

Grundlagen des Maschinellen Lernens

Clustering

Marc Hesenius



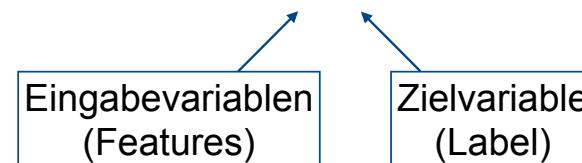
Photo by Clem Onojeghuo from Pexels

1. Unsupervised Learning und Clustering
2. K-Means-Algorithmus
3. Hierarchisches Clustering

Unsupervised Learning und Clustering

Supervised Learning

- (Trainings-)Daten liegen als Paar (X, Y) vor



- Die Zielvariable Y gibt das Lernen vor:

- Wir sind an der Beziehung zwischen Y und X interessiert
- Y ist das objektive Maß, ob die gelernte Beziehung gut oder schlecht ist, sprich:

Y überwacht das Lernen

Unsupervised Learning

- Daten X liegen ohne Zielvariable (Label) vor
- Wir sind an der zugrunde liegenden Struktur der Daten interessiert:
 - Allgemein: Wie sind die Daten im Feature-Raum verteilt?
 - Konkretes Beispiel: Wo befinden sich viele Datenpunkte, wo wenige?
- Es gibt kein objektives Maß für die Qualität der erlernten Struktur, sprich:

Das Lernen ist unüberwacht

Betrugserkennung

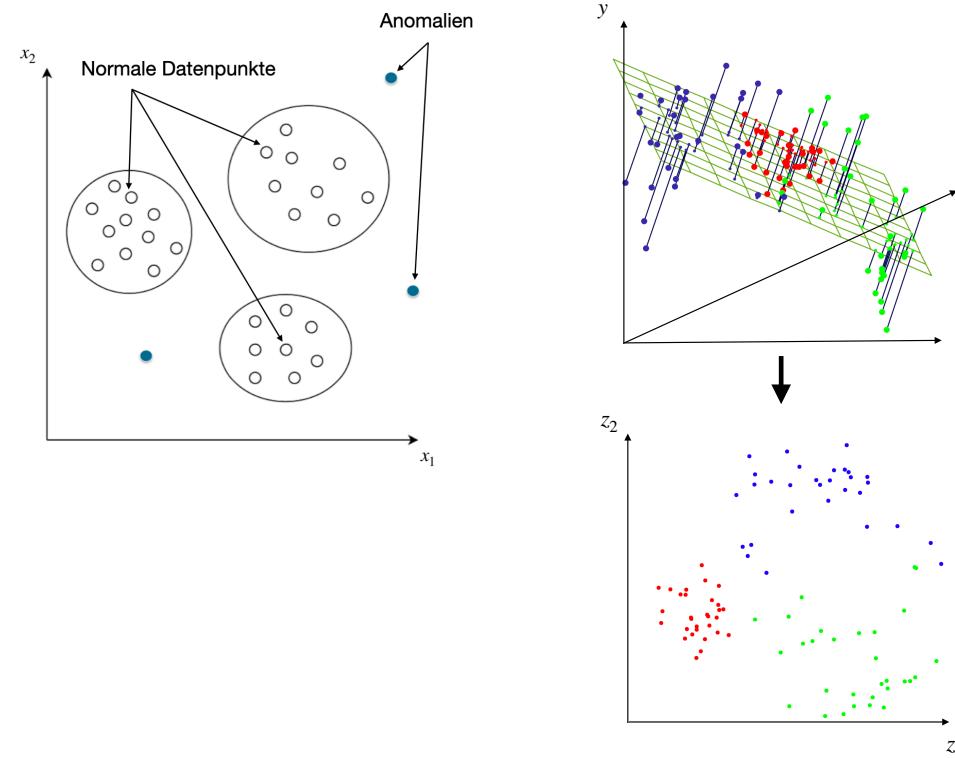
Es sollen ungewöhnliche Aktivitäten von gewöhnlichen unterschieden werden, damit mögliche Betrugsfälle entdeckt werden können. Oft mit *Outlier Detection* (Anomaliedetektion) umgesetzt.

Dimensionsreduktion

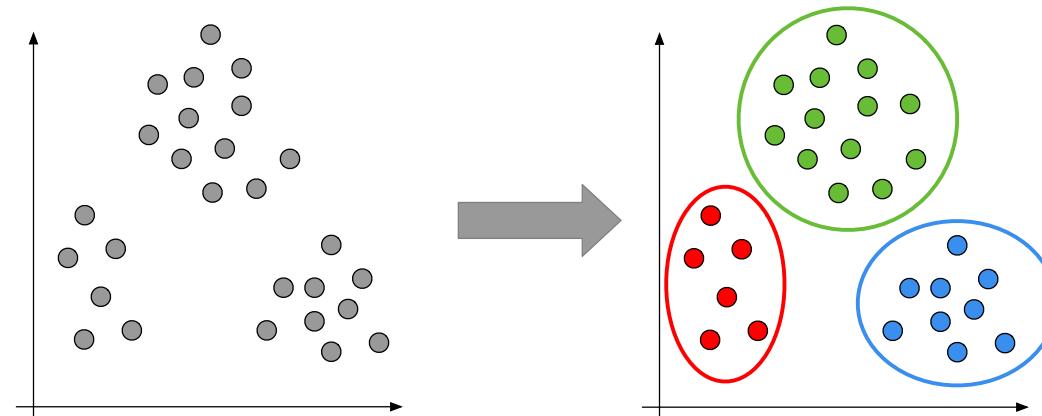
Es soll die Dimension hochdimensionaler Daten reduziert werden, damit sie visualisiert werden können.

Kundensegmentierung

Es sollen Gruppen einander ähnlicher Kunden erstellt werden, damit diese passendere Werbung erhalten können. Oder Kunden sollen anhand ihrer Kündigungswahrscheinlichkeit gruppiert werden.



- **Clusteranalysen** sind Verfahren, um Objekte in Gruppen zu teilen.
- Objekte derselben Gruppe sollen ähnlich zueinander sein. Objekte unterschiedlicher Gruppen sollen unähnlich zueinander sein.
- Die Gruppen werden **Cluster** genannt, die Gruppenzuordnung **Clustering**.
- Die Zuordnung zu den Gruppen geschieht ohne Labels: Clusteranalysen sind unüberwachte Verfahren.

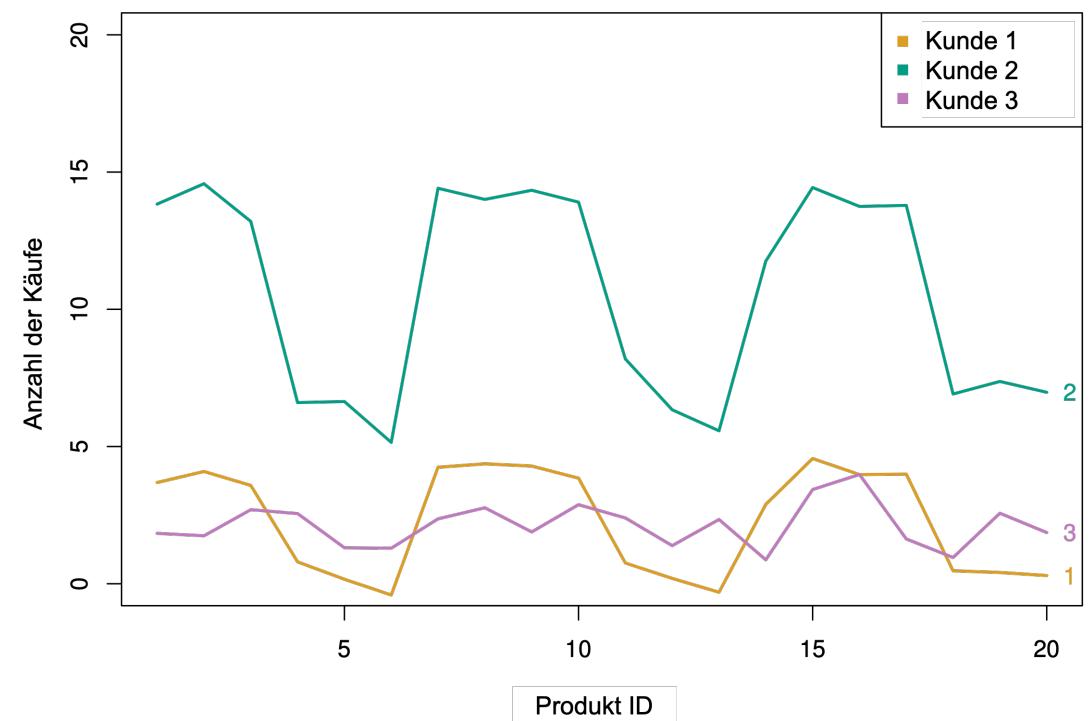


Herausforderungen

- Da Clusteranalysen unüberwacht sind, gibt es kein direktes Maß zur **Evaluierung** von Clusterings
- In jeder Clusteranalyse muss festgelegt werden, was **ähnlich** bzw. **unähnlich** bedeutet
 - Die Wahl des Ähnlichkeitsmaßes wirkt sich auf die entstehenden Clusterings aus
 - Welches Maß sinnvoll ist, hängt vom Anwendungsfall ab
 - Oft verwendet: Euklidische Distanz
 - Alternativen: Korrelationskoeffizient, Cosinusdistanz u. v. m.
- Beispiel: Online-Händler
 - Kunden sollen anhand ihrer Kaufhistorie in Cluster geteilt werden, um Kaufempfehlungen zu erstellen
 - Wann sind Kunden ähnlich? Wenn sie
 - in etwa gleich viel kaufen, also ähnlich viel Umsatz generieren?
 - in etwa die gleichen Produkte kaufen, ungeachtet der Menge?

Unähnlichkeitsmaße — Beispiel

- Verwendung von **euklidischer Distanz**:
 - Kunden 1 und 3 sind ähnlich zueinander, da sie ähnlich viel kaufen
 - Kunde 2 unterscheidet sich stark von den beiden anderen
- Verwendung des **Korrelationskoeffizienten**:
 - Kunden 1 und 2 sind ähnlich zueinander, da sie ähnliche Produkte kaufen, nur in anderen Mengen
 - Kunde 3 unterscheidet sich von den anderen beiden

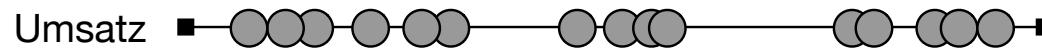


K-Means-Algorithmus

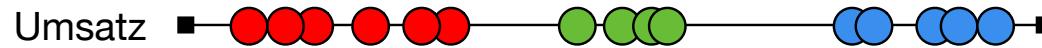
1. Beispiel
2. Kombinatorische Problemstellung
3. Der K-Means-Algorithmus
4. Praktische Überlegungen
5. Schwächen des K-Means-Algorithmus

Beispiel

- Wir betrachten eine Menge von Kunden, die in unserem Online-Shop einkaufen.



- Können wir diese Kunden in Kategorien einordnen, um ihnen z. B. gezielt Werbung zukommen zu lassen oder ihnen Incentives für weitere Käufe anzubieten?



- Gelegenheitskunde
- Basic-Kunde
- Premium-Kunde

- Wie können wir diese Einteilung mithilfe eines Algorithmus' automatisch bestimmen lassen?
- Hinweis: Das Beispiel ist stark vereinfacht — in der Realität würden wir Kunden nicht nur nach einer Dimension, sondern nach mehreren kategorisieren, wodurch die Ergebnisse nicht mehr so offensichtlich sind!

- Es liegt eine Datenmenge mit N Punkten vor: $\{x_1, \dots, x_N\}$
- Wir wollen die Datenmenge in K Cluster teilen, $1 < K < N$
- Wir suchen eine Funktion C , die die Zuordnung vornimmt:

$$\begin{aligned} C : \{1, \dots, N\} &\rightarrow \{1, \dots, K\} \\ C(i) &= k \end{aligned}$$

- Die Funktion C weist jedem Index (bzw. Datenpunkt) genau ein Cluster zu
- Ein Clustering ist dann gut, wenn Datenpunkt im selben Cluster ähnlicher zueinander sind als zu den Datenpunkten in anderen Clustern

- Ob zwei Punkte ähnlich sind, wird durch ein Distanz- bzw. Unähnlichkeitsmaß d bestimmt.
- Um die Güte eines Clusterings zu bewerten, bestimmen wir die Summe der Distanzen zwischen den Punkten desselben Clusters. Diese Größe nennen wir *Intra-Cluster-Streuung* W :

$$W(C) = \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(i')=k} d(x_i, x_{i'}), \text{ dabei ist}$$

- C unser aktuelles Clustering
- K die Anzahl aller Cluster
- k das aktuelle Cluster
- d ein Distanzmaß
- i der Index des betrachteten Datenpunkts
- i' der Index des verglichenen Datenpunkts
- **Ziel** ist es, ein Clustering C zu finden, das diese Streuung minimiert. Wie machen wir das?

Schwierigkeit des Problems

- Eine Möglichkeit wäre, alle möglichen Funktionen C auszuprobieren, d. h. alle möglichen Aufteilungen der N Datenpunkte auf K Cluster.
- Die Anzahl der Möglichkeiten beträgt $S(N, K) = \frac{1}{K!} \sum_{k=1}^K (-1)^{K-k} \binom{K}{k} k^N$
- Das bedeutet z. B. für $S(10,4) = 34.105$ und für $S(19,4) \approx 10^{10}$
- Für größere N und K explodiert die Anzahl der Möglichkeiten, daher ist reines Ausprobieren aller möglichen Lösungen nicht praktikabel.
- Das Problem, ein optimales Clustering zu finden, ist NP-schwer. Wir brauchen also eine Annäherung, mit der wir ein möglichst gutes Clustering ermitteln können.

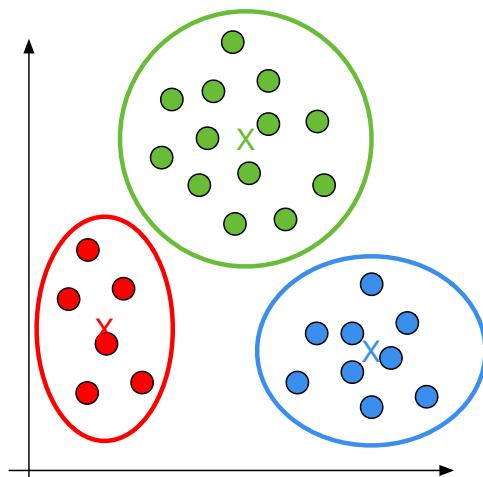
- Der K-Means-Algorithmus ist ein iteratives Clustering-Verfahren, das ein lokales Optimum findet
- Der Algorithmus ist für folgende Szenarien geeignet:
 - Features sind intervallskaliert
 - Distanzmaß ist die quadrierte euklidische Metrik $d(x_i, x_{i'}) = \|x_i - x_{i'}\|^2$
- Die Intra-Cluster-Streuung für dieses Szenario ist

$$W(C) = \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(i')=k} \|x_i - x_{i'}\|^2$$

- Die Anzahl der angestrebten Cluster (das K in *K-Means*) ist ein Hyperparameter und muss vorab angegeben werden. Da wir vorab nicht wissen können, wie viele Cluster *gut* sind, haben wir hier ein Problem, das wir uns später im Detail anschauen.

K-Means: Centroids

- Um die angestrebten k Cluster aufzubauen, versuchen wir, mithilfe des Algorithmus potentielle Mittelpunkte zu finden, um die die Datenpunkte herum angeordnet sind. Diese nenne wir *Centroids*.
- Die Centroids bilden den Durchschnitt der Datenpunkte im jeweiligen Cluster (das *Means* in K-Means).

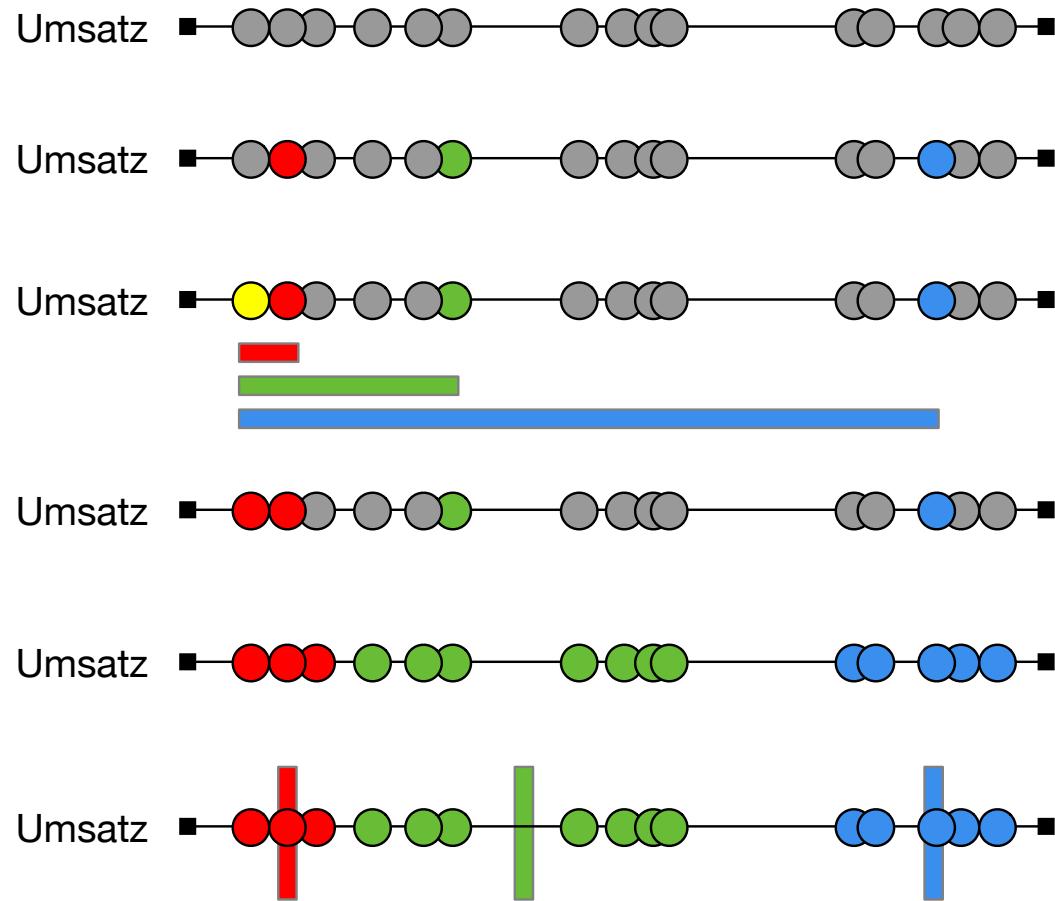


$$\begin{aligned} W(C) &= \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(i')=k} \|x_i - x_{i'}\|^2 \\ &= \sum_{k=1}^K |C_k| \sum_{C(i)=k} \|x_i - \bar{x}_k\|^2 \end{aligned}$$

- \bar{x}_k ist der mittlere Vektor der Punkte im k -ten Cluster
- $C_k = C^{-1}(\{k\})$ ist die Menge der Punkte im k -ten Cluster
- Die Intra-Cluster-Streuung zu minimieren bedeutet äquivalent, ein Clustering zu finden, sodass in den Clustern der durchschnittliche quadrierte Abstand zwischen den Punkten und dem Centroid des Clusters minimiert wird

K-Means

- 1. Wir definieren in diesem Beispiel $k = 3$, d. h. wir wollen unsere Daten in 3 Cluster aufteilen
- 2. Definiere zufällig k Datenpunkte als initiales *Centroid* (Zentrum) der Cluster
- 3. Bestimme die Distanz vom ersten Datenpunkt zu den Centroids der Cluster
- 4. Weise den ersten Datenpunkt dem nächsten Cluster zu
- 5. Wiederhole dieses Prozedere für alle Datenpunkte
- 6. Bestimme den Durchschnitt der Cluster als neue Zentren und wiederhole die Schritte 3-5 mit den neuen Zentren, bis sich an den Clustern nichts mehr ändert



- Unser neues Ergebnis unterscheidet sich jetzt von unserem initialen Beispiel:

initial: Umsatz ■—●●●●●●●●●●●●●●●●●●●●●●●●■

jetzt: Umsatz ■—●●●●●●●●●●●●●●●●●●●●●●●●■

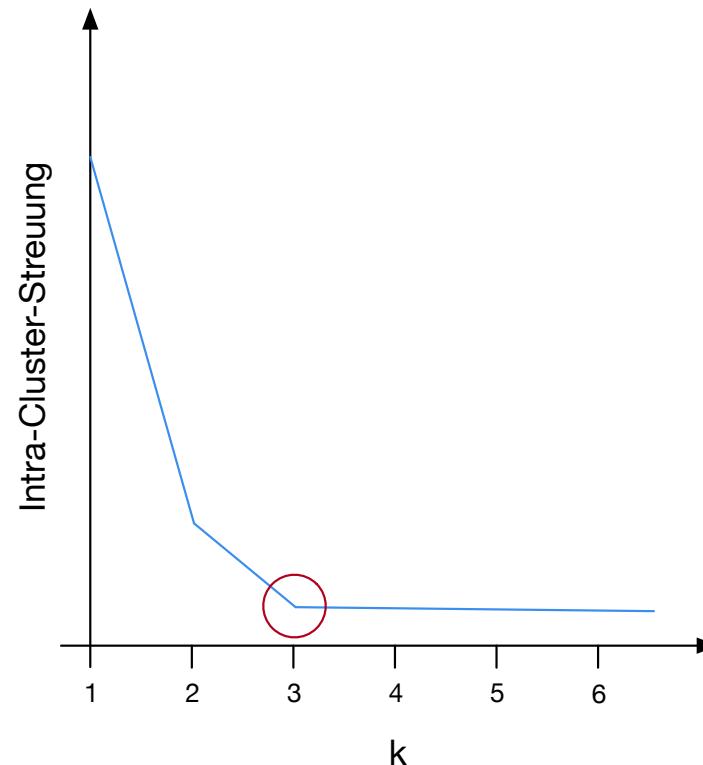
- Da wir das optimale Clustering nicht in akzeptabler Zeit bestimmen können (das Problem ist ja, wie vorher gezeigt, NP-schwer), müssen wir uns mit der bestmöglichen Lösung zufriedengeben.
- Wir haben aber die initialen Centroids der Cluster zufällig ausgewählt — wie finden wir die bestmögliche Lösung?
 - Wir berechnen mehrere Clusterings mit unterschiedlichen Initialwerten und berechnen die Intra-Cluster-Streuung.
 - Das Clustering mit der niedrigsten Intra-Cluster-Streuung ist unser bestmögliches Ergebnis.

Optimierte Initialisierung

- Die initiale Auswahl von Centroids hat offensichtlich großen Einfluss auf das Ergebnis
- Man kann die Initialisierung mit verschiedenen Verfahren optimieren und so die Laufzeit bis zu einem guten Ergebnis erheblich verkürzen. Ein weit verbreitetes Beispiel ist die Auswahl mit *K-Means++*:
 1. Wähle einen Datenpunkt zufällig als Centroid des ersten Clusters aus
 2. Berechne zu jedem noch nicht gewählten Datenpunkt die quadrierte Distanz zum Centroid
 3. Wähle als nächsten Centroid zufällig einen noch nicht gewählten Datenpunkt, wobei die Wahrscheinlichkeit proportional zur in Schritt 2 berechneten quadrierten Distanz sein soll (d.h., je weiter der Datenpunkt von bestehenden Centroids entfernt ist, desto größer ist die Wahrscheinlichkeit für die Auswahl als weiteres Centroid)
 4. Wiederhole die Schritte 2-3, bis K Center gewählt sind
 5. Führe K-Means aus

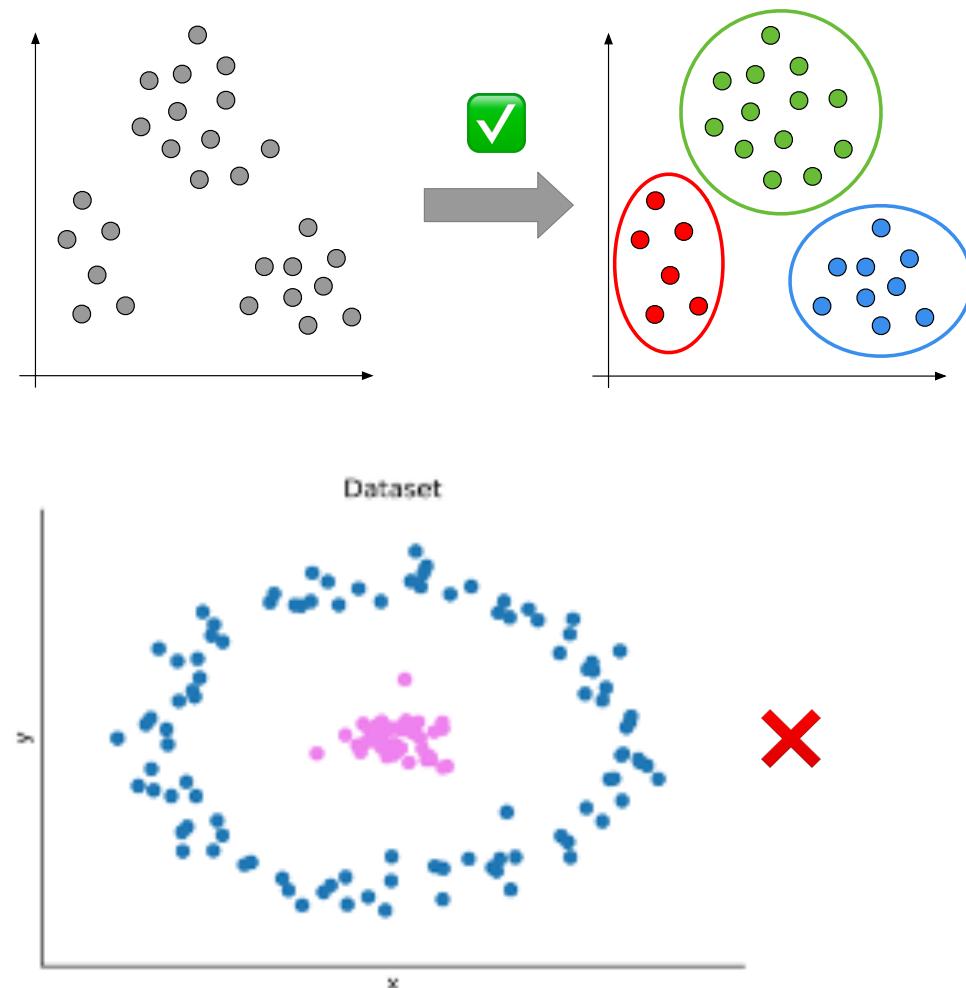
Wahl von k : Elbow Method

- Ein guter Wert von K (also die Anzahl der angestrebten Cluster) kann mit der *Elbow Method* ermittelt werden
- Dazu wird für verschiedene Werte von k die Intra-Cluster-Streuung berechnet und grafisch dargestellt. Das optimale k ist am Knick des *Ellbogens* in der Visualisierung, hier $k = 3$.
- Hinweis: Dasselbe Prinzip haben Sie in der Einheit zur Dimensionsreduktion schon kennengelernt, nur mit der *Proportion of Variance*.



Schwächen von K-Means

- Die Wahl von K muss vor der Durchführung des Algorithmus getroffen werden. Es können unterschiedlich gute Clusterings herauskommen.
- Ausreißer verzerren das Clustering, da die euklidische Metrik als Distanzmaß verwendet wird.
- Cluster, die
 - verschieden groß sind
 - unterschiedlich dicht sind
 - nicht kreisförmig sindsind für K-Means nicht gut geeignet.

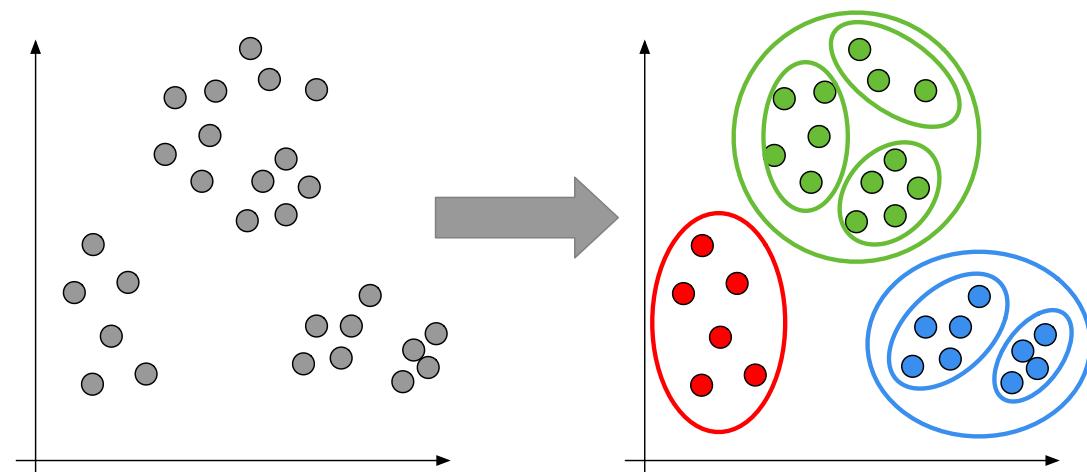


Hierarchisches Clustering

1. Konzept
2. Agglomeratives Clustering
3. Schwächen des hierarchischen Clustering

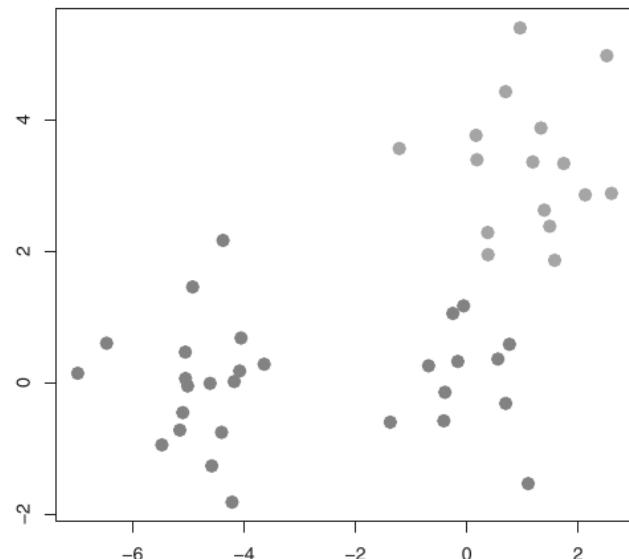
Hierarchisches Clustering

- Cluster sind ineinander verschachtelt, d. h. Cluster bestehen aus Unterclustern
- Größtes Cluster ist die gesamte Datenmenge, kleinstes Cluster ein einzelner Datenpunkt
- Hierarchische Clusterings lassen sich als Baum modellieren (Dendrogramm)
- Ein konkretes Clustering wird durch einen Schnitt im Dendrogramm erzeugt (jeder entstehende Unterbaum stellt ein Cluster dar)
- Wir unterteilen Verfahren in
 - divisiv: es wird top-down geteilt
 - agglomerative: es wird bottom-up zusammengeführt



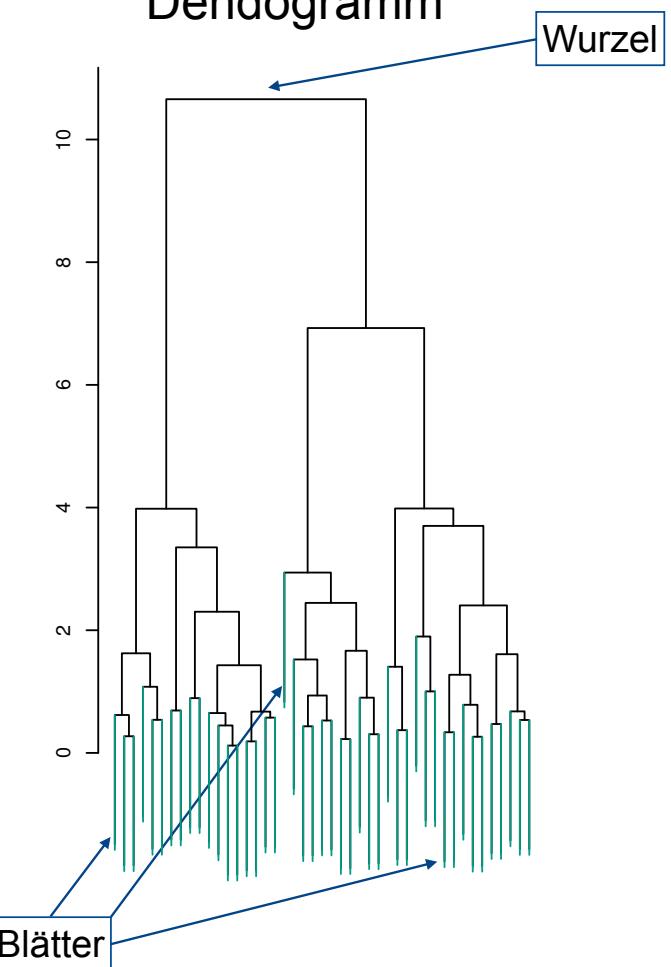
Dendogramme

Daten



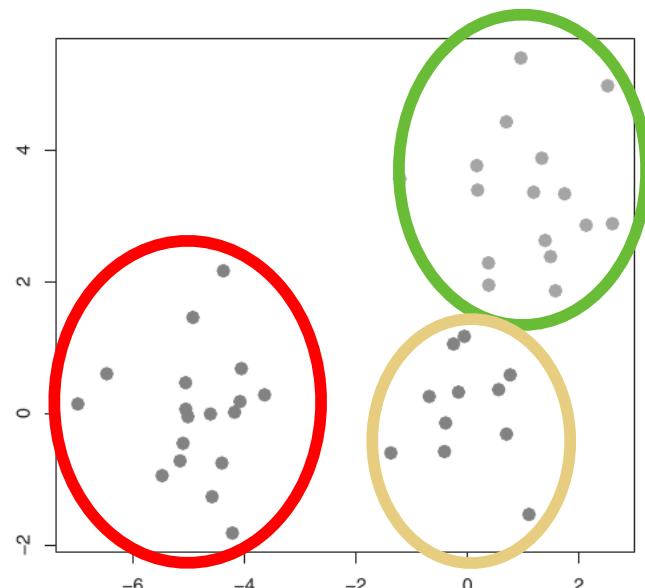
Die Punkte entsprechen den Blättern im Dendrogramm

Dendrogramm



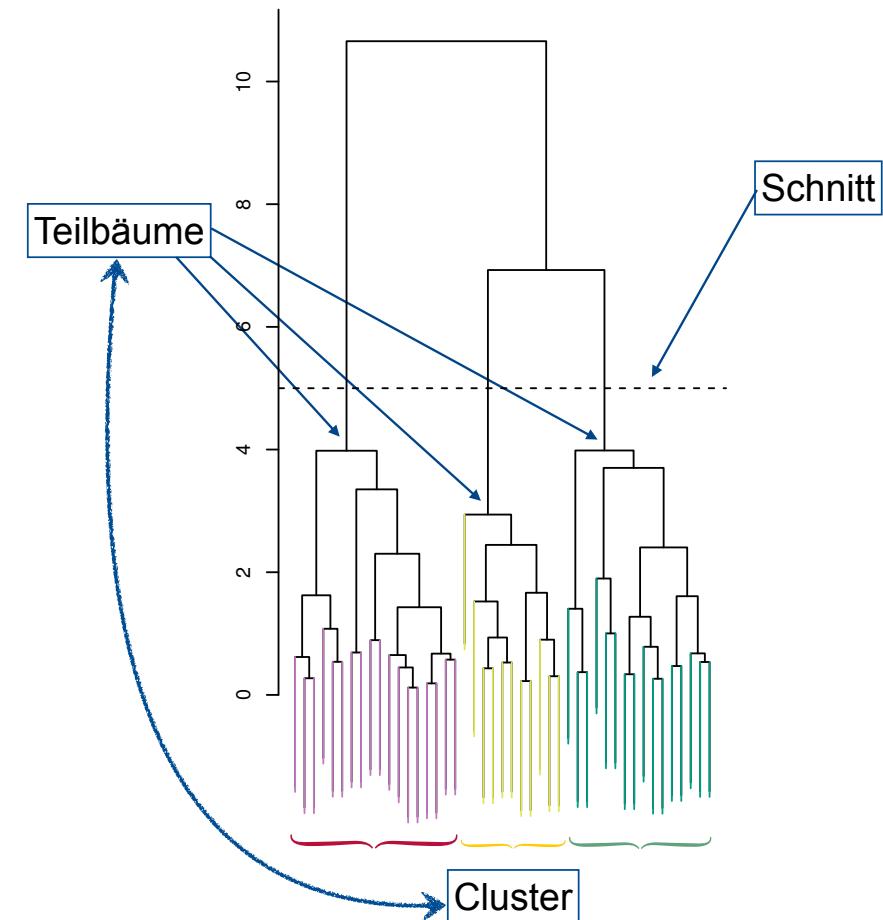
Clustering durch Schnitte

Daten



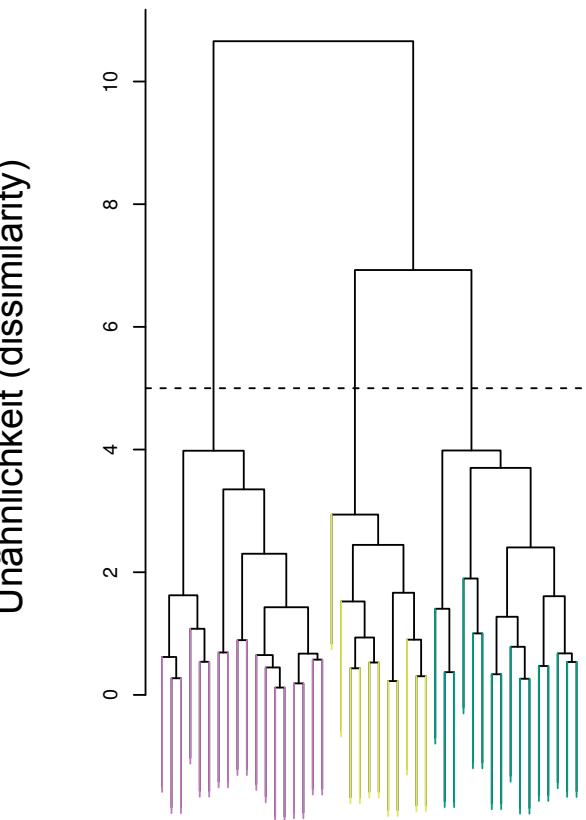
Durch den Schnitt sind drei Cluster entstanden

Dendogramm



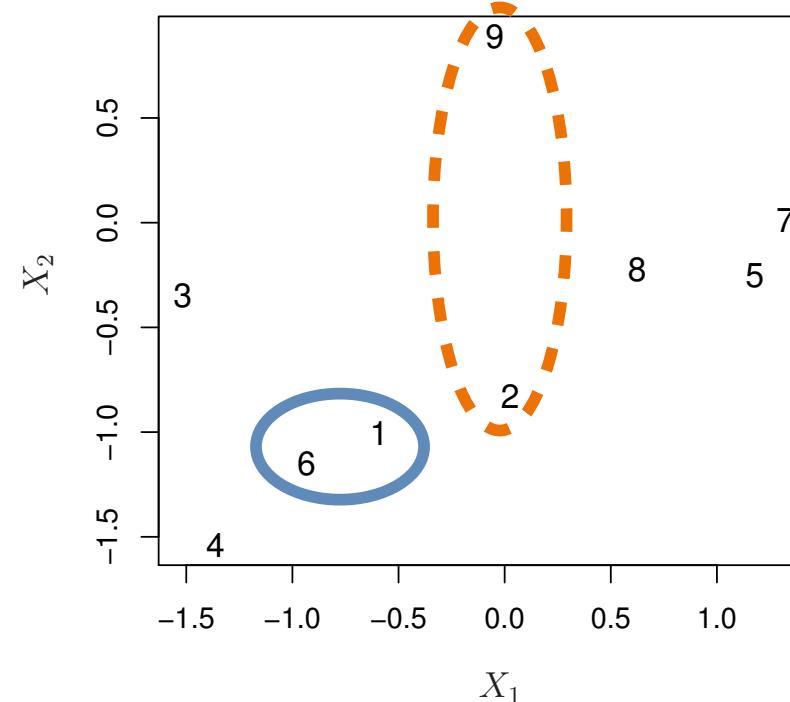
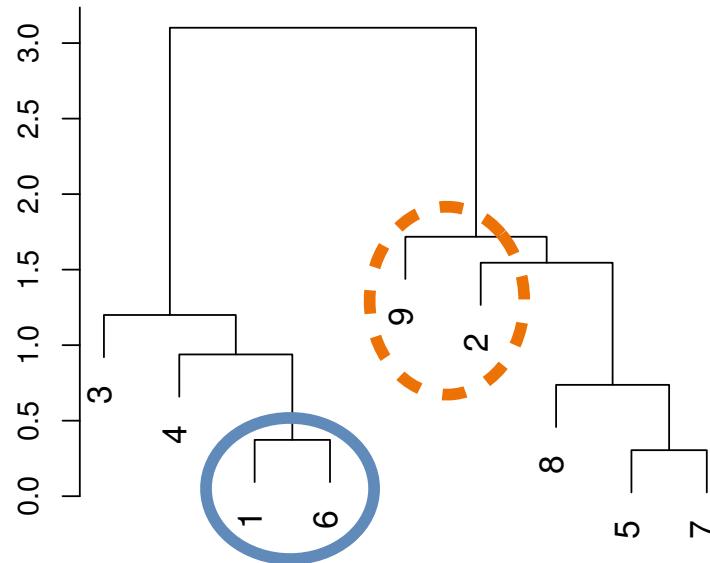
Interpretation

- Jeder Datenpunkt entspricht einem Blatt
- Blätter werden nach und nach zu Zweigen verbunden, je höher man in dem Baum geht
- Je früher Blätter bzw. Zweige verbunden werden (d.h. je weiter unten auf der y-Achse), desto ähnlicher sind sich die Blätter bzw. Unterbäume
- Für je zwei Blätter zeigt die Höhe ihrer Verbindung (auf der y-Achse) an, wie unähnlich sie sich sind



Beispiel

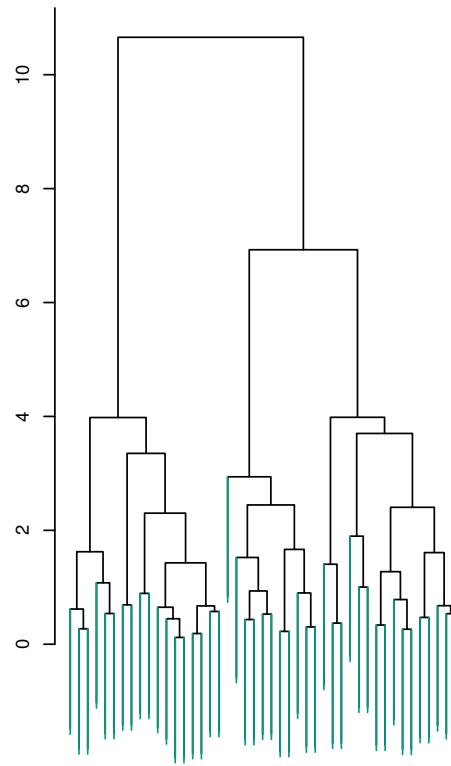
Wenn zwei Blätter direkt nebeneinander liegen, heißt das nicht unbedingt, dass sie sich ähnlich sind!



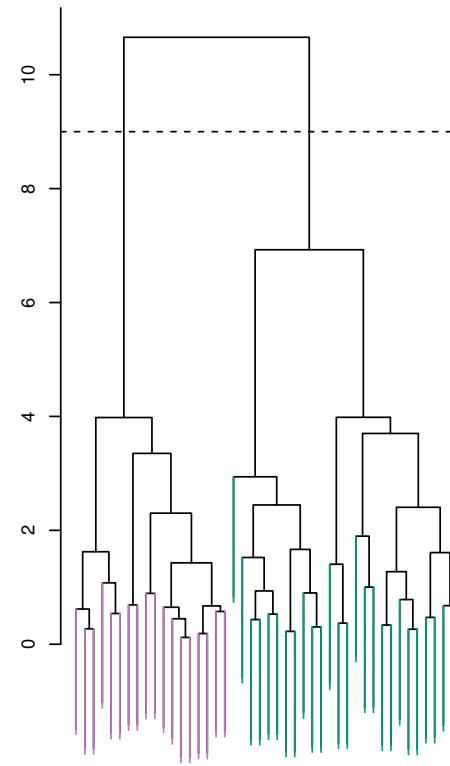
— 1 und 6 sind sich ähnlich, da ihre Verbindung niedrig auftritt
- - - 2 und 9 sind sich unähnlich, da ihre Verbindung hoch auftritt

Verschiedene Schnitte

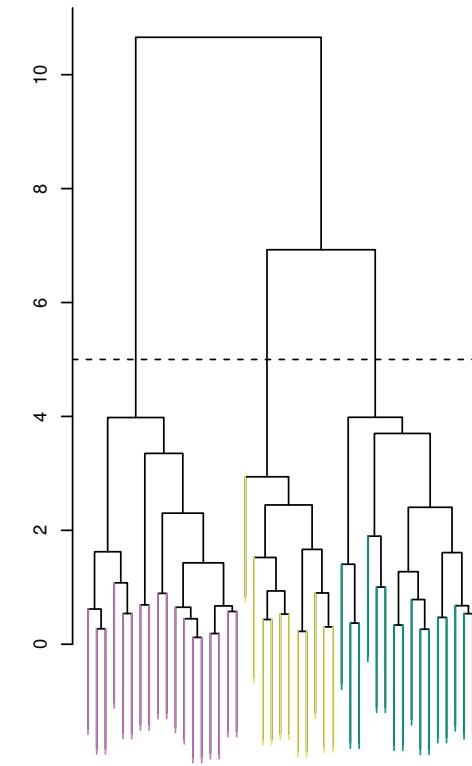
Die Höhe des Schnitts entscheidet, wie viele Cluster entstehen



Ein Cluster



Zwei Cluster



Drei Cluster

Agglomeratives Clustering

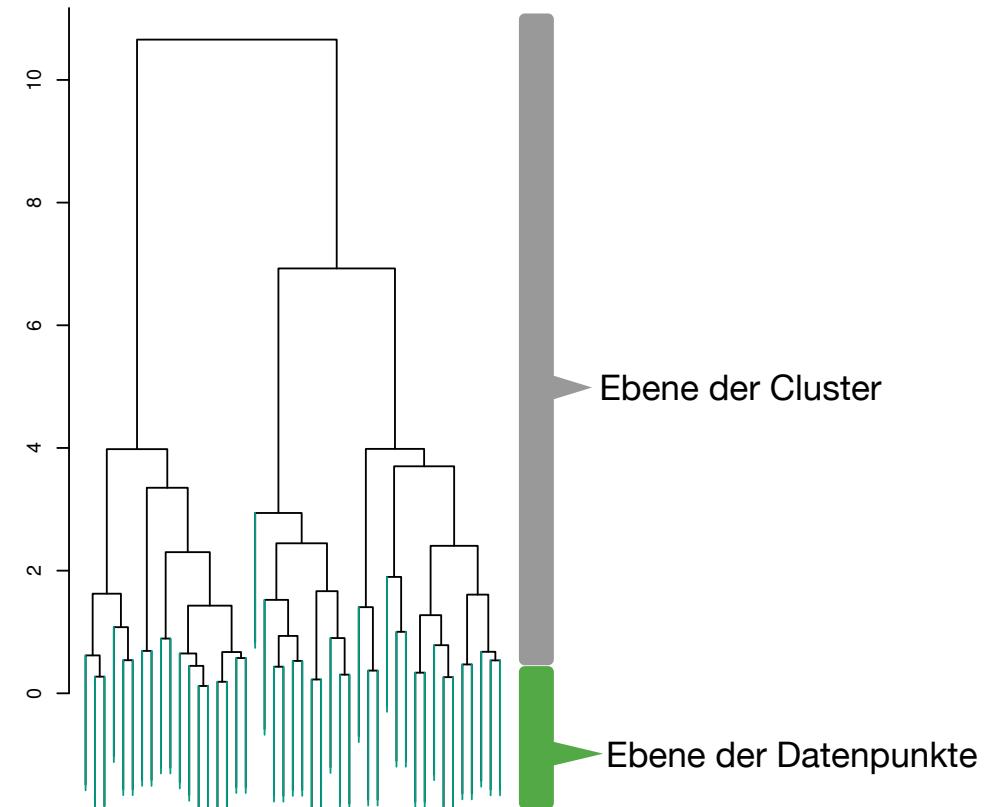
1. Weise jedem Paar von Datenpunkten ein Unähnlichkeitsmaß zu und fasse jeden Datenpunkt als ein Cluster auf.
2. Für $i = N, N - 1, \dots, 2$:
 - a) Vergleiche paarweise zwischen allen Clustern ihr Unähnlichkeitsmaß und fusioniere die beiden Cluster mit der geringsten Unähnlichkeit. Die Unähnlichkeit bestimmt die Höhe der Fusion im Dendrogramm.
 - b) Bestimme für die übrig gebliebenen $i - 1$ Cluster paarweise ihr Unähnlichkeitsmaß.

1. Unähnlichkeit auf Ebene der Datenpunkte

- wird durch **Unähnlichkeitsmaß** (auch Distanzmaß) bestimmt
- Häufig wird die euklidische Metrik gewählt

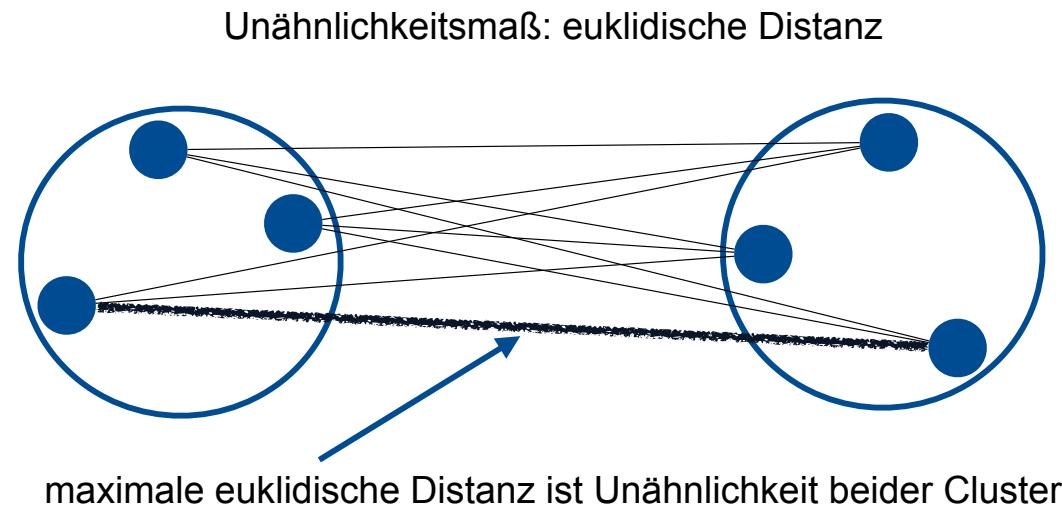
2. Unähnlichkeit auf Ebene der Cluster

- wird durch *Linkage* (Verknüpfung) bestimmt
- Unähnlichkeit von Clustern hängt von den enthaltenen Datenpunkten ab
- Auf welche Weise die einzelnen Punkte in die Unähnlichkeit der Cluster fließt, entscheidet die Linkage
- Hier werden drei Arten behandelt: *complete*, *single* und *average*



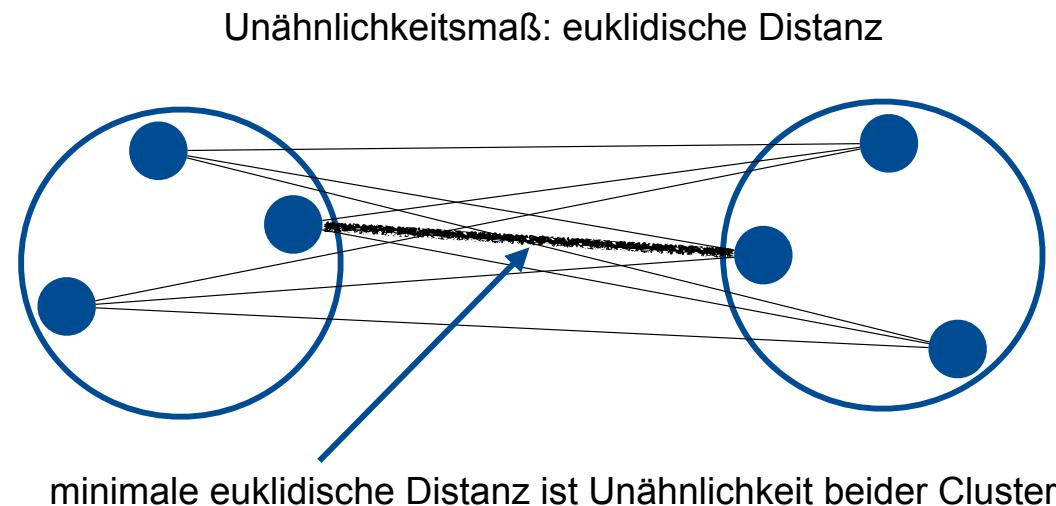
Maximale Inter-Cluster-Unähnlichkeit

- Bestimme die paarweisen Unähnlichkeiten zwischen den Punkten aus dem ersten und dem zweiten Cluster
- Die Unähnlichkeit beider Cluster wird definiert als der **größte** Wert unter den bestimmten Unähnlichkeiten



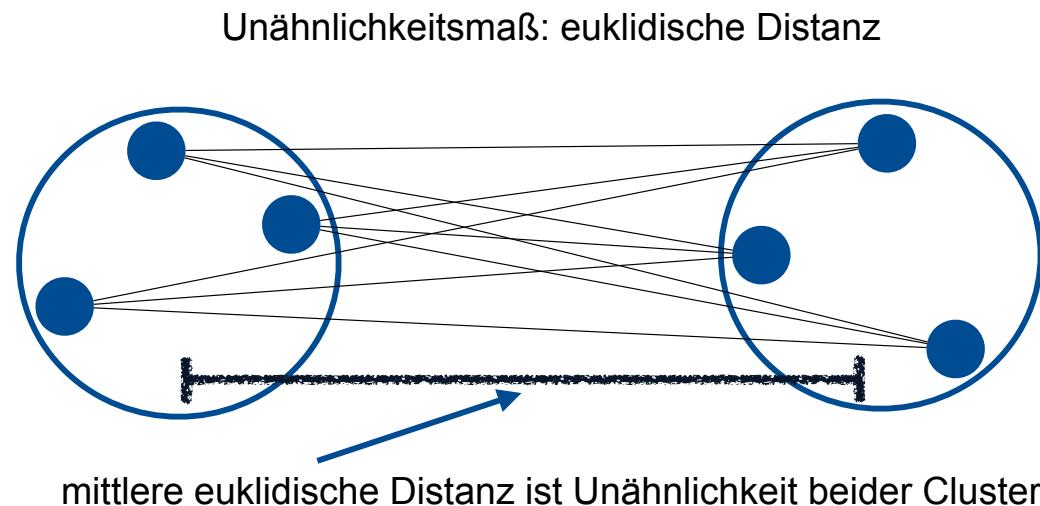
Minimale Inter-Cluster-Unähnlichkeit

- Bestimme die paarweisen Unähnlichkeiten zwischen den Punkten aus dem ersten und dem zweiten Cluster
- Die Unähnlichkeit beider Cluster wird definiert als der **kleinste** Wert unter den bestimmten Unähnlichkeiten



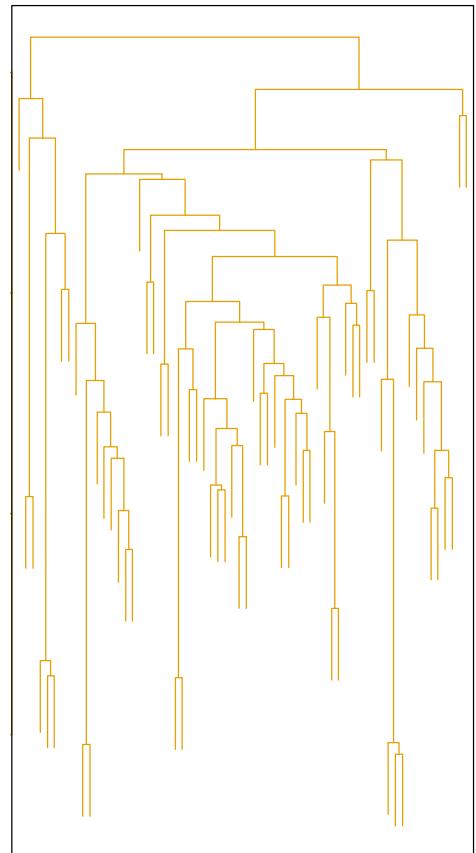
Mittlere Inter-Cluster-Unähnlichkeit

- Bestimme die paarweisen Unähnlichkeiten zwischen den Punkten aus dem ersten und dem zweiten Cluster
- Die Unähnlichkeit beider Cluster wird definiert als der **Mittelwert** aller bestimmten Unähnlichkeiten

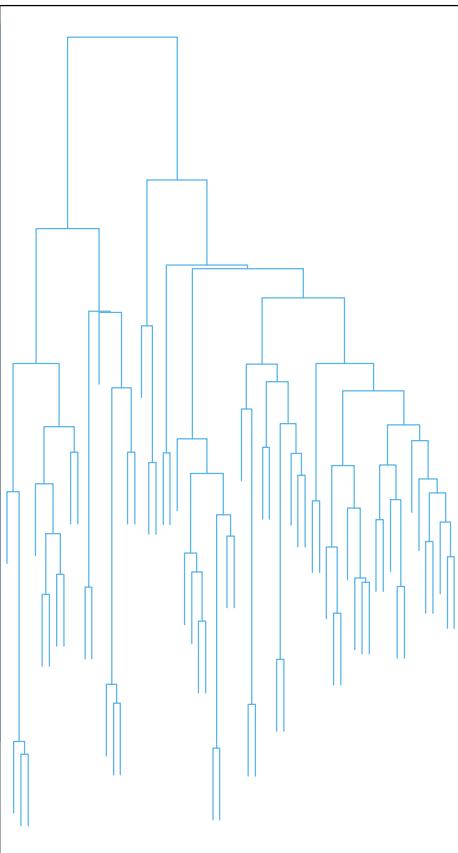


Vergleich Linkage-Methoden

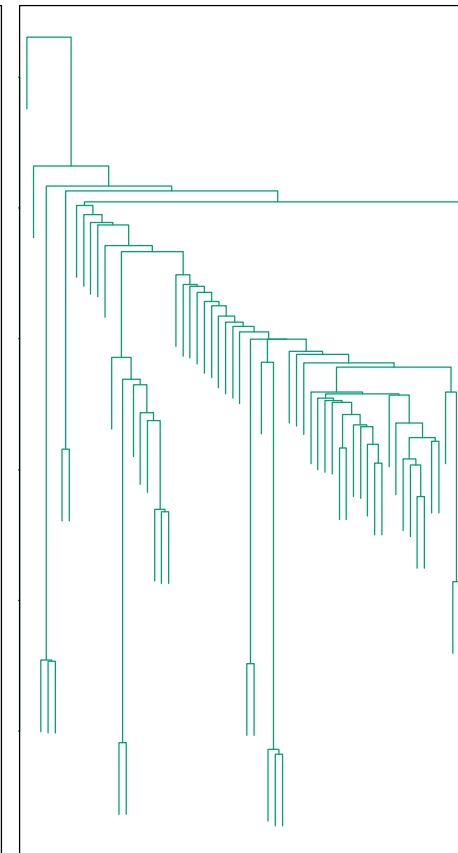
Average Linkage



Complete Linkage



Single Linkage



- Average (Mittelwert) und Complete (Maximal) Linkage erzeugen relativ ausgeglichene Cluster
- Single (Minimal) Linkage erzeugt häufig unausgeglichene Cluster:
 - Bestehende Cluster werden häufig mit dem nächstliegenden Punkt zusammengeführt

- Zwingt nicht hierarchisch organisierten Daten trotzdem eine hierarchische Struktur auf
- Sehr rechenintensiv für hohe Dimensionen bzw. große Datenmengen (Laufzeit ist $\mathcal{O}(n^3)$)
- Ungünstiges Zusammenführen zweier Cluster kann später nicht mehr korrigiert werden