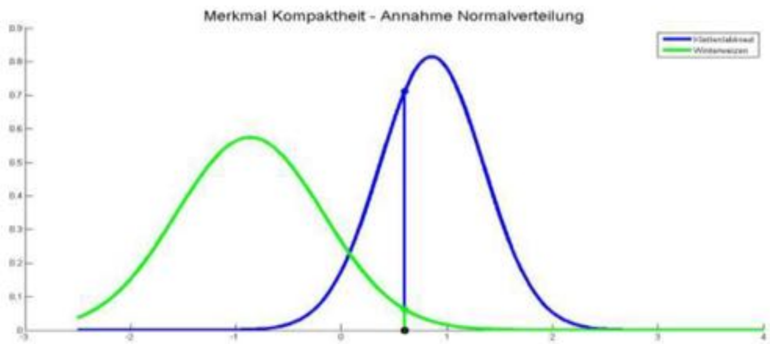
$$Klasse^* = \arg\max_{Klasse} Likelihood(Klasse) * Prior(Klasse)$$

Wenn wir die wirkliche „a posteriori“ Wahrscheinlichkeit berechnen wollen, können wir ausnutzen, dass P(Merkmal) eine „normalisierende Konstante“. Dabei hilft die Einsicht, dass die Gesamtwahrscheinlichkeit 1 ergeben muss

Zahlenbeispiel

p(0.6 | Klettenlabkraut) = 0.7119

p(0.6 | Winterweizen) = 0.0614



p(Winterweizen) = 0.95

p(Klettenlabkraut) = 0.05

p(Winterweizen |0.6) = 0.0614 * 0.95 / p(0.6) = 0.0583 / p(0.6)

p(Klettenlabkraut |0.6) = 0.7119 * 0.05 / p(0.6) = 0.0356 / p(0.6)

p(Winterweizen |0.6) = 0.0583 / (0.0583 + 0.0356) = 0.6212

p(Klettenlabkraut |0.6) = 0.0356 / (0.0583 + 0.0356) = 0.3788

Bayes´scher Klassifikator (II)

Der Bayes´sche Klassifikator setzt voraus, daß wir zwei Verteilungen kennen, den **Likelihood P(M| K)** und den **Prior P(K)**

Das "Lernen" des Klassifikators besteht in dem Fall im **Schätzen dieser beiden Verteilungen**

Z.B. bei einer **Gaußverteilung (Normalverteilung)** müssen wir **Mittelwert** und **Standardabweichung** schätzen

Bayes´scher Klassifikator (III)

Wir haben hier den **univariaten** Fall diskutiert (nur Kompaktheit)
Es geht aber auch bei mehreren Merkmalen (**multivariater** Fall).
Dann brauchen wir die **Kovarianzmatrix** (siehe vorherige Vorlesungen)

Oder wir treffen eine Unabhängigkeitsannahme (**Naive Bayes**):

$$P(M|k) = P(M_1, M_2, \ldots, M_l|k) = \prod_{i=1}^{l} P(M_i|k)$$

$$\frac{P(M|k_1) * P(k_1)}{P(M|k_1) * P(k_1) + P(M|k_2) * P(k_2)}$$

Bayes´scher Klassifikator (IV)

Wir haben hier den **univariaten** Fall diskutiert (nur Kompaktheit)

Es geht aber auch bei mehreren Merkmalen (**multivariater** Fall).

Dann brauchen wir die **Kovarianzmatrix** (siehe vorherige Vorlesungen)

Der Bayes´sche Klassifikator hat **drei wesentliche Vorteile**:

- Er kann “Vorwissen” in Form des Priors nutzen. Zum Beispiel die Tatsache, dass in einem Schlag “a priori” Klettenlabkraut seltener ist als Weizen
- Er liefert optimale Ergebnisse (unter den getroffenen Annahmen,)
- Er kann die Güte der Klassifikation in Form einer Wahrscheinlichkeit angeben:

$$\frac{P(M|k_1) * P(k_1)}{P(M|k_1) * P(k_1) + P(M|k_2) * P(k_2)}$$

Was wissen wir jetzt?

Bayes-Klassifikatoren betrachten die Verteilung, die die Daten generiert haben

Sie treffen „optimale“ Entscheidungen mit Hilfe der Bayes-Regel

Der Naive Bayes Klassifikator nimmt an, dass alle möglichen Teilmengen von Merkmalen voneinander unabhängig sind, geben das Klassenlabel

Was haben Sie kennengelernt?

- Graphische Modelle sind Arbeitstiere der Datenwissenschaften.
- Sie spezifieren Wahrscheinlichkeitsverteilungen und erlauben das Schlussfolgern unter Unsicherheit.
- Unvollständige Daten machen die Parameterschätzung schwierig, weil die Likelihood-Funktion multimodal ist und keine geschlossene Form hat.
- In diesem Falle können Sie die ML-Parameter mittels Expectation-Maximization (EM) schätzen.
- Dazu müssen wir schlussfolgern. Das ist (NP-)schwierig!

Von Bayes´schen zu Diskriminativen Ansätzen

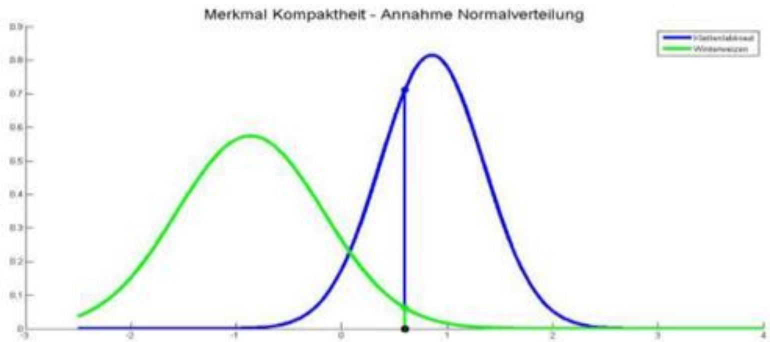
Der Bayes‘sche Ansatz:

Falls $P(x|k_1) * P(k_1) > P(x|k_2) * P(k_2)$ wähle k1, sonst k2

Das motiviert die Suche nach einem **Schwellwert x0** mit

$$P(x_0|k_1) * P(k_1) = P(x_0|k_2) * P(k_2)$$

Dieser Schwellwert ist ein (lineare) Diskriminator (siehe vorherige Vorl.)

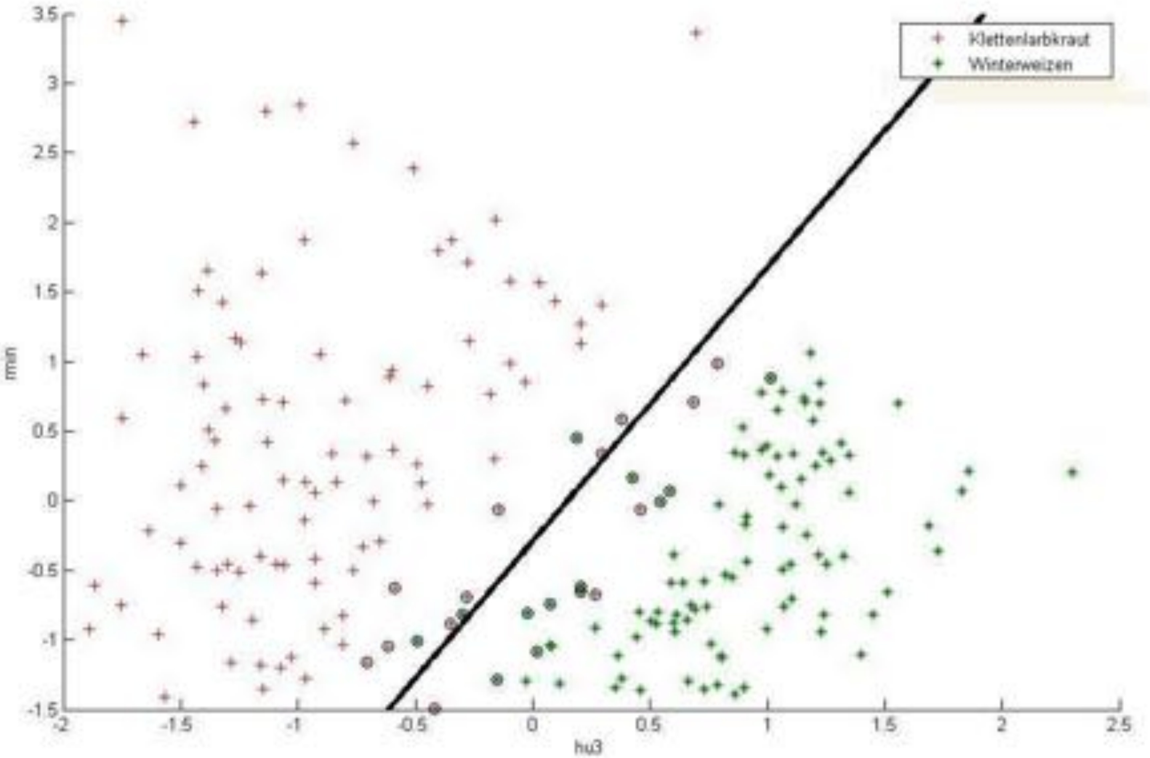


Im Allgemeinen, sind das:

- Ein Merkmal: Schwellwert
- Zwei Merkmale: Gerade
- Drei Merkmale: Ebene
- N Merkmale: (Lineare) Hyperebene

Lineare Trennung

Kombination beider Merkmale: die beiden Klassen sind durch eine Gerade offenbar recht gut trennbar



Lineare Trennung: Minimierung des beobachteten Fehlers

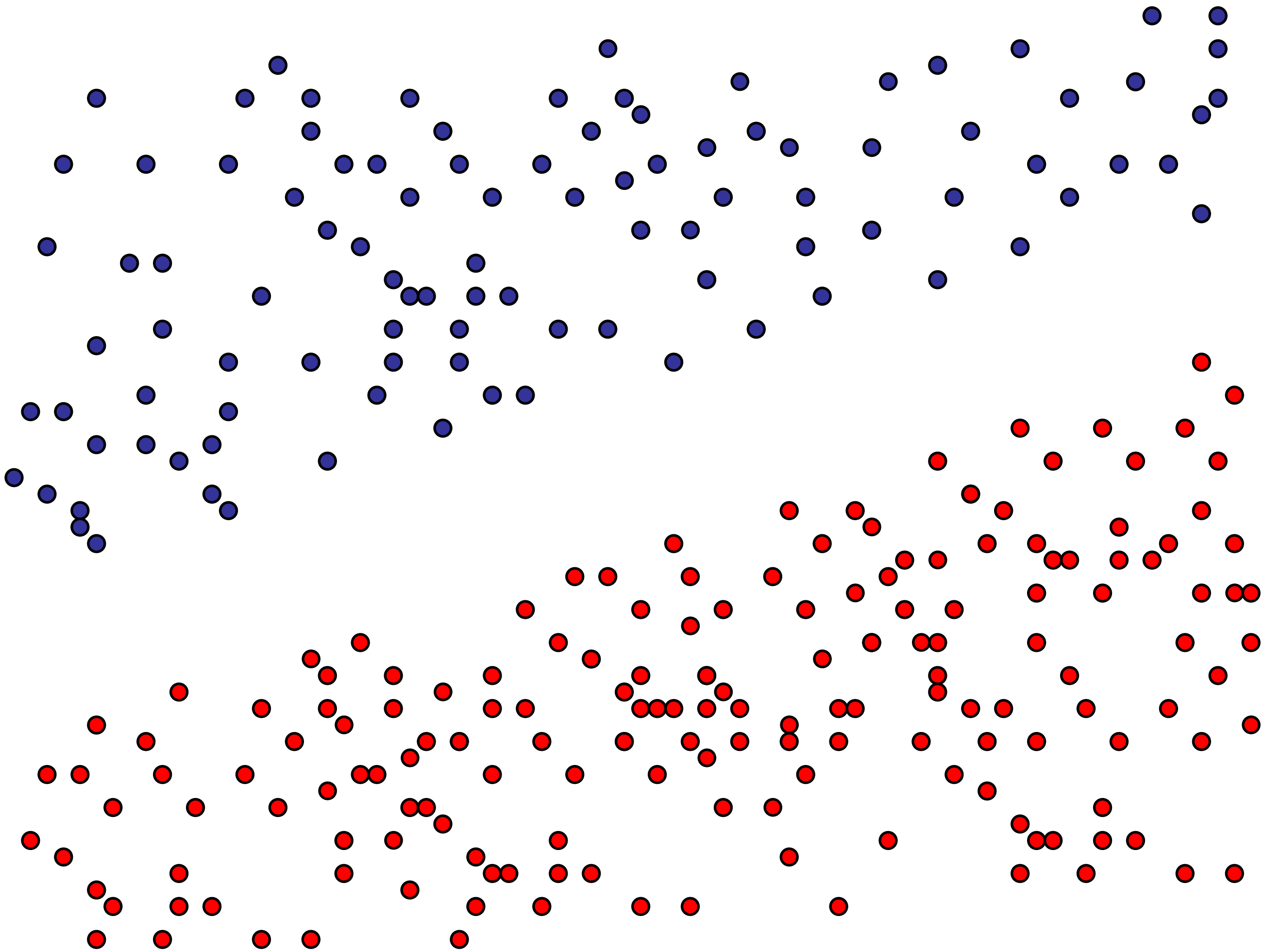
Funktionslernaufgabe nicht direkt lösbar. Problem:

- Die tatsächliche Funktion t(X) ist unbekannt.
- Die zugrunde liegende Wahrscheinlichkeit ist unbekannt.

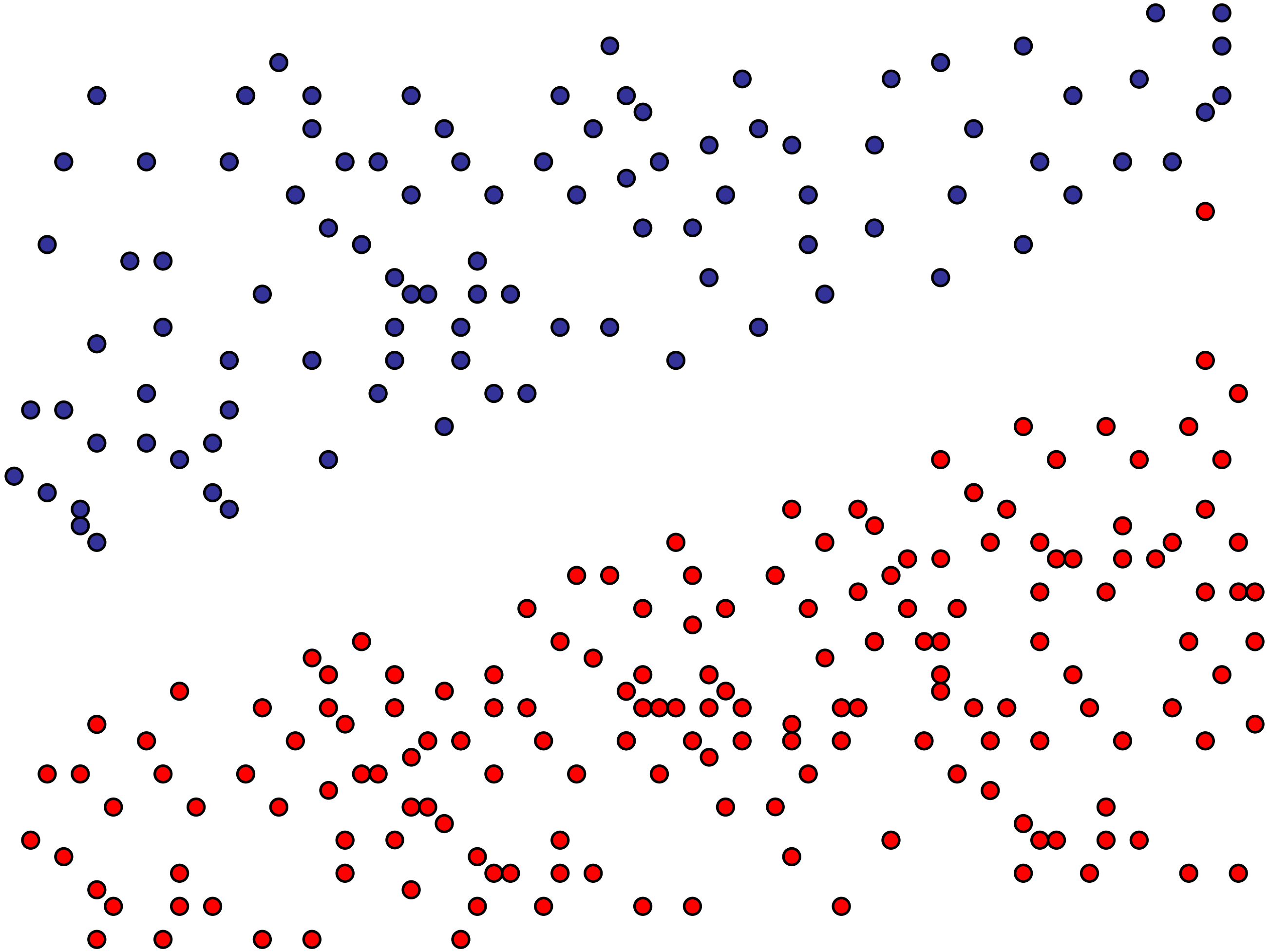
Ansatz der Empirical Risk Minimization (ERM):

eine hinreichend große Lernmenge nehmen und für diese den Fehler minimieren.

Lineare Trennung: Minimierung des beobachteten Fehlers — Beispiel 1



Lineare Trennung: Minimierung des beobachteten Fehlers — Beispiel 2

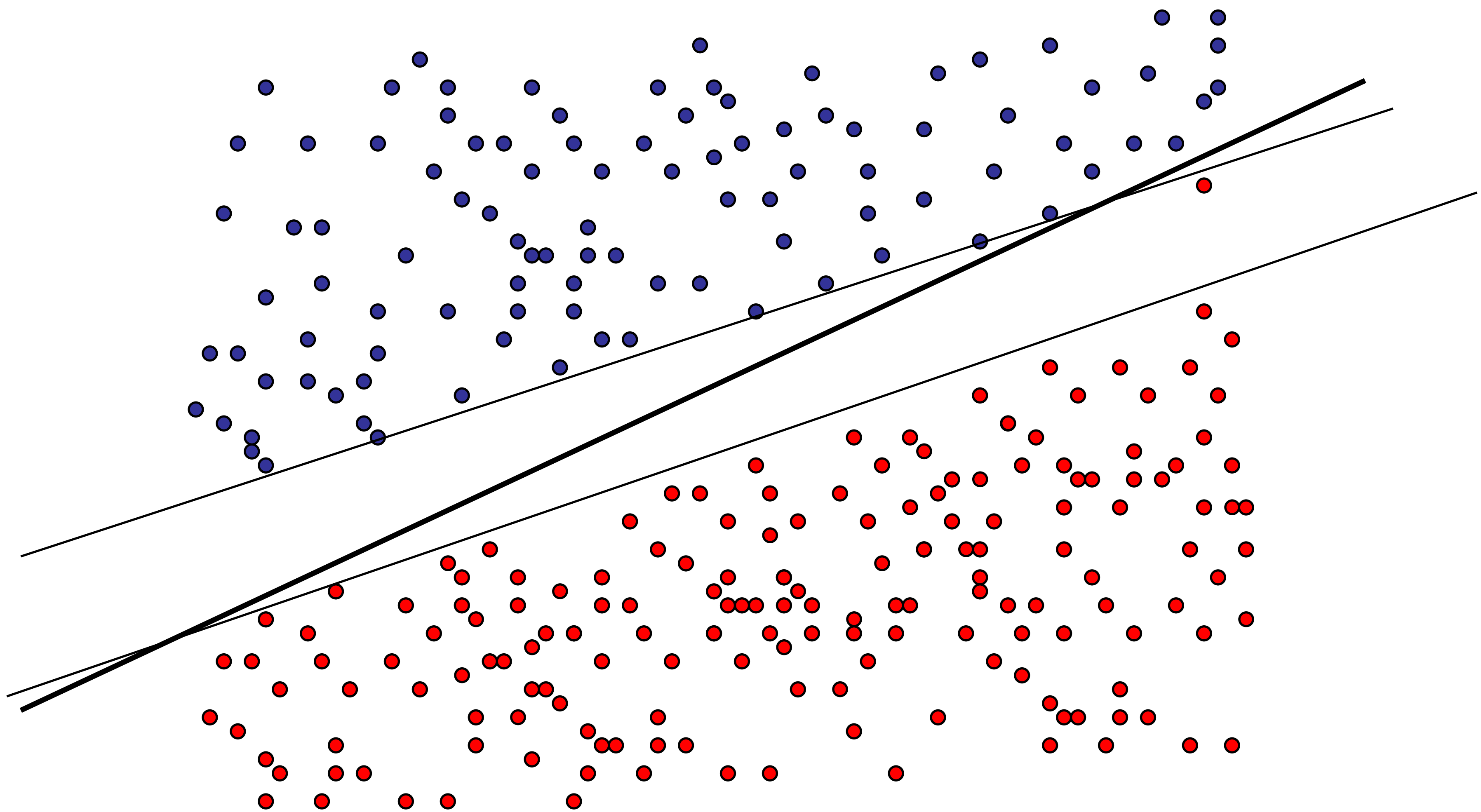


Probleme der ERM

Aufgabe ist nicht eindeutig beschrieben: Mehrere Funktionen mit minimalem Fehler existieren. Welche wählen?

Overfitting: Verrauschte Daten und zu wenig Beispiele führen zu falschen Ergebnissen.

Lineare Trennung: Minimierung des beobachteten Fehlers — Beispiel 3



Stützvektormethode: Grundlagen

Zwei-Klassen-Problem:

- Trainingsdaten $(x_1, y_1), \ldots, (x_m, y_m)$, $x_m \in X$, $y_m \in \{+1, -1\}$
- Ähnlichkeit eines neuen x_i bestimmt y_i
- Ähnlichkeitsmaß $k: X \times X \rightarrow \Re$

  $(x, x') \rightarrow k(x, x')$

  z.B. Skalarprodukt $x*x' := \sum [x]_i [x']_i$

Stützvektormethode: Grundlagen

Skalarprodukt x*y: Seien x und y Vektoren aus $\Re^p$

$$x * y = \sum_{i=1}^{p} [x]_i [y]_i$$

Euklidsche Länge (Betrag) eines Vektors ||x||:

$$\|x\| = \sqrt{x * x} = \left(\sum_{i=1}^{p} [x]_i^2 \right)^{\frac{1}{2}}$$

Hyperebene H: Sei w≠0 der Normalenvektor und b∈$\Re$ der bias

$$H(w,b) = \{x | w * x + b = 0\}$$

Warum Skalarprodukt?

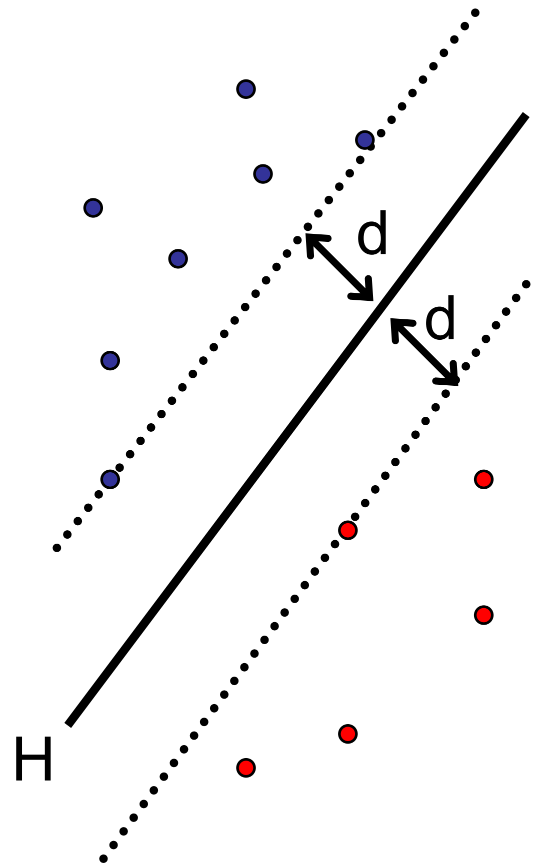
- Cosinus des Winkels zwischen x und x‘, wenn beide Vektoren auf die Länge 1 normiert sind.
- Abstand zwischen x und x‘ ist Länge des Differenzvektors.
- Voraussetzung: Beispiele sind Vektoren.
- Überführung in einen Raum mit Skalarprodukt $\Phi : X \rightarrow \mathcal{H}$
- Wenn X bereits ein Raum mit Skalarprodukt ist, kann nicht-lineare Abbildung Φ auch sinnvoll sein.

Die optimale Hyperebene

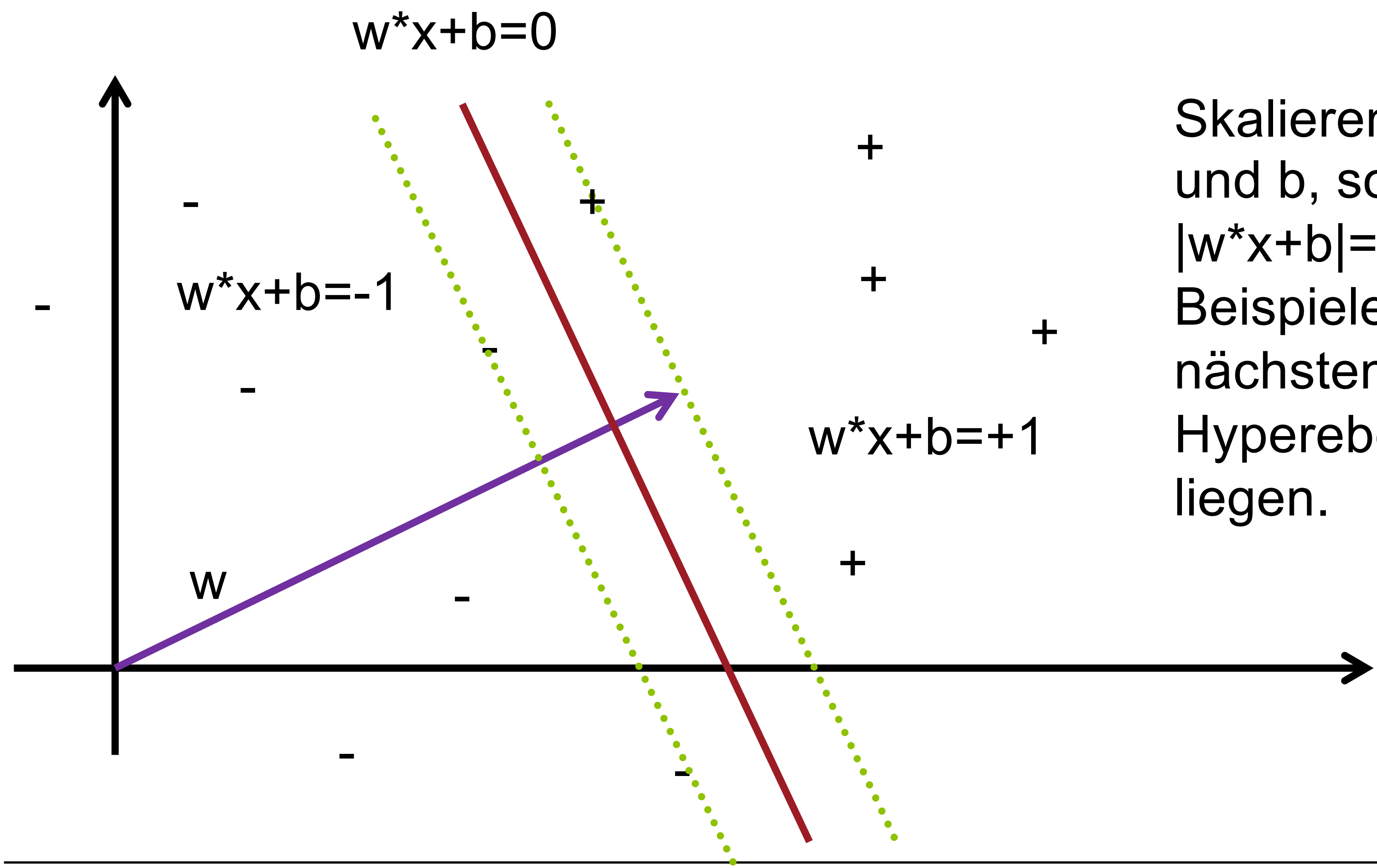
- Beispiele heißen linear trennbar, wenn es eine Hyperebene H gibt, die die positiven und negativen Beispiele voneinander trennt.
- H heißt optimale Hyperebene, wenn ihr Abstand d zum nächsten positiven und zum nächsten negativen Beispiel maximal ist.
- Satz: Es existiert eine eindeutig bestimmte optimale Hyperebene.

Der Normalenvektor steht senkrecht auf allen Vektoren der Hyperebene. Es gilt:

$$w * x + b \begin{cases} > 0 \ \textit{falls } x \textit{ im positiven Raum} \\ = 0 \ \textit{falls } x \textit{ auf } H \\ < 0 \ \textit{falls } x \textit{ im negativen Raum} \end{cases}$$



Verbildlicht

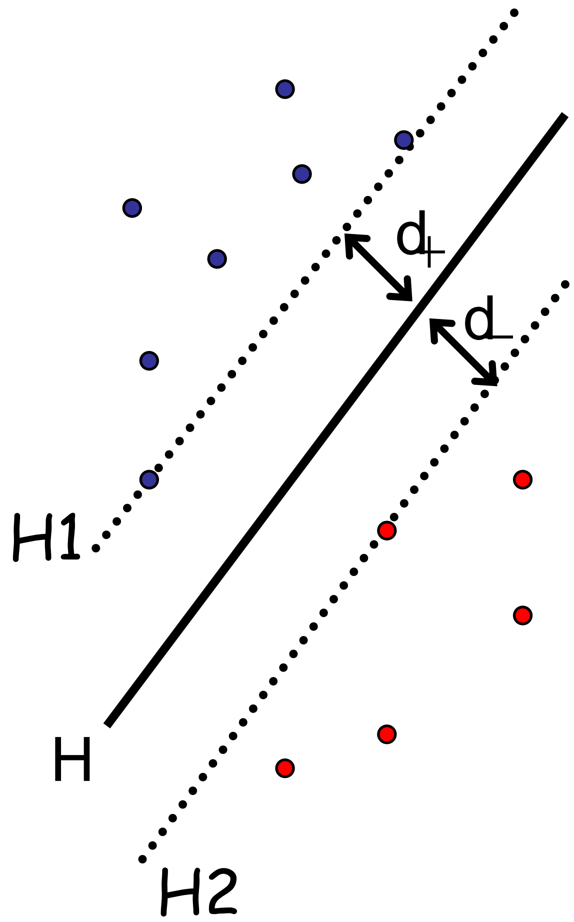


Skalieren von w und b, so dass |w*x+b|=1 für alle Beispiele, die am nächsten zur Hyperebene liegen.

Margin

- H1 und H2 sind parallel, haben denselben Normalenvektor w.
- Per Konstruktion liegt kein Beispiel zwischen H1 und H2.
- Um 2 / ||w|| zu maximieren, müssen wir ||w|| minimieren.
- Die Nebenbedingungen müssen eingehalten werden:

$$\forall i : y_i(x_i * w + b) - 1 \geq 0$$



Stützvektormethode: Optimierungsaufgabe

Minimiere $||w||^2$

so dass für alle i gilt:

$f(x_i) = w*x_i+b \geq 1$ für $y_i = 1$ und

$f(x_i) = w*x_i+b \leq -1$ für $y_i = -1$

Äquivalente Nebenbedingungen: $y_i*f(x_i) - 1 \geq 0$

Konvexes, quadratisches Optimierungsproblem $\Rightarrow$ eindeutig in $O(n^3)$ für n Beispiele lösbar.

Satz: $||w|| = 1/d$, d = Breite der optimalen Margin bzgl. der Beispiele.

SVM Optimierungsproblem
(mittels Lagrange und Dualität, hier nicht angesprochen)

Maximiere

unter $0 \leq \alpha_i$ für alle i und $\sum \alpha_i y_i = 0$

$$\boxed{L(\alpha) = \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \left(x_i * x_j\right)}$$

Für jedes Beispiel gibt es ein α in der Lösung.

- $0 = \alpha_i$ heißt, dass das Beispiel x_i im passenden Halbraum liegt.
- $0 < \alpha_i$ heißt, dass das Beispiel x_i auf H1 oder H2 liegt (Stützvektor).

Es gilt $w = \sum \alpha_i y_i x_i$ und somit $f(x) = \sum \alpha_i y_i (x_i * x) + b$. Also ist der beste Normalenvektor w eine Linearkombination von Stützvektoren ($\alpha_i \neq 0$).

Was wissen wir jetzt?

Maximieren der Margins einer Hyperebene ergibt eine eindeutige Festlegung der optimalen trennenden Hyperebene.

Dazu minimieren wir die Länge des Normalenvektors w.

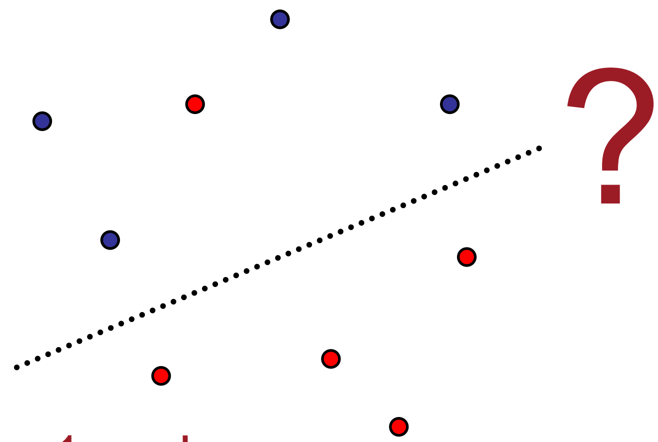
- Formulierung als Lagrange-Funktion
- Formulierung als duales Optimierungsproblem

Das Lernergebnis ist eine Linearkombination von Stützvektoren.

Mit den Beispielen müssen wir nur noch das Skalarprodukt rechnen.

Nicht linear trennbare Daten

In der Praxis sind linear trennbare Daten selten.



Wähle $C \in \Re_{>0}$ und minimiere

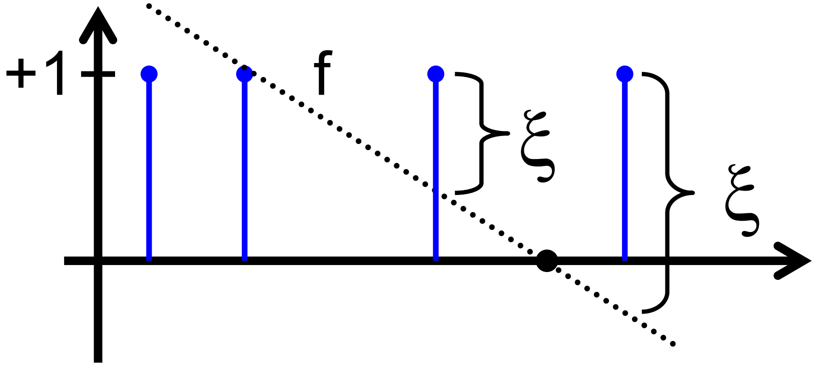
$$\boxed{\|w\|^2 + C\sum_{i=1}^{n} \xi_i}$$

so dass für alle i gilt:

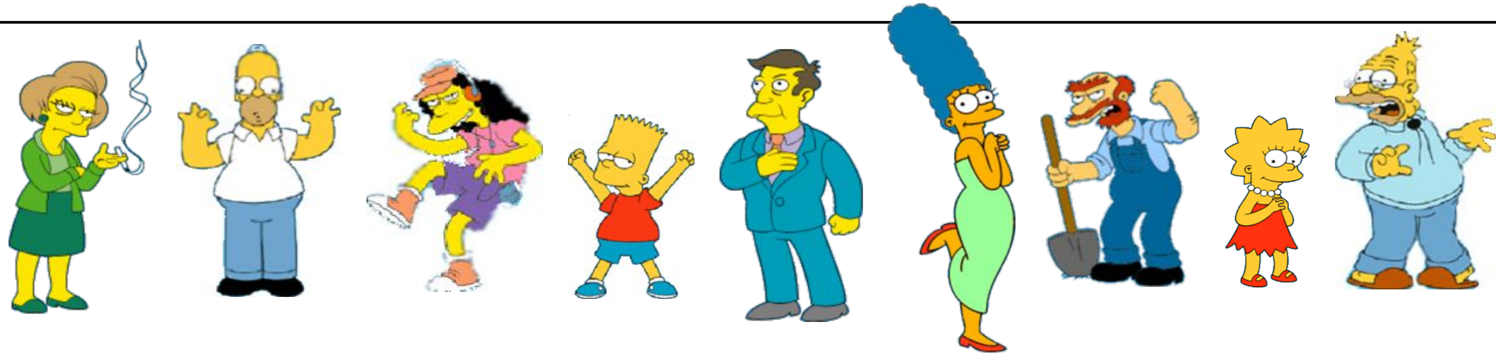
$f(x_i) = w^*x_i+b \geq 1-\xi_i$ für $y_i = 1$ und

$f(x_i) = w^*x_i+b \leq -1+\xi_i$ für $y_i = -1$

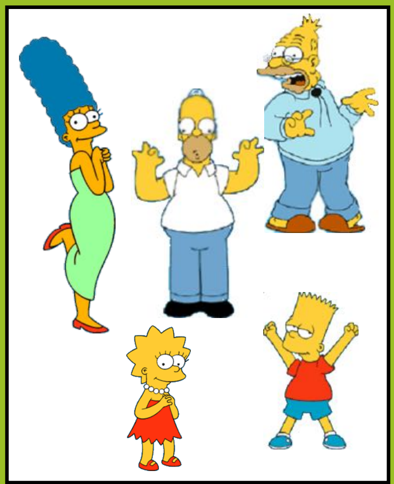
Äquivalent: $y_i^*f(x_i) \geq 1- \xi_i$



Cluster-Analyse:
Wie würden Sie die Simpsons gruppieren?

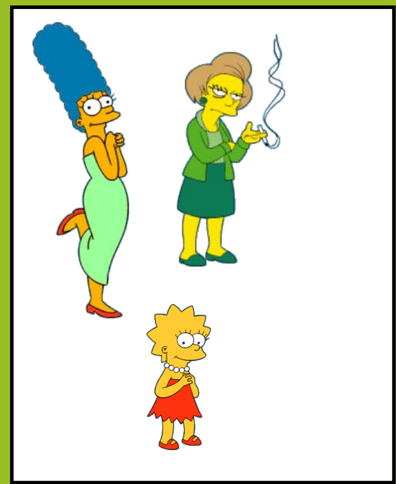


Clusteranalyse ist subjektiv



Die Simpsons

Schulangestellte



Weiblich

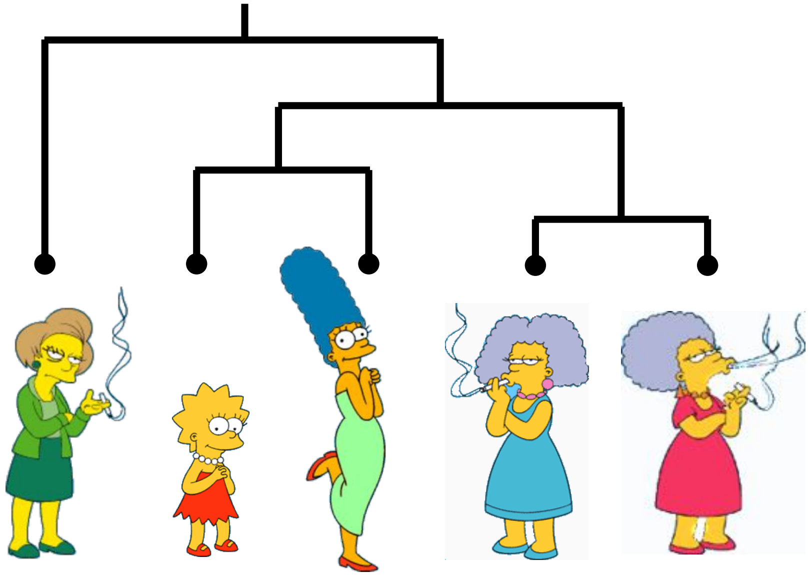
Männlich

Zwei Arten der Clusteranalyse

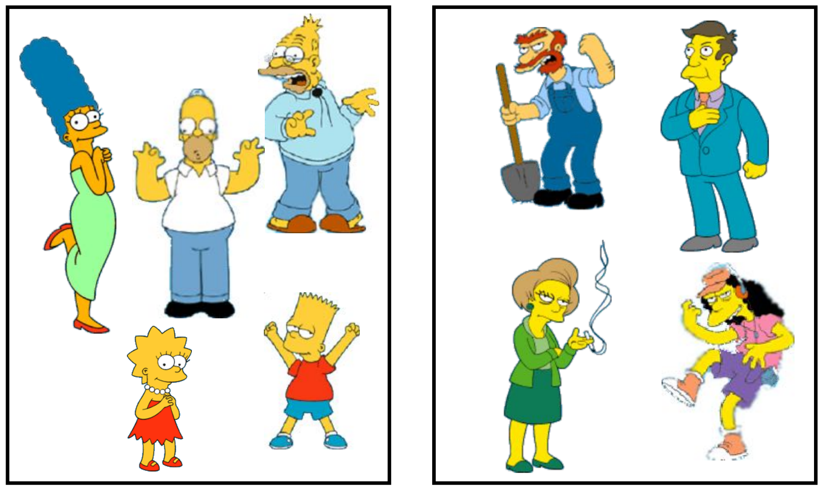
Partitionierungsansätze: Man konstruiert Partitionierungen (Aufteilungen) der Daten und bewertet sie mittels einer Bewertungsfunktion

Hierarchische Ansätze: Konstruiere eine hierarchische Aufteilung der Daten anhand eines Kriteriums

Hierarchisch



Partition



Distanzmaße

Beide Arten benötigen im Wesentlichen eine Distanzfunktion:

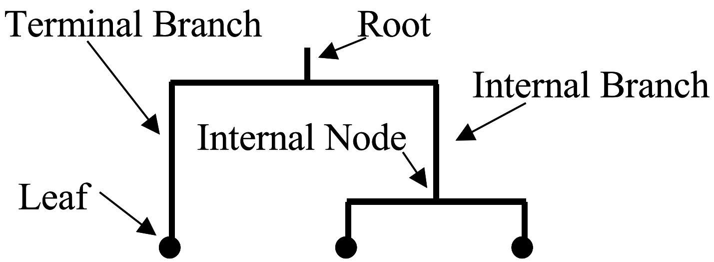
- D(A,B) = D(B,A) *Symmetrie*
- D(A,A) = 0 *Konstanz der Selbstähnlichkeit*
- D(A,B) = 0 If A= B *Positiv*
- D(A,B) $\leq$ D(A,C) + D(B,C) *Dreiecksunähnlichkeit*

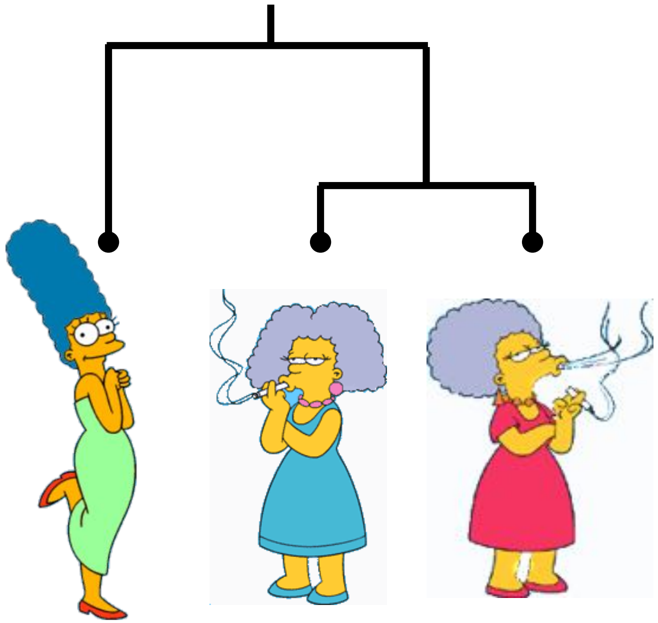
"We know it when we see it"

Dendrogramme

Ähnlichkeit können wir auch durch einen Baum darstellen:

Die Ähnlichkeit zweier Objekte wird in einem Dendrogram durch die Höhe (von den Blättern ausgesehen) des niedrigsten internen Knoten ausgedrückt, den beide Objekte gemeinsam haben

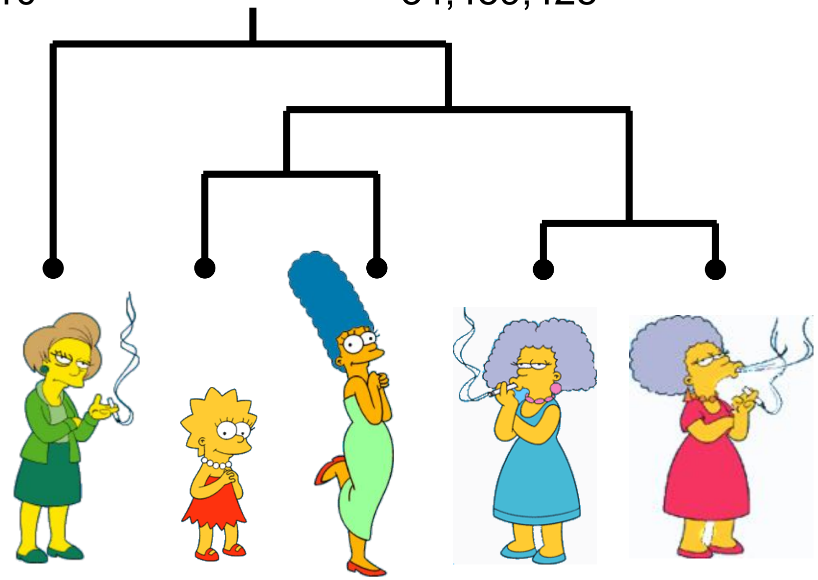




Hierarchisches Clustering

Die Zahl der Dendrogramme mit *n* Blättern
$$= (2n-3)!/[(2^{(n-2)})\,(n-2)!]$$

Zahl der Blätter	Zahl der möglichen Dendrogramme
2	1
3	3
4	15
5	105
...	…
10	34,459,425



Weil wir nicht alle durchtesten können, müssen wir uns auf Heuristiken beschränken:

Bottom-Up (Agglomerativ): Anfangs ist jedes Objekt sein eigenes Cluster. Finde die beiden Cluster, die sich am ähnlichsten sind, und vereinige (merge) sie. Wiederhole das solange, bis es nur noch ein Cluster gibt.

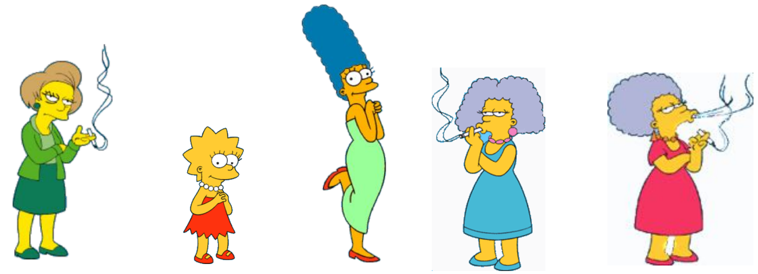
Top-Down (Aufteilend): Anfangs sind alle Objekte in einem Cluser. Finde den besten Split und führe diesen aus. Wiederhole das so oft, bis die Cluster nur noch aus einem Objekt bestehen

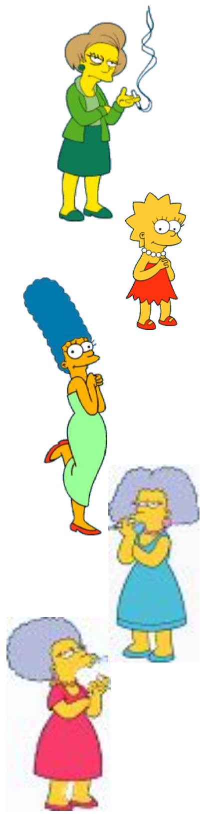
Beispiel Bottom-Up

Wir haben eine Distanzmatrix, die alle paarweisen Distanzen enthält

D() = 8

D() = 1



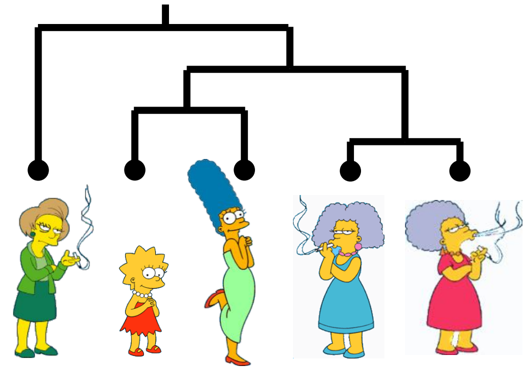


0	8	8	7	7
	0	2	4	4
		0	3	3
			0	1
				0

Beispiel Bottom-Up

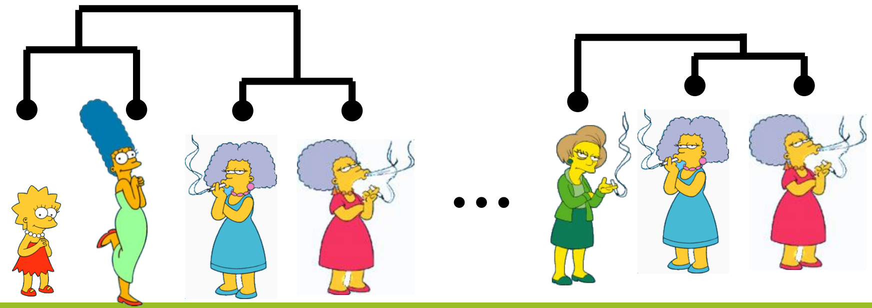
Anfangs ist jedes Objekt sein eigenes Cluster. (1) Finde die beiden Cluster, die sich am ähnlichsten sind, und vereinige (merge) sie. Wiederhole (2,3) das solange, bis es nur noch ein Cluster gibt (4).

(4)

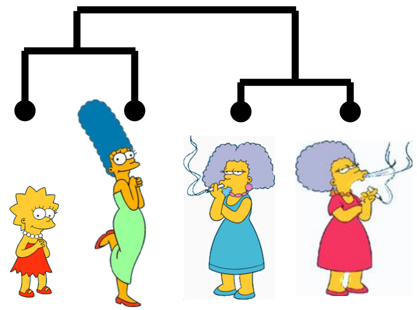


(3)

Betrachte alle möglichen Vereinigungen

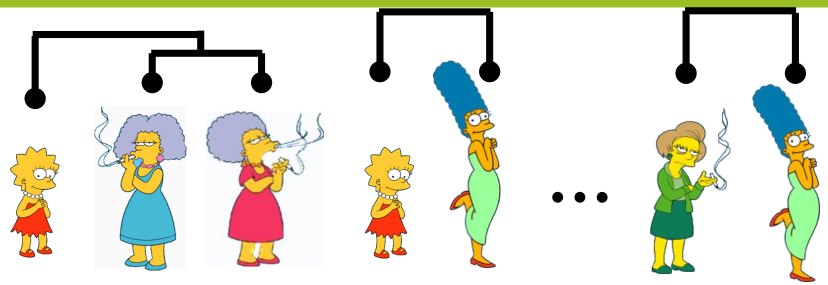


Wähle die beste

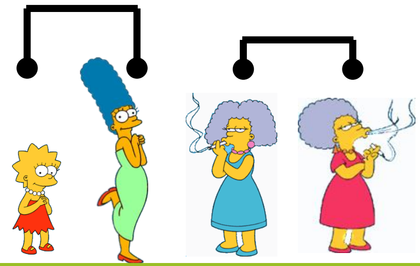


(2)

Betrachte alle möglichen Vereinigungen

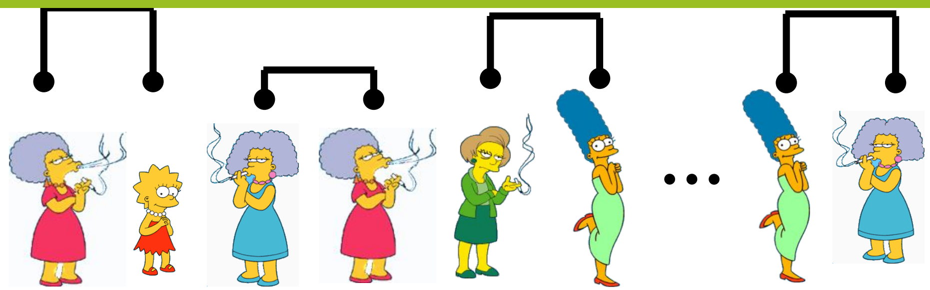


Wähle die beste



(1)

Betrachte alle möglichen Vereinigungen



Wähle die beste

Dendrograms / Hierarchiches Clustering

Pedro

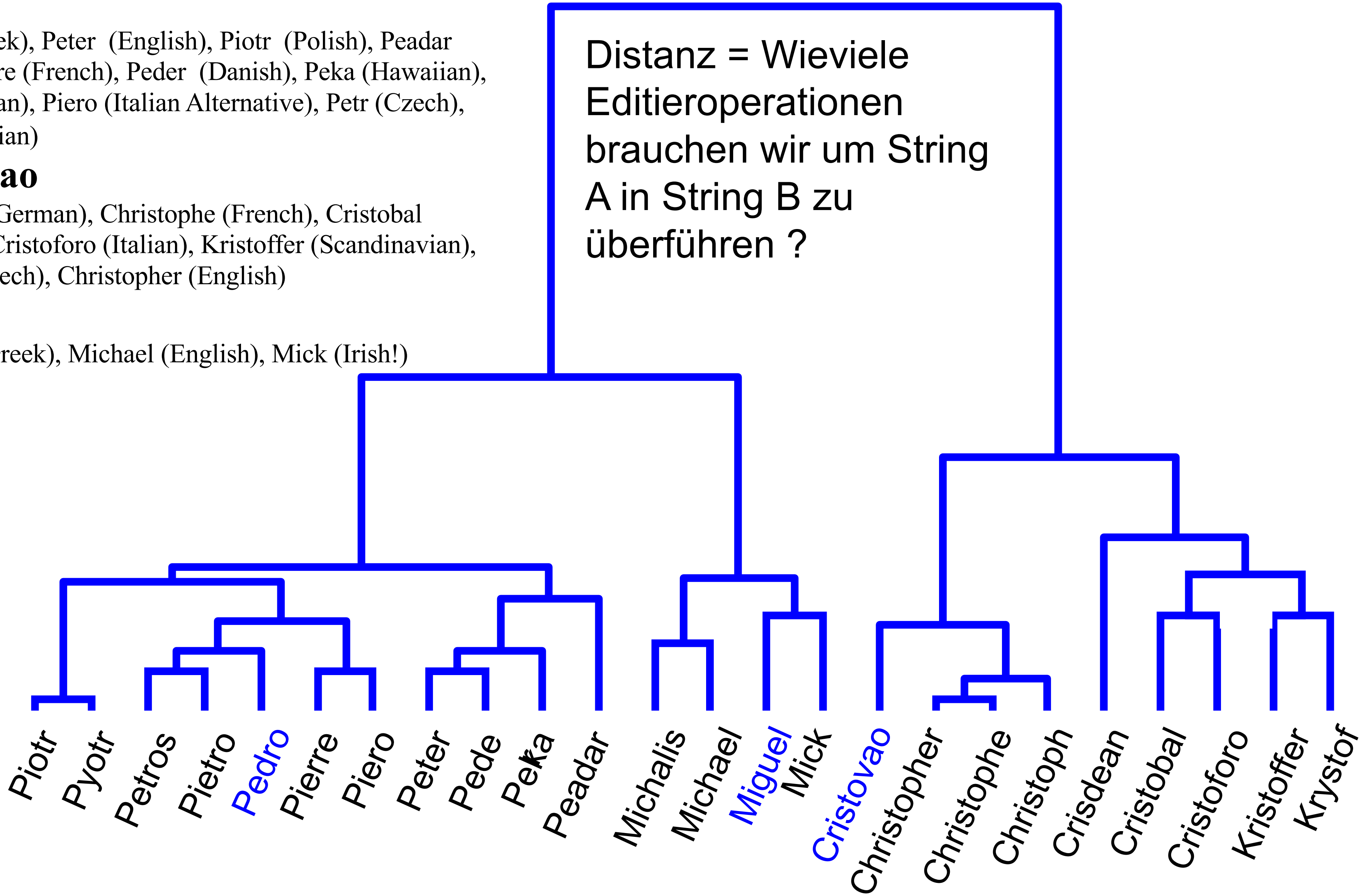
Petros (Greek), Peter (English), Piotr (Polish), Peadar (Irish), Pierre (French), Peder (Danish), Peka (Hawaiian), Pietro (Italian), Piero (Italian Alternative), Petr (Czech), Pyotr (Russian)

Cristovao

Christoph (German), Christophe (French), Cristobal (Spanish), Cristoforo (Italian), Kristoffer (Scandinavian), Krystof (Czech), Christopher (English)

Miguel

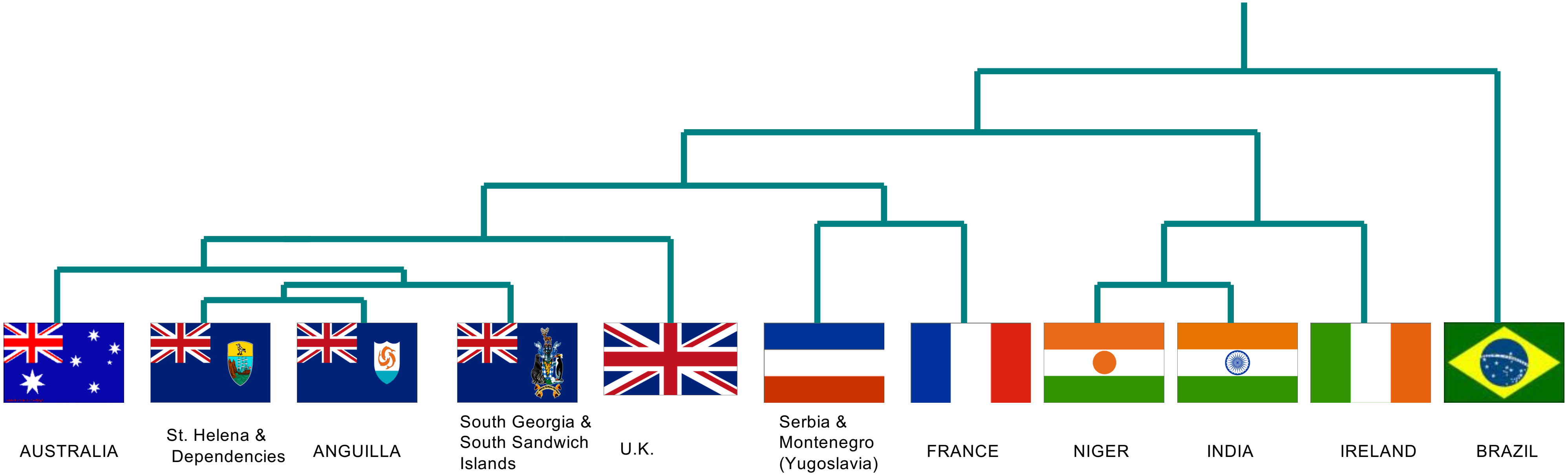
Michalis (Greek), Michael (English), Mick (Irish!)



Hierarchien können Strukturen aufdecken, aber auch vortäuschen

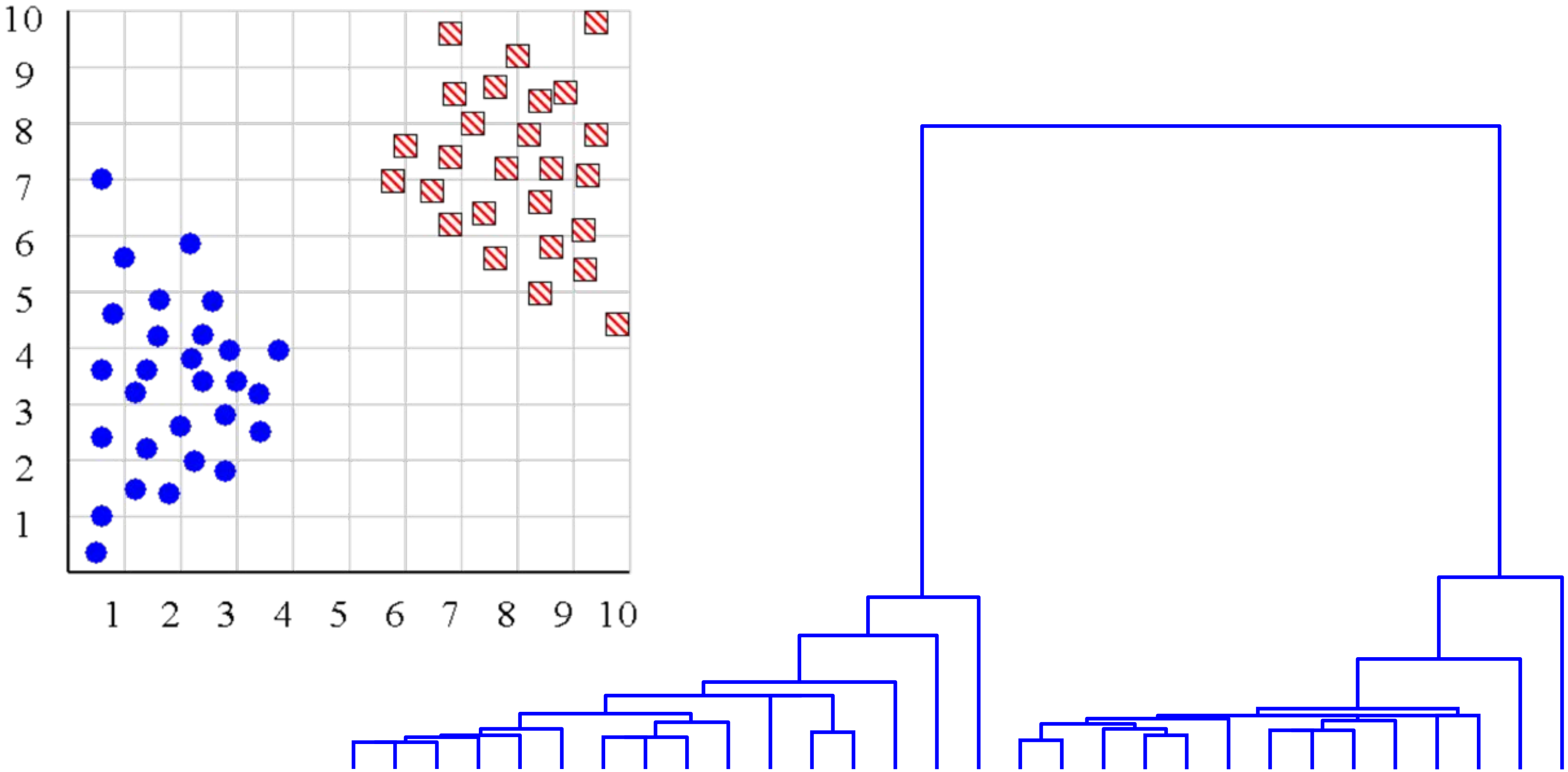
Die dichte Gruppe um Australien macht Sinn. Es sind alles Ländern aus den ehemaligen Britischen Kolonien.

Aber Nigeria und Indien (von Irland wollen wir mal gar nicht erst sprechen) haben nicht viel mit einander am Hut.



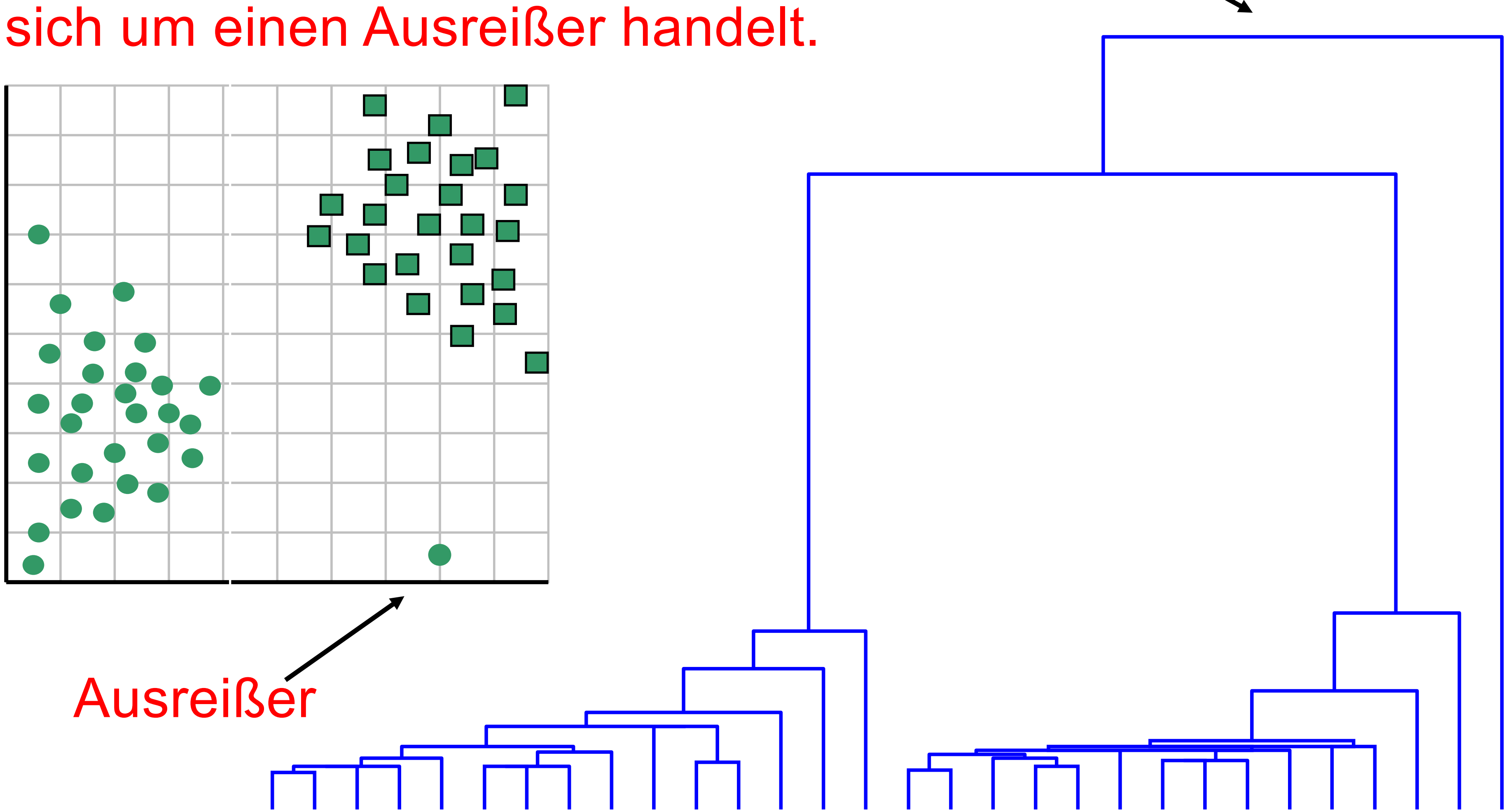
Dendrogramme können uns manchmal auch die richtige Zahl der Cluster sagen

Die zwei stark getrennten Teilbäume legen es nahe, dass es zwei Cluster gibt. Normalerweise ist das aber nicht so klar!



Sie können auch Ausreißer feststellen

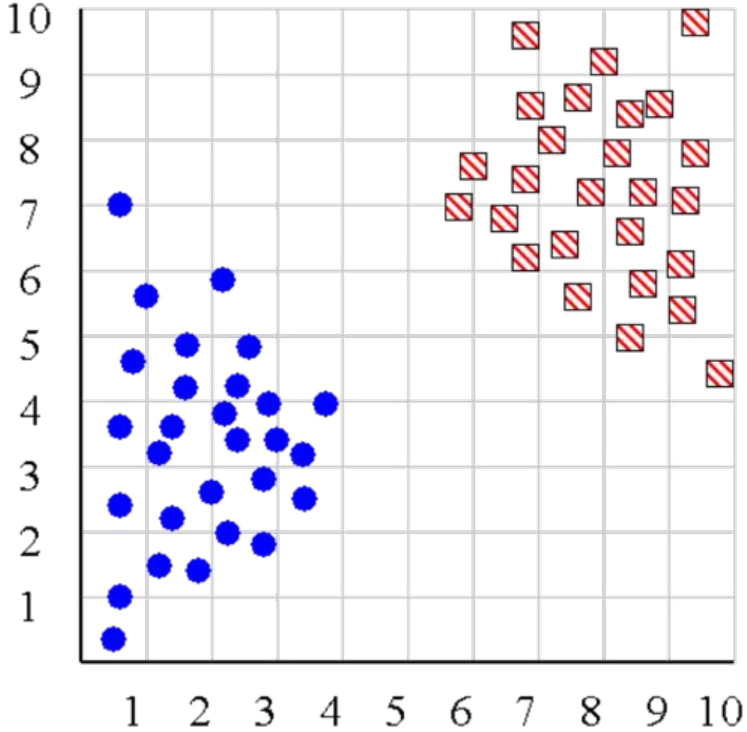
Dieser einzelne Ast legt es nahe, dass es sich um einen Ausreißer handelt.

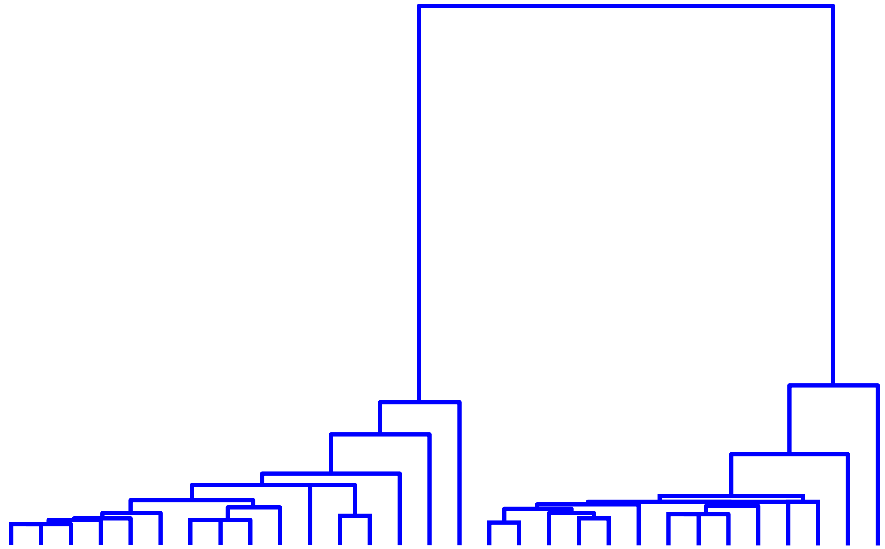


Clustering ist eine Kunst

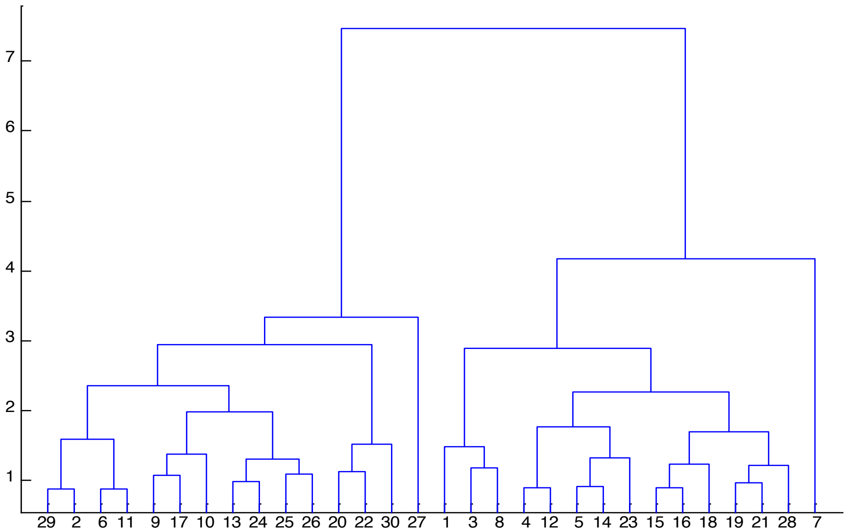
Selbst wenn wir eine gute Distanzfunktion habe, ist es nicht selbstverständlich, wie wir eine Distanz zwischen einem Objekt und einem Cluser bzw, zwischen Clustern definieren.

- **Single linkage (nearest neighbor):** Die Distanz ist die Distanz zwischen den beiden nächsten Nachbarn in den beiden unterschiedlichen Clustern.
- **Complete linkage (furthest neighbor):** Die Distanz ist die Distanz zwischen den beiden entferntesten Objekten
- **Group average linkage:** Durschnitt aller paarweisen Distanzen
- **Wards Linkage**: Man versucht die Varianz zwischen den Clustern zu minimieren.

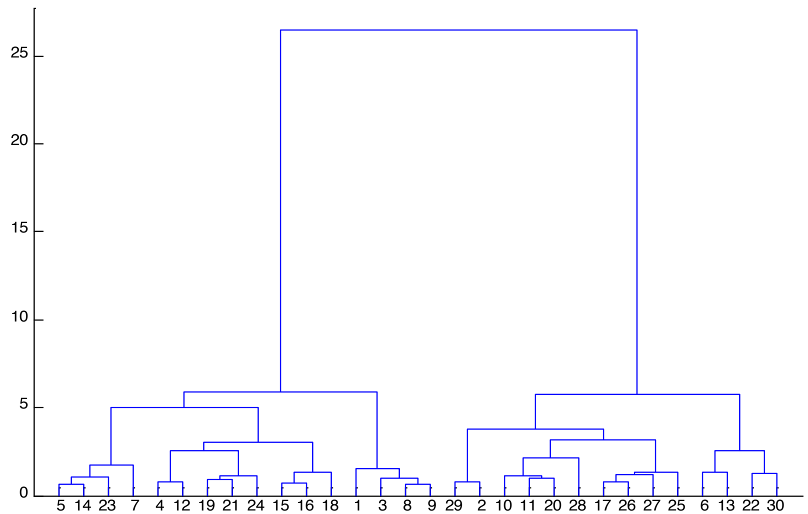




Single linkage



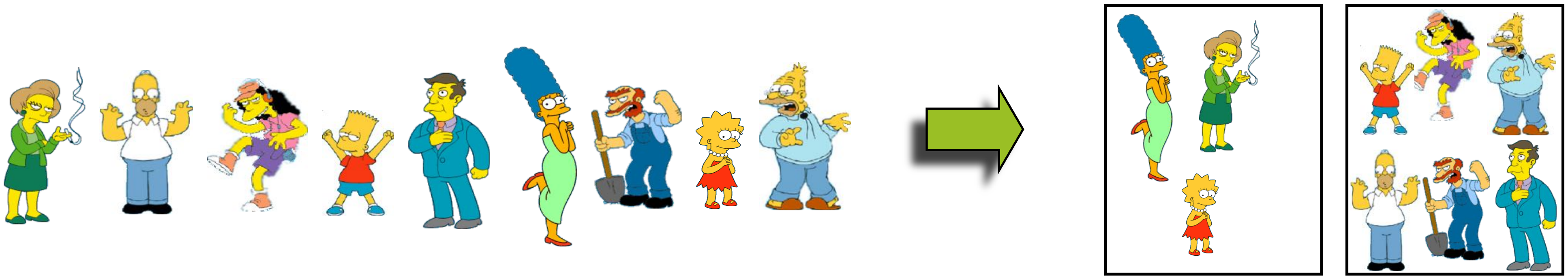
Average linkage



Wards linkage

Clusering mittels Partitionierungen

Keine Hierarchie. Jedes Objekt gehört zu genau einem Cluster. Die Cluster überlappen nicht



K-Means Algorithmus

K = Zahl der Clusters (das gebt ihr an)

Ein “Mittelwert” pro Cluster

(1) Initialisiere die Mitterlwert (z.B. in dem ihr K Datenpunkte zufällig wählt)

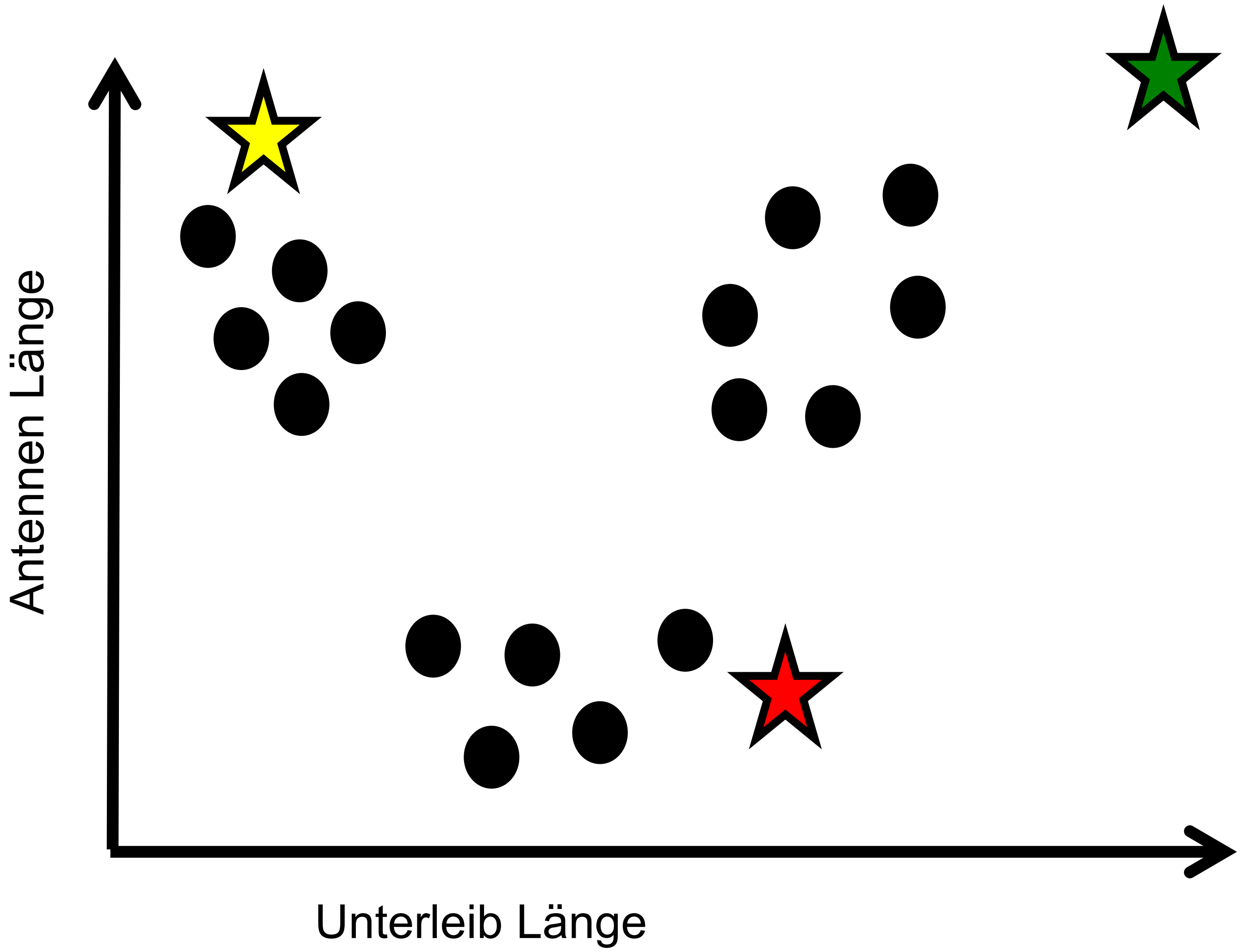
Jetzt, wiederholen wir die folgenden zwei Schritte bis zur Konvergenz:

(2) Ordne jeden Datenpunkt seinem nächsten Mittelwert zu

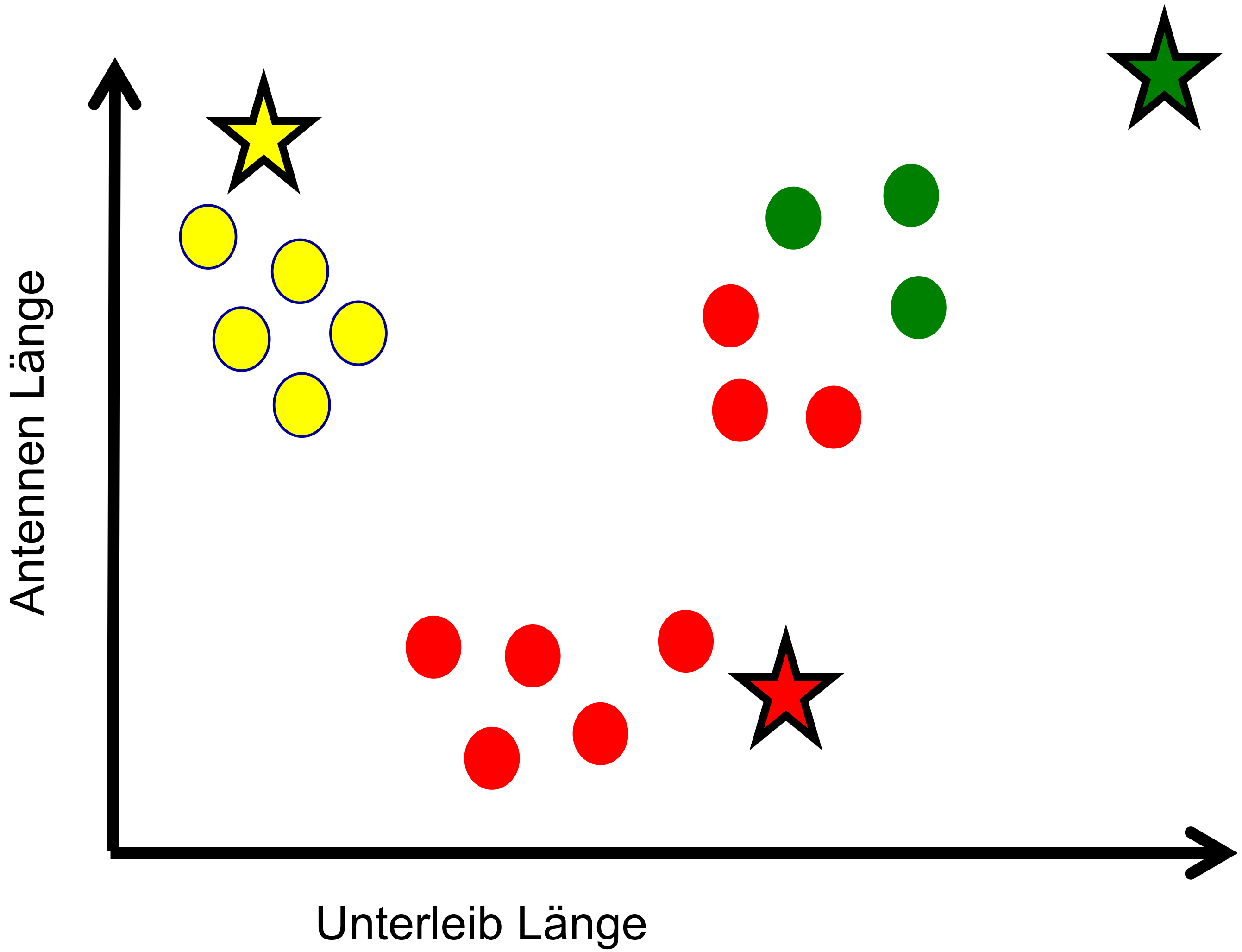
(3) Ersätze jeden Mittelwerte auf den Mittelwert seines Clustersove

“mean” to center of its cluster

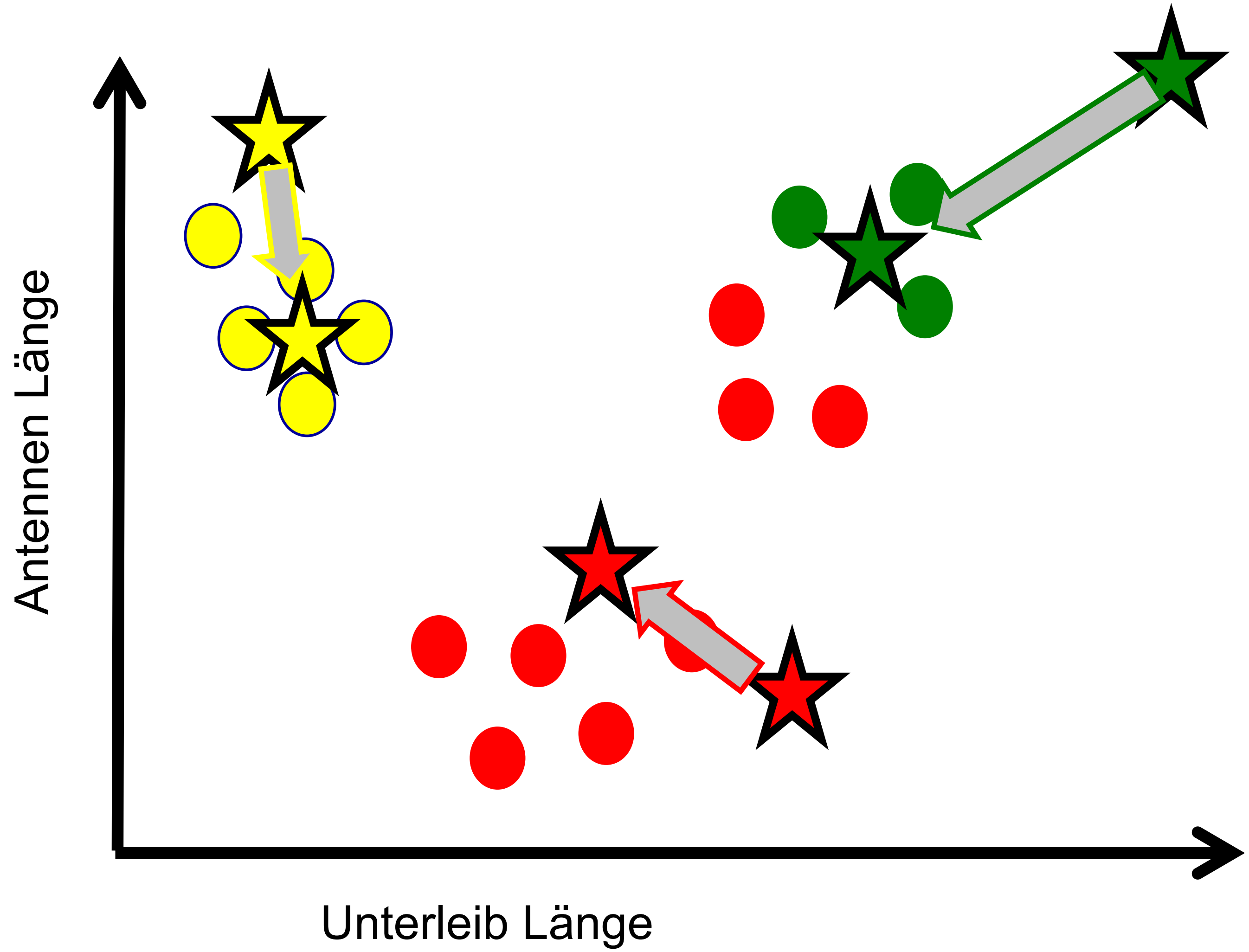
Schritt 1 : Initialisierung der Mittelwerte



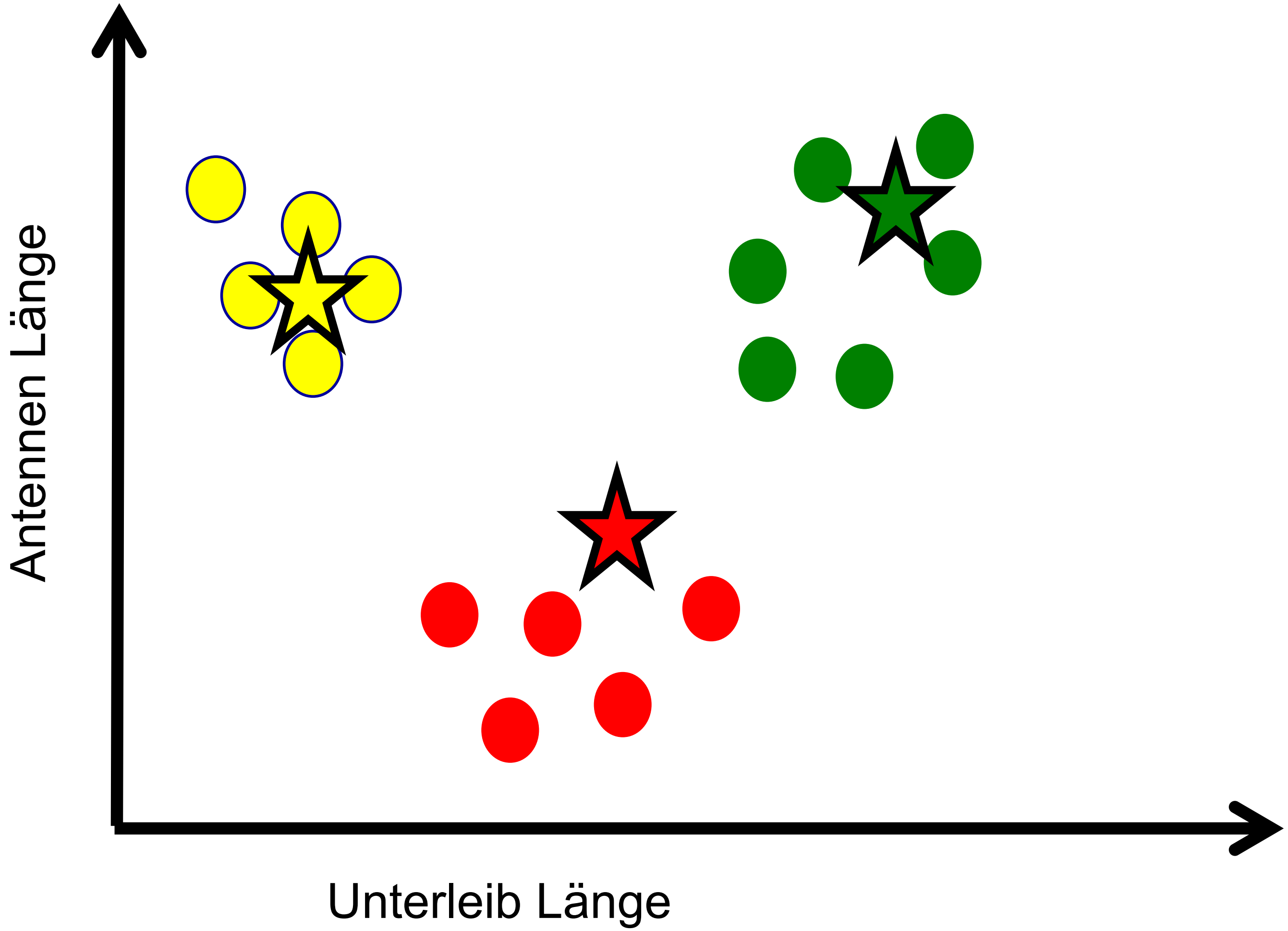
Schritt 2 : Nächster Mittelwert



Schritt 3 : Mittelwert=Mittelwert des Clusters

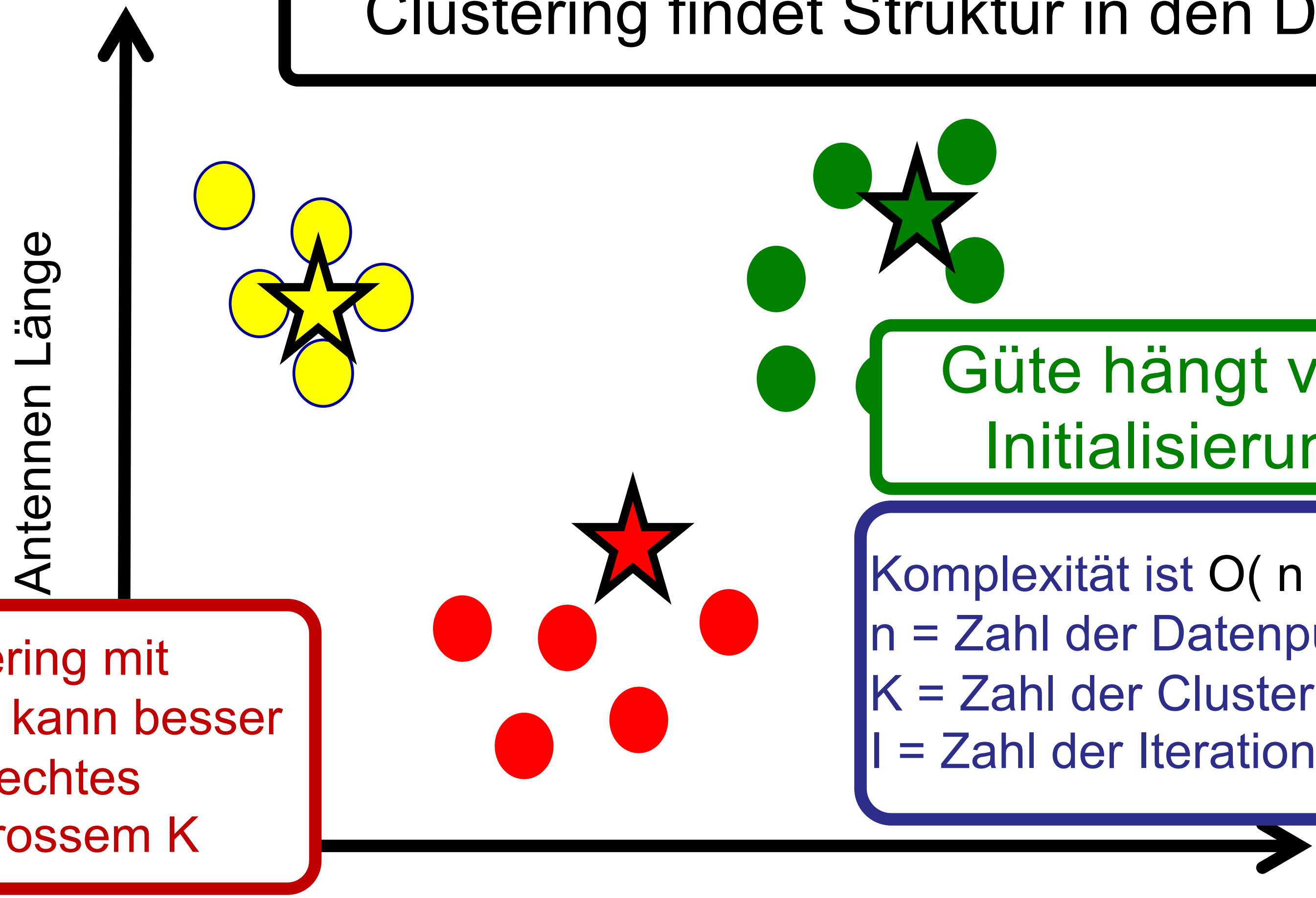


Schritt 2 : Nächster Mittelwert



Schritt 3 : Mittelwert=Mittelwert des Clusters

Clustering findet Struktur in den Daten



Güte hängt von der Initialisierung ab

Komplexität ist O(n * K * I)
n = Zahl der Datenpunkte
K = Zahl der Cluster
I = Zahl der Iterationen

Ein gutes Clustering mit einem kleinen K kann besser sein als ein schlechtes Clustering mit grossem K

kMeans ist konvergiert

Nachteile des kMeans-Verfahrens

kMeans bekommt Probleme, wenn die Cluster unterschiedliche

- Größe und
- Dichten

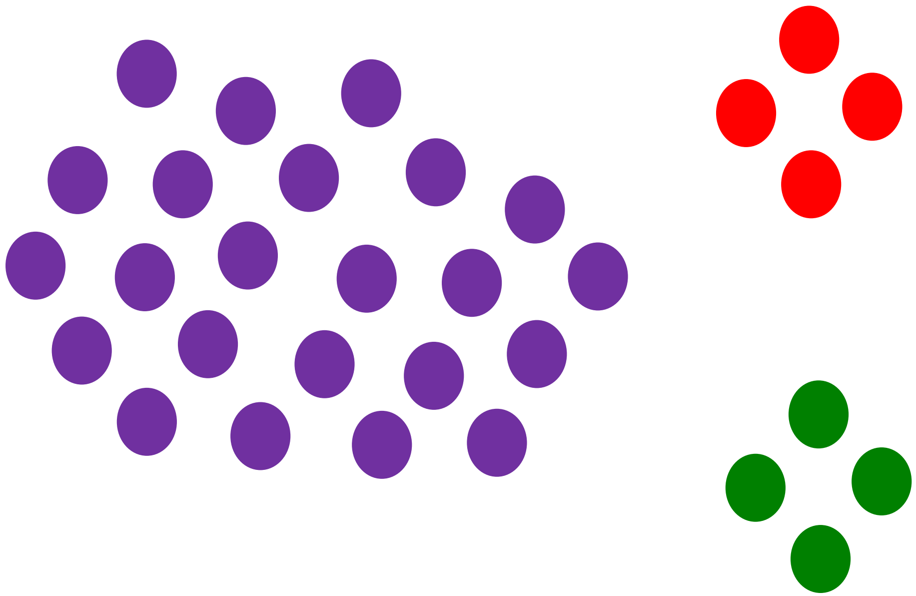
haben und/oder nicht „kugel-förmig“ sind

Weitere Probleme können sich aus

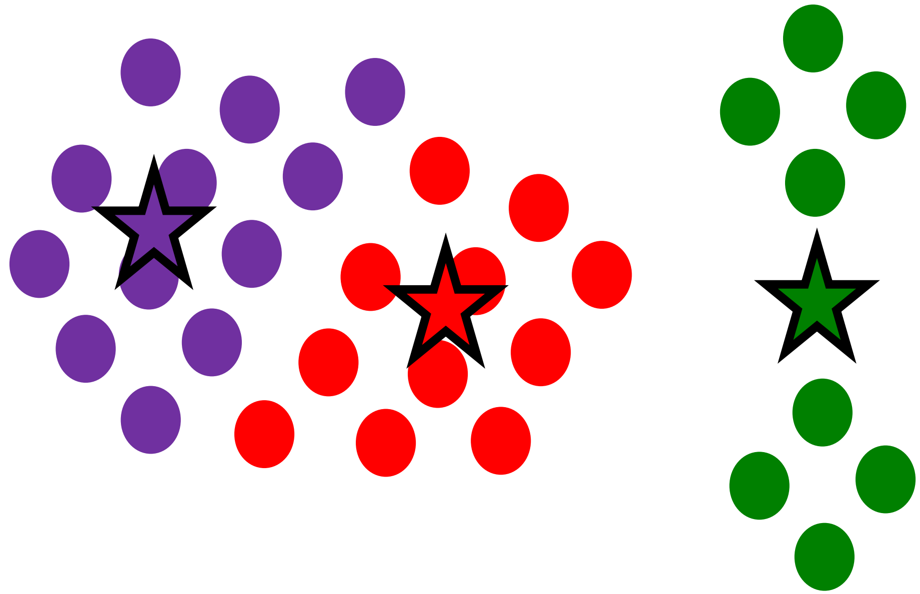
- Ausreißern,
- leere Cluster, und
- hoch-dimensionalen Daten (Fluch der hohen Dimensionen)

ergeben

Nachteile: Unterschiedliche Größen

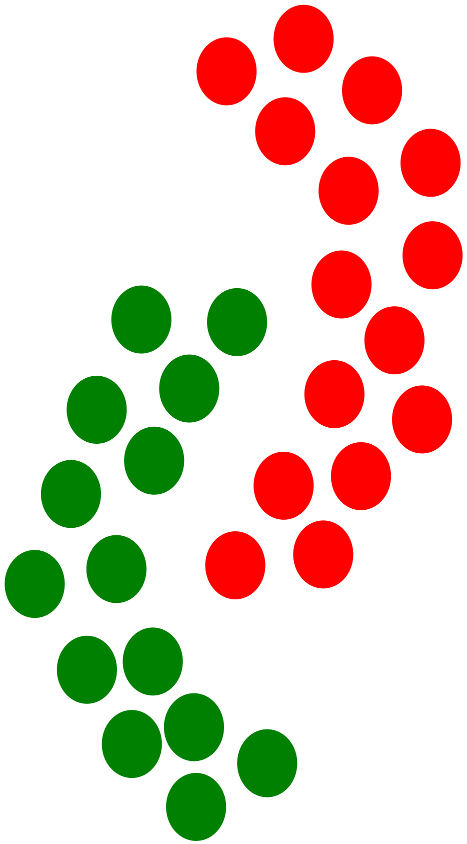


Daten

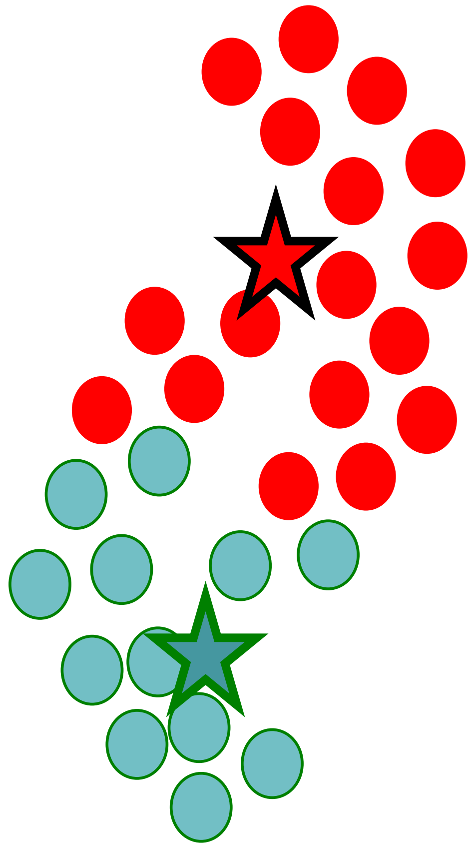


kMeans (3 Cluster)

Nachteile: Nicht-kugelförmig



Daten



kMeans (2 Cluster)

Strategien zum Umgehen der Nachteile

- Mehrfaches Durchführen + Behalten des besten Ergebnisses
- „Over-Clustering“ + Nachverarbeitung
- Probabilistische oder kernelized Varianten
- Andere Verfahren zur Clusteranalyse wie z.B. Spectral clustering, Random Projections, Clusteranalyse mit Randbedingungen, DBSCAN, bi-clustering, LDA, …
- …

Was wissen wir jetzt?

Die Lernaufgabe Clustering kennen Sie

Wir haben zwei Klassen von Methoden gesehen:
- hierarchisches Clustering,
- k-Means.

Die Wahl des Abstandmaßes ist entscheidend für das Clustering.