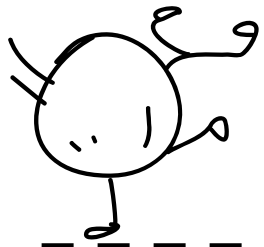


Clustering

Vorlesung 186.844

10.12.2015

TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

Überblick

I. Was ist nicht überwachtes Lernen?
II. Verwendung von nicht überwachtem Lernen
III. Grundlagen Clustering
IV. Distanzmaß
V. Partitionierungsverfahren
VI. Hierarchische Verfahren
VII. Graph-basierte Verfahren

I. Was ist nicht überwachtes Lernen?

Unterschied zwischen …

Überwachtem Lernen (Supervised Learning)

- Trainingsdaten haben **Klassenlabel**
- Anzahl der Klassen ist **bekannt**

Nicht überwachtem Lernen (Unsupervised Learning)

- Trainingsdaten haben **keine Klassenlabel**
- Anzahl der Klassen ist **nicht immer bekannt**

Grundidee

Wie werden Klassen (Gruppierungen, Cluster) festgelegt/bestimmt?

… an einem Beispiel aus dem Tierreich

Drei Merkmale

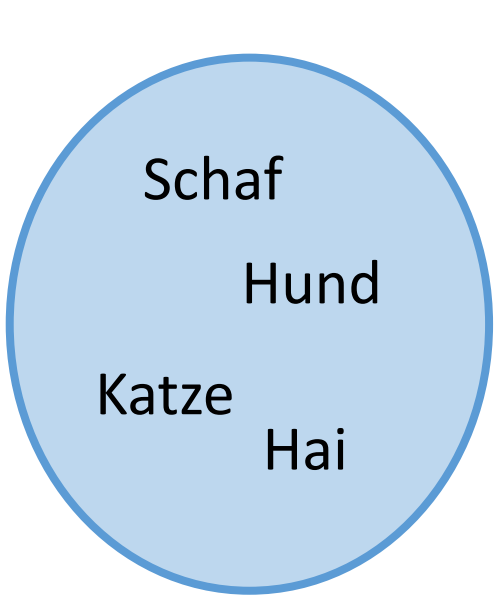
- Fortpflanzung
- Atmung
- Lebensraum

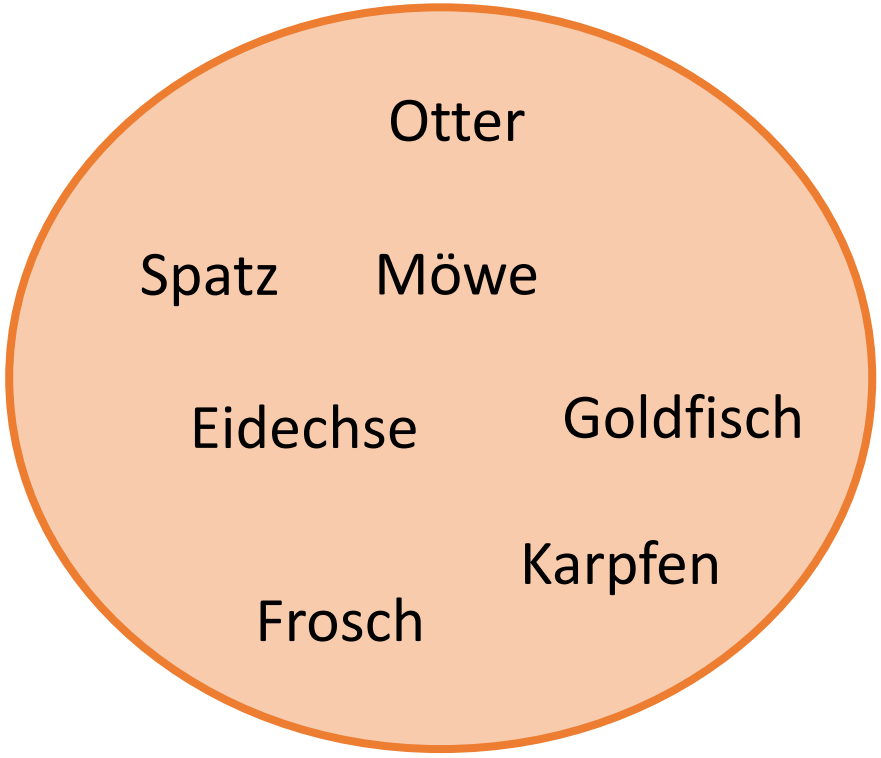
Elf Tiere (Trainingsdaten)

- Säugetiere: Schaf, Hund, Katze
- Vögel: Spatz, Möwe
- Reptilien: Otter, Eidechse
- Fische: Goldfisch, Hai, Karpfen
- Amphibien: Frosch

Grundidee

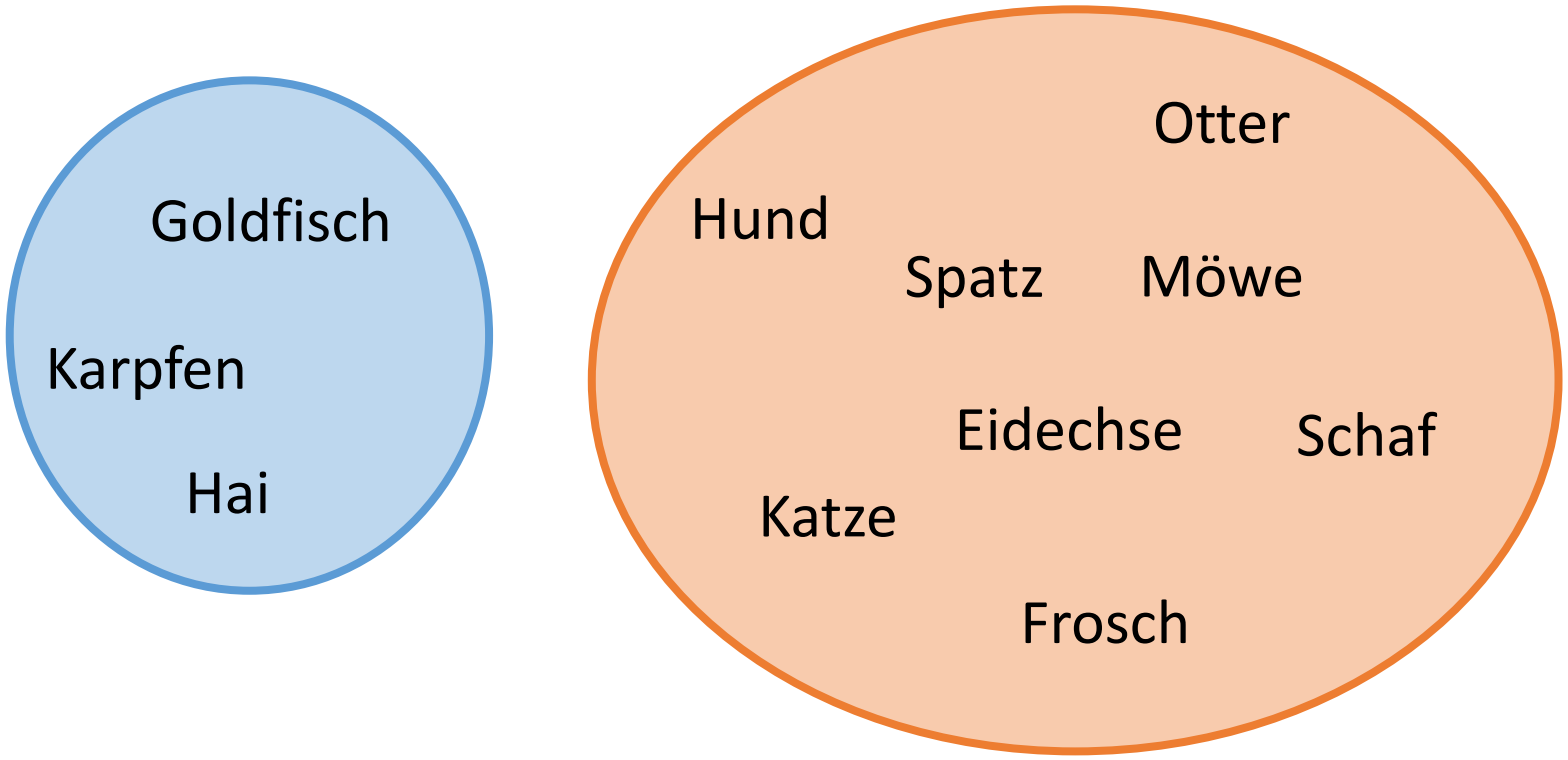
Merkmal : Art der Fortpflanzung → zwei Klassen





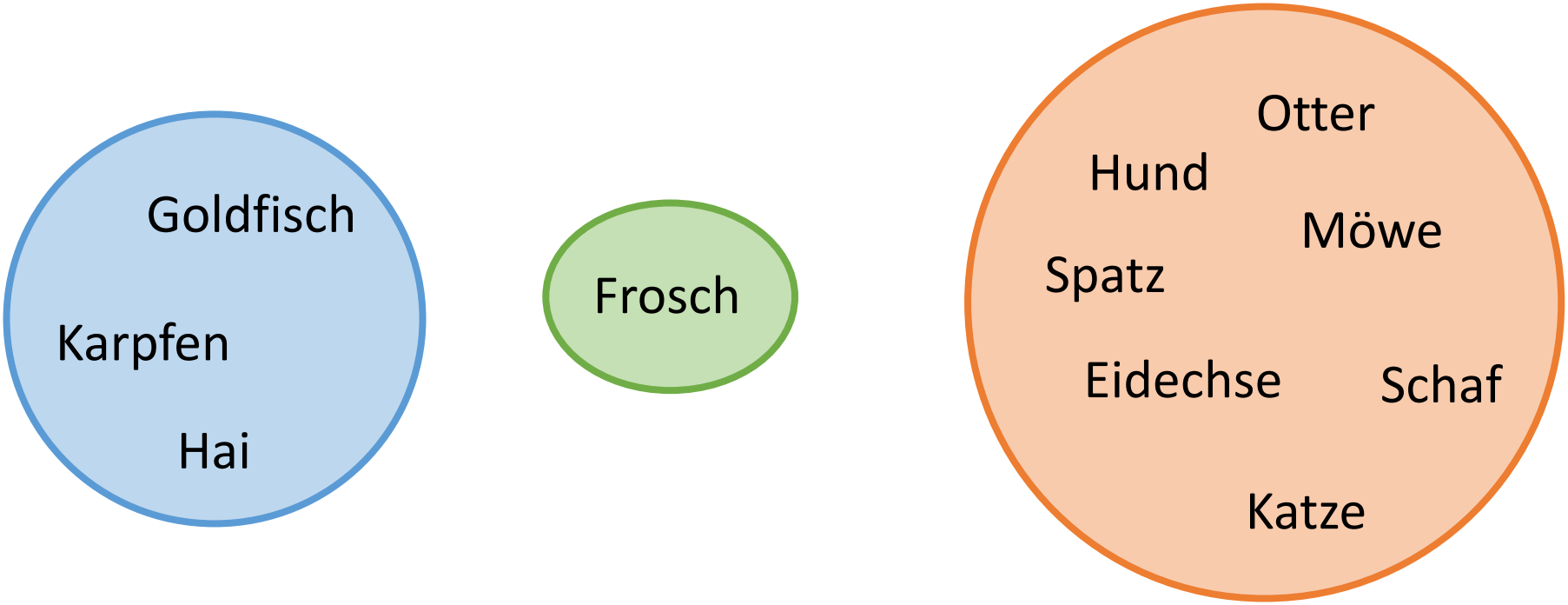
Grundidee

Merkmal : Hat das Tier eine Lunge? → zwei Klassen



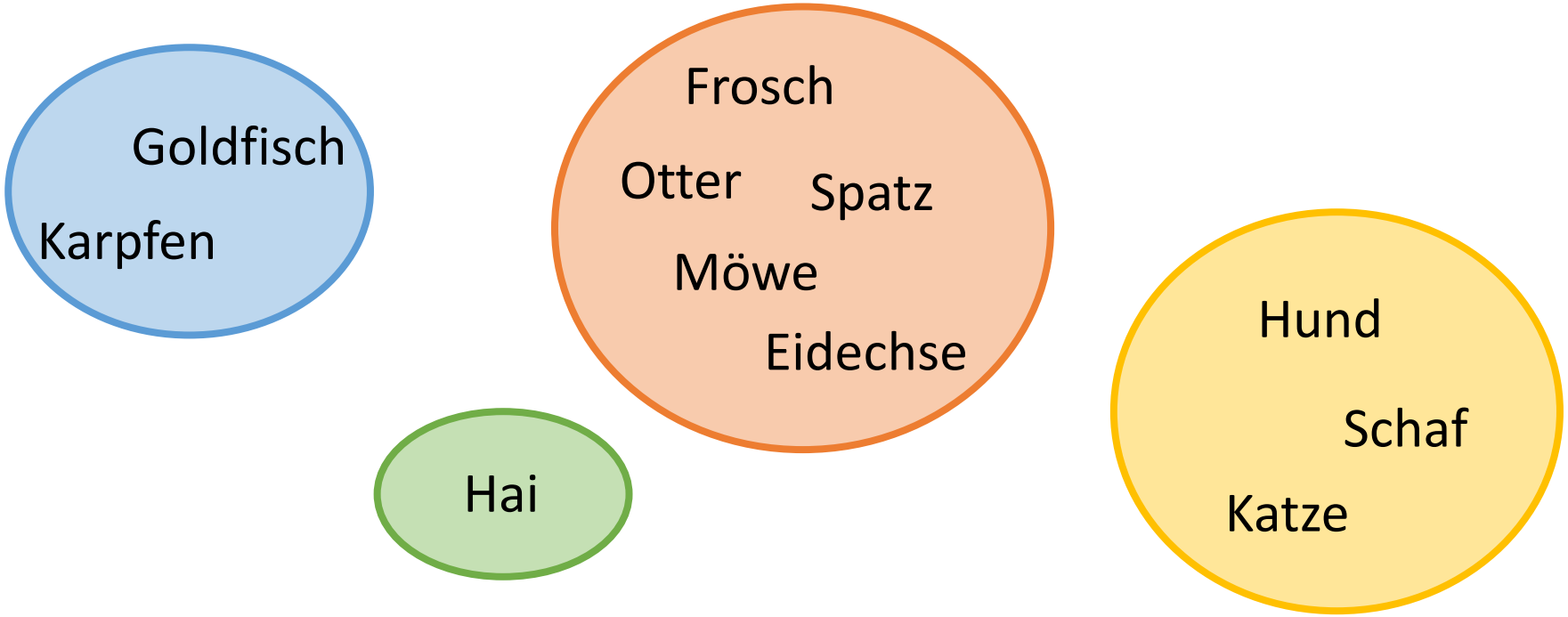
Grundidee

Merkmal: Lebensraum → drei Klassen



Grundidee

Kombination von zwei Merkmalen: Fortpflanzung und Existenz von Lungen → vier Klassen



II. Verwendung von nicht überwachtem Lernen

Ground Truth Erfassung

- kann für **große Datenmengen (Big Data)** teuer sein
- automatische Erfassung wäre wünschenswert
- Beispiele:
  - Daten im Internet (Bilder, Videos, Text, …)
  - Rundfunk (Radio, Fernsehen, …)

Klassifikator-Update

- Merkmalsausprägungen können sich über die Zeit verändern
- Klassifikator passt sich automatisch an
- Beispiele:
  - Nahrungsmittel (Saison, Reifungsprozess)
  - Personen (Wachstum, Alter, Gewichtsschwankungen)

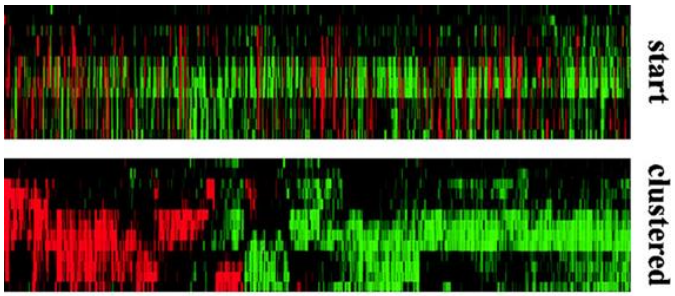
Data-Mining

Data-Mining bedeutet sinngemäß „in einem Datenberg nach wertvollem Wissen suchen“ [http://de.wikipedia.org/wiki/Data-Mining]

- automatische oder semi-automatische Erforschung und Analyse von großen Datenmengen
- Ziel:
  - bedeutsame Muster oder Regeln finden
  - Gruppen oder Trends identifizieren

Anwendung

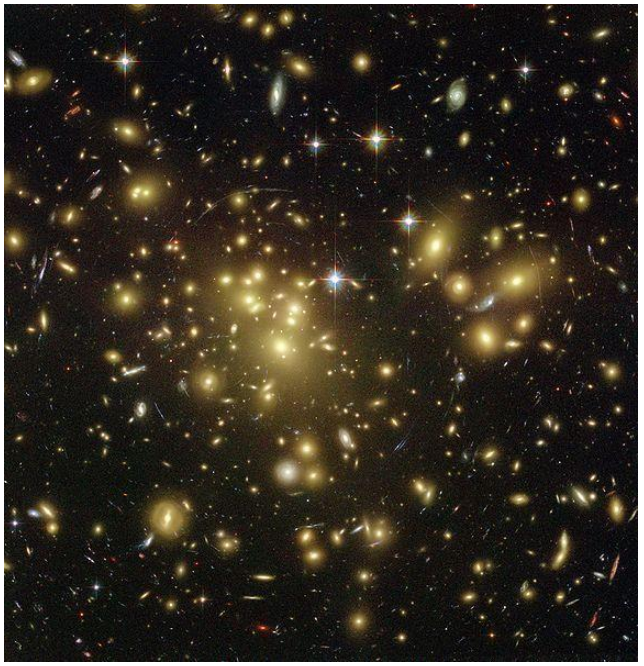
- Datenanalyse in der Wissenschaft
  - Gruppierung von Genen und Proteinen
  - Beispiel: Bierhefe



- Marketing
  - Produktanalyse: Gruppierung von Produkten mit ähnlichen Eigenschaften
  - Warenkorb-Analyse: Gruppierung von Käufern mit ähnlichem Kaufverhalten
- Datenbanken
  - Dokumentenklassifikation: thematische Gruppen

Anwendung: Astronomie

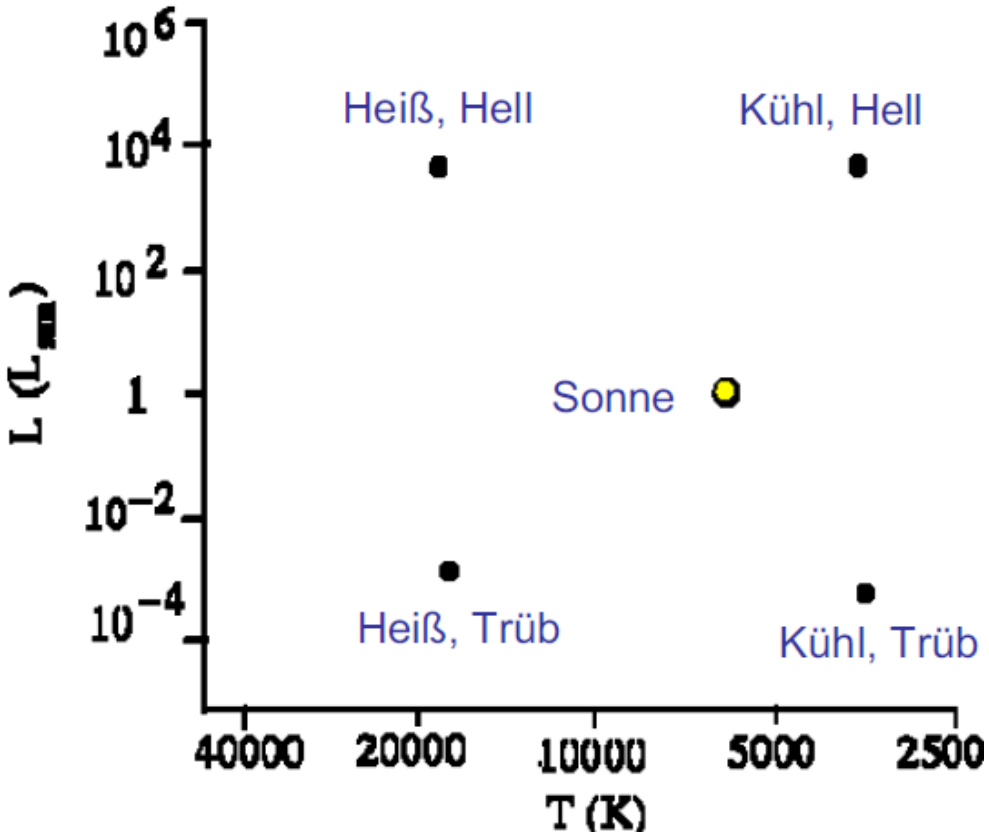
- Eigenschaften von Sternen werden von Astronomen gemessen oder von Messungen berechnet
  - Abstand von der Erde / Sonne
  - Helligkeit
  - Spektrum
  - Temperatur
  - Chemische Zusammensetzung
  - etc.



http://www.mpe.mpg.de

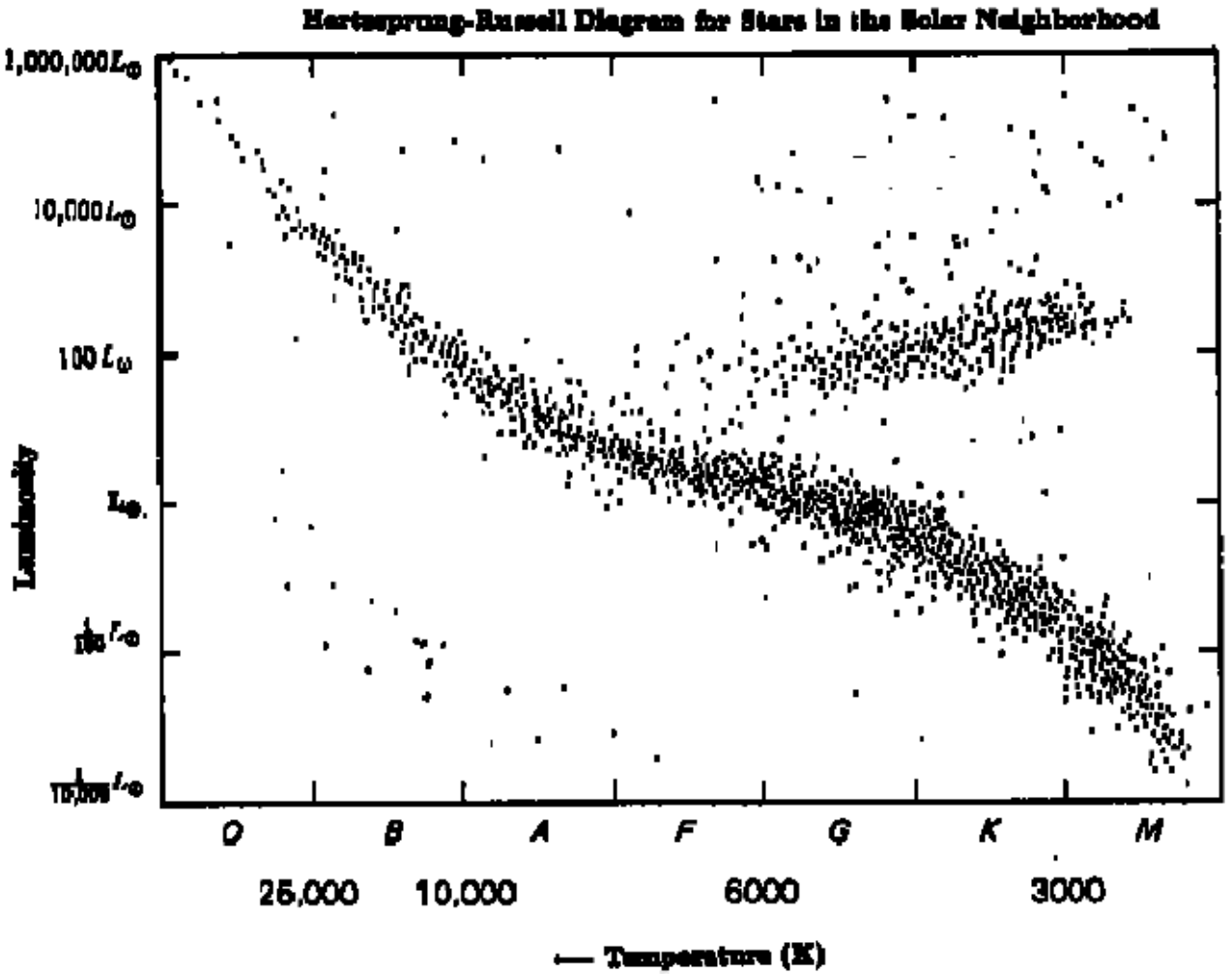
Anwendung: Astronomie

- Hertzsprung (Leiden, 1911) und Russell (Princeton, 1913) hatten die Idee Sterne anhand ihrer Helligkeit und Temperatur zu betrachten



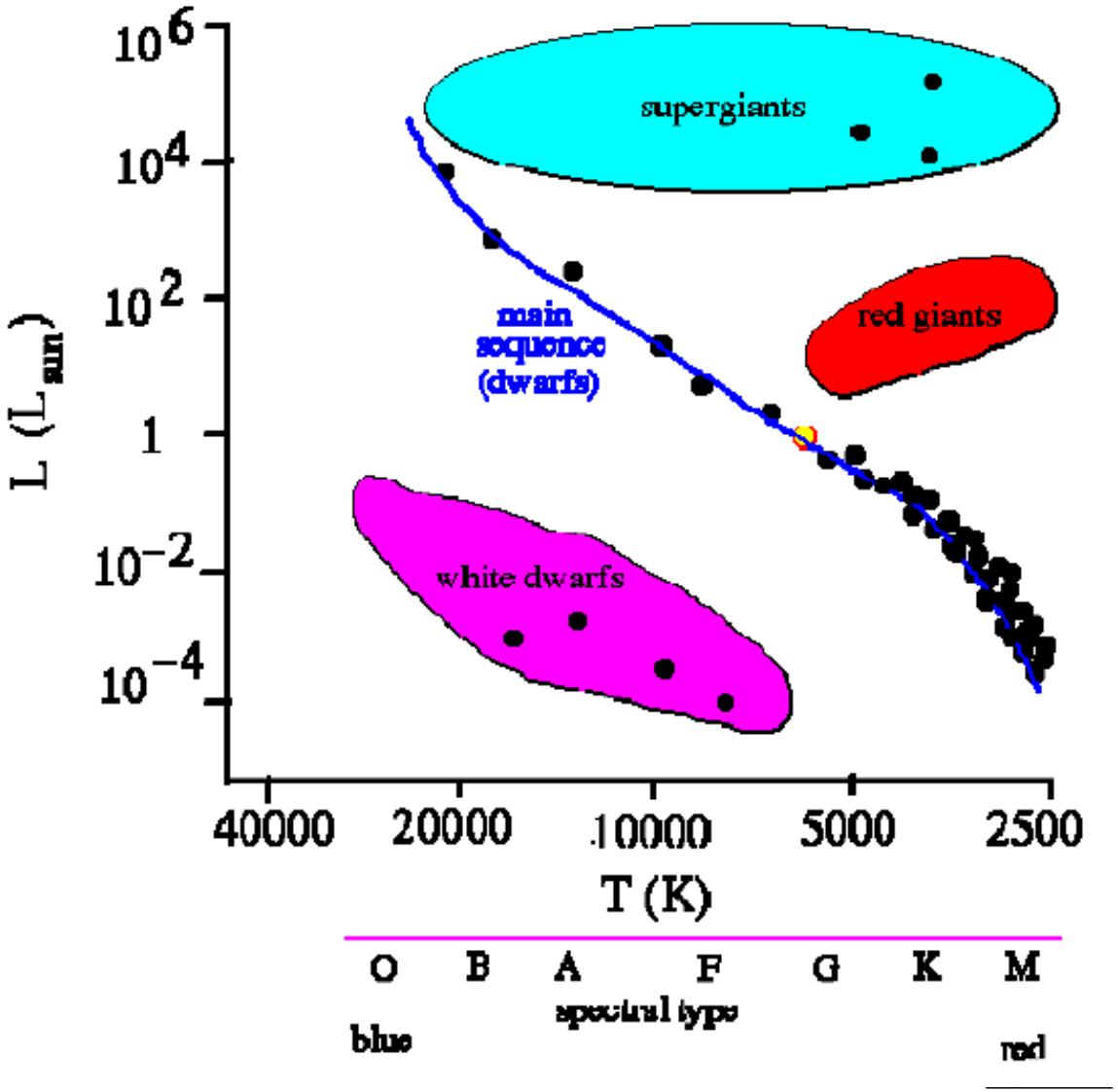
Anwendung: Astronomie

- für Sterne in der Nähe der Sonne sieht die Datenverteilung wie folgt aus:



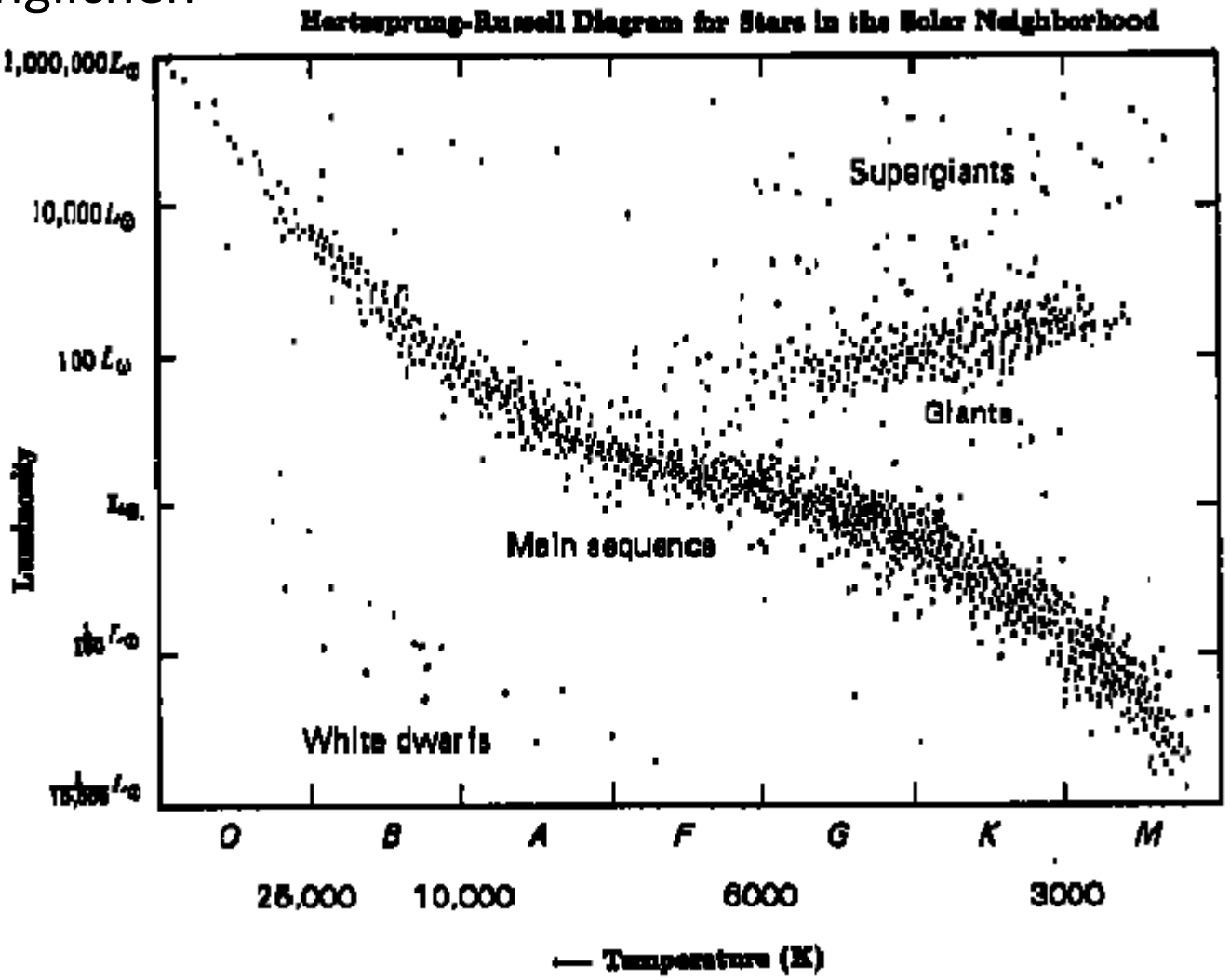
Anwendung: Astronomie

vier Cluster …



Anwendung: Astronomie

auf dem ursprünglichen Diagramm …



III. Grundlagen Clustering

Clustering

Clustering ist ein **nicht überwachtes Lernverfahren** oder Trainingsverfahren, um (Trainings-)**Daten in Klassen** oder sogenannte Gruppierungen (Cluster) zu **unterteilen**.

- für Menschen ist Clustering und Klassifizierung anhand von Eigenschaften einfach, z.B. Ente:





Definition Clustering I

Man bezeichnet als m-Clustering von $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$ die **Unterteilung von $\boldsymbol{X}$ in $\boldsymbol{m}$ Cluster** $C_1, \dots, C_m$, sodass gilt

- $C_i \neq \emptyset, i = 1, \dots, m$
- $\cup_{i=1}^{m} C_i = X$
- $C_i \cap C_j = \emptyset, \mathrm{i} \neq j, i, j = 1, \dots, m$

Die Vektoren $\boldsymbol{x}_i$ von Cluster C_i sind ähnlicher zueinander als zu den Vektoren in den anderen Cluster. Diese Art von Clustering bezeichnet man als „hard“ oder „crisp“.

Definition Clustering II

… eine Alternative

Ein **Fuzzy-Clustering** von X in m Cluster wird durch m **Mitglieds-funktionen** (membership functions) u_j beschrieben, wobei

$$u_j : X \to [0,1], j = 1, \dots, m$$

und

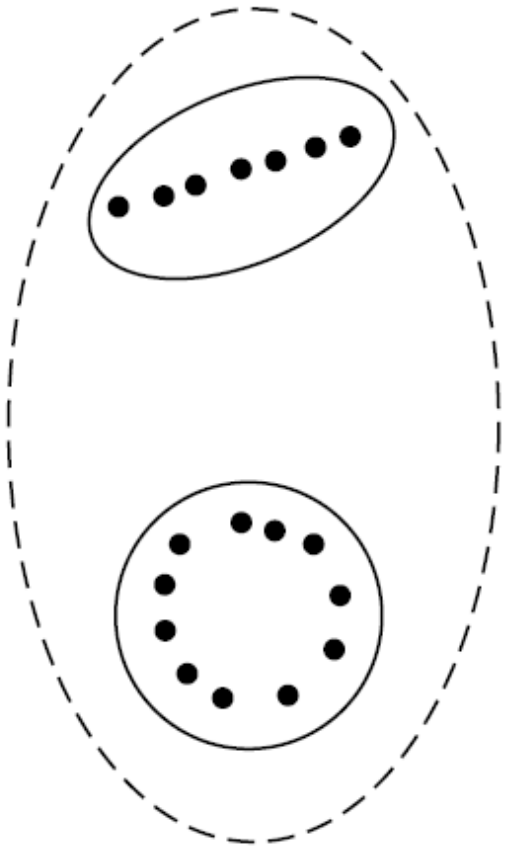
$$\sum_{j=1}^{m} u_j(\boldsymbol{x}_i) = 1, i = 1, \dots, N, \quad 0 < \sum_{i=1}^{N} u_j(\boldsymbol{x}_i) < N, j = 1, \dots, m$$

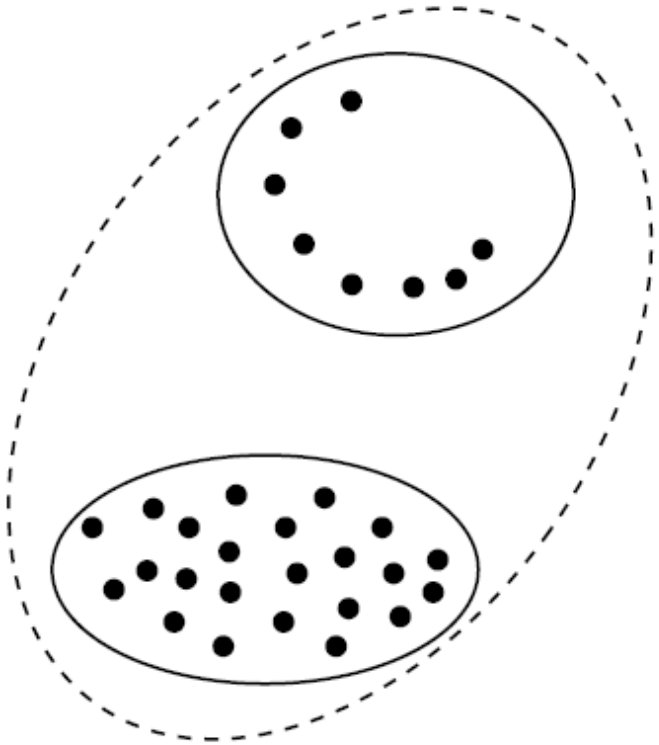
Jeder Vektor $\boldsymbol{x}_i$ gehört **gleichzeitig zu mehreren Clustern**, wobei der Wert der Mitgliedsfunktionen u_j die Zugehörigkeit quantifiziert. Werte nahe 1 deuten auf eine Mitgliedschaft hohen Grades hin.

Sinnvolle Cluster

mehr als eine Lösung: grob → zwei Cluster, fein → vier Cluster

[Quelle: Theodoris et al., 2009]





Clustering

Entwicklung eines Clustering-Verfahrens

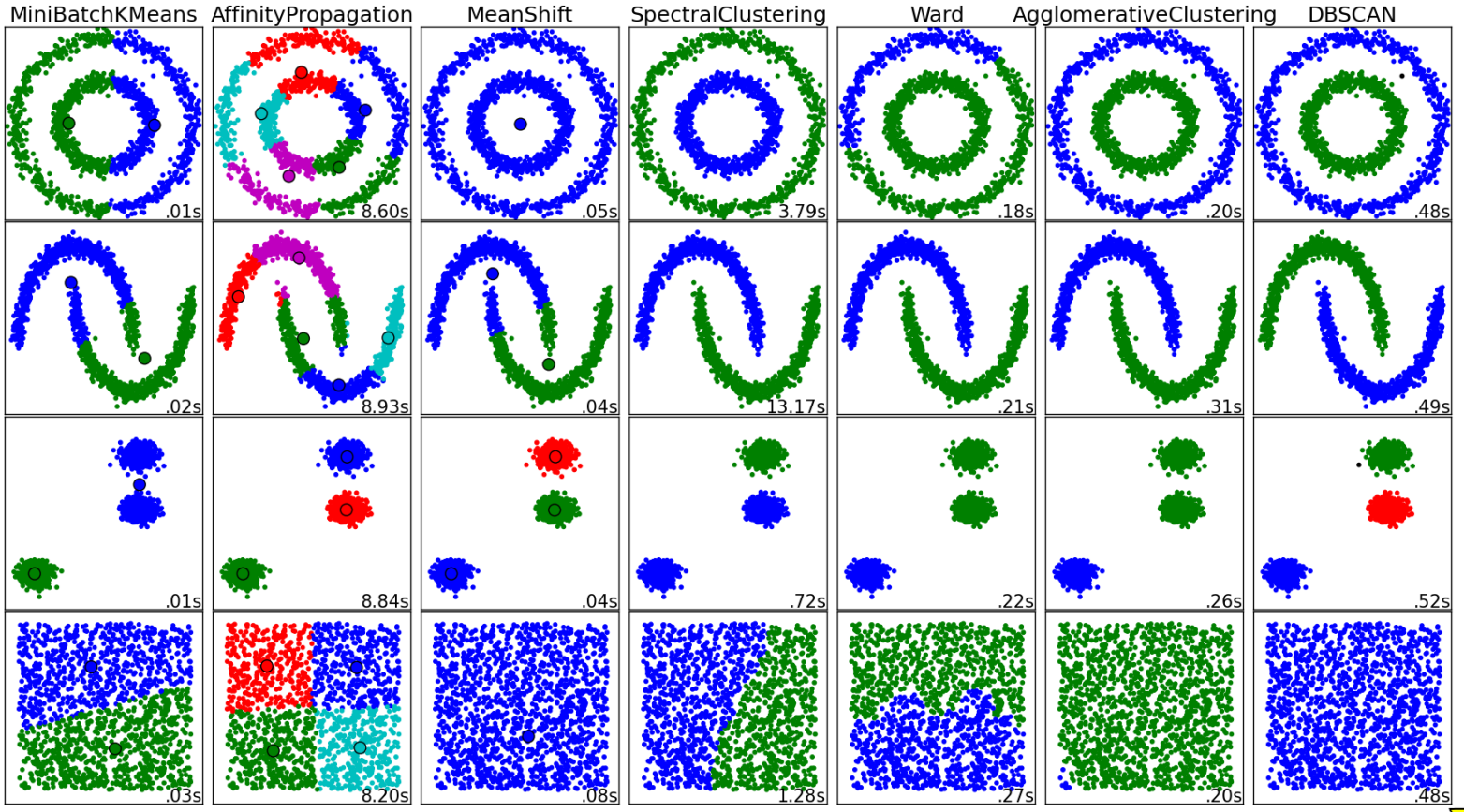
- Merkmalsauswahl und Vorverarbeitung
- **Distanzmaß**
- Clustering-Kriterium (Gruppierungskriterium)
- **Clustering-Algorithmus**
- Validierung und Interpretation der Ergebnisse

(mehr Informationen in: Pattern Recognition, Theodoris et al., 2009)

Ein Clustering-Verfahren kann, abhängig von den obigen Schritten, **sehr unterschiedlichen Ergebnisse** für die gleichen Daten liefern.

Clustering-Verfahren

- Vergleich unterschiedlicher Verfahren auf synthetischen Daten
  [Quelle: http://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html]



IV. Distanzmaß

Distanzmaß

Ein Distanzmaß **quantifiziert** wie „ähnlich“ oder „unähnlich“ zwei Merkmalsvektoren oder zwei Teilmengen von X sind.
Es ist wichtig darauf zu achten, dass alle ausgewählten Merkmale den **gleichen Einfluss** auf das Distanzmaß haben.

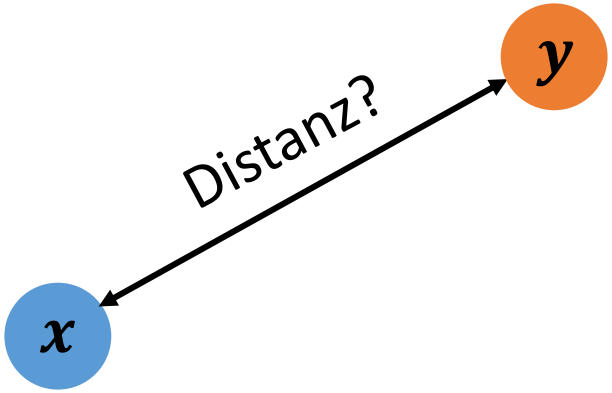
Ein **Unähnlichkeitsmaß** (dissimilarity) d auf X ist eine Funktion $d : X \times X \to \mathbb{R}$, wobei $\exists d_0 \in \mathbb{R} : -\infty < d_0 \leq d(\boldsymbol{x}, \boldsymbol{y}) < +\infty$, $\forall \boldsymbol{x}, \boldsymbol{y} \in X$. d_0 ist das kleinste Unähnlichkeitsmaß.

Ein **Ähnlichkeitsmaß** (similarity) s auf X ist eine Funktion $s : X \times X \to \mathbb{R}$, wobei $\exists s_0 \in \mathbb{R} : -\infty < s(\boldsymbol{x}, \boldsymbol{y}) \leq s_0 < +\infty, \forall \boldsymbol{x}, \boldsymbol{y} \in X$. s_0 ist das größte Ähnlichkeitsmaß.

Distanz zwischen

.. zwei (Merkmals-)**Vektoren** oder Punkten

Die Distanz zwischen Vektor $\boldsymbol{x}$ und $\boldsymbol{y}$ kann durch Unähnlichkeits- oder Ähnlichkeitsmaße bestimmt werden.



Unähnlichkeitsmaß

Unähnlichkeitsmaß für Vektoren: Gewichtete l_p Metrik

$$d_p(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_{i=1}^{l} w_i |x_i - y_i|^p \right)^{1/p}$$

wobei x_i, y_i die $i$te Dimension von $\boldsymbol{x}$ und $\boldsymbol{y}$ ist, $i = 1, \dots, l$ und $w_i \geq 0$ der $i$te Gewichtskoeffizient.

- z.B.: euklidische Distanz, l_2 Metrik: $p = 2, w_i = 1$

Ähnlichkeitsmaß

Häufig verwendete Ähnlichkeitsmaße s für Vektoren:

Skalarprodukt (inneres Produkt)

$$s_{skalar}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T \boldsymbol{y} = \sum_{i=1}^{l} x_i y_i$$

Pearsons Korrelationskoeffizient

$$s_{pearson}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x}_d^T \boldsymbol{y}_d}{\|\boldsymbol{x}_d\| \|\boldsymbol{y}_d\|}$$

wobei $\boldsymbol{x}_d = [x_1 - \bar{x}, \dots, x_l - \bar{x}]^T$ und $\boldsymbol{y}_d = [y_1 - \bar{y}, \dots, y_l - \bar{y}]^T$
mit $\bar{x} = \frac{1}{l} \sum_{i=1}^{l} x_i$ und $\bar{y} = \frac{1}{l} \sum_{i=1}^{l} y_i$.

Distanz zwischen

… einem **Vektor** und einer **Teilmenge** von X

Solche Distanzmaße sind notwendig, um zu entscheiden, ob man einen **Vektor (Punkt) einer Teilmenge (Cluster) C zuweist**.

Mögliche Distanzfunktionen zwischen Vektoren und Cluster:

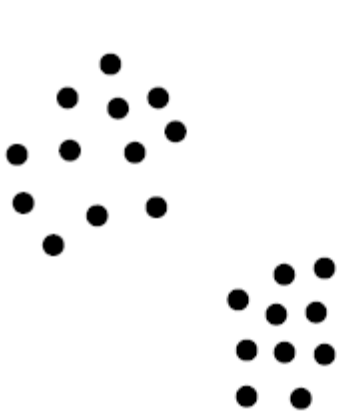
- **minimale** Unähnlichkeit $d(\boldsymbol{x}, C) = \min_{\boldsymbol{y} \in C} d(\boldsymbol{x}, \boldsymbol{y})$
- **maximale** Ähnlichkeit $\mathrm{s}(\boldsymbol{x}, C) = \max_{\boldsymbol{y} \in C} s(\boldsymbol{x}, \boldsymbol{y})$
- **durchschnittliche** Unähnlichkeit oder Ähnlichkeit

$$\bar{d}(\boldsymbol{x}, C) = \frac{1}{n_C} \sum_{i=1}^{n_C} d(\boldsymbol{x}, \boldsymbol{y_i}) \,, \bar{s}(\boldsymbol{x}, C) = \frac{1}{n_C} \sum_{i=1}^{n_C} s(\boldsymbol{x}, \boldsymbol{y_i})$$

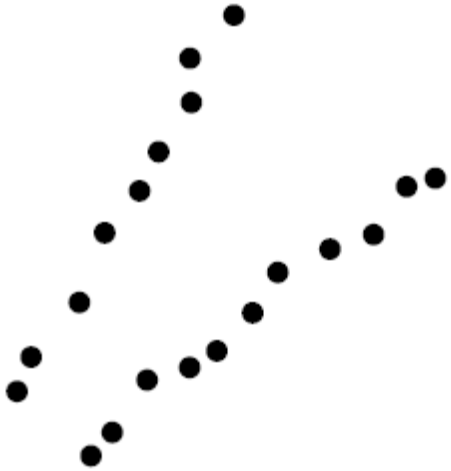
wobei n_C die Kardinalität von C ist.

Clustertypen

[Quelle: Theodoris et al., 2009]



kompakt



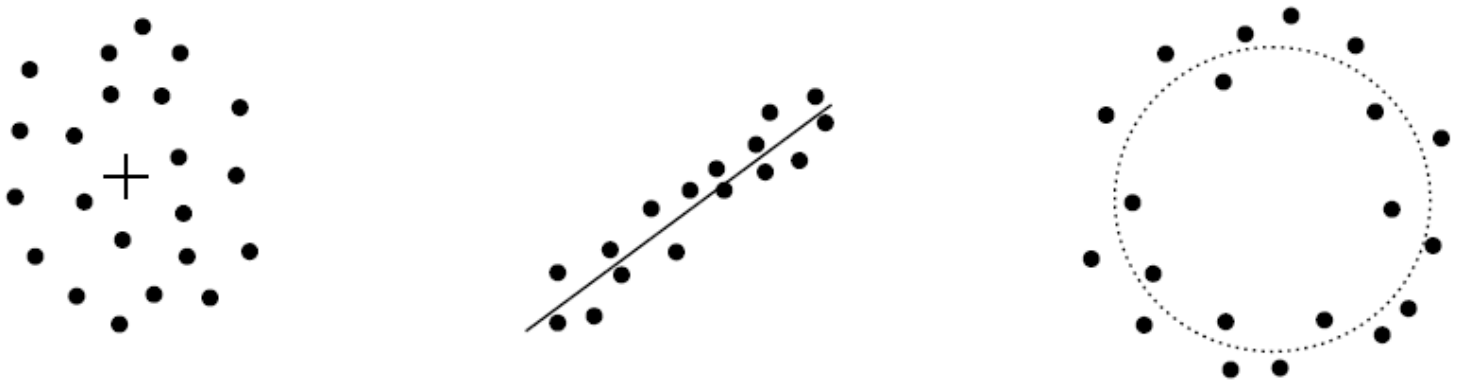
länglich

kreisförmig, elliptisch

Abhängig vom Clustertyp ist ein andere Art der Distanzbestimmung sinnvoll.

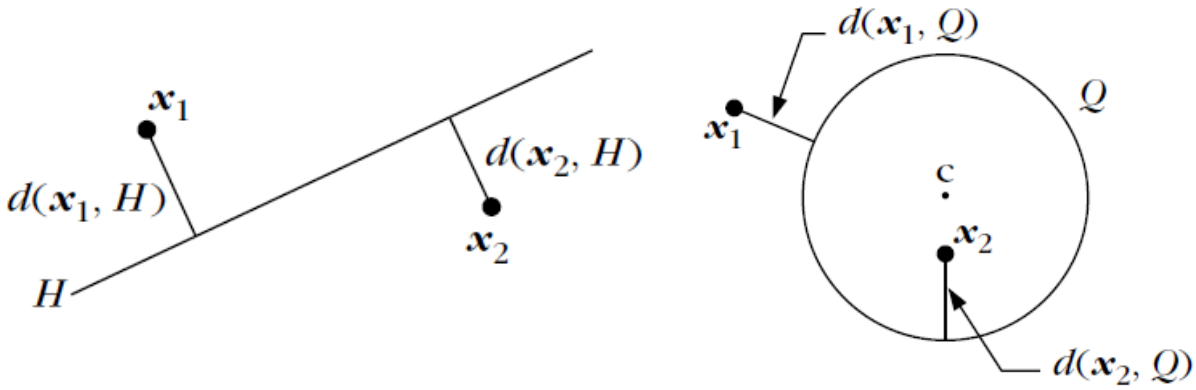
Distanz zwischen

… einem **Vektor** und einer **Repräsentation** von *C*



Repräsentationen: Punkt, Hyperebene, Hypersphäre

Distanzermittlung:



[Quelle aller Abbildungen: Theodoris et al., 2009]

Distanz zwischen

… zwei Teilmengen (Cluster) C_i, C_j

Mögliche Distanzfunktionen für zwei Cluster:

- **minimale** Unähnlichkeit $d(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(\boldsymbol{x}, \boldsymbol{y})$
- **maximale** Ähnlichkeit $s(C_i, C_j) = \max_{x \in C_i, y \in C_j} s(\boldsymbol{x}, \boldsymbol{y})$
- **durchschnittliche** Unähnlichkeit oder Ähnlichkeit

$$\bar{d}(C_i, C_j) = \frac{1}{n_{C_i} n_{C_j}} \sum_{x \in C_i} \sum_{y \in C_j} d(\boldsymbol{x}, \boldsymbol{y})$$

$$\bar{s}(C_i, C_j) = \frac{1}{n_{C_i} n_{C_j}} \sum_{x \in C_i} \sum_{y \in C_j} s(\boldsymbol{x}, \boldsymbol{y})$$

V. Partitionierungs-verfahren

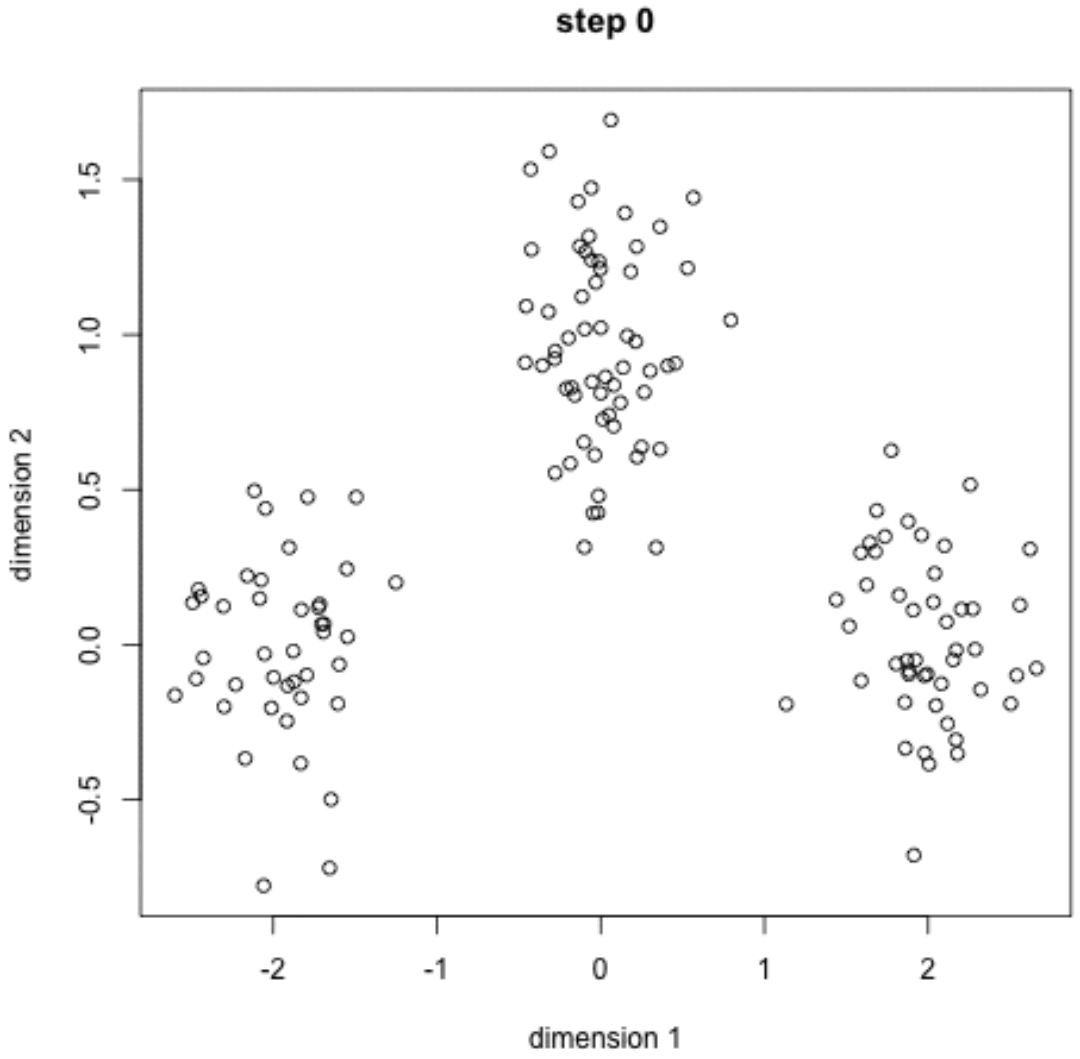
Partitionierungsverfahren

Die Grundidee von Partitionierungsverfahren ist es eine fix vorgegeben Anzahl von Cluster solange zu optimieren bis:

- die Punkte (Daten, Vektoren) innerhalb jedes Clusters möglichst homogen und
- Punkte aus unterschiedlichen Cluster möglichst heterogen sind.

- **k-Means** ist der am häufigsten verwendete Algorithmus zur Partitionierung
  - **Distanz** zwischen einem Punkt und einer Repräsentation
  - **Repräsentation**: Mittelpunkt des Clusters

DEMO



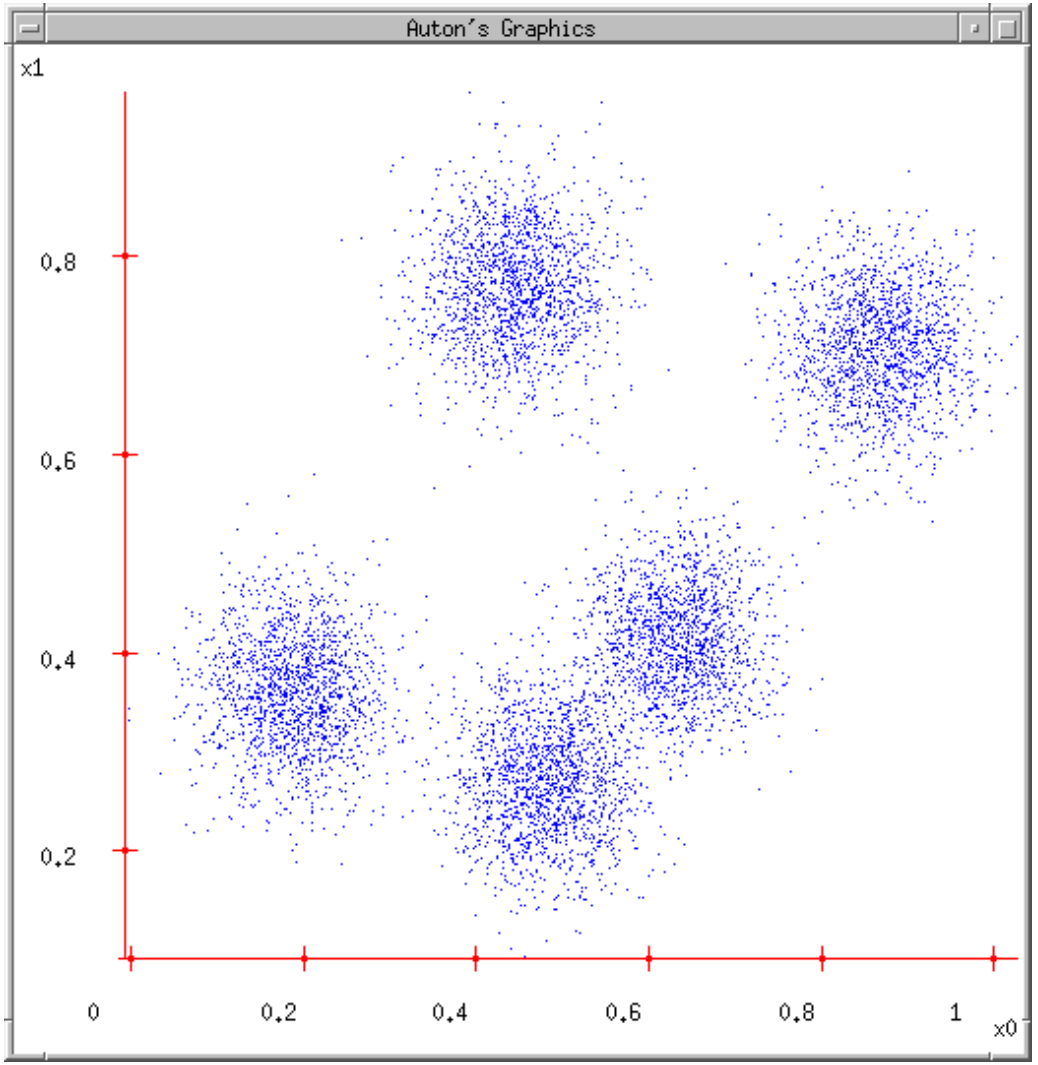
Quelle:
http://simplystatistics.org/2014/02/18/k-means-clustering-in-a-gif/

k-Means Algorithmus

Erklärung anhand eines 2D-Beispiels

von Andrew W. Moore

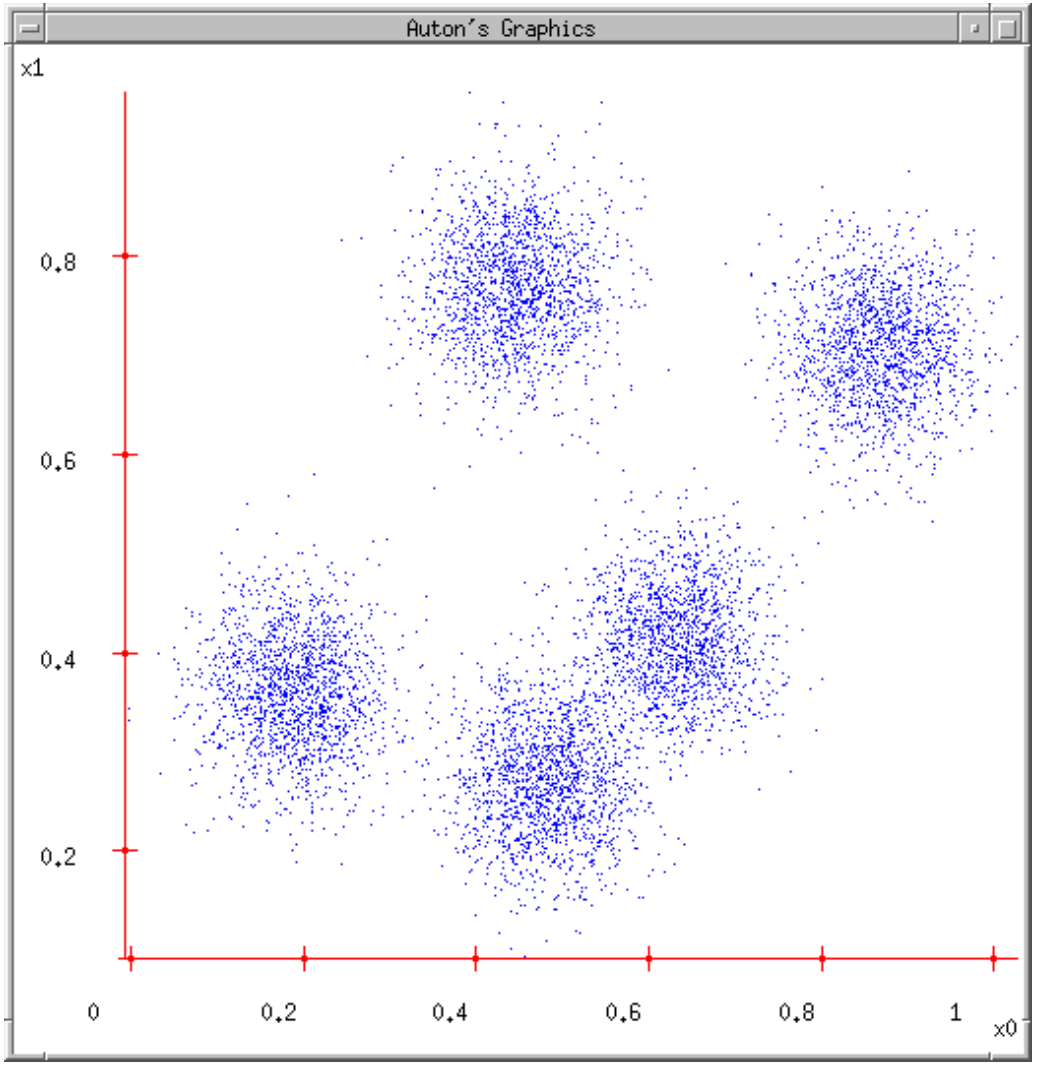
http://www.cs.cmu.edu/~awm/tutorials



k-Means Algorithmus

1. Schritt

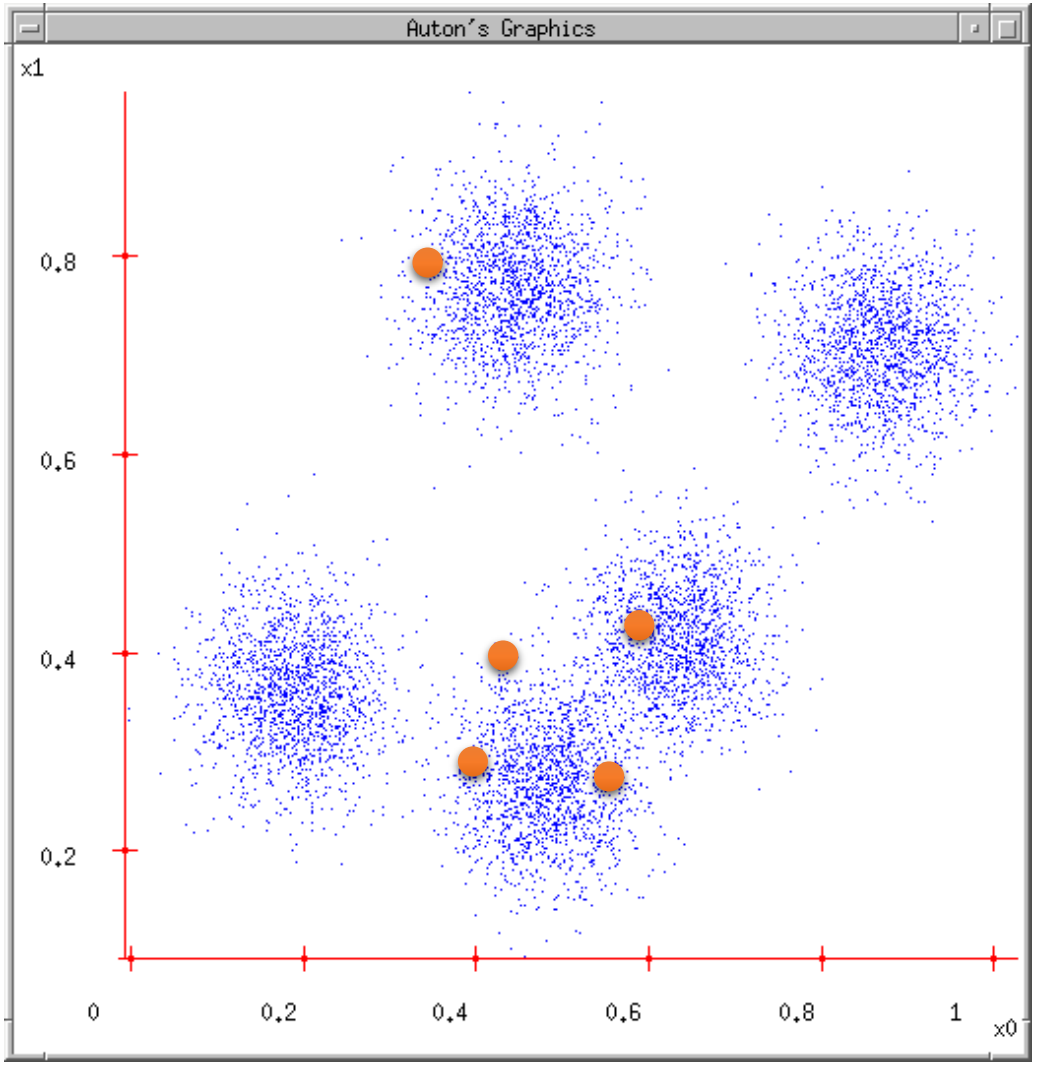
Benutzer wählt die Anzahl der Cluster → $k = 5$



k-Means Algorithmus

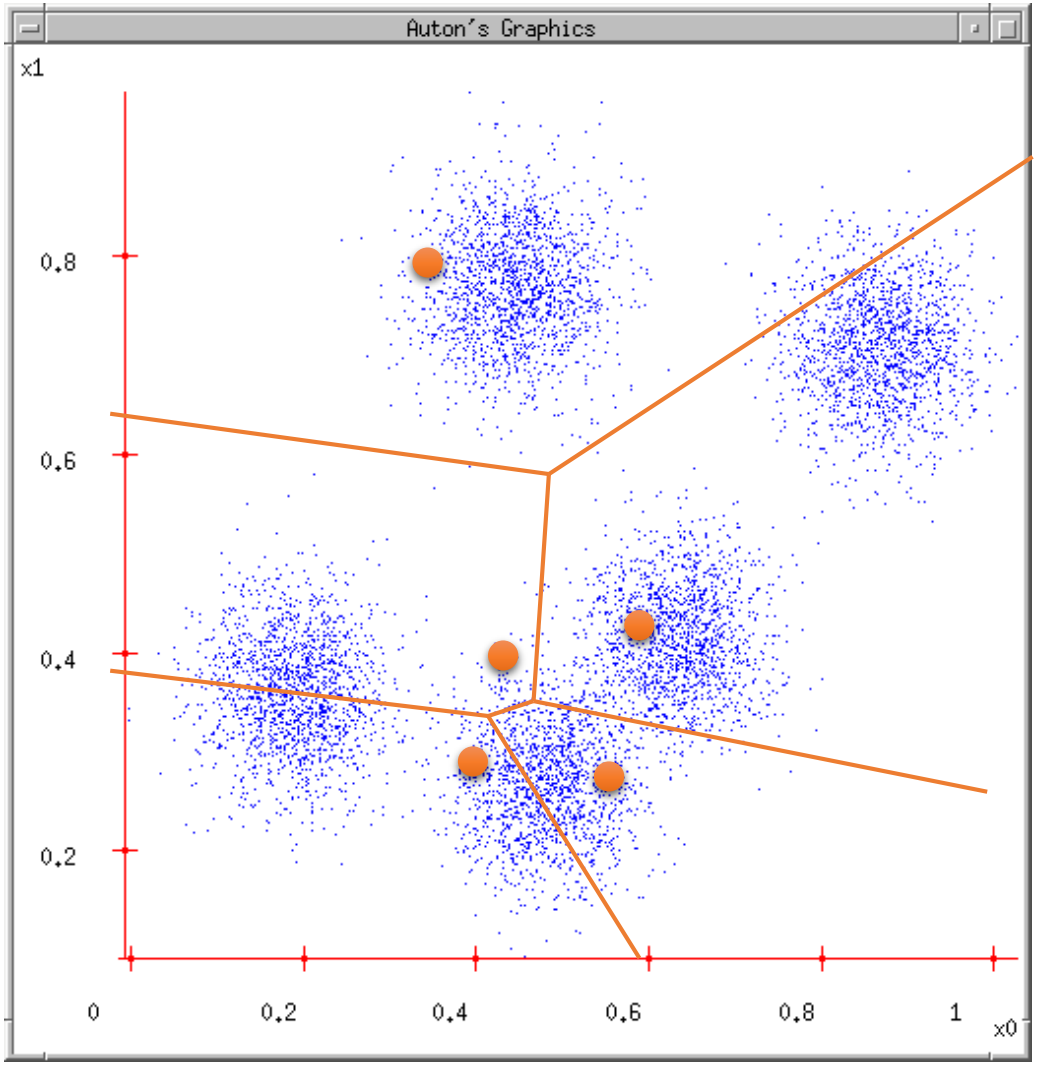
2. Schritt

Positionen von k Cluster-Mittelpunkten werden zufällig gewählt.



k-Means Algorithmus

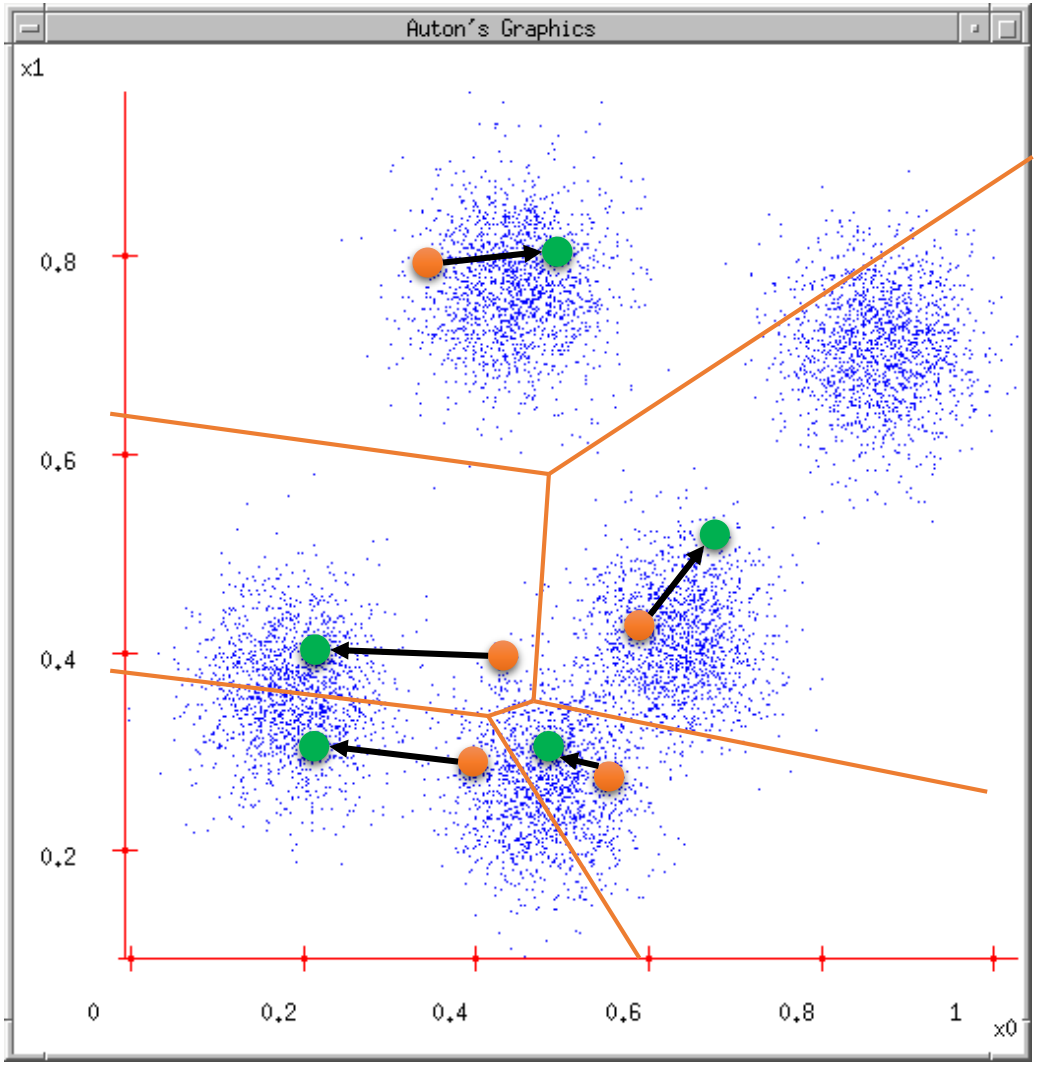
3. Schritt
Daten werden dem nächsten Cluster-Mittelpunkt zugeordnet



k-Means Algorithmus

4. Schritt

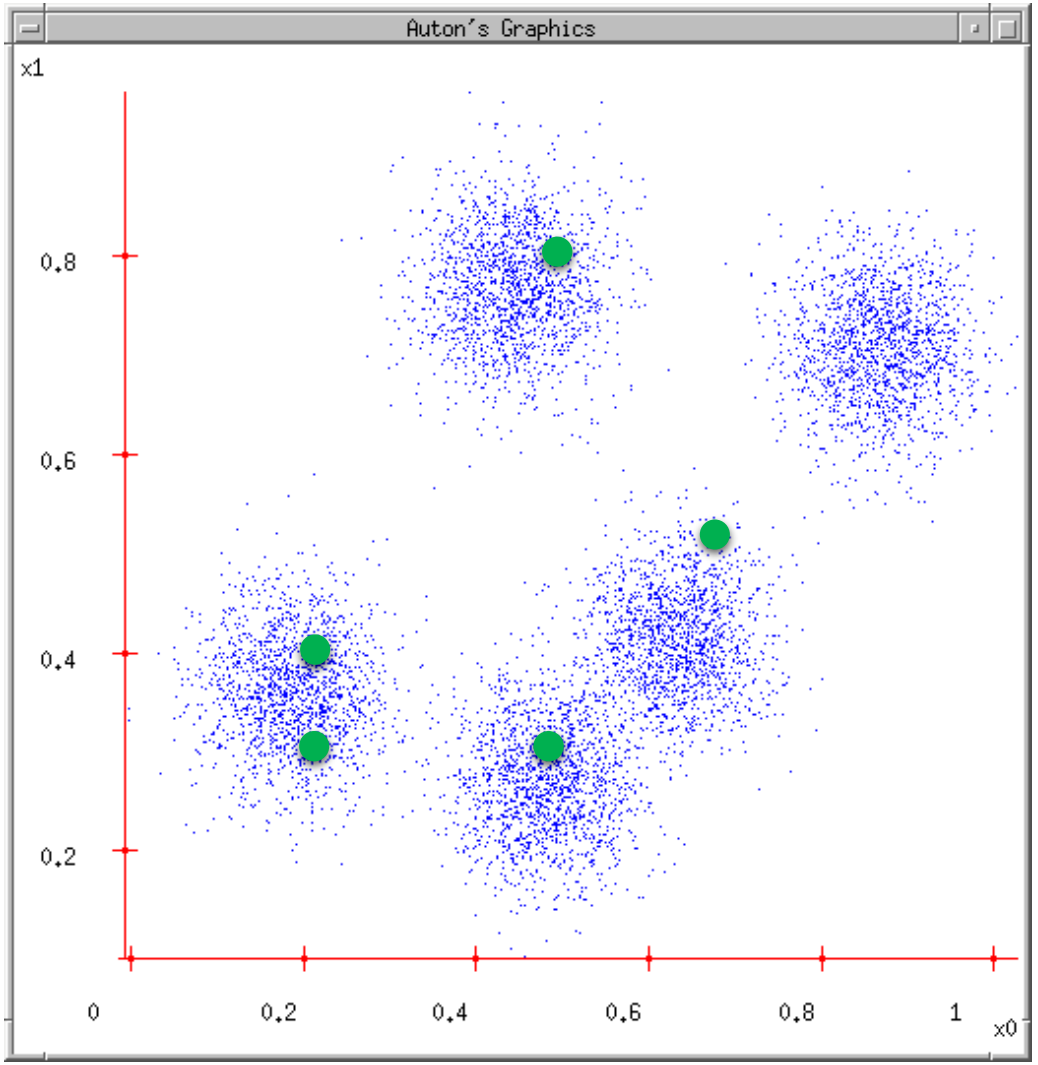
Innerhalb jedes Clusters wird der Schwerpunkt der Daten ermittelt



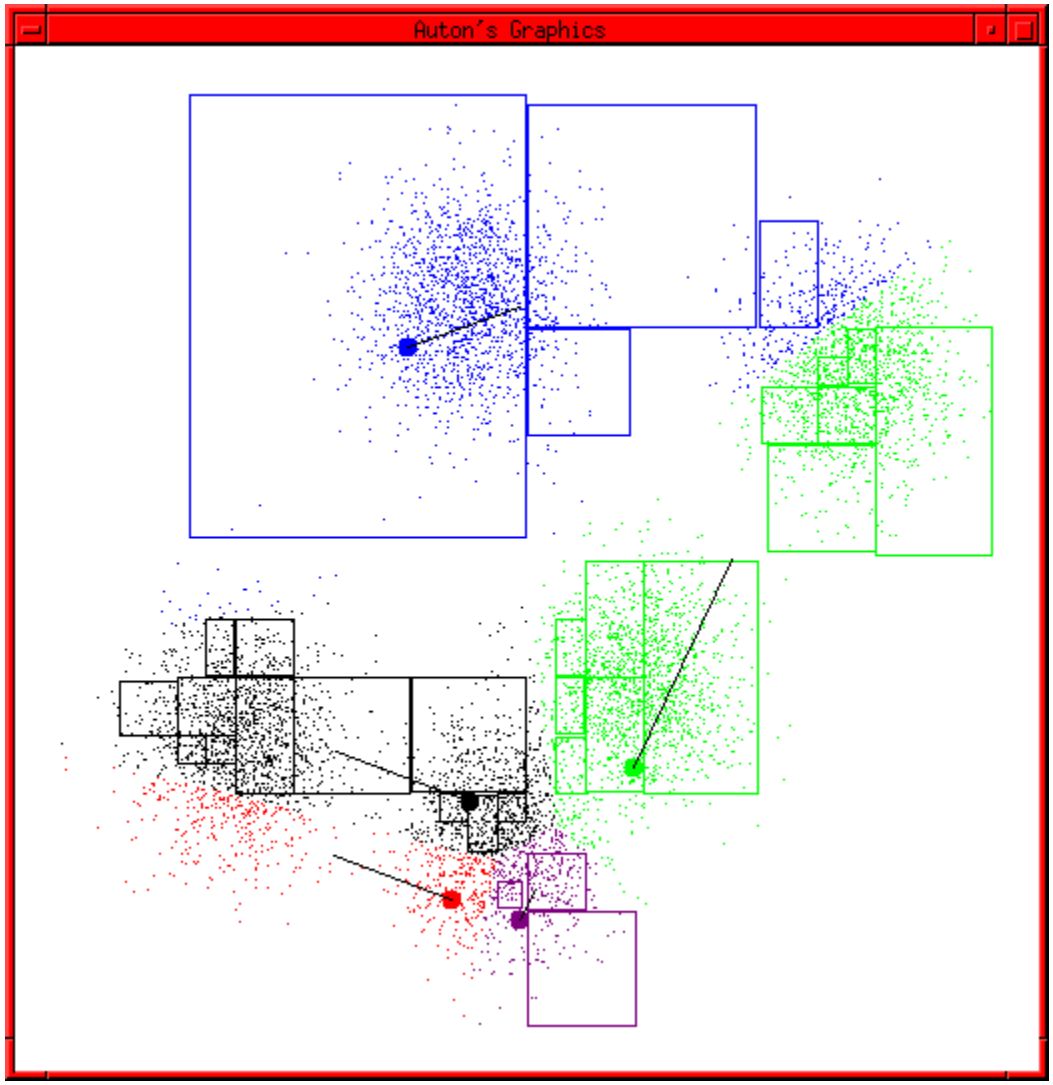
k-Means Algorithmus

5. Schritt
Schwerpunkte der Cluster
→ neue Mittelpunkte
(neue Repräsentation)

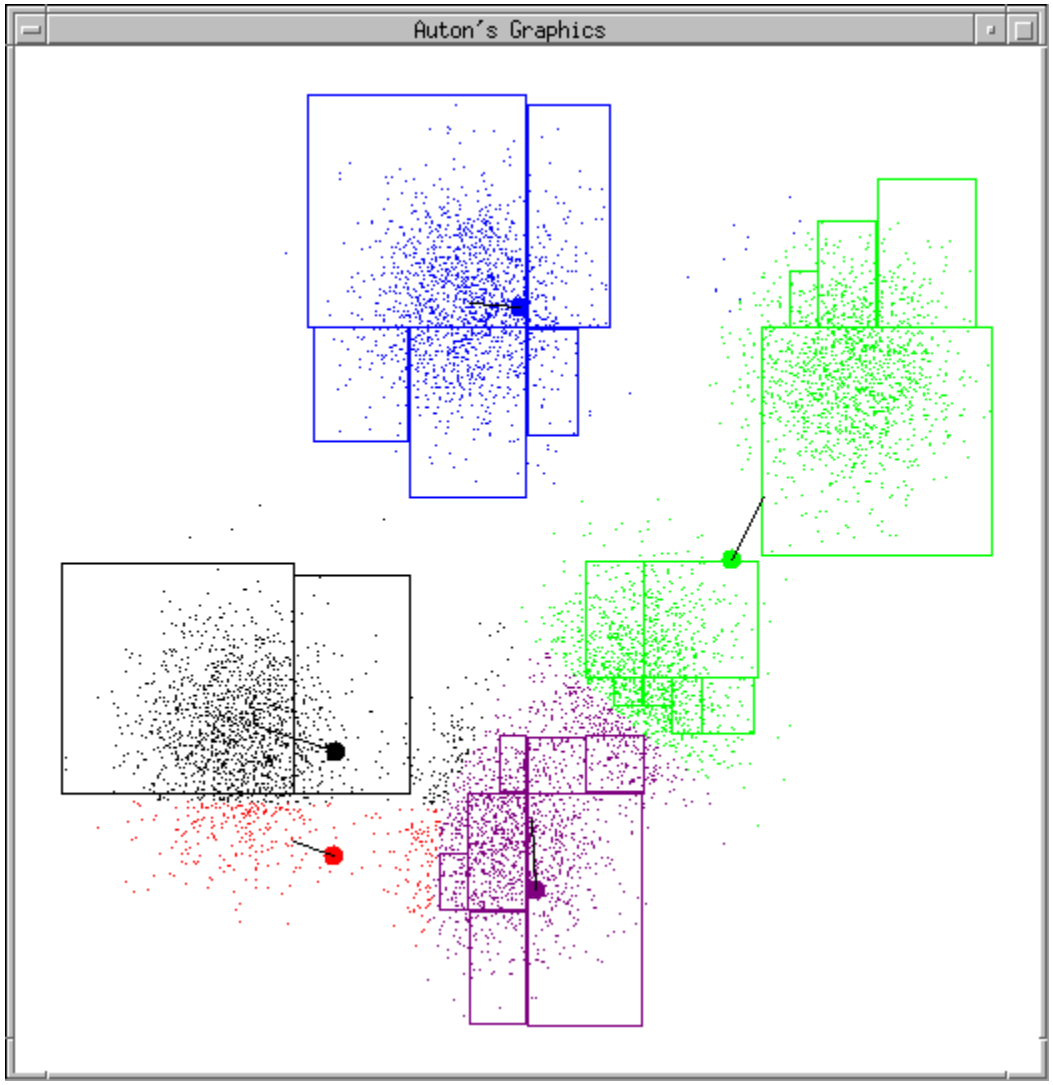
Schritte 3 – 5 werden solange wiederholt bis sich die Mittelpunkte nicht mehr verändern.



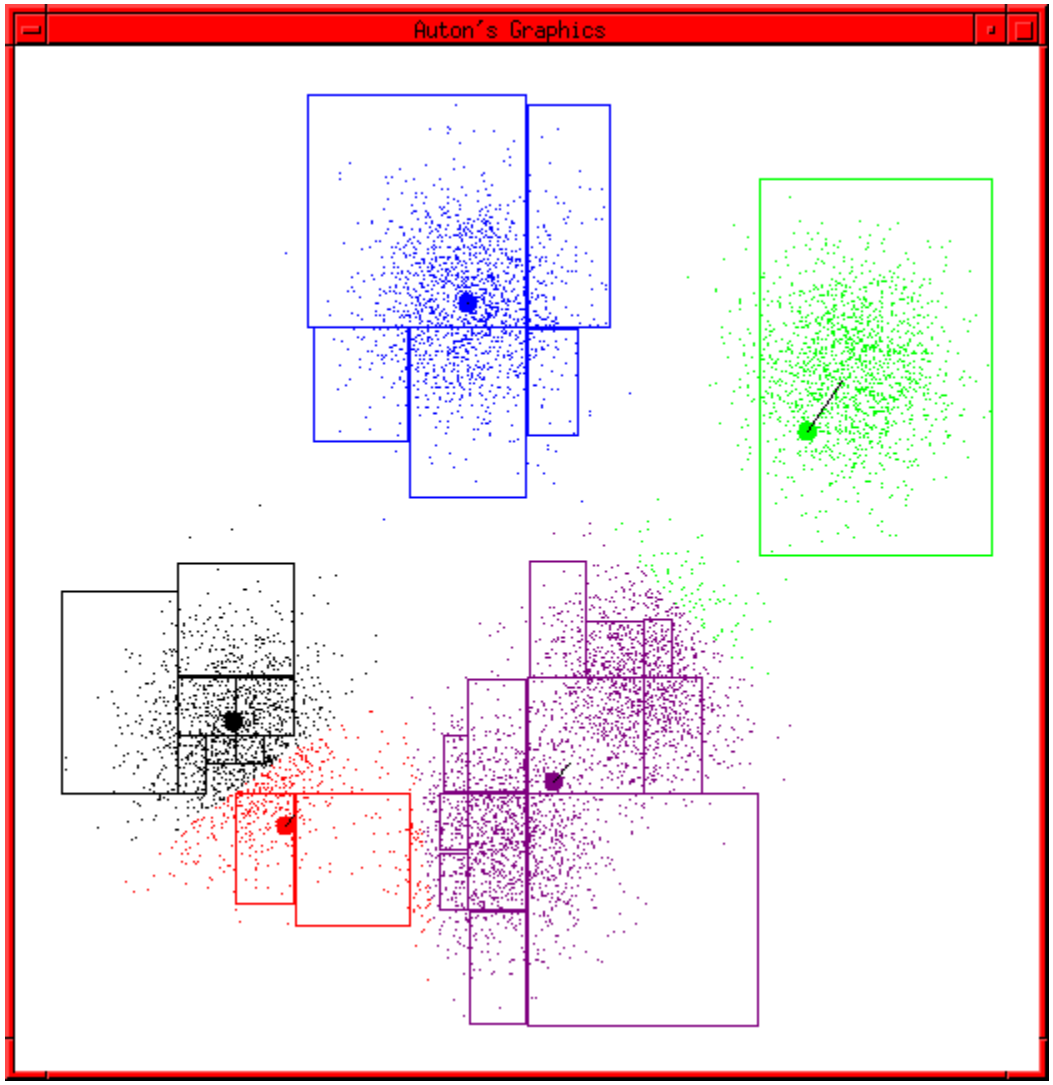
k-Means Algorithmus



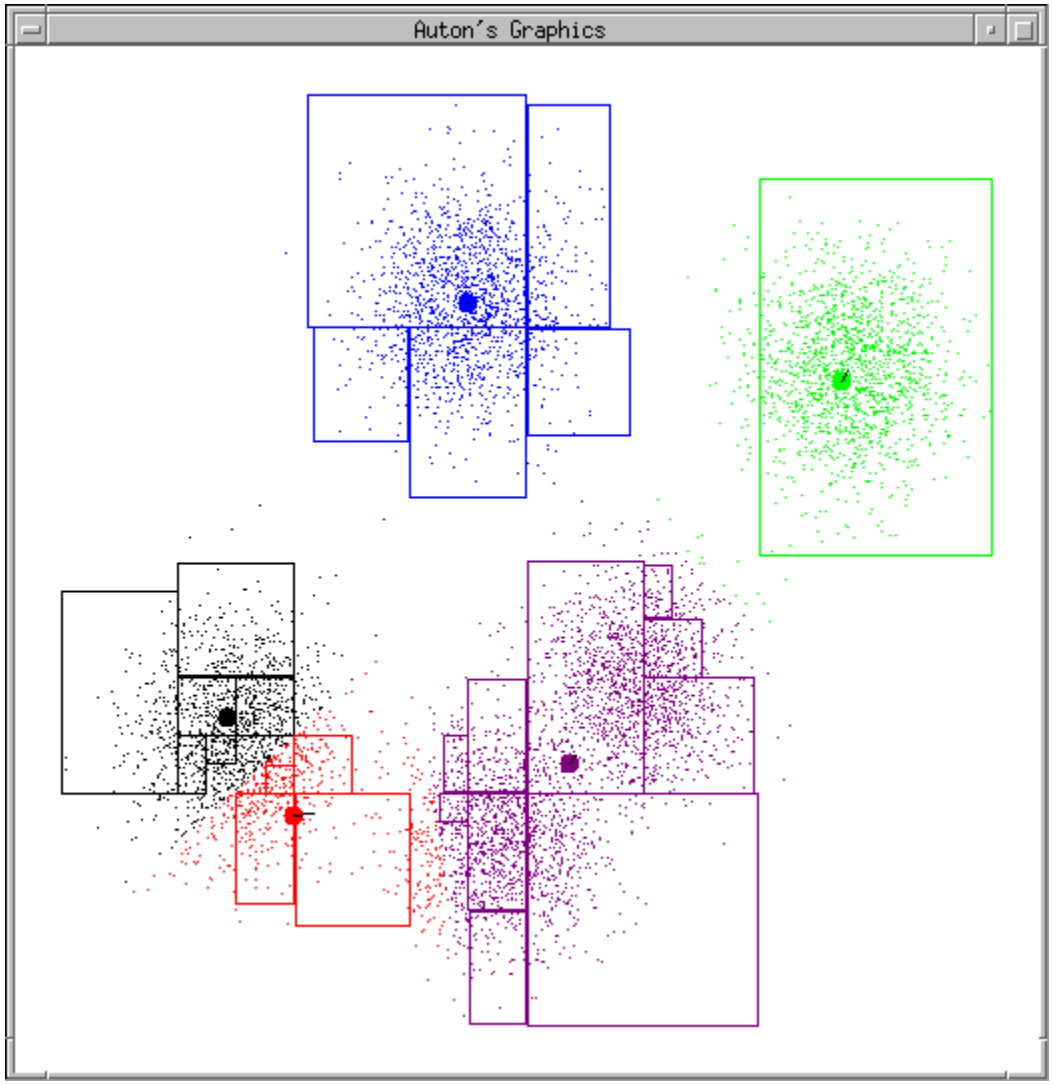
k-Means Algorithmus



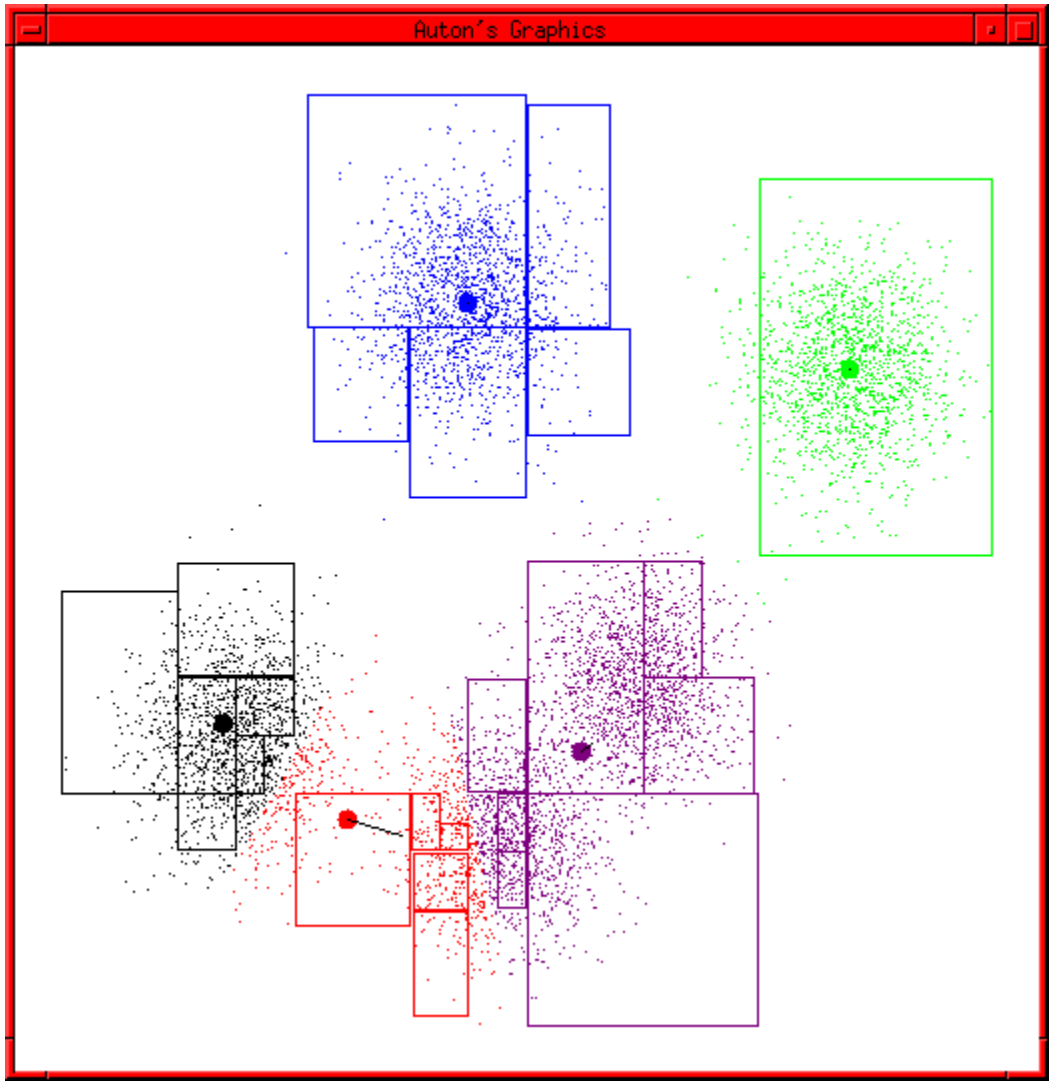
k-Means Algorithmus



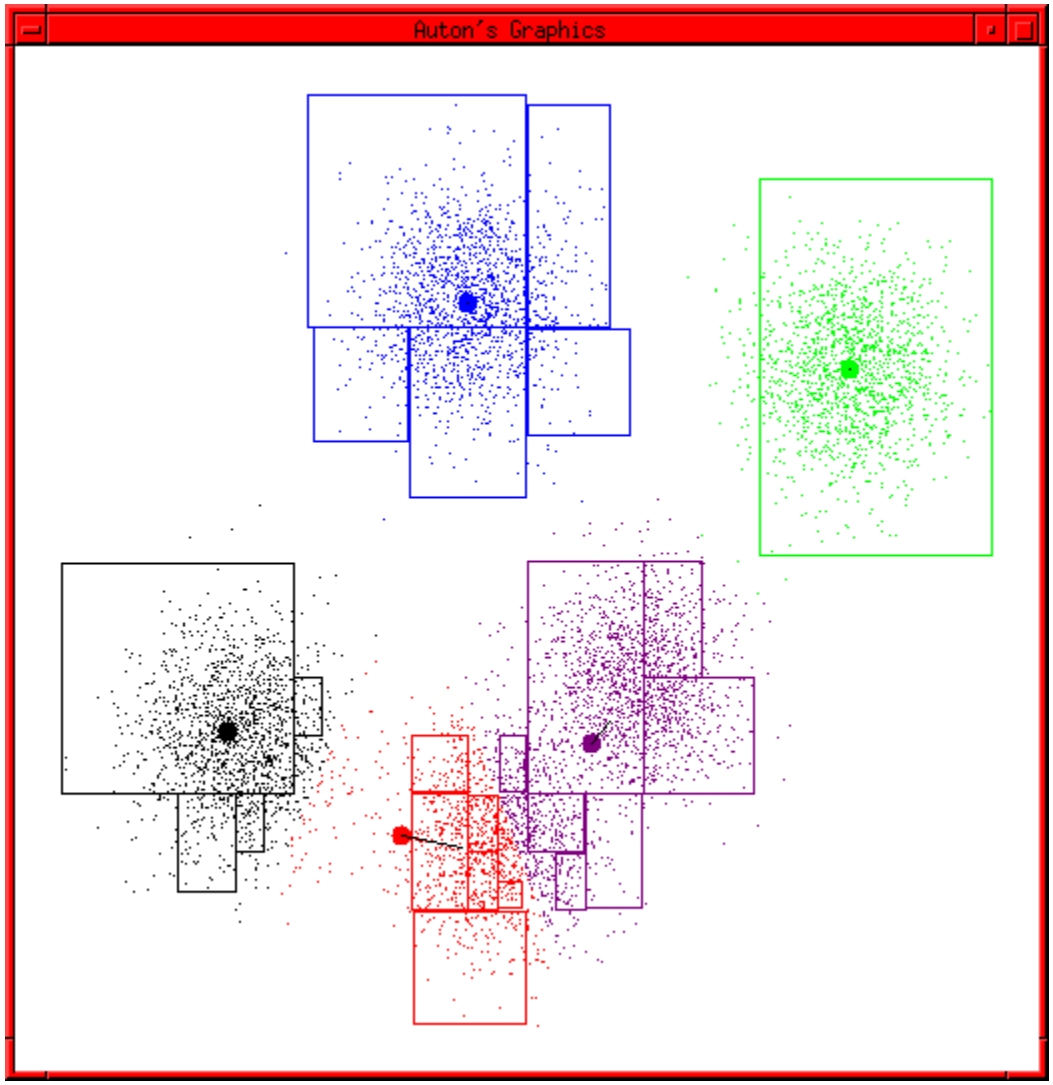
k-Means Algorithmus



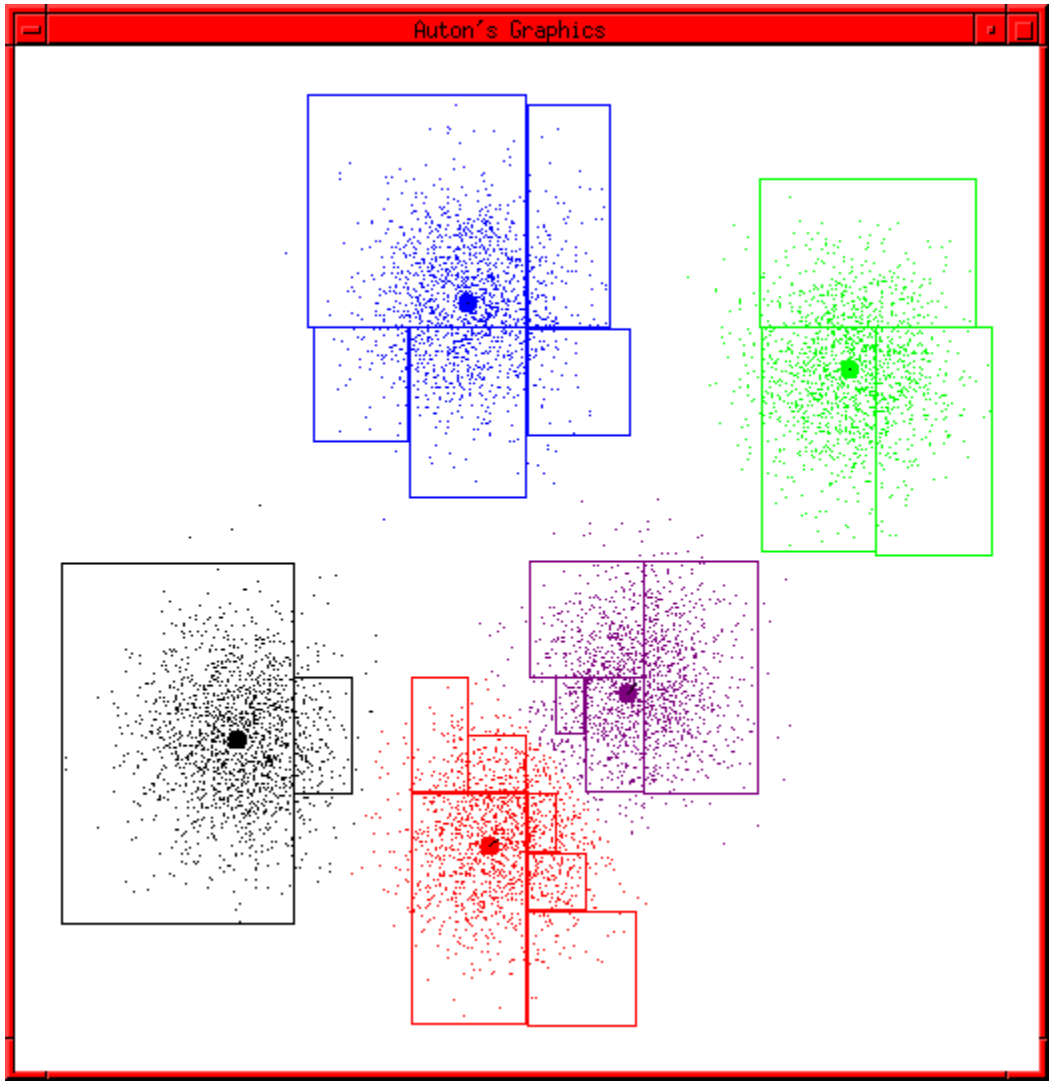
k-Means Algorithmus



k-Means Algorithmus

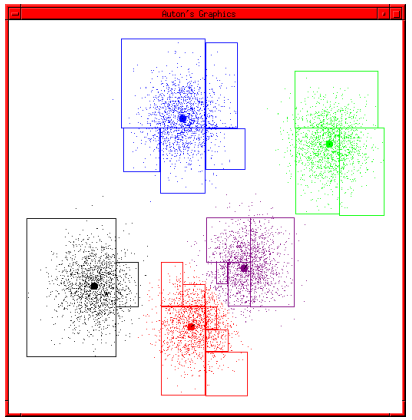


k-Means Algorithmus



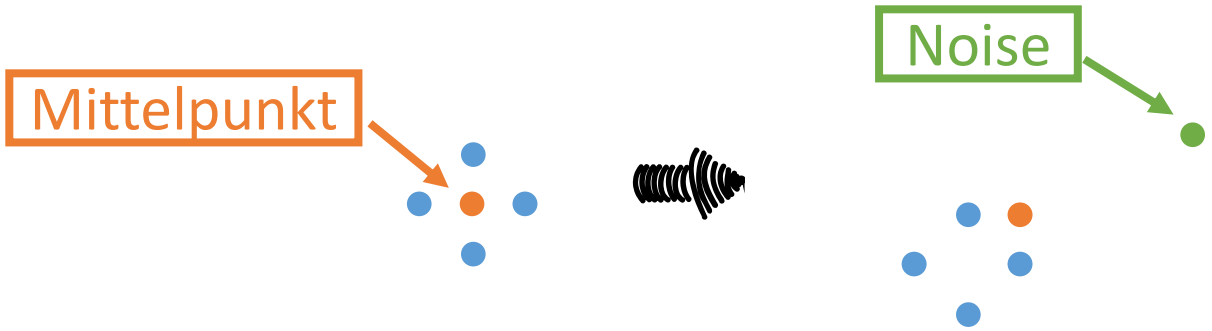
k-Means Eigenschaften

- optimiert die Distanz von den Mittelpunkten der Cluster
  → nur für **numerische Merkmale** geeignet
- Anzahl k der Cluster muss im vorhinein bekannt sein
- am besten geeignet für kompakte, gleich große Cluster
  → „Entscheidungsgrenze“ liegt in der Mitte zwischen zwei Cluster-Mittelpunkten



k-Means Nachteile

- findet nicht die beste Lösung
- Lösung hängt stark von den gewählten Start-Mittelpunkten ab
- bei unpassendem k können sich sinnlose Cluster ergeben
- Stördaten (Noise) können das Clustering-Ergebnis stark beeinflussen



k-Means Varianten

k-Median

- Median als Repräsentation für Cluster statt Mittelwert
- dadurch robuster gegen Stördaten (Noise)

k-Means++

- Regeln zum Bestimmen der „Start“-Mittelpunkte
- restliche Schritte wie bei k-Means
- konvergiert schneller als k-Means

k-Means Anwendung

Segmentierung von Farbbildern → Datenpunkt = Farbvektor

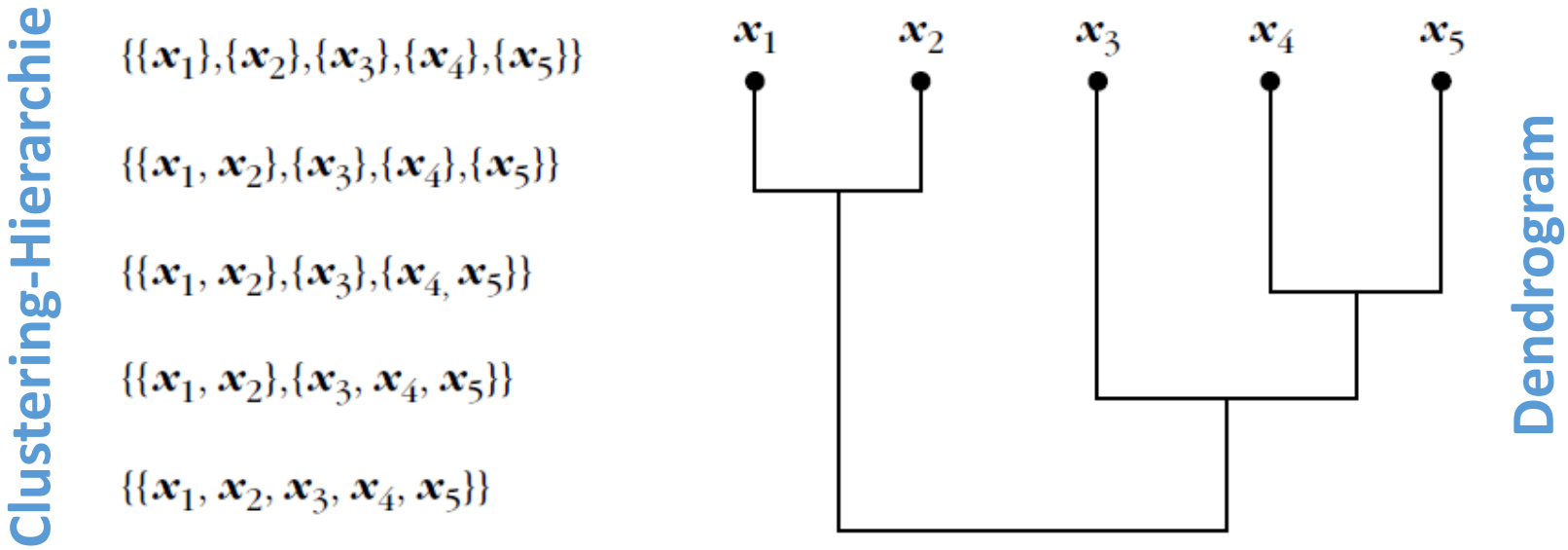
$k = 2$ | $k = 3$ | $k = 10$ | Original

[Quelle: https://www.projectrhea.org/]

VI. Hierarchische Verfahren

Hierarchische Verfahren

Hierarchische Verfahren können in „**Agglomerative**“ und „**Divisive**“ unterteilt werden. Neben homogenen Clustern liefern hierarchische Verfahren eine **hierarchische Struktur** der Daten, ein sogenanntes **Dendrogram** (Verzweigungsbaum).



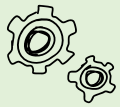
[Quelle: Theodoris et al., 2009]

„Divisive“

… spaltend (frei übersetzt)

Divisive Verfahren verfolgen die **umgekehrte Strategie** von „Agglomerativen“ Verfahren. Beginnend mit einem **einzigen Cluster** der alle Trainingsdaten X enthält, wird dieser Cluster **immer weiter unterteilt** bis in jedem Cluster nur mehr ein Punkt (Vektor) liegt.

Naive Lösung:
Suche die beste Unterteilung von $X = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ in zwei Cluster. Wähle aus den $2^{N-1} - 1$ Möglichkeiten die beste mit Hilfe eines Distanzmaßes. Dieser Vorgang wird wiederholt bis man N Cluster erhält.



„Agglomerative“

… anhäufend (frei übersetzt)

Alle Vektoren in $X = \{x_1, \dots, x_N\}$ repräsentieren jeweils einen von N **Clustern**. Diese werden dann mit Hilfe eines agglomerativen Verfahrens **immer weiter gruppiert** bis alle Daten in einem **einzigen Cluster** liegen.

Abhängig vom **Gruppierungskriterium** unterscheidet man:

- Single Linkage Clustering: kleinster Abstand zwischen Vektoren in Clustern
- Complete Linkage Clustering: größter Abstand zwischen Vektoren in Clustern
- Average Linkage Clustering: durchschnittlicher Abstand zwischen Vektoren in Clustern

Agglomeratives Verfahren

Initialisierung

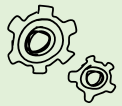
Clustering $\Re_0 = \{C_i = \{\boldsymbol{x}_i\}, i = 1, \dots, N\}$

Level der Hierarchie $t = 0$

$\Re_0$

Agglomeratives Verfahren

1. Schritt



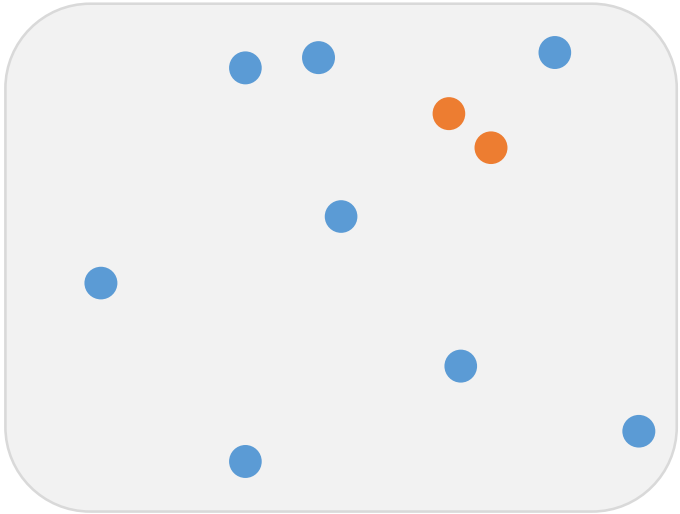
$t = t + 1$

2. Schritt

Aus allen möglichen Clusterpaaren (C_r, C_s) in $\Re_{t-1}$ finde (C_i, C_j), sodass

$g(C_i, C_j) = \min_{\boldsymbol{x} \in C_i, \boldsymbol{y} \in C_j} d(\boldsymbol{x}, \boldsymbol{y})$ oder

$g(C_i, C_j) = \max_{\boldsymbol{x} \in C_i, \boldsymbol{y} \in C_j} s(\boldsymbol{x}, \boldsymbol{y})$.

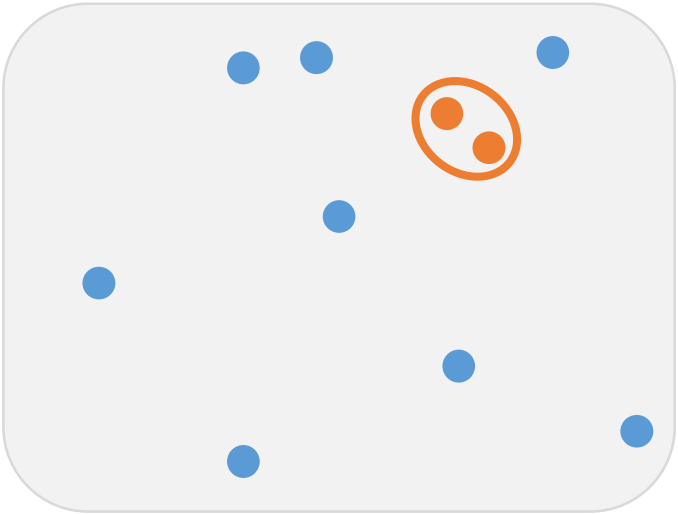


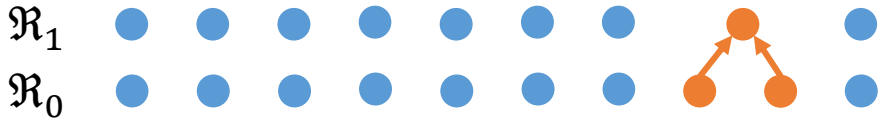
$\Re_0$

Agglomeratives Verfahren

3. Schritt

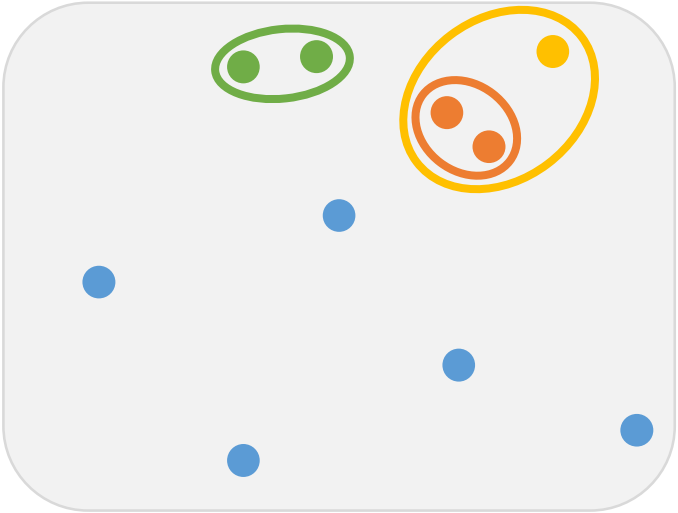
Definiere $C_p = C_i \cup C_j$ und erzeuge neues Clustering $\Re_t = (\Re_{t-1} - \{C_i, C_j\}) \cup \{C_p\}$.

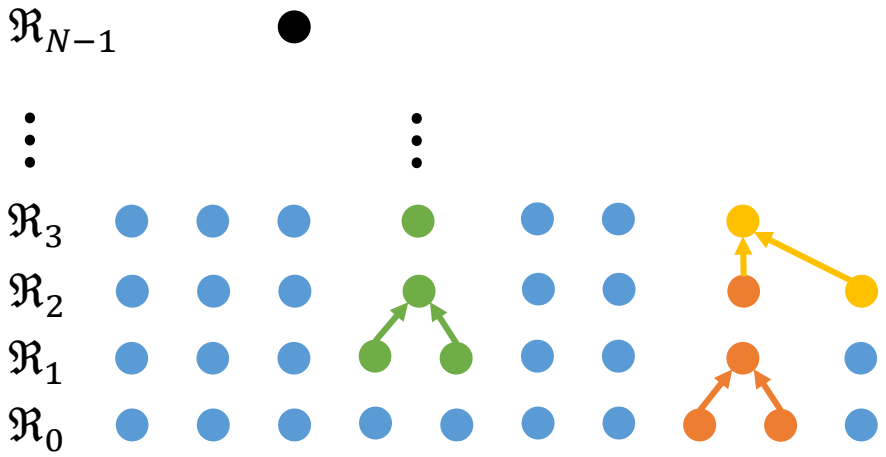




Agglomeratives Verfahren

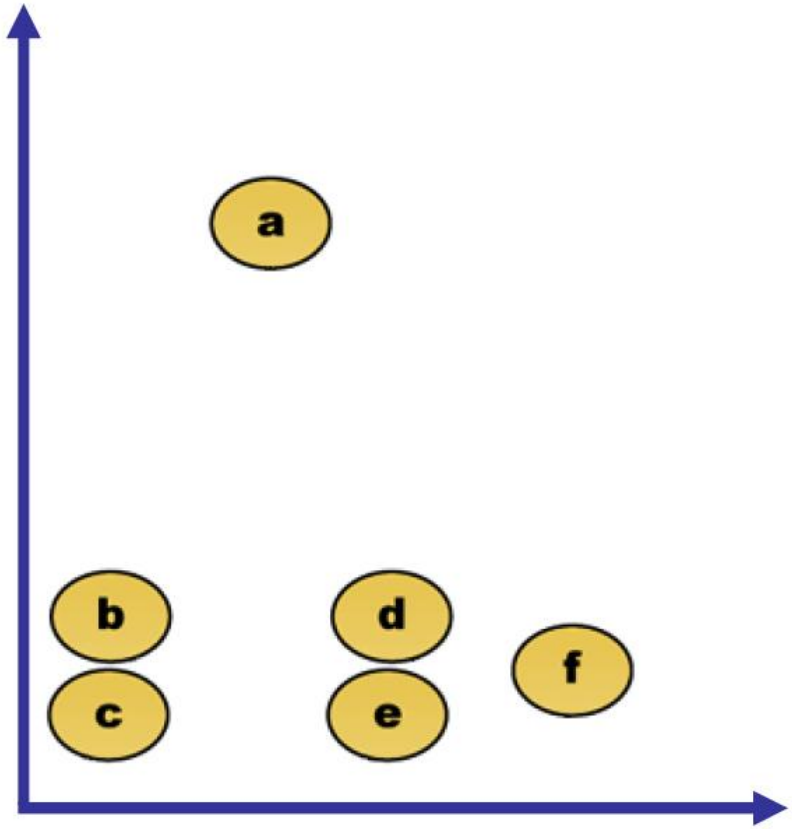
Schritte 1 – 3 werden wiederholt bis alle Punkte (Vektoren) in einem Cluster liegen.



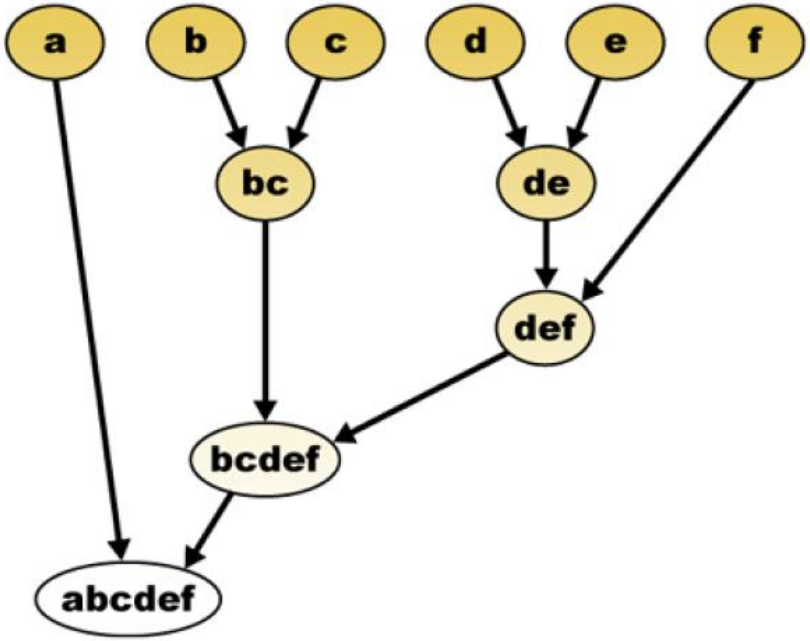


Weiteres Beispiel

[Quelle: http://de.wikipedia.org/wiki/Hierarchische_Clusteranalyse]

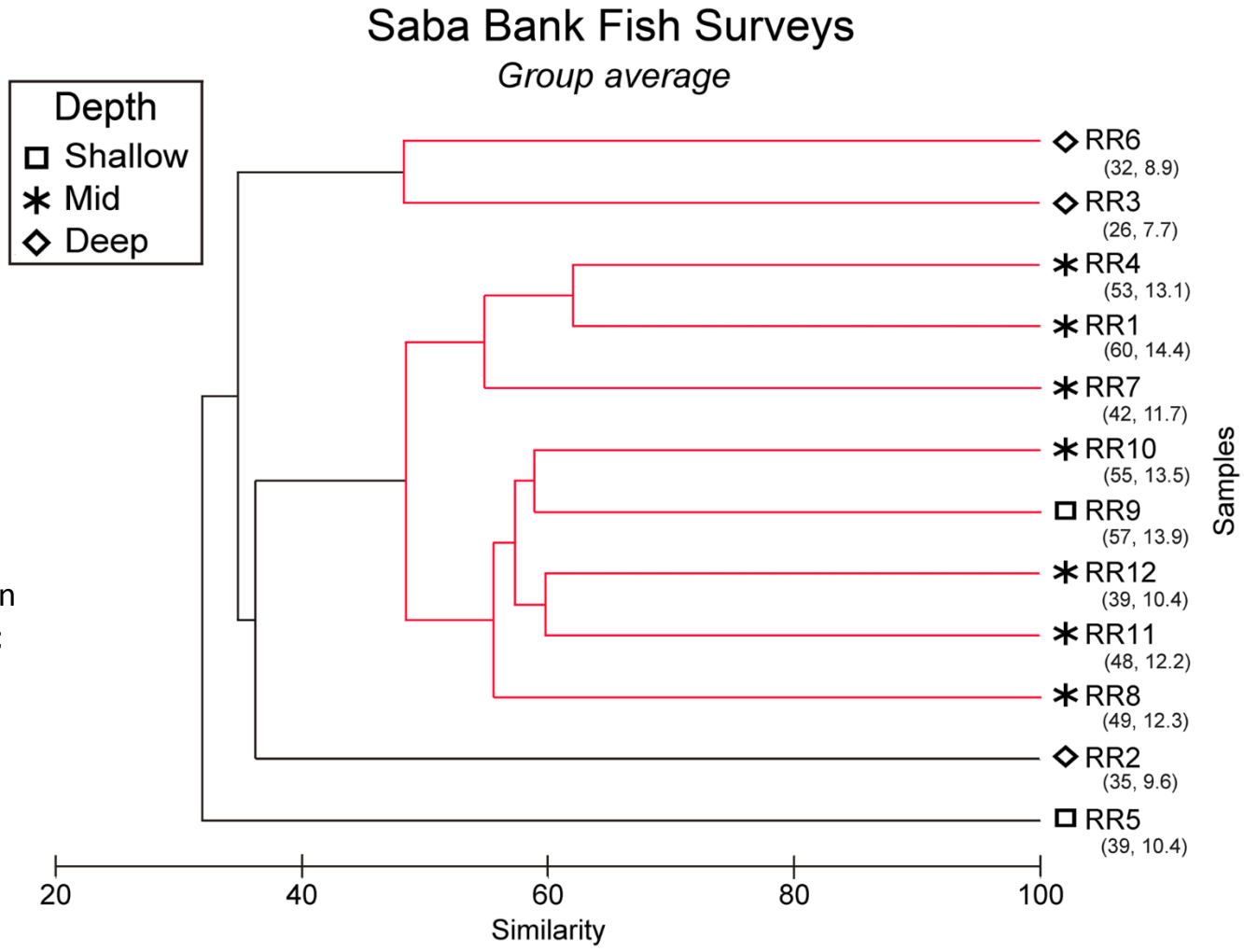


Datenpunkte



Verzweigungsbaum

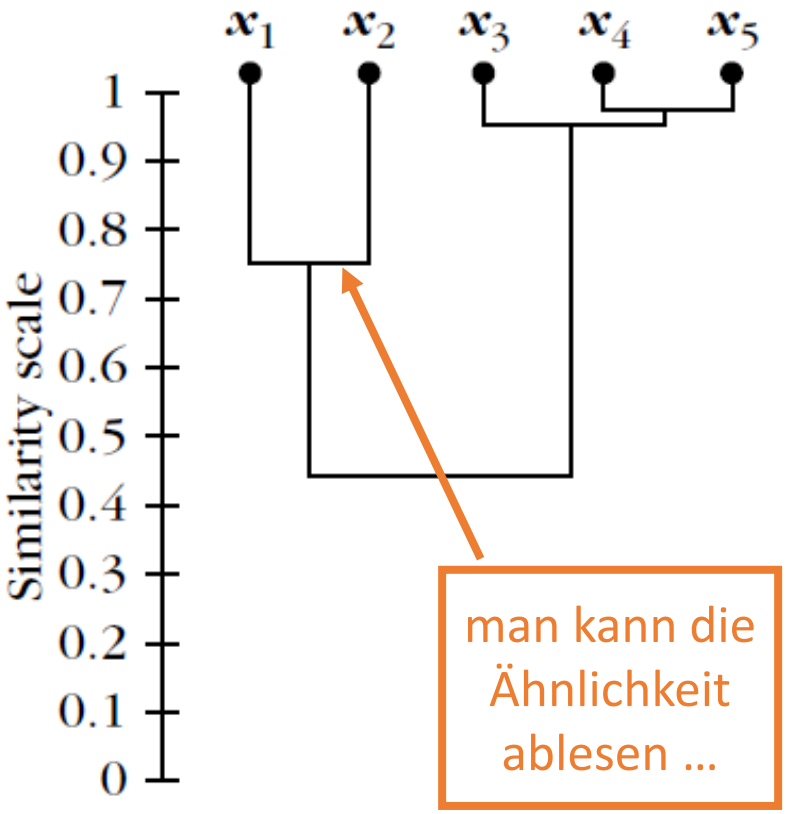
Weiteres Beispiel

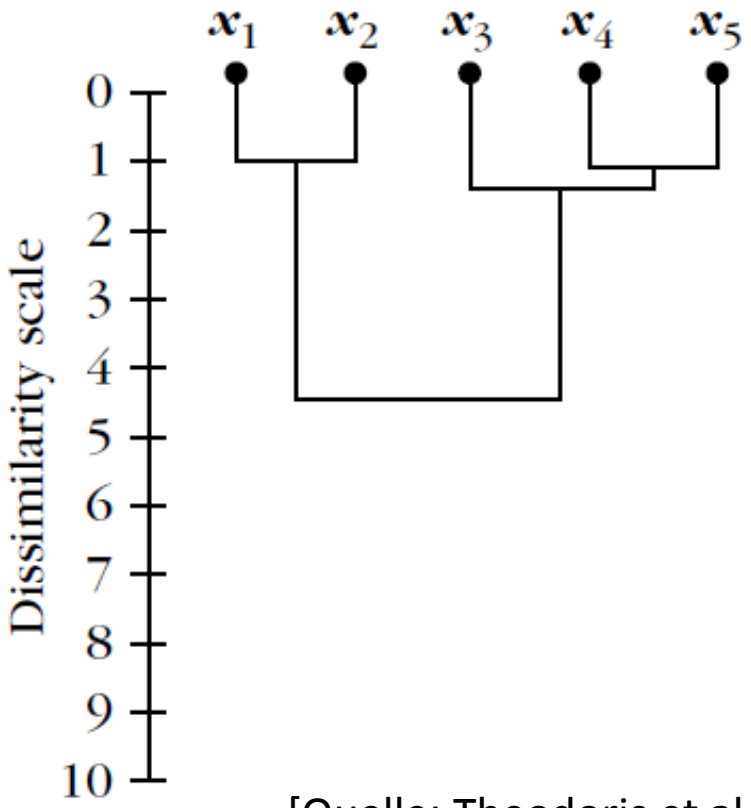


Quelle:
Williams, J. T.; Carpenter, K. E.; Van Tassell, J. L.; Hoetjes, P.; Toller, W.; Etnoyer, P.; Smith, M. (2010). "Biodiversity Assessment of the Fishes of Saba Bank Atoll, Netherlands Antilles". *PLoS ONE* **5** (5): e10676.

Dendrogram

… und was man daraus ablesen kann





[Quelle: Theodoris et al., 2009]

Anmerkungen

… zu hierarchischen Verfahren

- Nachteil: es gibt keine Möglichkeit sich von einer schlechten Gruppierung zu „erholen“
- Vorteil: man erhält eine Hierarchie
- um k Cluster zu erhalten, stoppt man das Verfahren an der richtigen Stelle oder „schneidet“ am Ende einfach die passende Verbindung

Anwendungsbeispiel

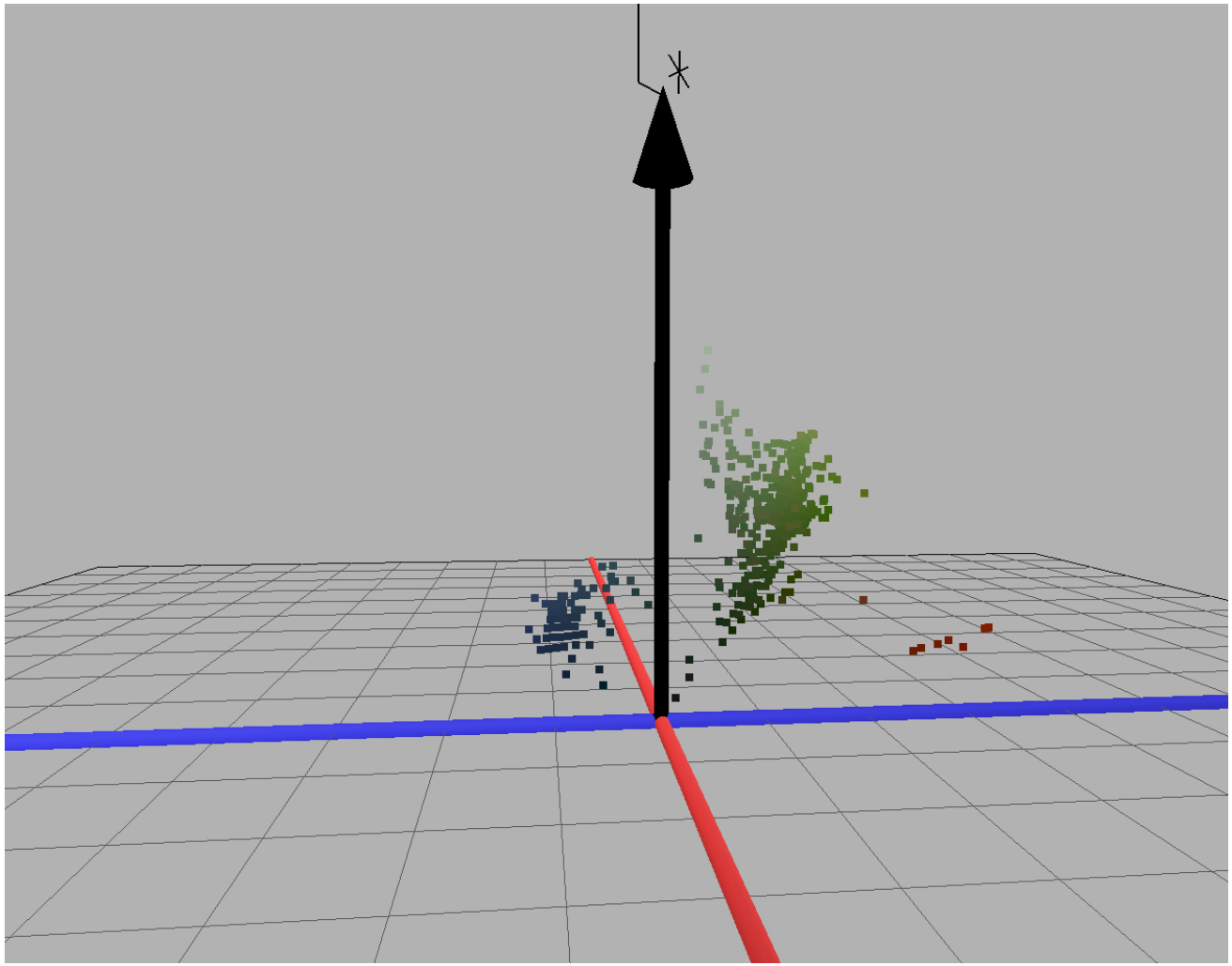
… wieder für die Segmentierung von Farbbildern

Anwendungsbeispiel

- Reduktion auf 500 Farben
- Transformation vom RGB- in den CIELAB-Farbraum

Anwendungsbeispiel

500 verschiedene Farben (Cluster) im CIELAB-Farbraum



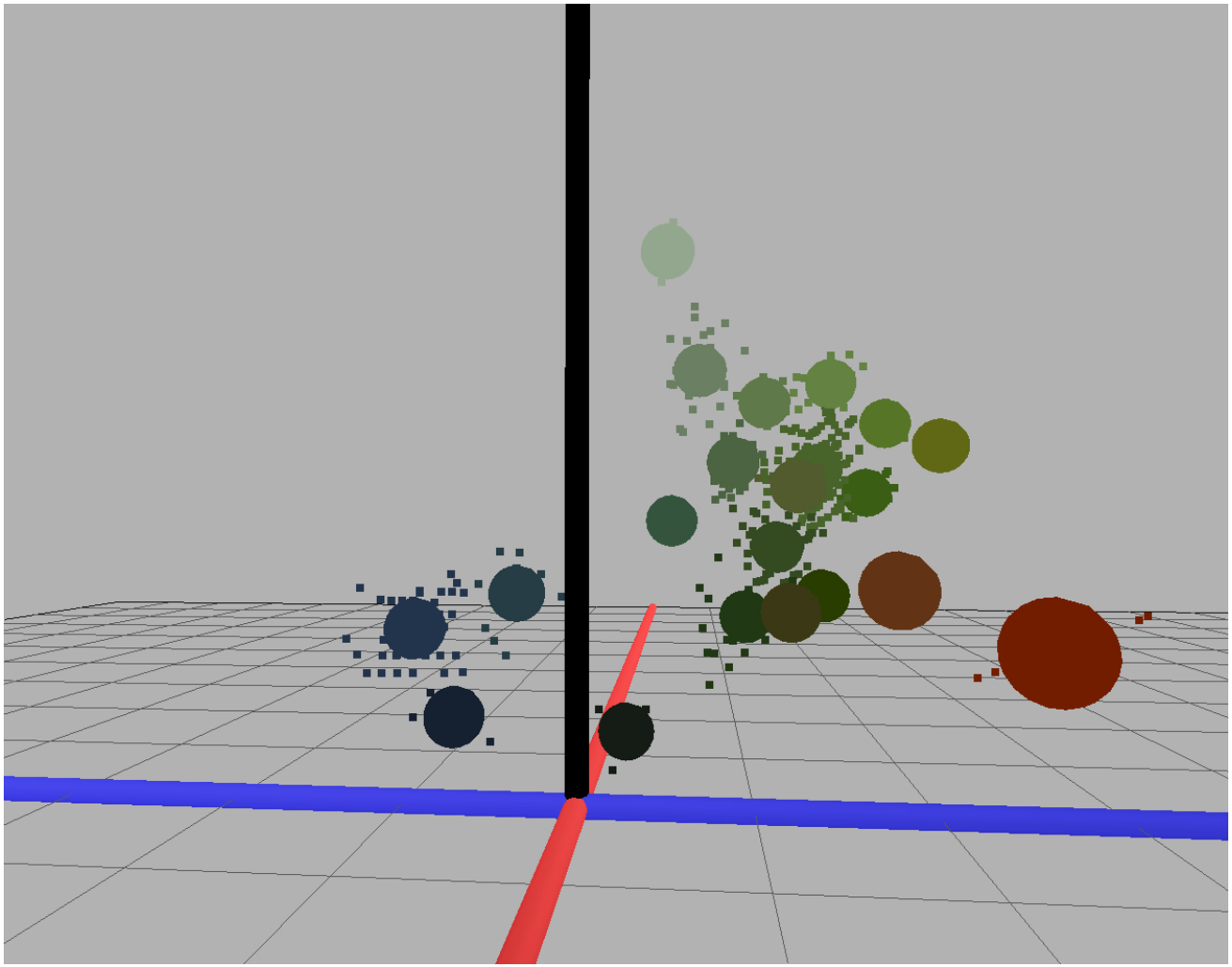
Anwendungsbeispiel

- für die Segmentierung wird hierarchisches „ Single Linkage Clustering“ verwendet
- Stoppkriterium: euklidische Distanz zwischen den zwei am nächst gelegenen Clustern > 10
- Ergebnis: 21 Cluster

Anwendungsbeispiel

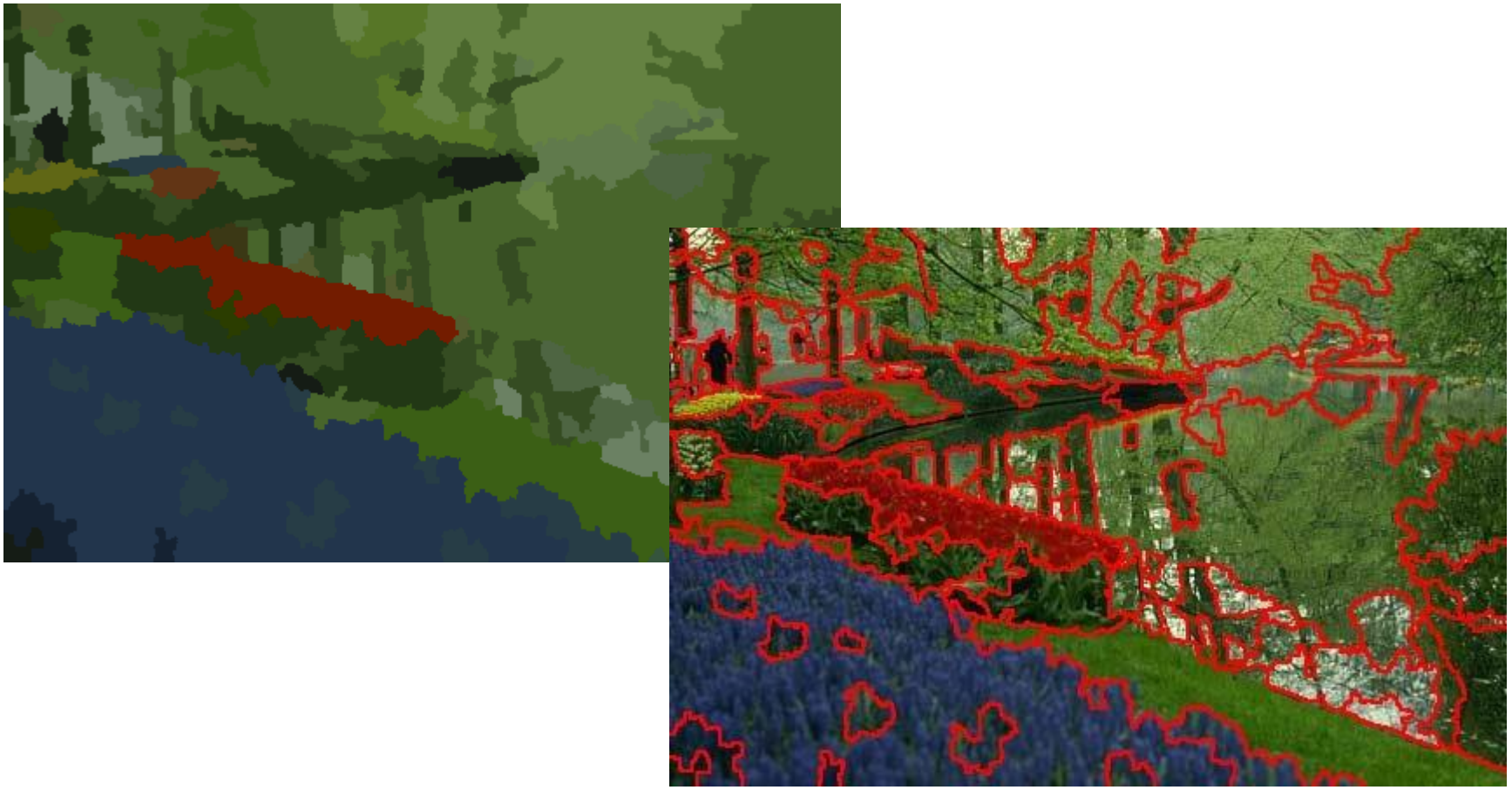
Ergebnis:

21 Cluster



Anwendungsbeispiel

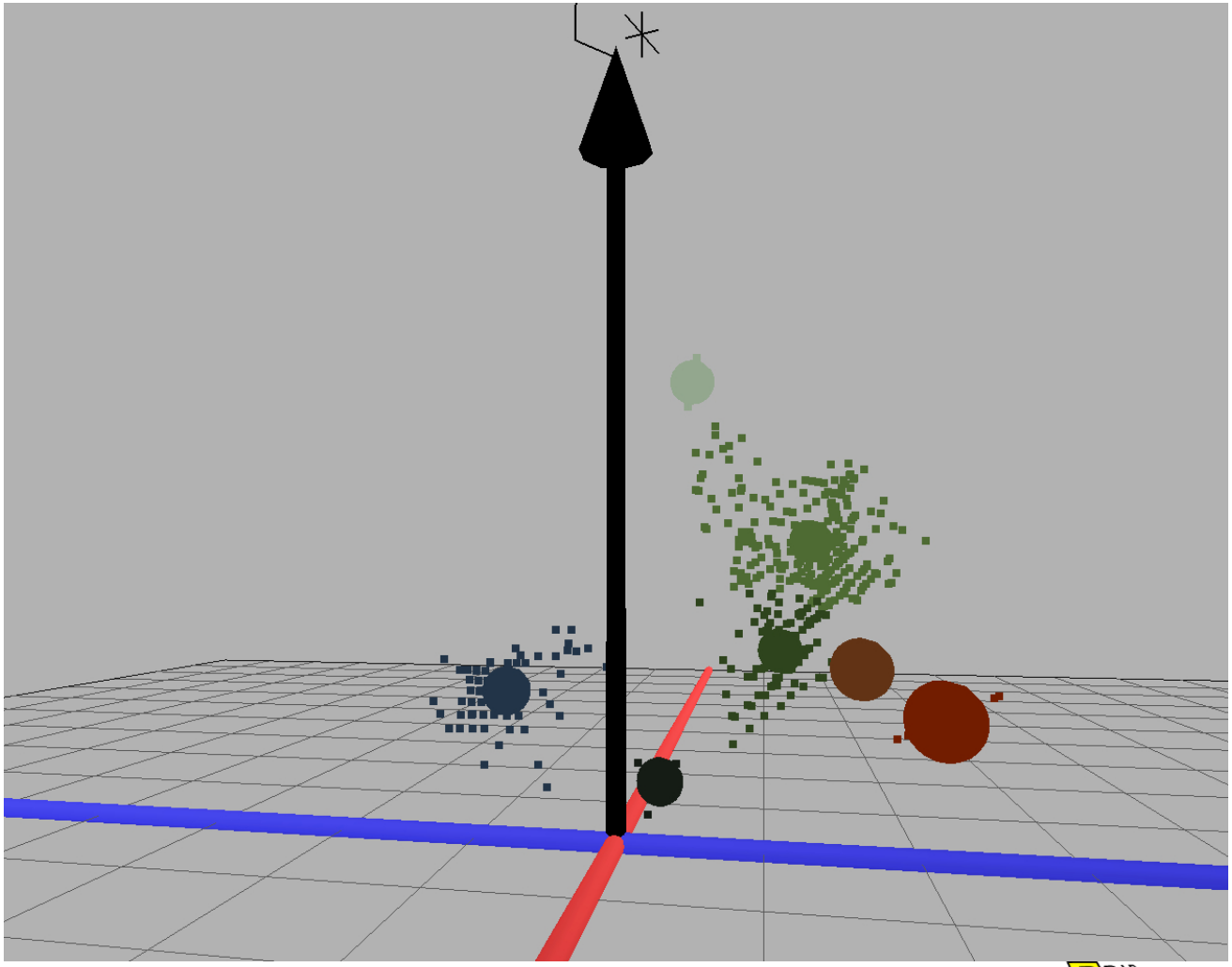
- jedes Pixel wurde einem der 21 Farb-Cluster zugeordnet



Anwendungsbeispiel

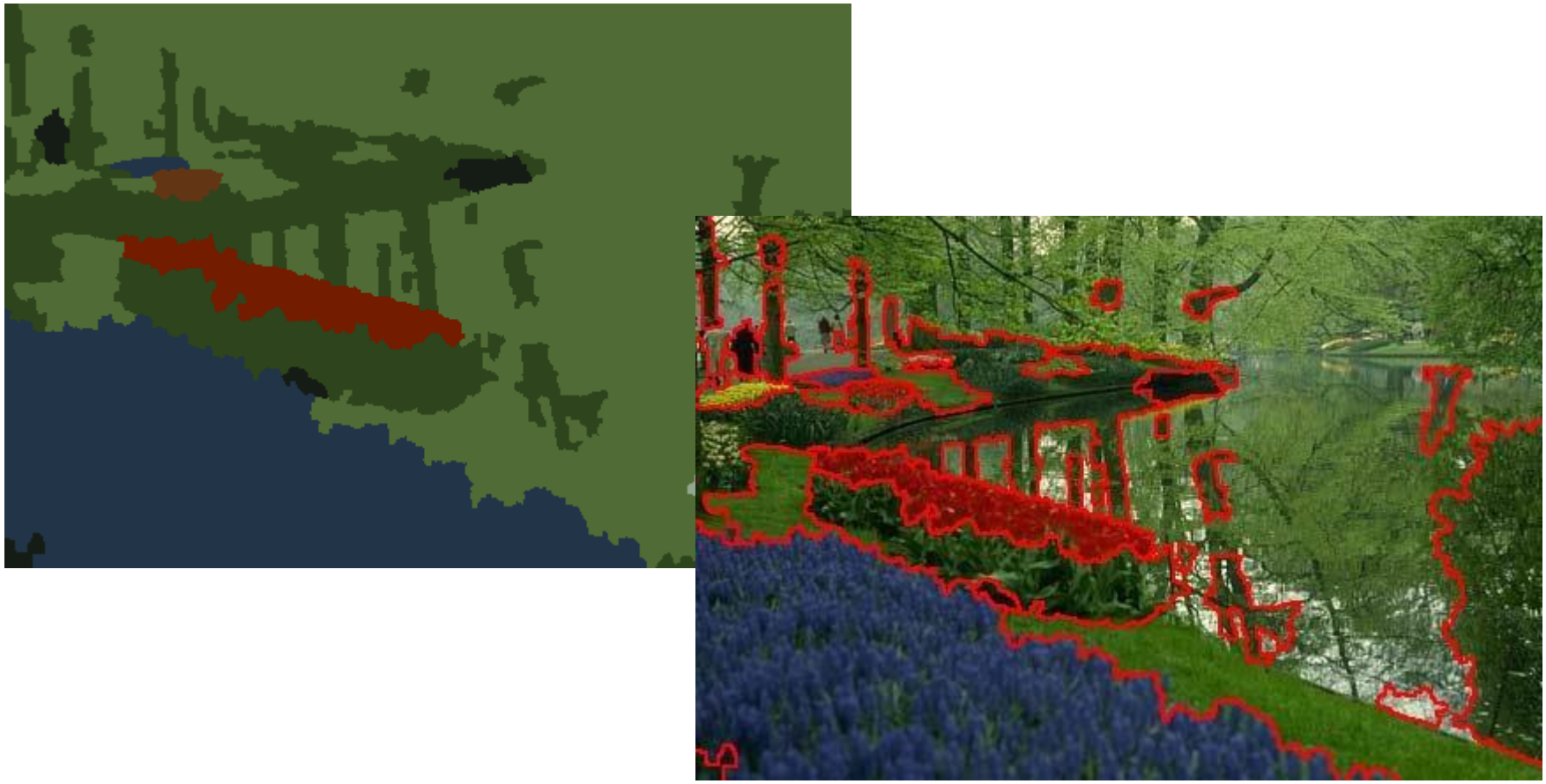
Änderung des Stoppkriteriums: euklidische Distanz > 20

→ nur 7 Cluster



Anwendungsbeispiel

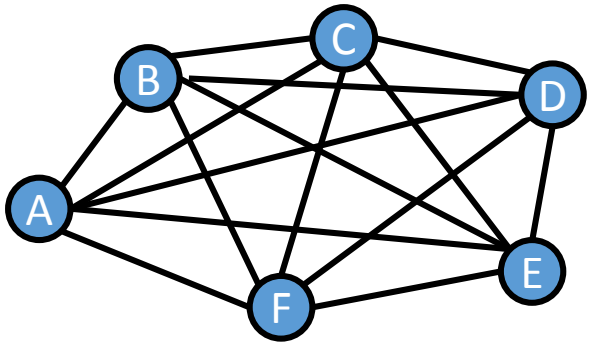
- Ergebnis im Bild …



VII. Graph-basierte Verfahren

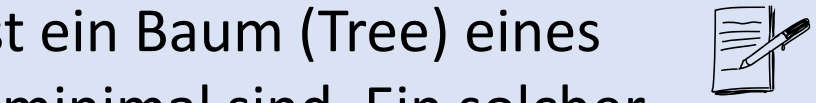
Graph-basierte Verfahren

Graph-basierte Verfahren repräsentieren die Trainingsdaten $X = \{x_1, \dots, x_N\}$ mit Hilfe eines Graphen $G = \{V, E\}$ mit N Knoten und bis zu $\frac{N(N-1)}{2}$ Kanten (man spricht von einem „vollständigen Graphen“). In den Kanten wird die Ähnlichkeit/Unähnlichkeit als Gewicht gespeichert. Das Ergebnis des Clusterings ist abhängig vom Verfahren, vom Layout und von den Gewichten des Graphen.

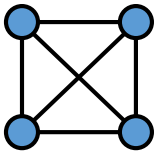


Minimum Spanning Tree

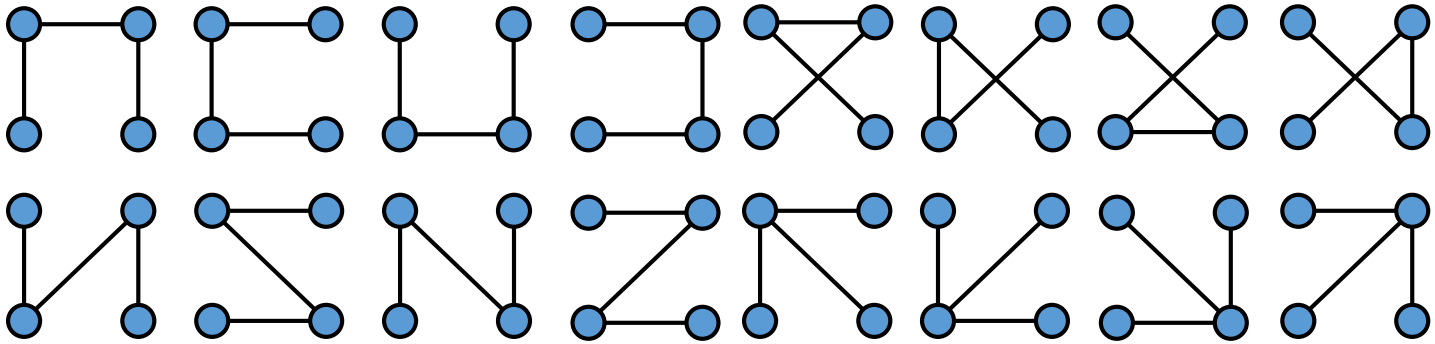
Der **M**inimum **S**panning **T**ree (MST) ist ein Baum (Tree) eines Graphen dessen Gewichte in Summe minimal sind. Ein solcher Baum enthält alle Knoten des Graphen und hat keine Schleifen.



… ein Graph hat viele „Spanning Trees“



vollständiger Graph



16 „Spanning Trees“

MST Clustering-Algorithmus

1. Erstelle einen vollständigen Graphen G, sodass seine Knoten den Vektoren (Trainingsdaten) in $X = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ entsprechen und die Gewichte $w_{(\boldsymbol{x}_i, \boldsymbol{x}_j)} = d(\boldsymbol{x}_i, \boldsymbol{x}_j) \quad i, j = 1, \dots, N \quad i \neq j$.
2. Bestimme MST.
3. Finde und entferne inkonsistente Kanten im MST.
4. Cluster ergeben sich als verbundene Komponenten (connected components) des MST.

Kruskals Algorithmus

… zur Bestimmung des MST

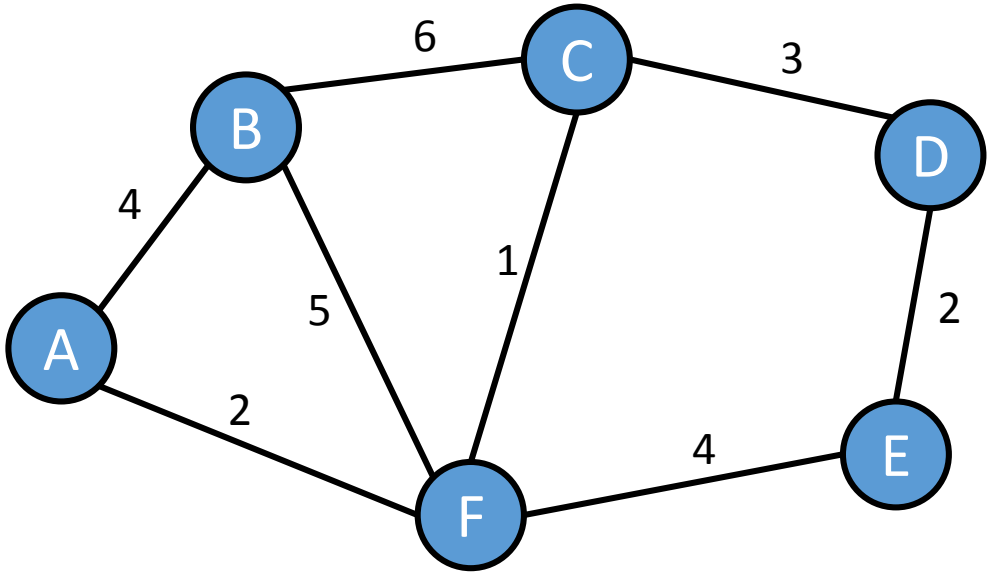
1. Sortiere Kanten des Graphen G in aufsteigender Reihenfolge (basierend auf den Gewichten)
2. Erstelle eine „Kanten-Liste“ A (anfangs leer)

für jede Kante in sortierter Reihenfolge:

3. überprüfe, ob Knoten der Kante über einen Pfad in A verbunden sind
4. falls NEIN: dann füge Kante in A hinzu

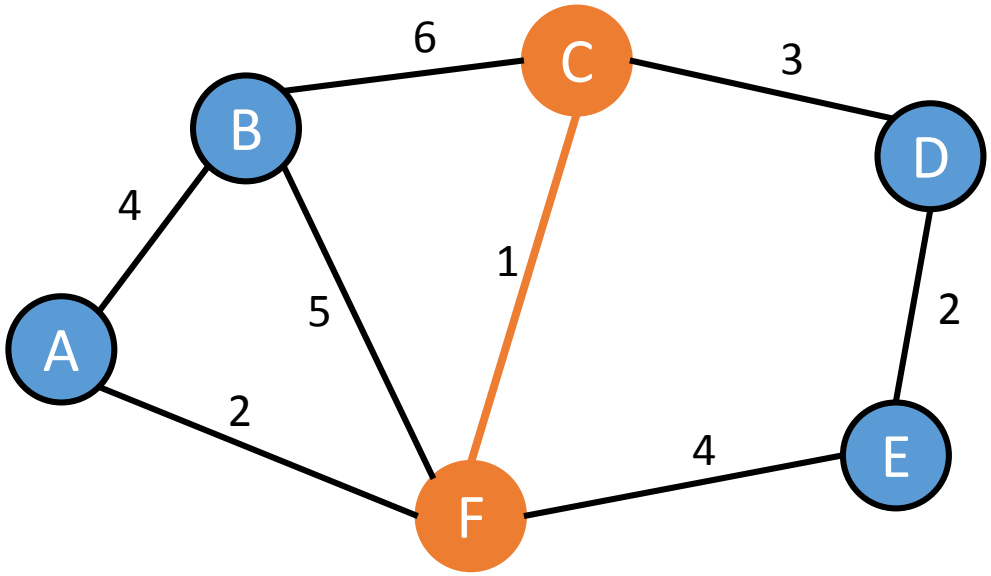
Beispiel

$A = \{\quad\}$
$V = \{A, B, C, D, E, F\}$
$E = \{(C, F); (A, F); (D, E); (C, D); (A, B); (E, F); (B, F); (B, C)\}$



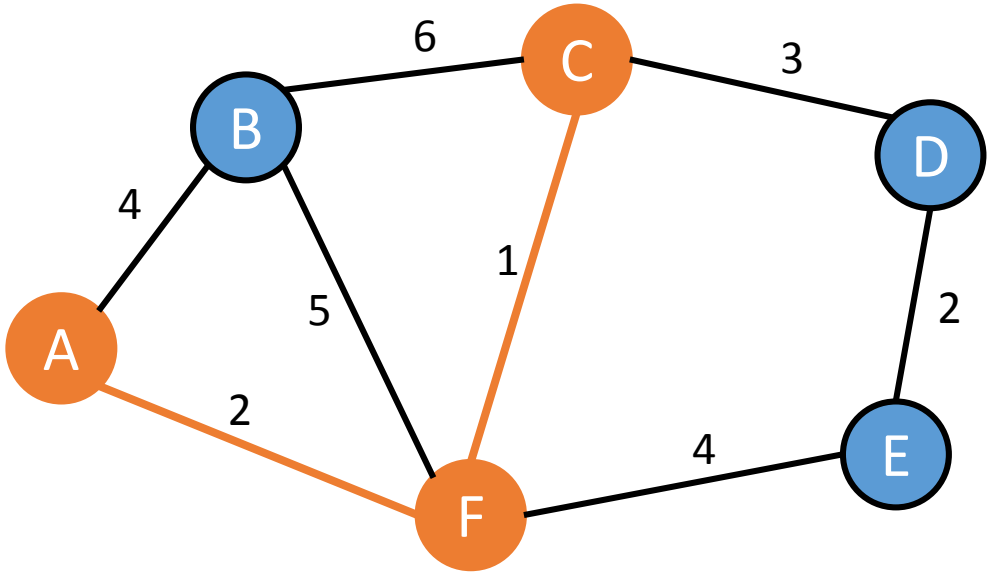
Beispiel

$A = \{(C, F)\}$
$V = \{A, B, C, D, E, F\}$
$E = \{\cancel{(C, F)}; (A, F); (D, E); (C, D); (A, B); (E, F); (B, F); (B, C)\}$



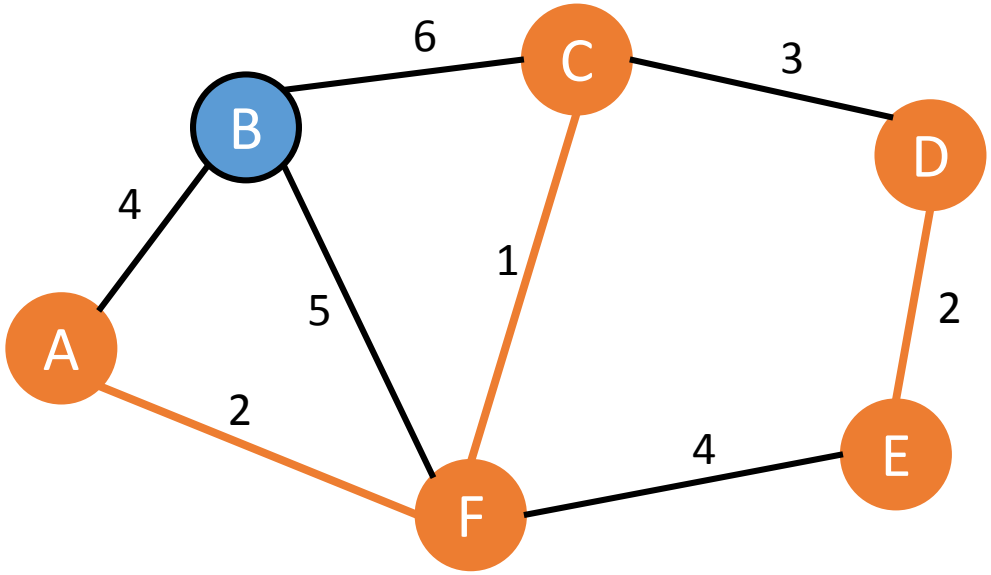
Beispiel

$A = \{(C, F); (A, F)\}$
$V = \{A, B, C, D, E, F\}$
$E = \{\cancel{(C, F)}; \cancel{(A, F)}; (D, E); (C, D); (A, B); (E, F); (B, F); (B, C)\}$



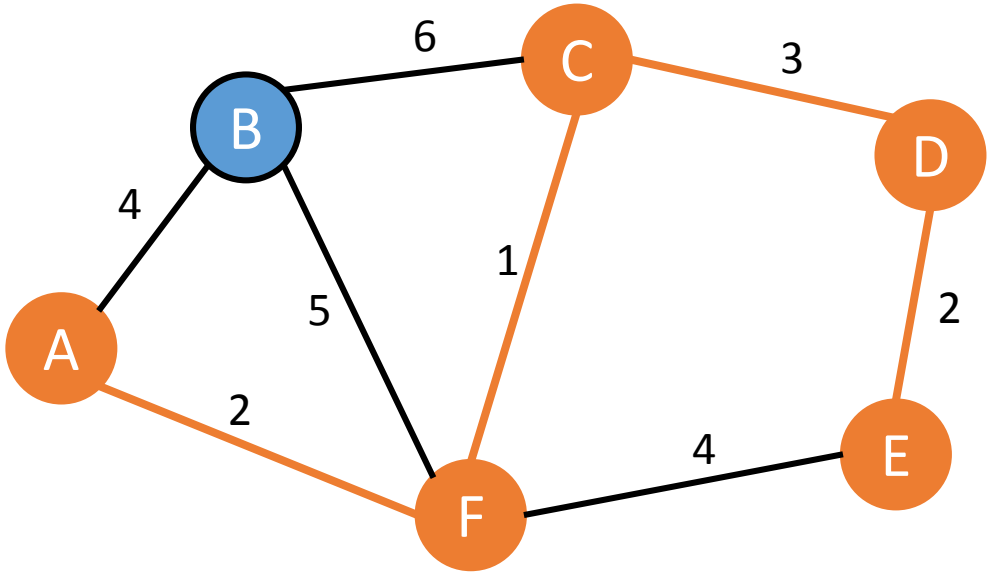
Beispiel

$A = \{(C,F);(A,F);(D,E)\}$
$V = \{A,B,C,D,E,F\}$
$E = \{\cancel{(C,F)};\cancel{(A,F)};\cancel{(D,E)};(C,D);(A,B);(E,F);(B,F);(B,C)\}$



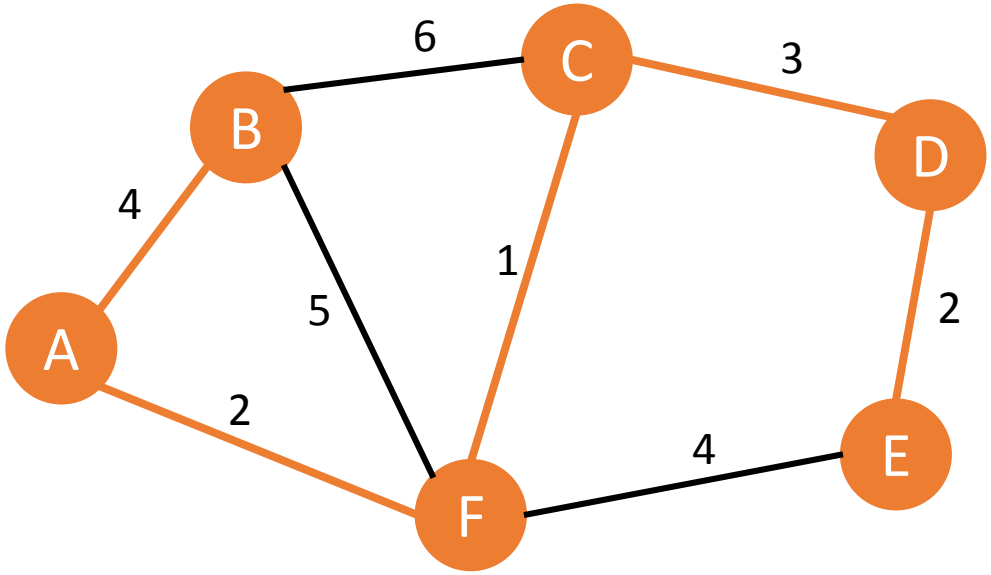
Beispiel

$A = \{(C,F);(A,F);(D,E);(C,D)\}$
$V = \{A,B,C,D,E,F\}$
$E = \{\sout{(C,F)};\sout{(A,F)};\sout{(D,E)};\sout{(C,D)};(A,B);(E,F);(B,F);(B,C)\}$



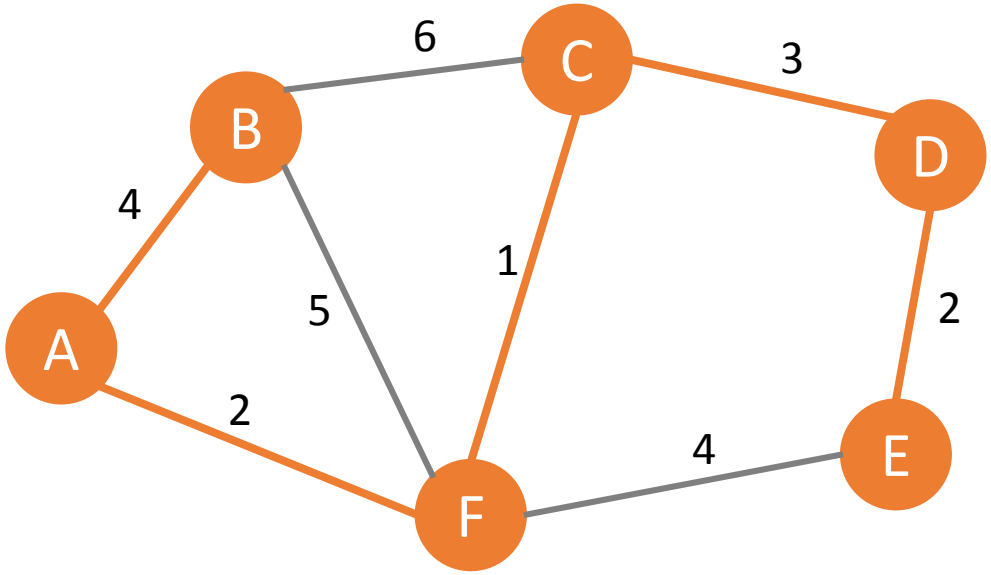
Beispiel

$A = \{(C,F); (A,F); (D,E); (C,D); (A,B)\}$
$V = \{A, B, C, D, E, F\}$
$E = \{\cancel{(C,F)}; \cancel{(A,F)}; \cancel{(D,E)}; \cancel{(C,D)}; \cancel{(A,B)}; (E,F); (B,F); (B,C)\}$



Beispiel

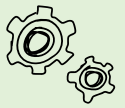
$A = \{(C,F);(A,F);(D,E);(C,D);(A,B)\}$
$V = \{A,B,C,D,E,F\}$
$E = \{\cancel{(C,F)};\cancel{(A,F)};\cancel{(D,E)};\cancel{(C,D)};\cancel{(A,B)};\cancel{(E,F)};\cancel{(B,F)};\cancel{(B,C)}\}$



Inkonsistente Kanten

… sind Kanten die ein ungewöhnlich großes Gewicht im Vergleich zu ihren Nachbar aufweisen

Ein möglicher Ansatz:



- Für jede Kante e betrachtet man alle Kanten e_i die maximal k Schritte entfernt liegen.
- Aus dieser „Nachbarschaft“ berechnet man das durchschnittliche (mittlere) Gewicht μ_w und damit die Standardabweichung σ_w.
- Falls $|w_e - \mu_w| > q \cdot \sigma_w$, dann ist e inkonsistent

Bestimmung der inkonsistenten Kanten hängt stark von den gewählten Parametern k und q ab.

Beispiel

für $k = 2$ und $q = 3$

[Quelle: Theodoris et al., 2009]

Überprüfung e_0:

$\mu_w = 2{,}3$

$\sigma_w = 0{,}95$

$17 - 2{,}3 > 3 \cdot 0{,}95$

Ergebnis: inkonsistent

Überprüfung e_{11}:

$\mu_w = 2{,}5$

$\sigma_w = 2{,}12$

$3 - 2{,}5 < 3 \cdot 2{,}12$

Ergebnis: konsistent