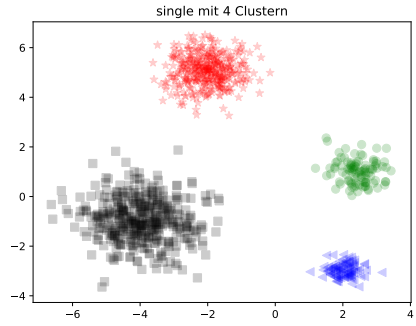
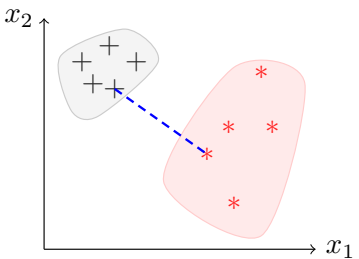
Hierarchische Clusteranalyse und Evaluierung von Clustern

Prof. Dr. Jörg Frochte

Maschinelles Lernen

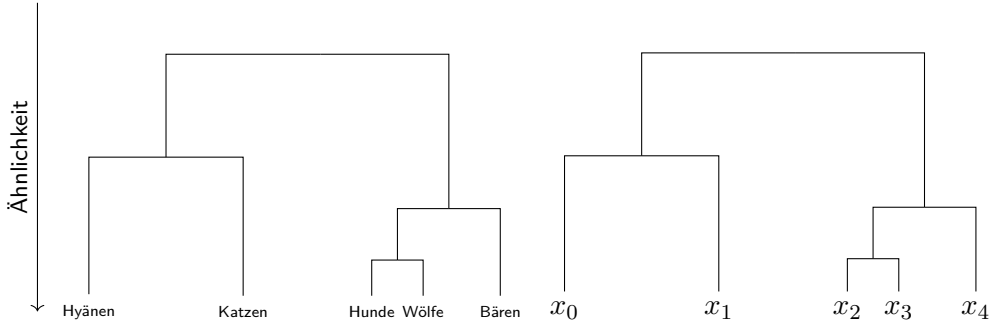


Hochschule Bochum
Bochum University
of Applied Sciences
Campus Velbert/Heiligenhaus



Hierarchische Clusteranalyse

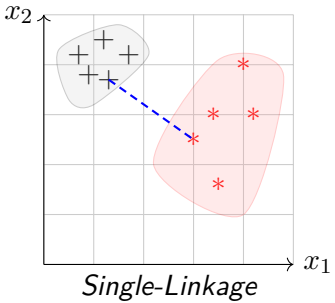
- Hierarchisch organisierten Clustern-Ansätze erinnern an die die Art wie man in der Biologie Tierarten und deren Verwandtheitsgrade aufgezeigt bekommt.
- Hierbei entstehen **Dendrogramme**, welche oft für die Visualisierung einen Mehrwert ergeben.
- Auch müssen wir uns nicht vorher festlegen, wie viele Cluster es geben soll.



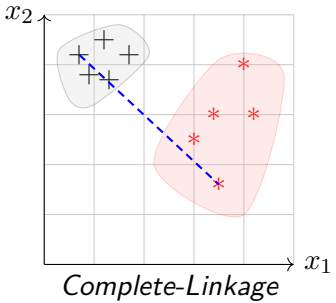
Agglomerative und divisive Clusterverfahren

- Im Vorgehen unterscheidet man beim hierarchischen Clustering zwei Ansätze: **agglomerative** und **divisive Clusterverfahren**
- Zu Beginn eines divisiven (eng. *spaltend*) Clusterverfahrens steht ein großer Cluster, welcher alle Elemente beinhaltet. Dann wird dieser Cluster schrittweise in kleinere Untereinheiten aufgespalten.
- Am Schluss steht ein Zustand, in dem jeder Cluster genau ein Objekt beinhaltet.
- Wir konzentrieren uns auf den agglomerativen Ansatz, der genau umgekehrt vorgeht.
- Hier erhält zunächst jedes Objekt einen eigenen Cluster. Das entspricht der untersten Ebene im Dendrogramm.
- Nun suchen wir die zwei davon heraus, die sich am nächsten sind, und bilden aus den beiden einen neuen Cluster. So geht man immer weiter vor.
- Der kritische Aspekt ist hier wieder das Konzept *Ähnlichkeit* und zwar auf Element- und Cluster-Ebene.

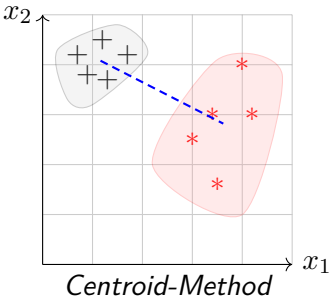
- Zwischen einzelnen Objekten nutzen wir erneut entsprechende Metriken.
- Die Metrik funktioniert zunächst nur auf der Ebene des einzelnen Objektes bzw. Vektors. Für Cluster müssen wir das Konzept erweitern.
- Es gibt verschiedene Ansätze. Wir betrachten hier vier populäre, nämlich: **Single-Linkage**, **Complete-Linkage**, **Centroid-Method** und **Average-Linkage**.



Single-Linkage



Complete-Linkage



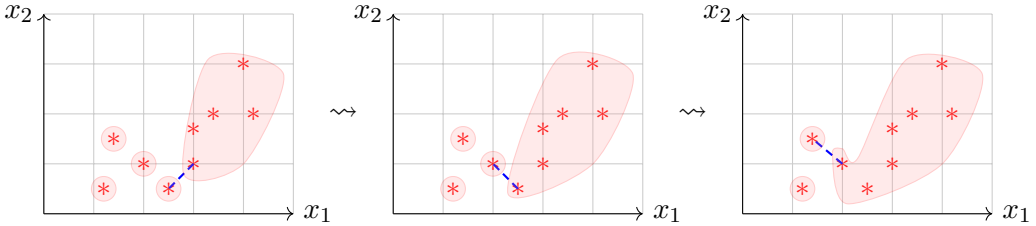
Centroid-Method

Single-Linkage und Kettenbildung

- **Single-Linkage** nutzt die Distanz, welche beiden nächsten Objekte haben:

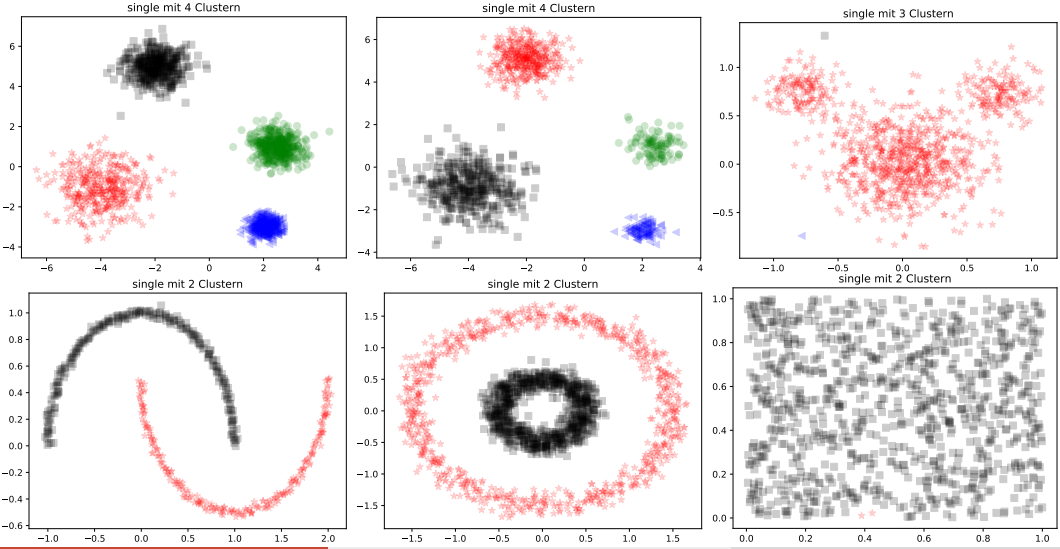
$$D_{\text{sl}}(C_1, C_2) = \min_{a \in C_1, b \in C_2} \{d(a, b)\}$$

- Ein großes Problem beim Single-Link ist die Tendenz zur Kettenbildung.
- Dieses Ausbreiten eines Clusters entspricht oft nicht der Intention einer Clusteranalyse und den Ähnlichkeiten, die man als Mensch glaubt zu erkennen.



Kettenbildung bei Verwendung von Single-Linkage

Ergebnisse für Single-Linkage



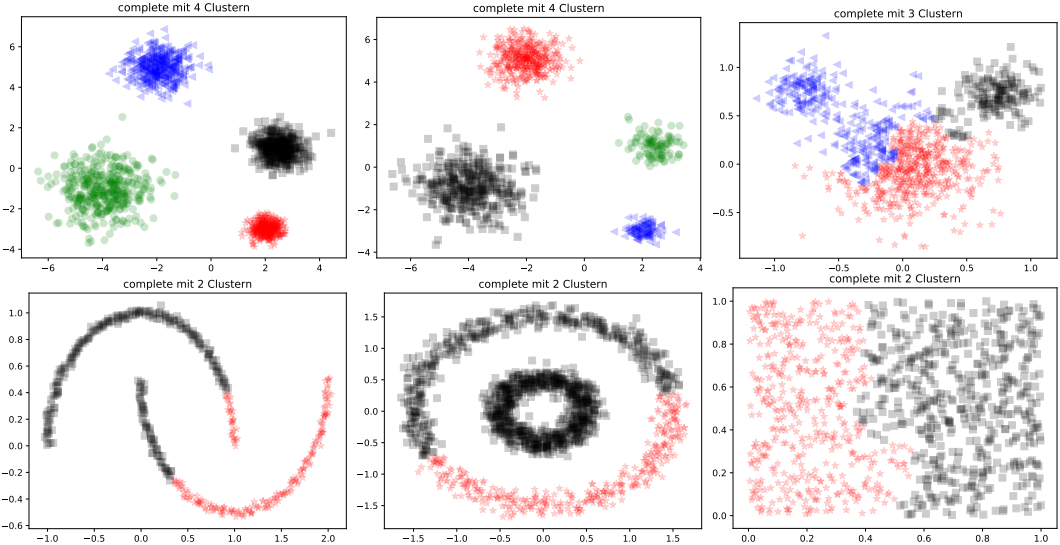
Complete-Linkage

- Der Gegenentwurf ist der Ansatz des **Complete-Linkage**.
- Hier wird die Entfernung zwischen zwei Clustern C_1 und C_2 durch den maximalen Abstand aller Elementpaare aus den beiden Clustern definiert:

$$D_{\text{cl}}(C_1, C_2) = \max_{a \in C_1, b \in C_2} \{d(a, b)\}$$

- Während Single-Linkage zu Ketten tendiert, weil dieser Ansatz mehr Ähnlichkeiten sieht als Unterschiede, ist es bei Complete-Linkage genau umgekehrt.
- Man konzentriert sich auf die Unterschiede und gelangt folglich zu vielen kleinen Gruppen, die erst sehr spät zusammenwachsen.

Ergebnisse für Complete-Linkage



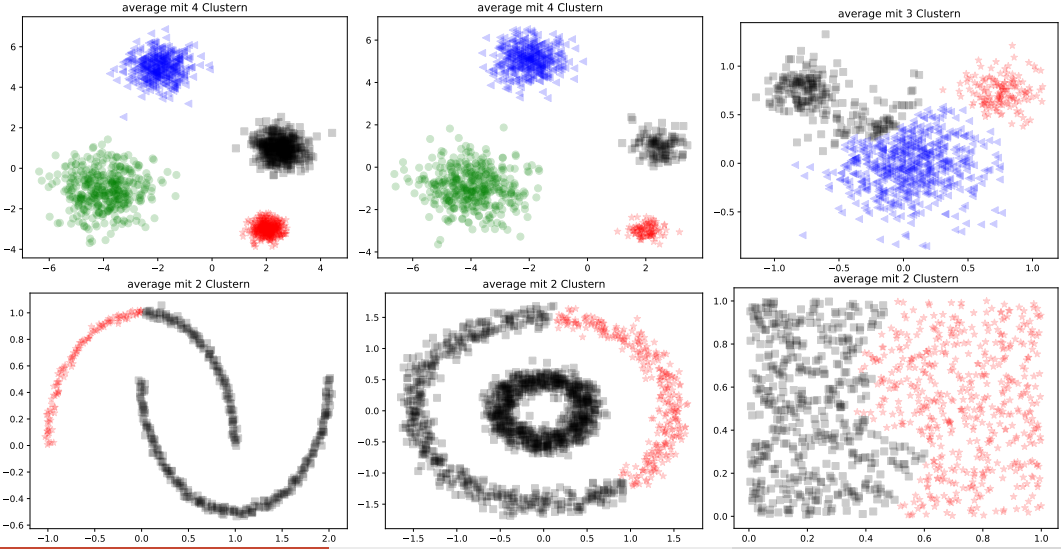
Average-Linkage

- Bei **Average-Linkage** werden die Abstände von allen Elementen beider Klassen zueinander gebildet.
- Daraus wird dann durch Mittelung der durchschnittliche Abstand aller Elementpaare berechnet. Als Distanz formalisiert erhalten wir:

$$D_{\text{al}}(C_1, C_2) = \frac{1}{\#C_1 \cdot \#C_2} \sum_{a \in C_1, b \in C_2} d(a, b)$$

- Mit $\#C_1$ wird die Mächtigkeit der Menge (die Anzahl ihrer Elemente) bezeichnet.
- Dieser gemittelte Ansatz tendiert zwar nicht zu extremem Verhalten wie Single-Linkage oder Complete-Linkage, hat jedoch einen anderen Nachteil.
- Ebenso wie k-Means tendiert er dazu, konvexe Mengen auszubilden.
- Es ist nicht in dem Sinne eingebaut wie bei k-Means und auch kann dieses Verfahren auch leichte Formänderungen ausbilden (Kartoffel vs. Kugel) die Tendenz ist aber da.

Ergebnisse für Average-Linkage

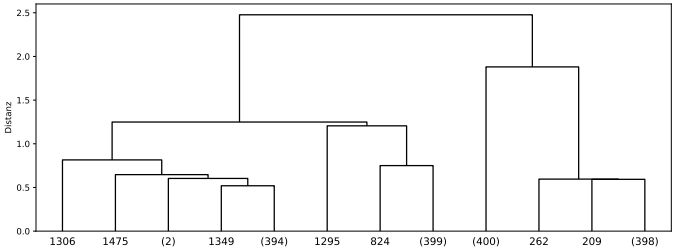


Laufzeitverhalten

- Das hierarchische Clustern ist nicht billig.
- Grund sind die Distanzen, die für die Fusionen berechnet werden müssen.
- Der Algorithmus für ein hierarchisches Clustering hat i. A. eine kubische Komplexität bzgl. der Rechenzeit und eine quadratische bzgl. des Speicherverbrauchs.
- Es gibt jedoch Möglichkeiten Single- & Complete-Linkage Ansätze günstiger umzusetzen und zwar quadratisch umzusetzen.
- Da **Average-Linkage** teuer ist als k-Means, jedoch ähnliche Eigenschaften aufweist, wird diese Variante in der Praxis selten eingesetzt.

Umgang mit Ausreißern

- Durch die Analyse der Dendrogramme können wir den Aspekt von Ausreißern adressieren.
- Eine einzelne Zahl steht für ein spezielles Element und eine Zahl in Klammern für die Anzahl in einem Cluster.



Dendrogramm für Beispiel mit vier Clustern & Single-Linkage

- Suchen wir z. B. 4 Cluster so sind wir auf der untersten Ebene schon i. W. fertig und können den Rest als Ausreißer einordnen.
- Können wir die Anzahl nicht schätzen so können wir über die Distanz gehen und z. B. Schwellen festlegen ab denen nicht mehr zusammengeschlossen wird.
- Der *Abstand* bezieht sich auf die gewählte Vorgehensweise (hier Single-Linkage).

Evaluierung von Clustern

Ziel einer Clusteranalyse ist, Mengen so einzuteilen, dass:

- Die Ähnlichkeit innerhalb der Gruppen maximiert wird (**Konsistenz**) und
- die Unterschiede zwischen den Gruppen maximiert werden (**Separation**).

- Dies geschieht z.B. bei anhand eines Optimierungsproblems. kann.
- Das Optimum davon ist jedoch nur optimal für ein festes k. Es ist nicht das Optimum der Aufgabe, Cluster zu bilden, sondern das k Cluster entsprechend dem k-Means-Funktional zu bilden.
- Wie vergleichen wir denn allgemein, ob das Clustering der Daten gelungen ist?
- Das hängt zunächst vom Ziel ab und ist oft objektiv sehr schwierig zu sagen, denn es gibt kein allgemein gültiges Qualitätsmaß.

Evaluierung von Clustern – extrinsische Qualitätsmaß

Es gibt zwei Hauptarten, erreichte Cluster von Daten zu bewerten:

- Ein von außen gegebenes, also extrinsisches, Qualitätsmaß (eng. extrinsic) sowie
- Ein aus dem Cluster selbst gegebenes intrinsisches Qualitätsmaß (eng. intrinsic)

- Oft sind Cluster-Algorithmen kein Selbstzweck, sondern sie werden verwendet
  - als Zwischenschritt für eine weitere Verarbeitung,
  - zum Auffinden bzw. Aussortieren von Outliern
  - ...

  Das alles sind von außen gegebene Qualitätsmaße.
- Wenn der Prozess mit Algorithmus A messbar besser funktioniert als mit Algorithmus B, ist der eben um diesen Unterschied besser geeignet.
- Ein Beispiel ist die Homogenisierung von Trainingsdaten für ein überwachtes Lernen.

Evaluierung von Clustern – intrinsische Qualitätsmaß

Was sind Qualitätsmaße, die aus den Clusterdaten selber kommen?

Das können sein:

- Hilft Ihnen das Clustering die Daten besser zu verstehen?
- Ähnlichkeit zu einer verwandten Klassifikationsaufgabe
- oder Indikatoren, die anzeigen, ob Konsistenz und Separation gut gelungen sind.

- Der erste Punkt ist sehr wichtig und oft hilfreich, aber schwer objektivierbar.
- Wir haben z.B. eine Fragestellung mit Länderdaten.
- Eine Erwartungshaltung ist hier nicht leicht zu formulieren, denn wir betrachten unterschiedliche Eigenschaften. W
- ürden wir uns primär Wirtschaftsdaten ansehen, könnten wir nach den G7 oder G20 suchen und schauen, was mit ihnen passiert ist.

Evaluierung von Clustern – Indikatoren

- Es gibt sehr viele Indikatoren und die meisten sind schwer vergleichbar über Algorithmengrenzen hinweg.
- Nehmen wir als Beispiel den **Dunn Index** aus dem Jahr 1974.
- Die Idee besteht darin, den Quotienten aus Separation und Konsistenz zu bilden:

$$\mathrm{dunn}(C) = \frac{\mathrm{Sep}(C)}{\mathrm{Comp}(C)}$$

- Das Problem liegt darin, wie man Separation und Konsistenz definiert.
- Nehmen wir die Separation, dann können und werden alle Ansätze benutzt, die wir bei hierarchischen Ansätzen diskutiert haben: Single-Linkage, Complete-Linkage, Centroid-Method und Average-Linkage.
- Beispiel: Hierarchisch gebildete Cluster mit Single-Linkage und Complete-Linkage vergleichen. Welches Maß soll man hier für die Sepration wählen?
- Derjenige, der nach dem entsprechenden Maß die Cluster gebildet hat, ist immer im Vorteil. Ähnliches gilt für die Kompaktheit.

Evaluierung von Clustern – K-Nearest-Neighbor-Consistency

- Ich möchte Ihnen daher einen Ansatz aus dem Jahr 2014 von C. Ding et al vorstellen, der zumindest für unsere Algorithmen ein fairer Vergleich ist.
- Es geht um die **K-Nearest-Neighbor-Consistency**, welche die Dichte eines Clusterings betrachtet.
- Dazu wird die Umgebung jedes Elementes analog zum k-NN betrachtet, nur dass wir weder klassifizieren noch eine Regression durchführen wollen.
- Wir lassen uns die k Nachbarn jedes Objektes geben und schauen, ob diese alle zum selben Cluster gehören.
- Diesen Umstand gilt es zu maximieren, ein höherer Index ist wünschenswert.

Evaluierung von Clustern – K-Nearest-Neighbor-Consistency

- Der Index, wobei ich eine leichte normierte Abwandlung empfehle, berechnet sich wie folgt:

$$\frac{1}{k}\frac{1}{K}\sum_{c_i \in C}\frac{n}{\#c_i}$$

- n ist dabei die Anzahl der konsistenten Nachbarn in einem Cluster und
- $\#c_i$ die Anzahl der Elemente im Cluster c_i.
- K ist die Anzahl der Cluster und k die Anzahl der Nachbarn, die jeweils überprüft werden.
- Da jeder Punkt mit k Nachbarn in die Berechnung eingehen kann, ergibt sich als maximal möglicher Durchschnitt über alle Cluster k als Obergrenze.
- Dividiert man, wie oben, das Ergebnis noch durch k, so entsteht ein Index, bei dem 1 für die größte mögliche Konsistenz steht.
- Die Separation wird durch diesen Index nicht gut erfasst.