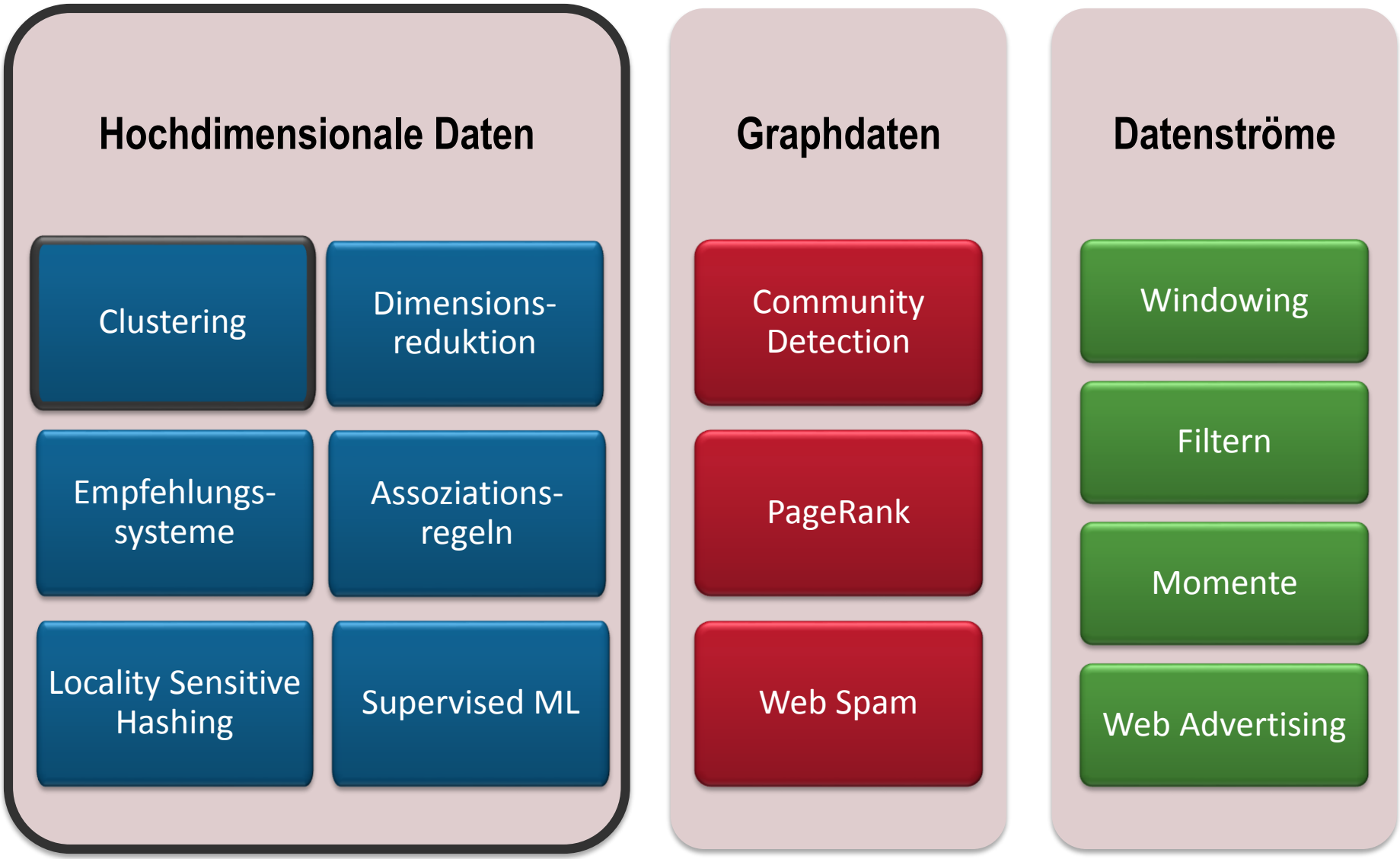
Data Mining

Clustering

Dr. Hanna Köpcke
Wintersemester 2020

Abteilung Datenbanken, Universität Leipzig
http://dbs.uni-leipzig.de

Übersicht


Hochdimensionale Daten
Clustering
Dimensions-reduktion
Empfehlungs-systeme
Assoziations-regeln
Locality Sensitive Hashing
Supervised ML
Graphdaten
Community Detection
PageRank
Web Spam
Datenströme
Windowing
Filtern
Momente
Web Advertising

Inhaltsverzeichnis

Clustering

- Gegeben einer Menge von $\boldsymbol{N}$ **Datenpunkten** im $\mathbb{R}^d$

$$\boldsymbol{x_1} = (\mathrm{x}_{11}, \mathrm{x}_{12}, \ldots, \mathrm{x}_{1\mathrm{d}}),$$
$$\boldsymbol{x_2} = (\mathrm{x}_{21}, \mathrm{x}_{22}, \ldots, \mathrm{x}_{2\mathrm{d}}),$$
$$\ldots,$$
$$\boldsymbol{x}_N = (\mathrm{x}_{N1}, \mathrm{x}_{N2}, \ldots, \mathrm{x}_{\mathrm{Nd}})$$

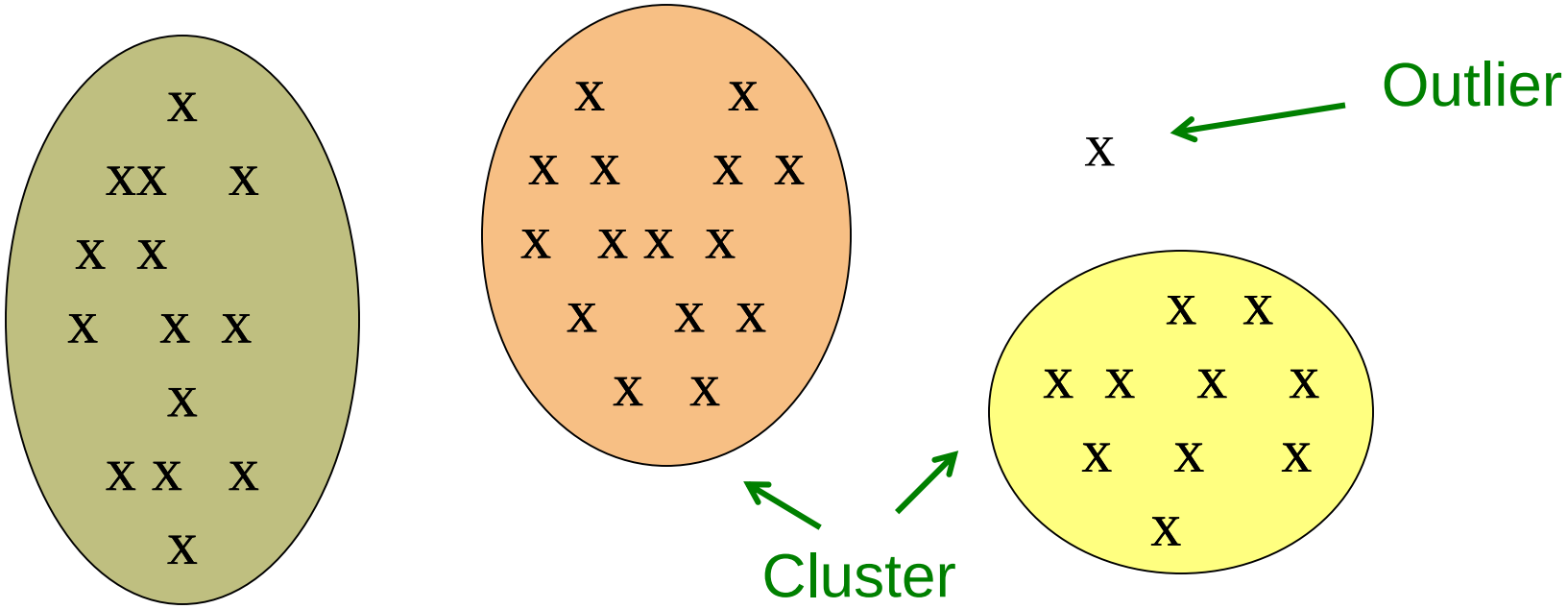
- Distanzfunktion $\boldsymbol{d}(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}_j)$, z.B. Euklidisch:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{d} (x_{ik} - x_{jk})^2}$$

- **Ziel**: Gruppierung der Datenpunkte in **Cluster**, so dass
  - Mitglieder eines Cluster weisen eine geringe paarweise Distanz auf
  - Mitglieder verschiedener Cluster weisen eine hohe paarweise Distanz auf

Clustering

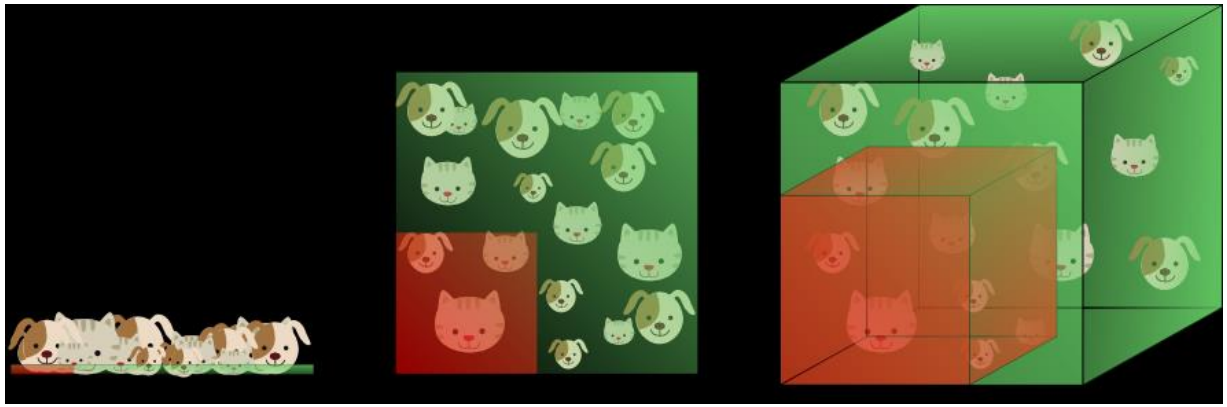
- Visualisierung möglich bei nur 2 Dimensionen:



Das Problem

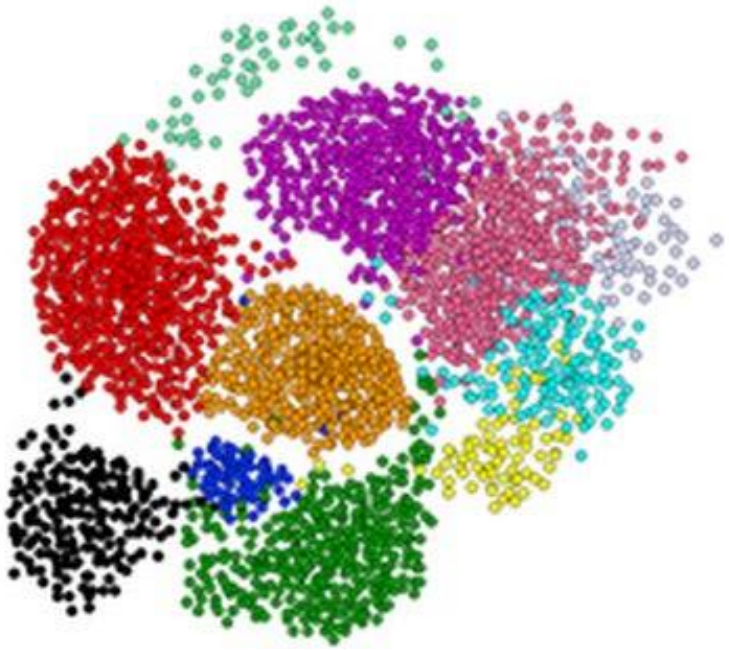
- Clustering ist *anspruchsvoll* im Fall großer Datenmengen
  - Gegebene Anzahl an Cluster k
  - $\boldsymbol{k^N}$ **Möglichkeiten** die N Punkte in k Cluster zu ordnen
  - Paarweiser Vergleich erfordert Berechnung von $\binom{N}{2}$ Ähnlichkeiten

- Clustering ist *anspruchsvoll* bei hoher Dimension der Datenpunkte
  - Oft: 10-10 000 Dimensionen
  - **The Curse of Dimensionality**: Im Falle einer sehr hohen Dimension haben fast alle Paare von Datenpunkten eine ähnliche Distanz



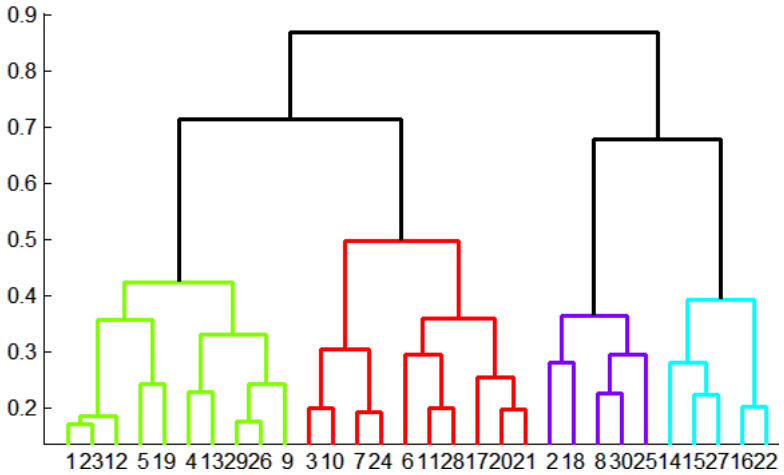
Beispiele

- Gruppierung von Musikalben nach Käufern
  - Zwei Alben sind ähnlich, wenn sie von den selben Personen gekauft wurden
  - Eine Dimension pro Käufer
  - Amazon: mehrere Millionen Dimensionen
- Gruppierung von Dokumenten nach Thema
  - Zwei Dokumente behandeln das selbe Thema, wenn sie die gleichen Wörter enthalten
  - Unbegrenzte Anzahl von Dimensionen möglich
- DNA Sequence Clustering
  - Gruppierung von DNA-Sequenzen
  - Menschliches Genom: > 3 Mrd. Basenpaare
- Segmentierung von Bildern
  - Markierung zusammengehörender Bild-regionen über ähnliche Pixel
  - Merkmale: Farbton, Helligkeit, Textur, Lage, …

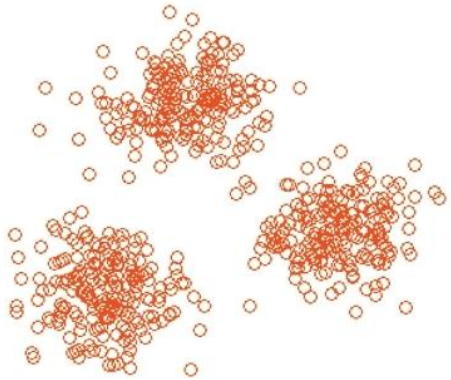


Übersicht: Clusterverfahren

- **Hierarchisch:**
  - **Agglomerativ** (Bottom-up):
    - Zu Beginn bildet jeder Punkt ein Cluster
    - Wiederholtes Kombinieren von zwei ähnlichen Clustern zu einem neuen Cluster
  - Divisiv (Top-down):
    - Ein großes Cluster zu Beginn
    - Wiederholtes Aufteilen eines großen Clusters in zwei unähnliche kleinere Cluster



- **Partitionierend:**
  - Feste Anzahl an Cluster
  - Zuordnen der Punkte zu den Clustern

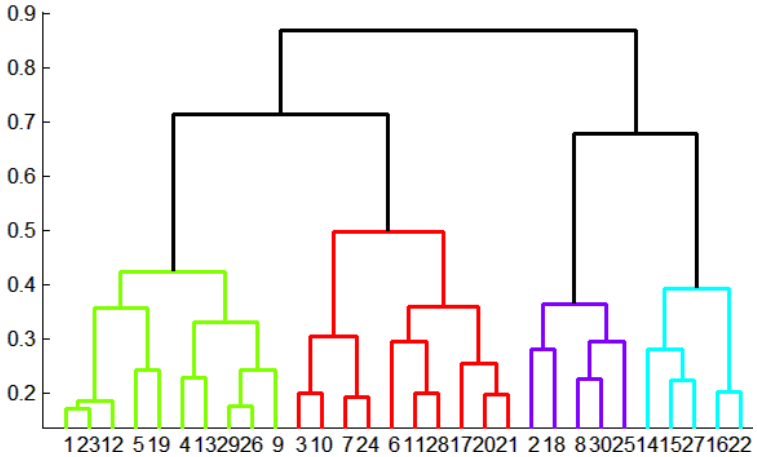


Inhaltsverzeichnis

Hierarchische Clusteranalyse

- **Agglomerativ:** Wiederholtes Kombinieren der beiden Cluster mit geringster Distanz zu einem neuen Cluster

- **Dendrogramm:**
  - Blätter (unterste Ebene) repräsentieren die Datenpunkte
  - Jeder Punkt ist ein Cluster
  - Höhe der Vereinigung gibt die Distanz zwischen den beiden Clustern an



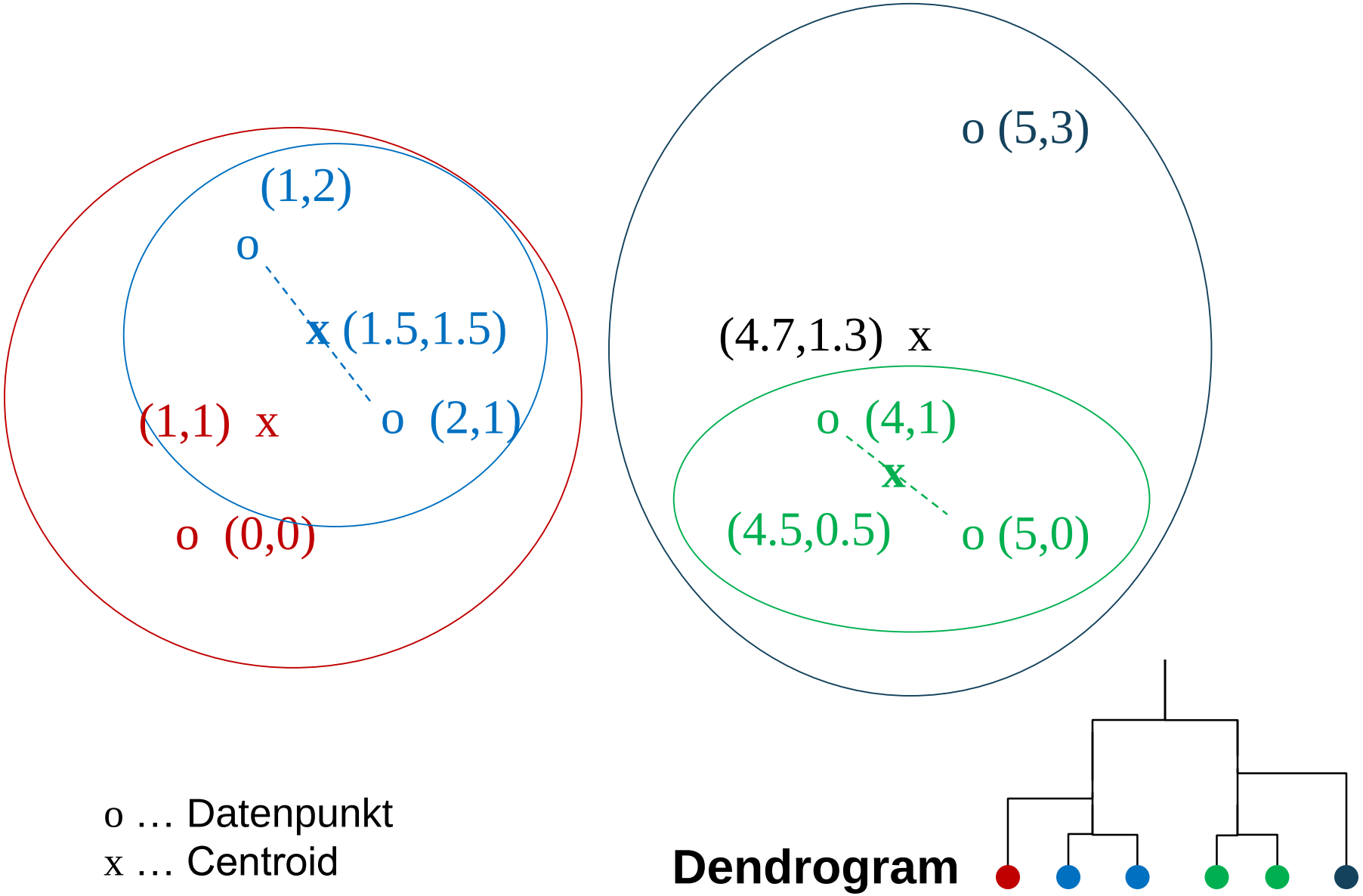
- **Zwei Kriterien:**
  1. Distanz zwischen zwei Clustern
  2. Stoppregel

Hierarchische Clusteranalyse (agglomerativ)

1. Mögliche Definitionen der Distanz zwischen zwei Clustern:
   a) Distanz zwischen den beiden *Centroiden* der Cluster (*Centroid* = Arithmetisches Mittel aller Punkte des Clusters)
   b) Maximale paarweise Distanz zwischen allen Mitgliedern der beiden Cluster
   c) Minimale paarweise Distanz zwischen allen Mitgliedern der beiden Cluster
   d) Durchschnittliche paarweise Distanz zwischen allen Mitgliedern der beiden Cluster
2. Mögliche Stoppregeln:
   a) Anzahl der Cluster
   b) Maximale Distanz innerhalb des neu entstandenen Clusters übersteigt Schwellenwert
   c) Durchschnittliche maximale Distanz steigt stark an
   d) „Dichte“ der Cluster liegt unter einem Schwellenwert

- Auch die *Skalierung* der Dimensionen hat Einfluss auf das Clustering
  - Bei Daten mit verschiedenen Attributen (z.B. Einkommen und Körpergröße)
  - Daten sollten standardisiert werden, so dass alle Attribute eine Standardabweichung von Eins aufweisen

Beispiel: Verwendung der Centroiden



o … Datenpunkt
x … Centroid

Dendrogram

Hierarchische Clusteranalyse

- Auch in Nicht-Euklidischen Räumen möglich
  - z.B. über Jaccard-Metrik
  - Anstatt Centroid: **Clustroid** = Punkt aus dem Cluster mit
    - Minimaler Summe aller Distanzen zu den anderen Punkten des Clusters, oder
    - Minimaler maximale Distanz zu den anderen Punkten des Clusters
  - Oder Anwendung eines Verfahrens ohne Centroid

- Bestes Verfahren zur Auswahl zweier Cluster hängt von der Form der tatsächlichen Cluster ab (welche man nicht kennt)

Komplexität (Bearbeitungszeit)

- Auswahl ohne Centroid über minimale Distanz: $O(N^2)$
  - Einmalige Berechnung aller paarweiser Distanzen und Sortierung der Paare aufsteigend nach Distanz
  - Zusammenfügen der Cluster in dieser Reihenfolge

- Alle anderen Verfahren: $O(N^2 \log N)$
  - Naiv: In jedem Schritt müssen Distanzmaße zwischen allen Clustern neu berechnet werden: $O(N^2)$, $O((N-1)^2)$, $O\big((N-2)^2\big)$, …,
  - Da N Schritte, also $O(N^3)$ insgesamt
  - Bei Verwendung eines **Priority Queue**: $O(N^2 \log N)$

Komplexität (Bearbeitungszeit)

- **Priority Queue**: Veränderungen in $O(\log N)$
- Vorgehen:
  - Sortierung der Paare nach Distanz in Priority Queue P: $O(N^2)$

C,D	C,E	A,B	D,A	D,B	A,E	C,B	…
1.2	2.1	2.3	2.8	3.3	3.5	4.0	…

  - Wiederhole:
    - Finden des Minimums (in $O(1)$), z.B. Paar (C, D)
    - Entfernen aller Elemente aus P, welche sich auf C oder D beziehen, z.B. (C,D), (C,E), (D,A), …: max. $2N$ Veränderungen, also $O(N \log N)$
    - Berechnung aller Distanzen zwischen neuem Cluster X = (C,D) und anderen Clustern, sowie Hinzufügen dieser Paare zu P: $O(N \log N)$
  - Maximal N Wiederholungen, also $O(N^2 \log N)$ insgesamt

Inhaltsverzeichnis

k-Means

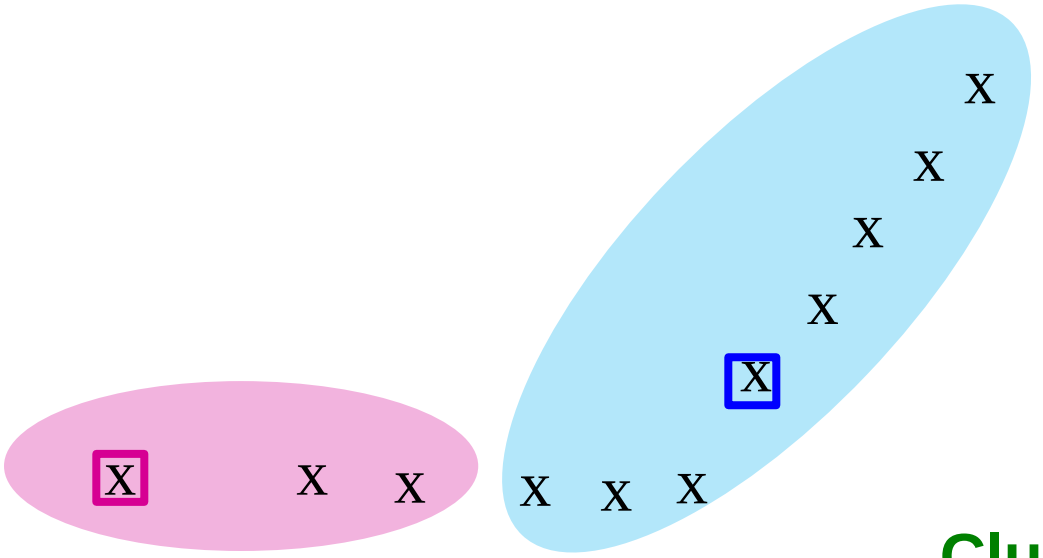
- **Anzahl der Cluster k ist vorgegeben**
- Seien $C_1, C_2, \ldots, C_k$ Mengen von Datenpunkten mit
  - $C_1 \cup C_2 \cup \cdots \cup C_k$ = Menge aller Datenpunkte
  - $C_i \cap C_j = \emptyset$ für alle $i \neq j$.
- k-Means-Clustering versucht ein Clustering $C_1, C_2, \ldots, C_k$ mit möglichst geringer durchschnittlicher Distanz innerhalb der Cluster zu finden:

$$\underset{C_1, C_2, \ldots, C_k}{\text{minimiere}} \left\{ \sum_{j=1}^{k} \frac{1}{|C_j|} \sum_{i,i' \in C_j} d(x_i, x_{i'}) \right\}$$

- Sehr schwieriges Optimierungsproblem für große Datenmengen
- Der k-Means-Algorithmus approximiert dieses Ziel *ziemlich gut*
- Komplexität: *O(N)*

k-Means-Algorithmus

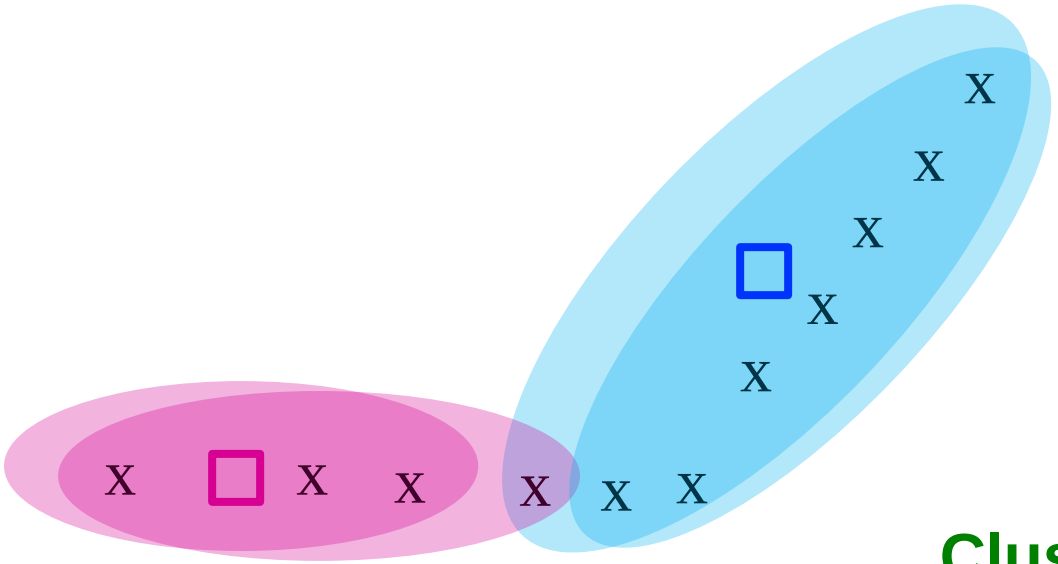
- Gegeben einer initialen Wahl von k Centroiden
- Hinzufügen aller Punkte zum Cluster mit nächstgelegenem Centroiden
- Wiederholung bis Konvergenz (keine Änderungen):
  1. Neuberechnung der Centroiden
  2. Zuordnung aller Punkte zum Cluster mit nächstgelegenem Centroiden



Cluster nach 1. Runde

k-Means-Algorithmus

- Gegeben einer Initialen Wahl von k Centroiden
- Hinzufügen aller Punkte zum Cluster mit nächstgelegenem Centroiden
- Wiederholung bis Konvergenz (keine Änderungen):
  1. Neuberechnung der Centroiden
  2. Zuordnung aller Punkte zum Cluster mit nächstgelegenem Centroiden

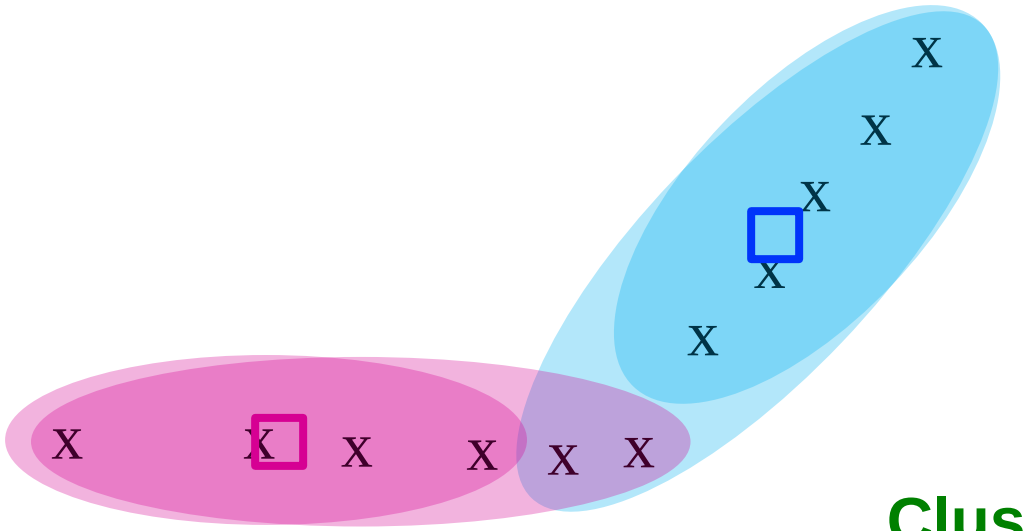


x … Datenpunkte
□ … Centroid

Cluster nach 2. Runde

k-Means-Algorithmus

- Gegeben einer Initialen Wahl von k Centroiden
- Hinzufügen aller Punkte zum Cluster mit nächstgelegenem Centroiden
- Wiederholung bis Konvergenz (keine Änderungen):
  1. Neuberechnung der Centroiden
  2. Zuordnung aller Punkte zum Cluster mit nächstgelegenem Centroiden



x … Datenpunkte
□ … Centroid

Cluster nach 3. Runde

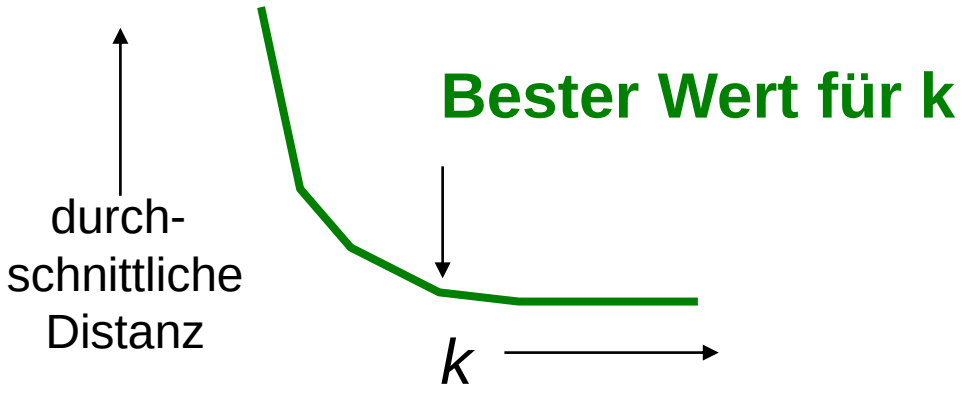
Initialisierung: Wahl von k Centroiden

- **1. Möglichkeit:** Jeder Punkt wird zufällig einem von k Clustern zugeordnet und Berechnung der dazugehörigen Centroiden
- **2. Möglichkeit:** Auswahl von k Punkten mit größtmöglichen paarweisen Entfernungen
  - Wähle ersten Punkt zufällig
  - Wiederhole: Wähle den Punkt mit der maximalen minimalen Distanz zu allen schon gewählten Punkten
- **3. Möglichkeit:** Hierarchische Clusteranalyse auf Stichprobe (so dass k Cluster entstehen) und Auswahl der jeweiligen Clustroiden

- Ergebnis des Algorithmus hängt von der Initialisierung ab
  - Die durchschnittliche Distanz ist zwar garantiert minimal aber nur lokales Minimum
  - Wiederholung mit verschiedenen Initialisierungen und Wahl des besten Clustering

Wahl von k

- Ausprobieren verschiedener Werte k = 2, 4, 8, 16, 32, …
- Schrittweises Erhöhen von k solange „sich etwas Relevantes ändert“
- Beispiel: durchschnittliche Distanz fällt signifikant

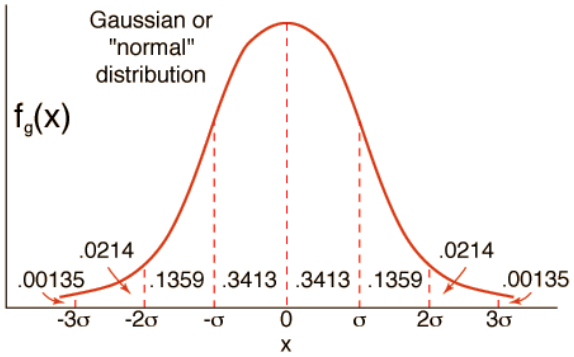


- Binäre Suche um Rechenaufwand gering zu halten
  - Annahme: Signifikante Änderung von k = 8 zu k = 16, aber keine signifikante Änderungen von k = 16 zu k = 32 → Setze k = 12
  - Falls signifikante Änderungen von k = 12 zu k = 16, setze k = 14 , …
  - Falls keine signifikante Änderungen von k = 12 zu k = 16, setze k = 10 , …

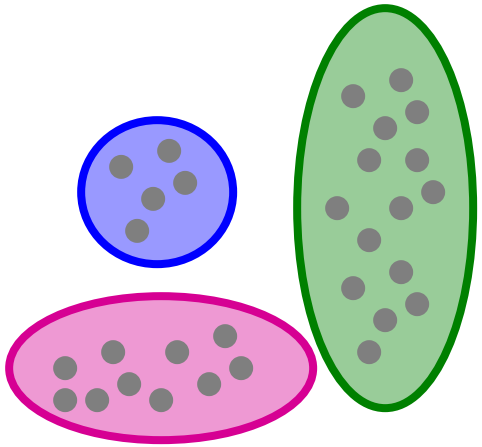
Inhaltsverzeichnis

BFR-Algorithmus

- **BFR** [Bradley-Fayyad-Reina] ist eine Variante des k-Means-Algorithmus, welche die Verarbeitung sehr **umfangreicher** Datensätze (die nicht in den Hauptspeicher passen) erlaubt



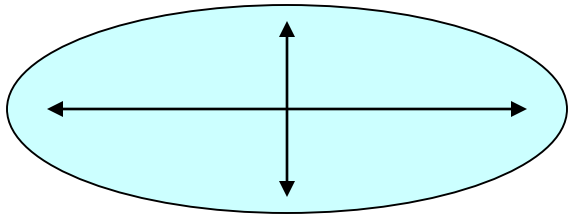
- **Annahme:** Cluster sind (multivariat) normal verteilt
  - um den Centroiden
  - mit stochastisch unabhängigen Dimensionen



- **Ziel**: Bestimmung der k Centroiden und dazugehörigen Varianzen

Repräsentation der Cluster

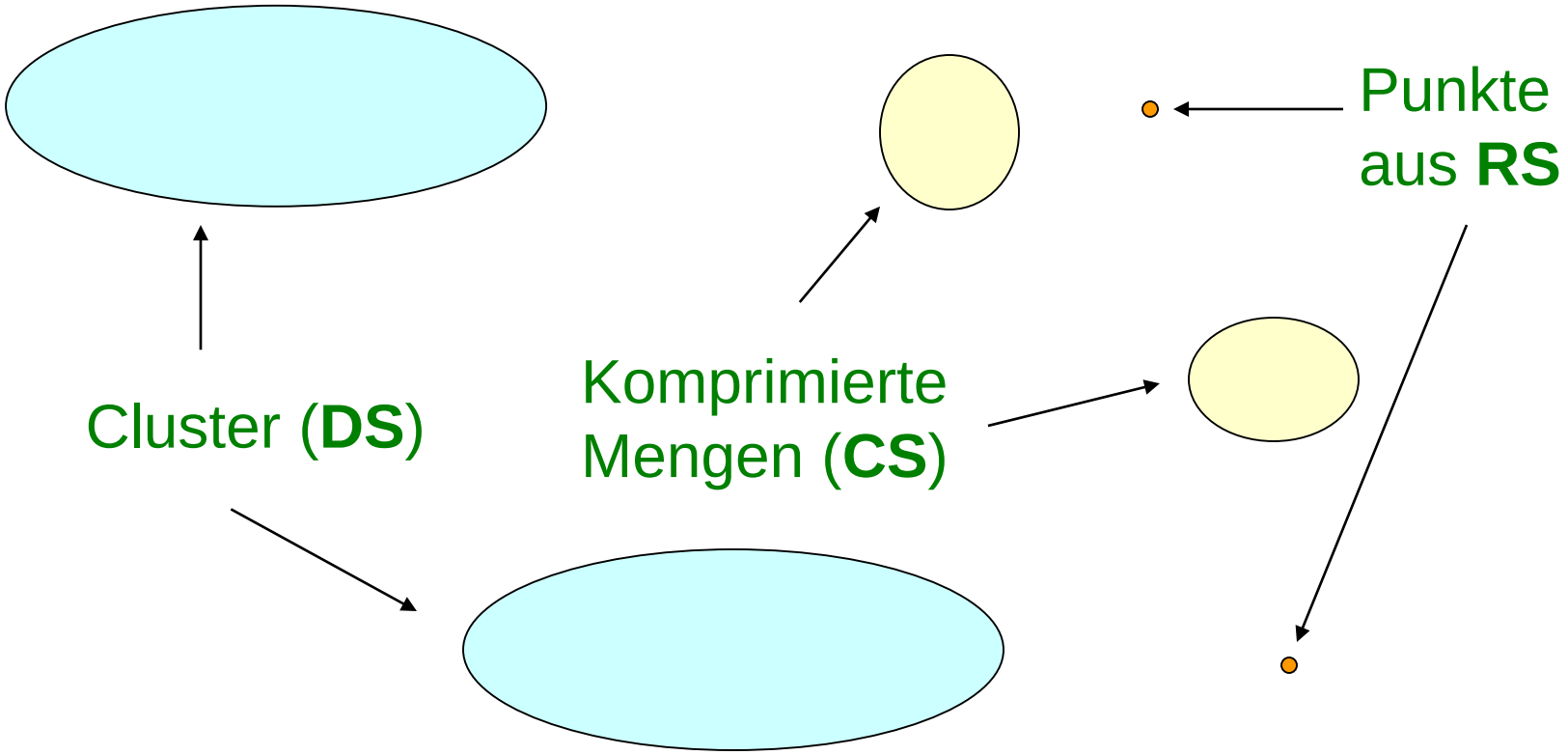
- Anzahl der Dimensionen: d
- Effiziente Zusammenfassung eines Clusters mittels $2d + 1$ Zahlen
  - n: Anzahl der, im Cluster enthaltenen, Punkte
  - $SUM_1, SUM_2, \ldots, SUM_d$ die Summen der Komponenten dieser Punkte
  - $SUMSQ_1, SUMSQ_2, \ldots, SUMSQ_d$ die Summen der Quadrate der Komponenten dieser Punkte



- Hinzufügen eines Punktes $(x_1, x_2, \ldots, x_d)$
  - n + 1
  - $SUM_1 + x_1, SUM_2 + x_2, \ldots, SUM_d + x_d$
  - $SUMSQ_1 + x_1^2, SUMSQ_2 + x_2^2, \ldots, SUMSQ_d + x_d^2$
- Centroid: $\left(\frac{SUM_1}{n}, \frac{SUM_2}{n}, \ldots, \frac{SUM_d}{n}\right)$
- Varianz: $\left(\frac{SUMSQ_1}{n}, \frac{SUMSQ_2}{n}, \ldots, \frac{SUMSQ_d}{n}\right) - \left(\frac{SUM_1}{n}, \frac{SUM_2}{n}, \ldots, \frac{SUM_d}{n}\right)^2$

Drei Mengen

- **Discard set (DS):** Punkte, die einem Cluster zugeordnet wurden
- **Retained set (RS):** Punkte, die bisher keinem Cluster zugeordet wurden
- **Compression set (CS):** Punkte aus RS die nahe genug beieinander liegen um sie zusamenzufassen (Mini-Cluster)



BFR-Algorithmus

1. Initialisiere k Cluster, z.B. Clusteranalyse auf Stichprobe
2. Wiederhole:
   a. Lade einen Chunk mit Punkten (Fülle Hauptspeicher mit Daten von Festplatte)
   b. Hinzufügen der Punkte zu den k vorhandenen Clustern (**DS**), falls deren Distanz innerhalb eines Schwellenwerts liegen
   c. Clusteranalyse auf übrigen Punkten, inkl. der Punkte aus **RS**
      - Zusammenführen der entstandenen „Mini-Cluster“ mit **CS**
      - z.B. Zusammenführen zweier Cluster, falls Varianz deren Kombination unter einem Schwellenwert liegt
      - Manche Punkte bleiben einzeln und somit in RS
   d. Evtl. Zusammenführen einiger „Mini-Cluster“ aus **CS** mit Clustern aus **DS**
3. Hinzufügen der Cluster aus **CS** und Punkte aus **RS** zu nächstliegenden Clustern aus **DS**

BFR-Cluster

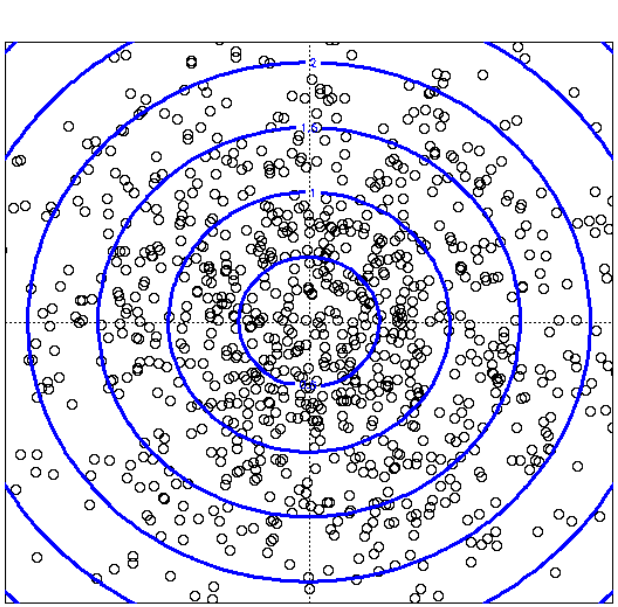
- *Nach welchem Kriterium wird ein Punkt einem vorhandenen Cluster hinzugefügt?*
- Kriterium: **Mahalanobis-Abstand** zwischen Punkt und Centroid eines Clusters ist minimal und liegt unter *einem Schwellenwert*
  - Punkt $(x_1, x_2, \ldots, x_d)$
  - Centroid $(c_1, c_2, \ldots, c_d)$
  - Standardabweichungen $(\sigma_1, \sigma_2, \ldots, \sigma_d)$

$$M(x, c) = \sqrt{\sum_{i=1}^{d} \left(\frac{x_i - c_i}{\sigma_i} \right)^2}$$

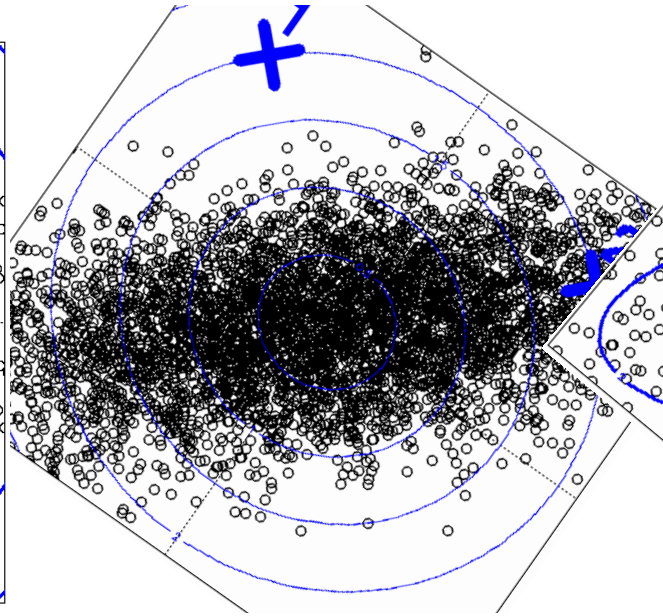
- Für 68% der Punkte gilt: $M(x, c) < \sqrt{d}$
- Für 95% der Punkte gilt: $M(x, c) < 2\sqrt{d}$
- Wahl des Schwellenwerts, so dass Punkt mit hoher Wahrscheinlichkeit zu Cluster gehört

Vergleich: Euklidisch vs Mahalanobis

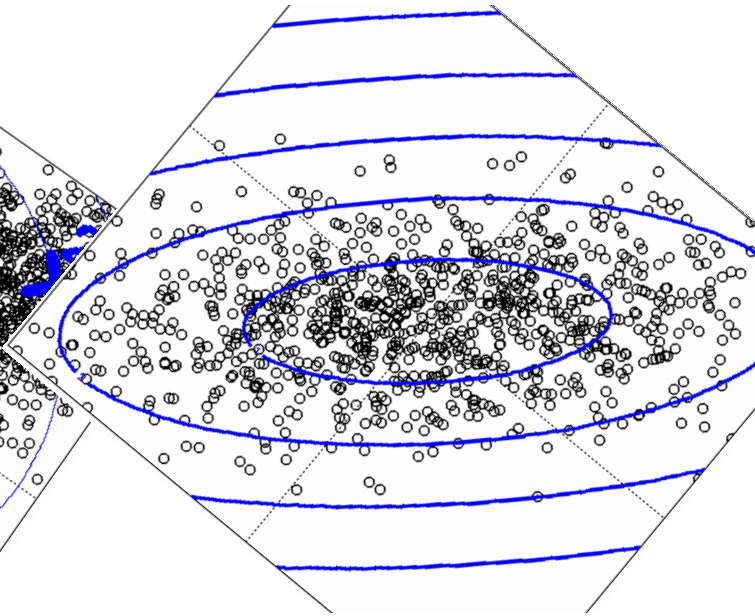
Konturlinien der Punkte mit gleichem Abstand zum Ursprung



Gleichverteilung,
Euklidische Distanz



Normalverteilung,
Euklidische Distanz

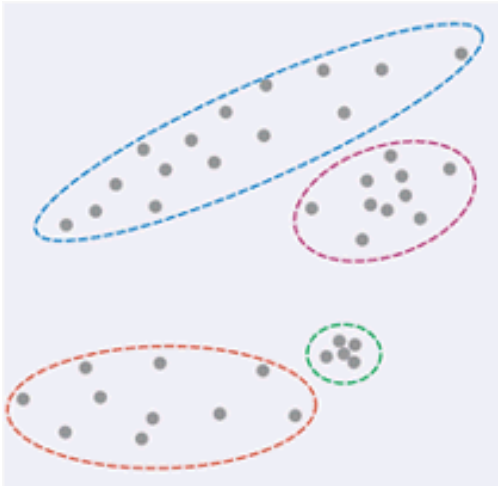


Normalverteilung,
Mahalanobis-Distanz

Inhaltsverzeichnis

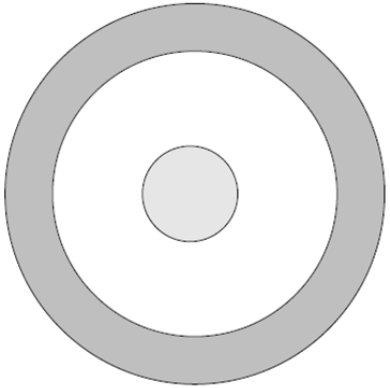
CURE-Algorithmus

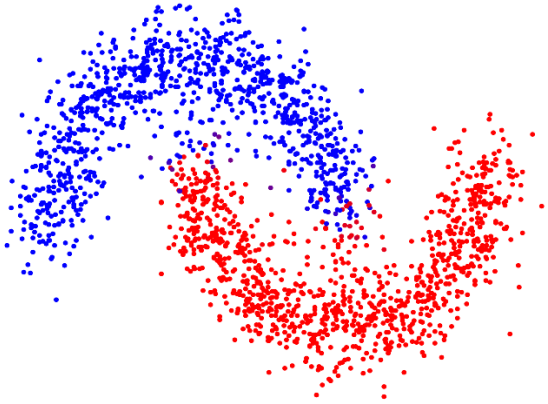
- Erweiterung von k-Means zu Clustern beliebiger Form



- **Probleme bei BFR:**
  - Annahme der Normalverteilung und unabhängige Dimensionen
  - Ellipsen sind möglich, doch keine rotierten Ellipsen

- **CURE (Clustering Using REpresentatives):**
  - Cluster beliebiger Form möglich
  - Cluster werden über eine *Menge repräsentativer Punkte* beschrieben





CURE-Algorithmus

a) Ziehe eine Zufallsstichprobe, die in den Hauptspeicher passt

b) Initialisierung: Hierarchische Clusteranalyse, wobei eine agglomerative Methode ohne Centroid bevorzugt werden sollte

c) Auswahl von repräsentativen Punkten für jedes Cluster: Innerhalb eines Clusters sollten die Punkte möglichst weit auseinander liegen

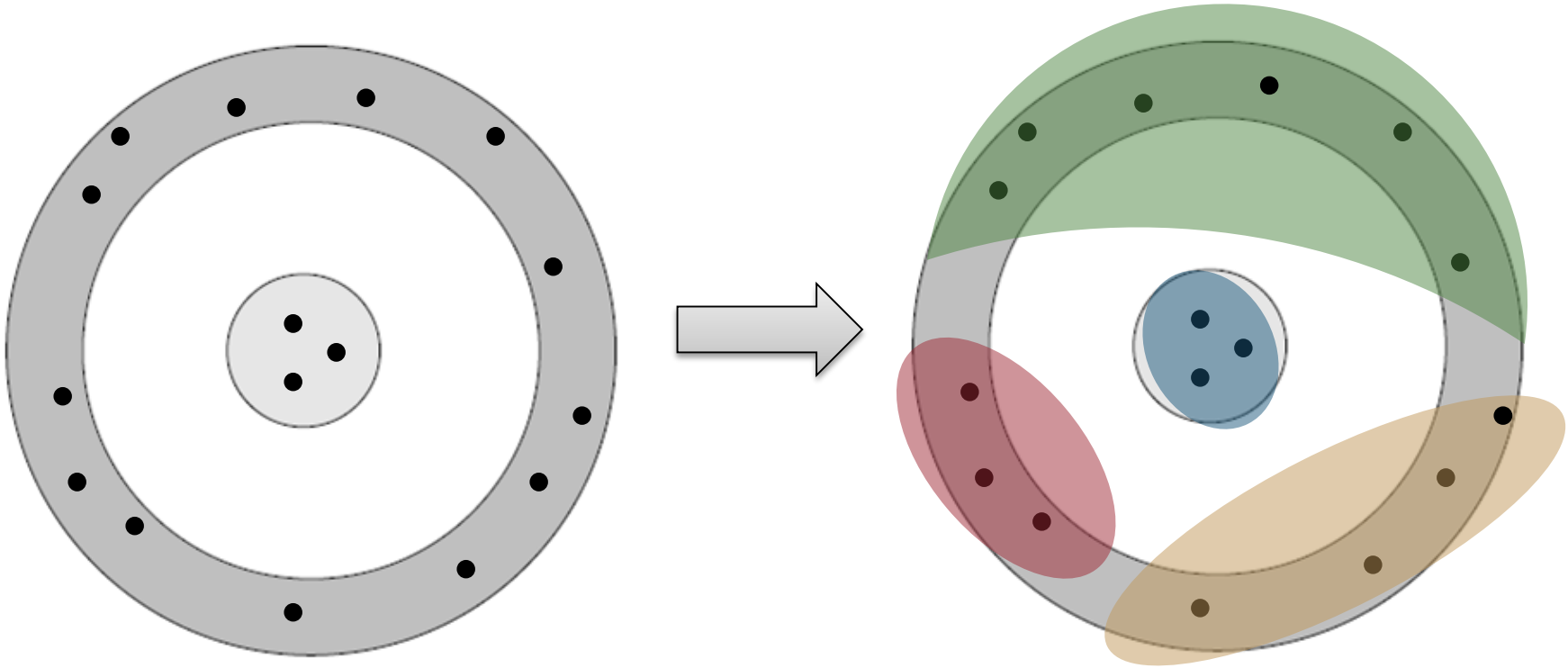
d) Verschiebung der repräsentativen Punkte um einen bestimmten Anteil (z.B. 20%) hin zu den jeweiligen Centroiden der Cluster

e) Zusammenführung zweier Cluster, deren repräsentative Punkte eine geringe maximale paarweise Distanz aufweisen (Schwellenwert)

f) Durchlauf aller Punkte und Zuordnung zu Cluster mit geringstem Abstand zu einem repräsentativen Punkt

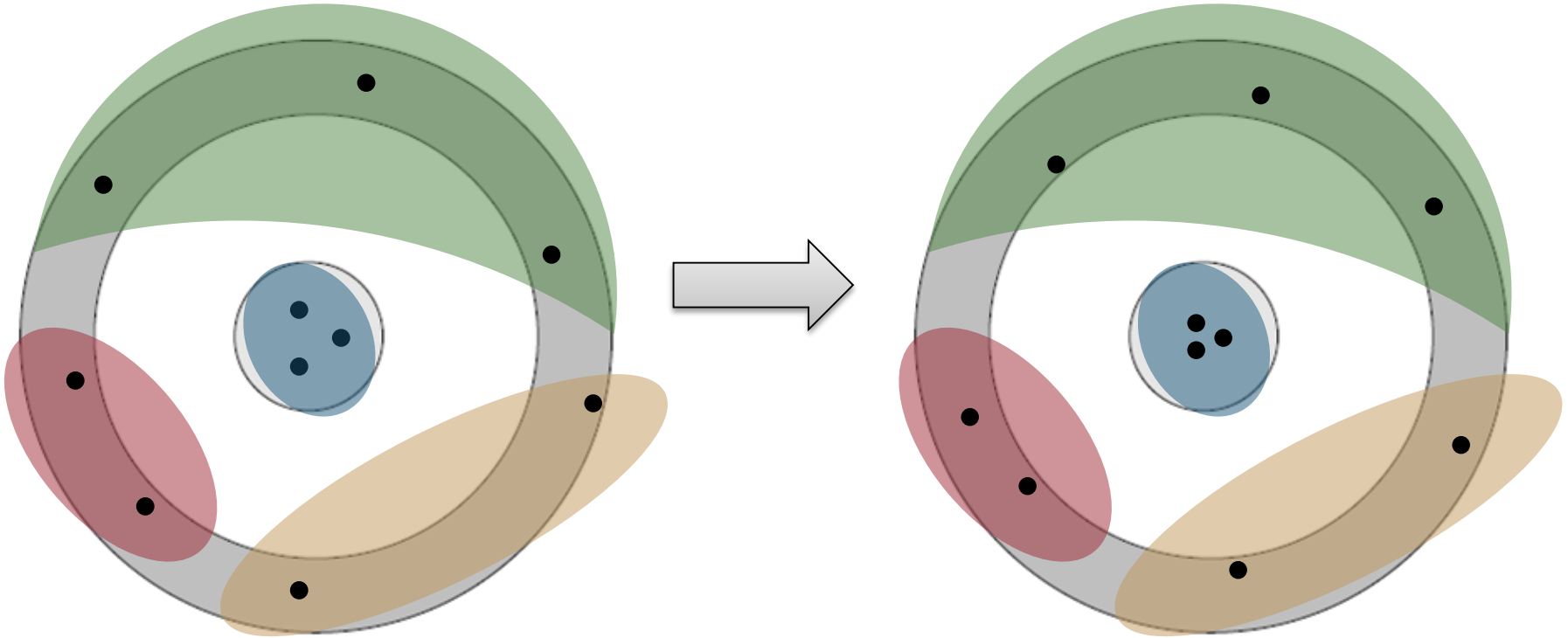
Beispiel

Ziehen einer Zufallsstichprobe & hierarchische Clusteranalyse



Beispiel

Auswahl repräsentativer Punkte & Verschiebung zu Centroiden



Beispiel

Zusammenführung zweier Cluster, deren repräsentative Punkte eine geringe Distanz aufweisen