INTELLIGENTE DATENANALYSE IN MATLAB

Unüberwachtes Lernen

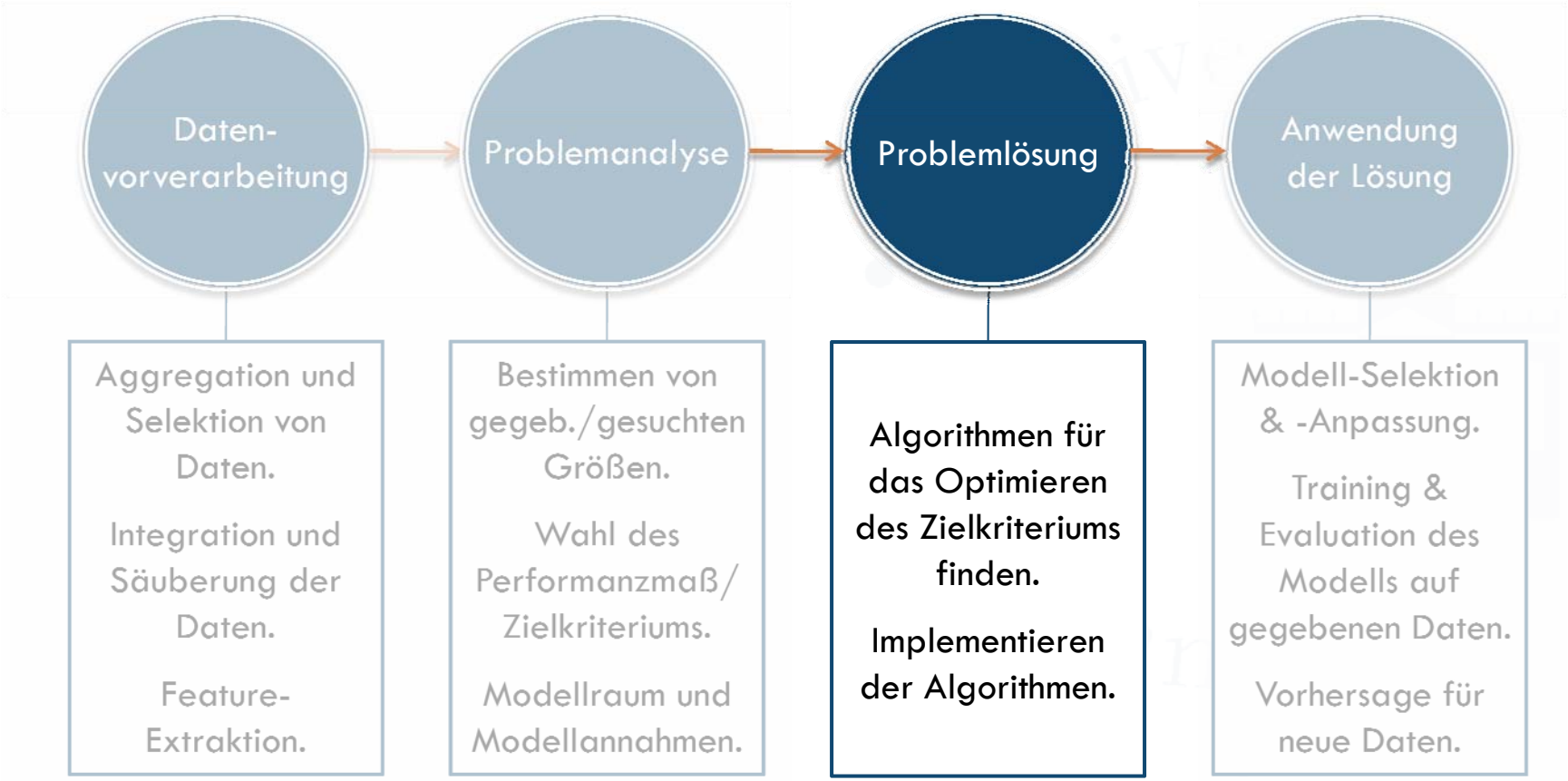
Michael Brückner/Tobias Scheffer

Literatur

- Chris Bishop: Pattern Recognition and Machine Learning.
- Jiawei Han und Micheline Kamber: Data Mining – Concepts and Techniques.
- Ulrike von Luxburg: A Tutorial on Spectral Clustering.
  http://www.kyb.mpg.de/publications/attachments/Luxburg06_TR_%5B0%5D.pdf
- Matteo Matteucci: A Tutorial on Clustering Algorithms.
  http://home.dei.polimi.it/matteucc/Clustering/tutorial_html

Überblick

- Schritte der Datenanalyse:



Unüberwachtes Lernen:

Problemstellung

- Gegeben: Trainingsdaten mit unbekannten Zielattributen (ungelabelte Daten).
- Eingabe: Instanzen.
- Ausgabe: Belegung der Zielattribute.
  - Clustern: Nominaler Wertebereich des Zielattributs (z.B. Jahreszeiten, {Cluster1, Cluster2, …}), d.h. jeder Datenpunkt wird zu genau einem Cluster zugeordnet.
  - Dichteschätzung: Numerischer Wertebereich des Zielattributs, d.h. jedem Datenpunkt wird Wahrscheinlichkeitsverteilung über alle Cluster zugeordnet.

Unüberwachtes Lernen:

Beispiel

- Beispiel *Clustern* (Tabellendarstellung):

Monat	Bewölkung	Temperatur	Luftfeuchtigkeit	Wind	Jahreszeit
Juli	sonnig	warm	hoch	wenig	?
September	sonnig	warm	hoch	stark	?
August	bedeckt	warm	hoch	wenig	?
April	Regen	mild	hoch	wenig	?
Oktober	Regen	kühl	normal	wenig	?
Dezember	Regen	kühl	normal	stark	?
Januar	bedeckt	kühl	normal	stark	?
Juli	sonnig	mild	hoch	wenig	?
Februar	sonnig	kühl	normal	wenig	?
März	Regen	mild	normal	wenig	?
November	sonnig	mild	normal	stark	?
August	bedeckt	mild	hoch	stark	?
Juni	bedeckt	warm	normal	wenig	?
April	Regen	mild	hoch	stark	?

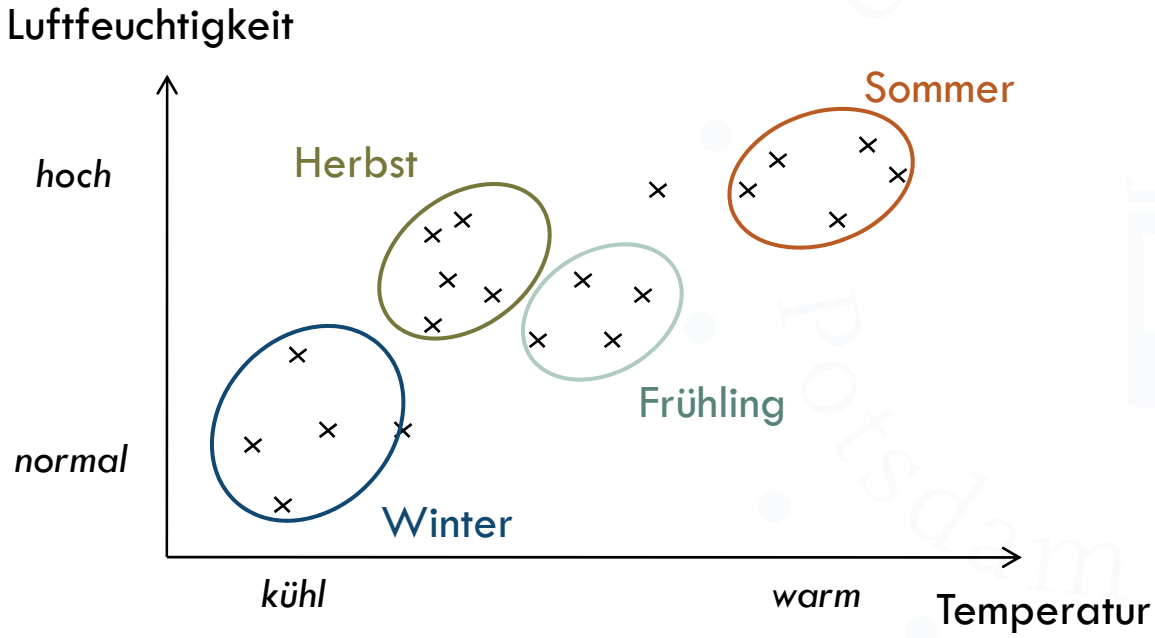
Trainingsdaten

Zielgröße

Unüberwachtes Lernen:

Beispiel

- Beispiel *Clustern* (Diagramm bzgl. der Attribute Luftfeuchtigkeit und Temperatur):



Motivation

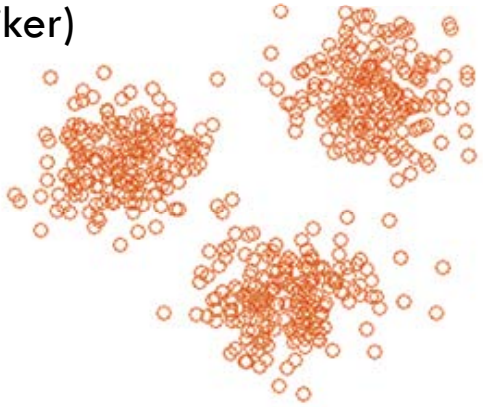
- Besseres Verständnis/Beschreibung der Daten:
  - Kundensegmentierung: Zielgerichtetes Marketing, Produktentwicklung usw. für einzelne Zielgruppen.
  - Landnutzung/Stadtplanung: Identifizieren von Regionen mit ähnlichen geographischen, klimatischen, städtebaulichen Eigenschaften.
  - Risikoanalyse: Klassen von Kunden mit unterschiedlichen Versicherungsrisiken erkennen.
- Teil der Datenvorverarbeitung für weitere Analyse:
  - Diskretisierung von numerischen Attributen.
  - Outlier-Detection.

Motivation:

Anwendung

- Überblick über eine Dokumentenkollektion.
  - Suche nach Stichwort „Kohl“ liefert viele Dokumente.
  - Idee: Zeige dem Nutzer Cluster um genauere Auswahl des Themas zu ermöglichen.

Helmut Kohl (Politiker)



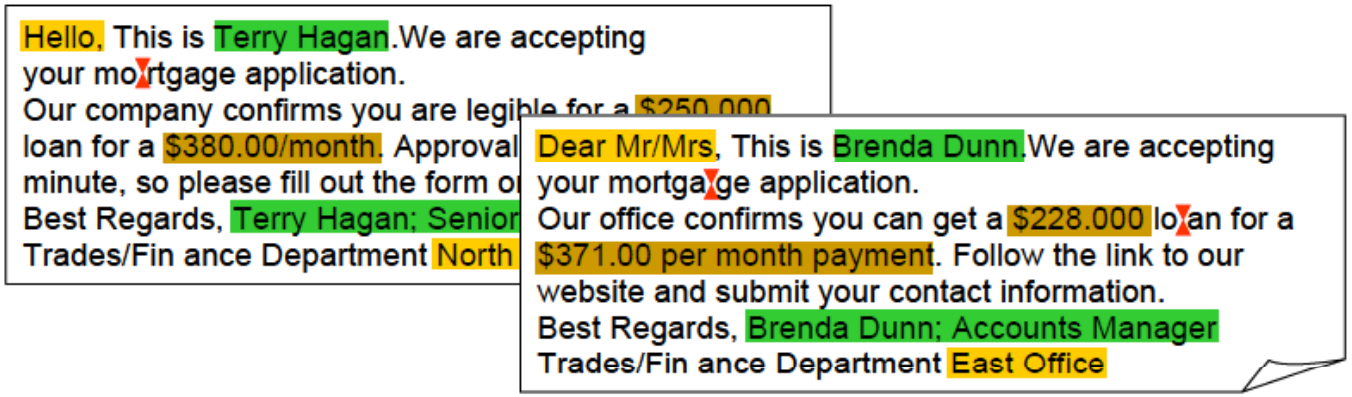
Kohl‘s (US Kaufhaus)

Kohl (Gemüse)

Motivation:

Anwendung

- Spam-Kampagnen identifizieren.
  - Spam-Kampagne ist große Menge ähnlicher (aber nicht gleicher) Emails.
  - Idee: Email-Attribute (z.B. enthaltene Wörter) clustern und Spam-Cluster durch nachgeschalteten Klassifikator erkennen.

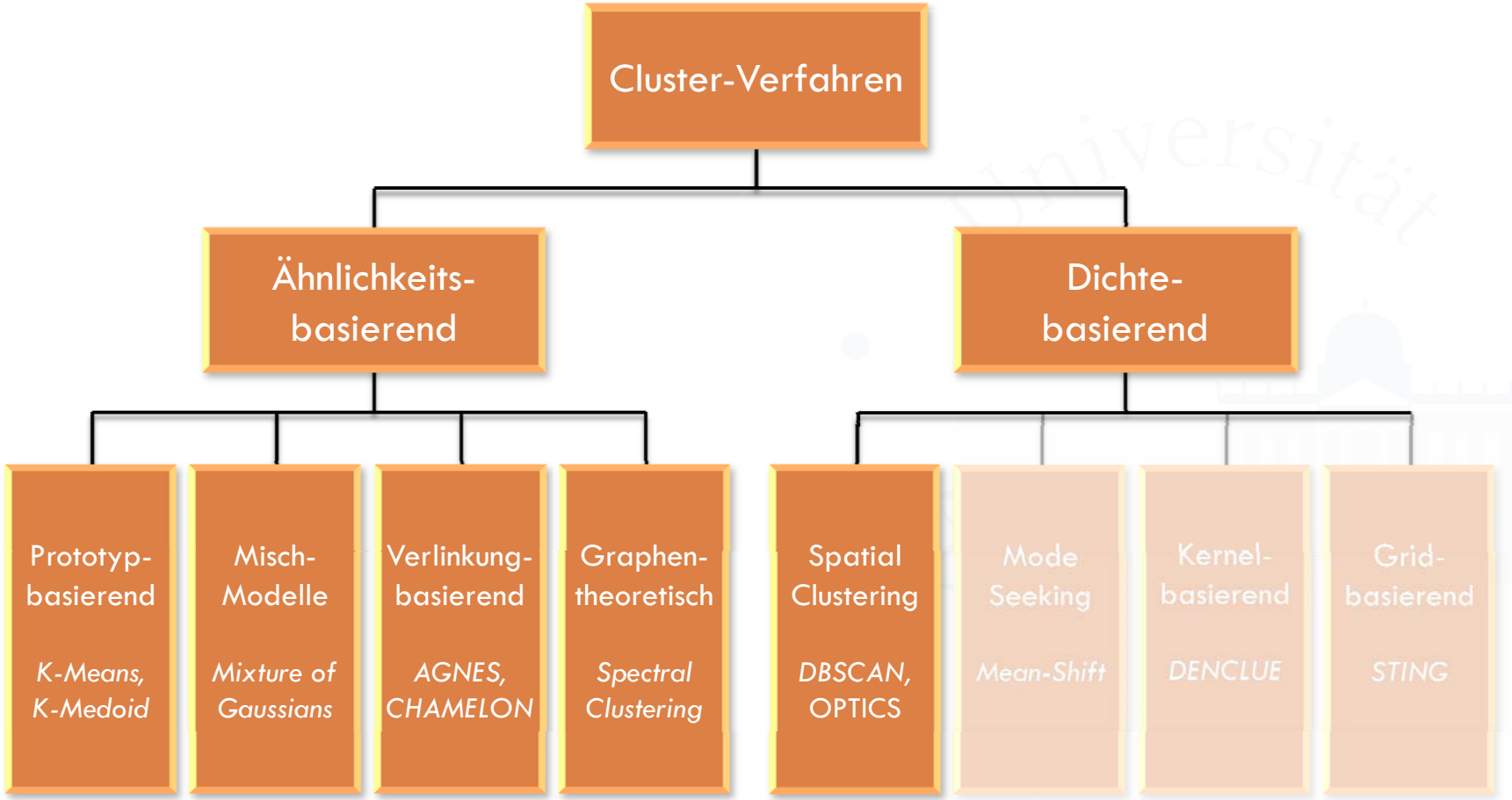


Evaluierung

- Qualitätsmerkmale eines Clusterings:
  - Hohe Ähnlichkeit zweier Datenpunkte eines Clusters (*intra-cluster similarity*).
  - Geringe Ähnlichkeit zwischen Datenpunkten verschiedener Cluster (*inter-cluster similarity*).
  - Anzahl, Form und Größenvarianz der Cluster.
  - Interpretierbarkeit, d.h. gefundene Cluster entsprechen echten (versteckten) Clustern.

Stichprobenartig durch Experten prüfen.

Cluster-Verfahren



Ähnlichkeitsbasierendes Clustern:

Prototyp-Verfahren

- Gegeben:
  - Ungelabelte Daten $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$.
  - Anzahl vermuteter Cluster k mit $1 < k < n$.
  - Ähnlichkeitsmaß zwischen Datenpunkten.
- Gesucht: Partitionierung der Daten in k Cluster.
- Ziel: Kleiner Abstand zw. Punkten im selben Cluster und großer Abstand zw. Punkten verschiedener Cluster.
  - Exponentiell viele Partitionierungen $\Rightarrow$ Suche NP-hart!
  - Heuristische Suche (lokal optimal): *K-Means* und *K-Medoids*.

Ähnlichkeitsbasierendes Clustern:

Prototyp-Verfahren: K-Means

- Gesucht ist eine Zuweisung $\mathbf{R} = \{\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_n\}$ der Daten $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ zu den Clustern mit

$$\mathbf{r}_i \in \{0,1\}^k \qquad r_{ij} = \begin{cases} 1 & \mathbf{x}_i \text{ in Cluster } j \\ 0 & \text{sonst} \end{cases}$$

  z.B. $\mathbf{r}_5 = [0 \quad 0 \quad 1 \quad 0]^T$ falls das 5. Beispiel in Cluster 3 liegt.

- Seien $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_k\}$ Cluster-Mittelpunkte (Prototypen).
- Ziel: Minimaler quadratischer Abstand zu den Cluster-Mittelpunkten:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{k} r_{ij} \left\| \mathbf{x}_i - \boldsymbol{\mu}_j \right\|^2$$

Ähnlichkeitsbasierendes Clustern:

Prototyp-Verfahren: K-Means

- Gleichzeitiges Minimieren über $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_k\}$ und $\{\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_n\}$ sehr schwierig.
- Idee: Abwechselnde Minimierung.
- Algorithmus:

K-Means(*Instanzen* $\mathbf{x}_i$, *Clusteranzahl* k)

Setze $l = 0$ und wähle zufällig $\forall i \ \boldsymbol{\mu}_i^0 = \mathbf{x}_j$

DO

$$\{\mathbf{r}_1^{l+1}, ..., \mathbf{r}_n^{l+1}\} = \arg\min_{\mathbf{r}_1,...,\mathbf{r}_n} \sum_{i=1}^{n} \sum_{j=1}^{k} r_{ij} \left\| \mathbf{x}_i - \boldsymbol{\mu}_j^l \right\|^2$$

Expectation

$$\{\boldsymbol{\mu}_1^{l+1}, ..., \boldsymbol{\mu}_n^{l+1}\} = \arg\min_{\boldsymbol{\mu}_1,...,\boldsymbol{\mu}_n} \sum_{i=1}^{n} \sum_{j=1}^{k} r_{ij}^{l+1} \left\| \mathbf{x}_i - \boldsymbol{\mu}_j \right\|^2$$

Maximization

$$l = l + 1$$

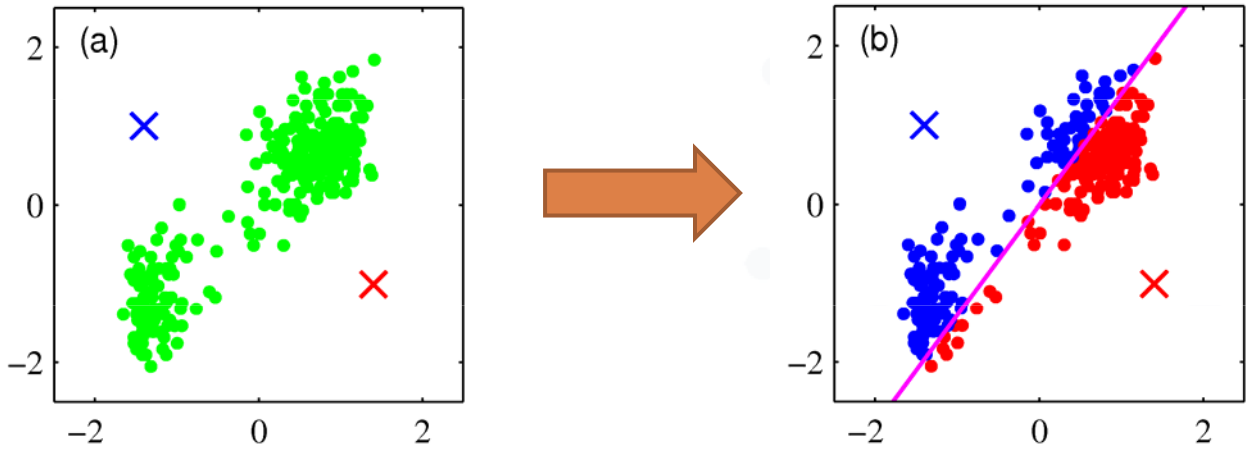
WHILE $\{\mathbf{r}_1^l, ..., \mathbf{r}_n^l\} \neq \{\mathbf{r}_1^{l-1}, ..., \mathbf{r}_n^{l-1}\}$

RETURN $\{\mathbf{r}_1^l, ..., \mathbf{r}_n^l\}$

Ähnlichkeitsbasierendes Clustern:

Prototyp-Verfahren: K-Means

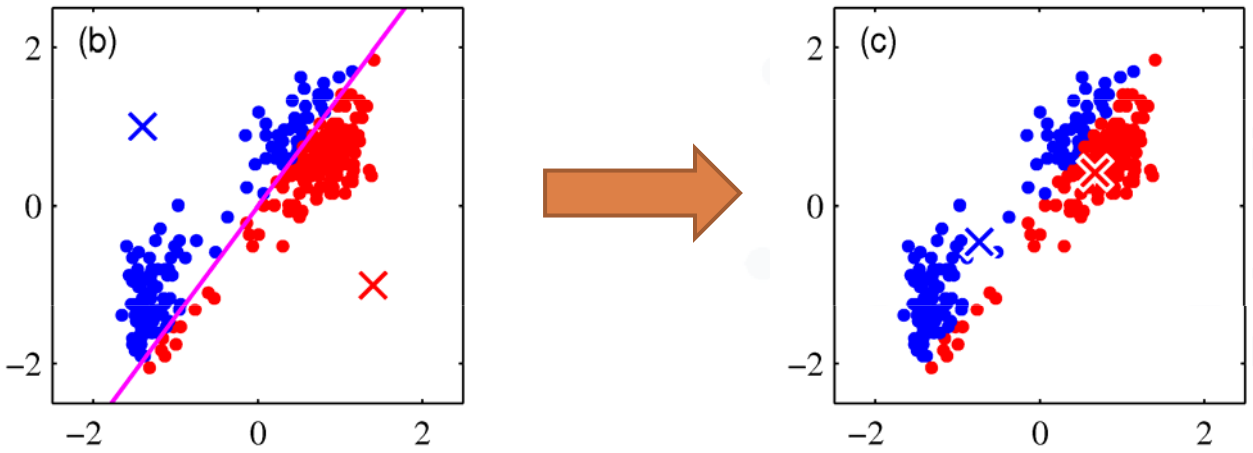
- Expectation-Schritt: $$\{\mathbf{r}_1^{l+1},\ldots,\mathbf{r}_n^{l+1}\} = \arg\min_{\mathbf{r}_1,\ldots,\mathbf{r}_n} \sum_{i=1}^{n}\sum_{j=1}^{k} r_{ij} \left\|\mathbf{x}_i - \boldsymbol{\mu}_j^l\right\|^2$$
  - Ordne jeden Punkt dem ihm nächsten Cluster-Mittelpunkt zu.



Ähnlichkeitsbasierendes Clustern:

Prototyp-Verfahren: K-Means

- Maximization-Schritt: $\{\boldsymbol{\mu}_1^{l+1},\ldots,\boldsymbol{\mu}_n^{l+1}\} = \arg\min_{\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_n} \sum_{i=1}^{n}\sum_{j=1}^{k} r_{ij}^{l+1} \left\|\mathbf{x}_i - \boldsymbol{\mu}_j\right\|^2$
  - Bestimme neue Cluster-Mittelpunkte $\boldsymbol{\mu}_j^{l+1} = \sum_{i=1}^{n} r_{ij}^{l+1}\mathbf{x}_i \Big/ \sum_{i=1}^{n} r_{ij}^{l+1}$.



Ähnlichkeitsbasierendes Clustern:

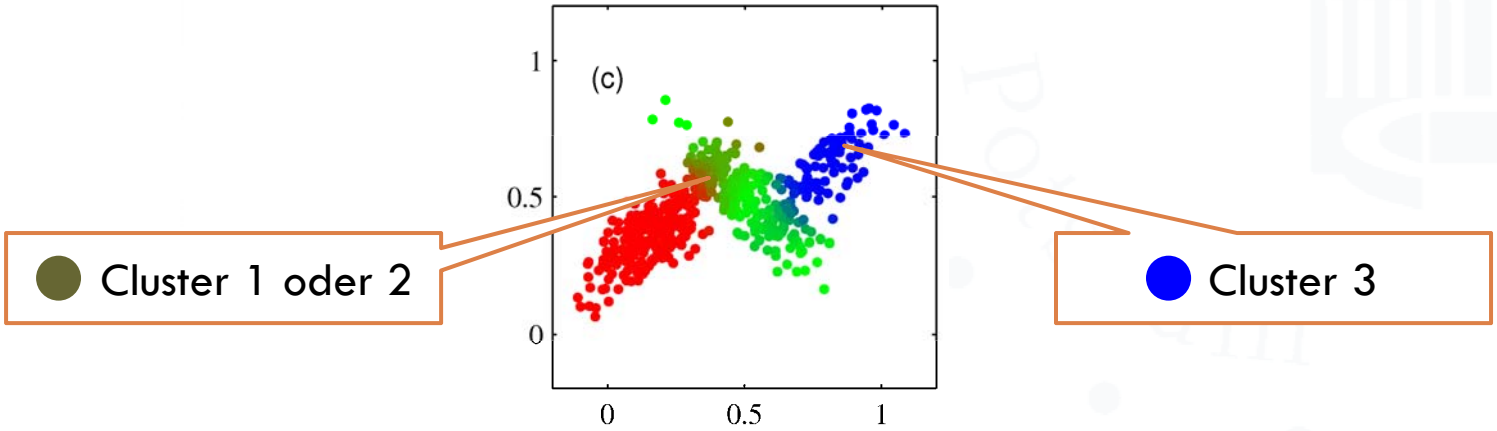
Prototyp-Verfahren: K-Means

- Vorteile:
  - Einfach zu implementieren
  - Relativ schnell.
    - In $O(n \cdot k)$ pro Iteration.
    - Effizienten Berechnung neuer Cluster-Mittelpunkte möglich.
- Nachteile:
  - Nur lokales Optimum garantiert (unterschiedliche Startwerte = unterschiedliche Lösungen).
  - Harte Zuweisung zu Clustern (nicht-probabilistisch).
  - Anzahl Cluster muss vorgeben werden.
  - Nur für nummerische Attribute geeignet.

Ähnlichkeitsbasierendes Clustern:

Mischmodelle

- Gegeben:
  - Ungelabelte Daten $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$.
  - Anzahl vermuteter Cluster k muss nicht bekannt sein.
  - Verteilungsannahme der Datenpunkte.
- Gesucht: Probabilistische Partitionierung der Daten.



Ähnlichkeitsbasierendes Clustern:

Mischmodelle

- Idee:
  - Generatives (Misch-)Modell welches Daten $\mathbf{X}$ erzeugt hat mit Modell-Paramter Θ.
  - Cluster-Zuordnungen $\mathbf{R} = \{\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_n\}$ sind versteckte Variablen des Modells.
- Ziel:
  - Parameter Θ mit maximaler A-Posteriori-Warscheinlichkeit (MAP):

$$\Theta^* = \arg\max_{\Theta} p(\Theta \mid \mathbf{X}) = \arg\max_{\Theta} p(\mathbf{X} \mid \Theta)\, p(\Theta)$$

Ähnlichkeitsbasierendes Clustern:

Mischmodelle: Mixture of Gaussians

- Gesucht ist eine Zuweisung $\{\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, ..., \boldsymbol{\pi}_n\}$ der Daten $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ zu den Clustern mit

$$\boldsymbol{\pi}_i \in [0,1]^k \qquad \sum_{j=1}^{k} \pi_{ij} = 1$$

- Annahme:
  - Daten-erzeugendes Modell ist Kombination von Gauß-Verteilungen mit unterschiedlichen Mittelwerten und Kovarianzen

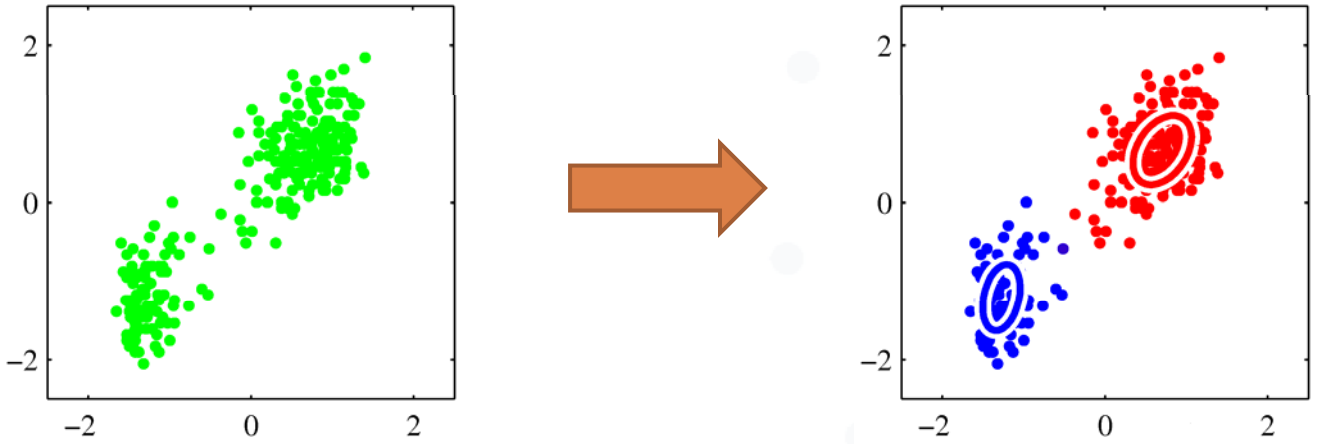
$$p(\mathbf{X} \mid \Theta) = \prod_{i=1}^{n} p(\mathbf{x}_i \mid \Theta) = \prod_{i=1}^{n} \left(\sum_{j=1}^{k} \pi_{ij} N(\mathbf{x}_i \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \right)$$

mit Parametern $\Theta = \left(\{\pi_{ij}\}, \{\boldsymbol{\mu}_j\}, \{\boldsymbol{\Sigma}_j\} \right)$.

Ähnlichkeitsbasierendes Clustern:

Mischmodelle: Mixture of Gaussians

- Schätzen der Parameter durch abwechselnde Optimierung analog zu K-Means (EM-Algorithmus).



Ähnlichkeitsbasierendes Clustern:

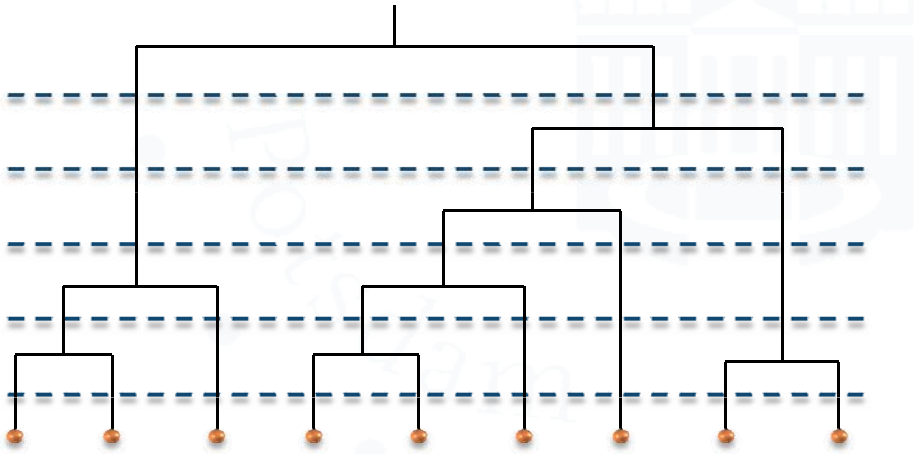
Mischmodelle: Mixture of Gaussians

- Vorteile:
  - Probabilistische Zuweisung zu Clustern.
  - Anzahl Cluster muss nicht vorgeben werden.
    - Automatischer Trade-off zwischen Anzahl Clustern und Anpassung an Daten.
- Nachteile:
  - Langsamer und komplexer als K-Means.
  - Nur für nummerische Attribute geeignet.

Ähnlichkeitsbasierendes Clustern:

Verlinkungsbasierte Verfahren

- Gegeben:
  - Ungelabelte Daten $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$.
  - Anzahl vermuteter Cluster k muss nicht bekannt sein.
  - Abstandsmaß *dist* zwischen Datenpunkten.
- Gesucht:
  - Darstellung der Daten in Form eines *Dendrogramms*.



Ähnlichkeitsbasierendes Clustern:

Verlinkungsbasierte Verfahren

- Idee:
  - *Agglomerative Hierarchical Clustering.*
    - Zu Beginn bildet jeder Datenpunkt einen eigenen Cluster.
    - Benachbarte Cluster werden sukzessive verschmolzen (bottom-up).
  - *Divisive Hierarchical Clustering.*
    - Zu Beginn bilden alle Daten einen gemeinsamen Cluster.
    - Cluster werden sukzessive gesplittet (top-down).
- Ziel:
  - Iteratives Aufbauen des Clusterings bis vorgegebene Qualität erreicht ist.

Ähnlichkeitsbasierendes Clustern:

Verlinkungsbasierte Verfahren: Agglomerative Nesting (AGNES)

- Algorithmus:

AGNES(*Instanzen* $\mathbf{x}_i$)

Setze $C_i = \{\mathbf{x}_i\} \ \forall i$

DO

$D_{ij} = dist(C_i, C_j) \ \forall i, j$

$(i^*, j^*) = \arg\min_{i,j} \left(D_{ij} \right)$

Verschmelze C_{i^*} und C_{j^*}

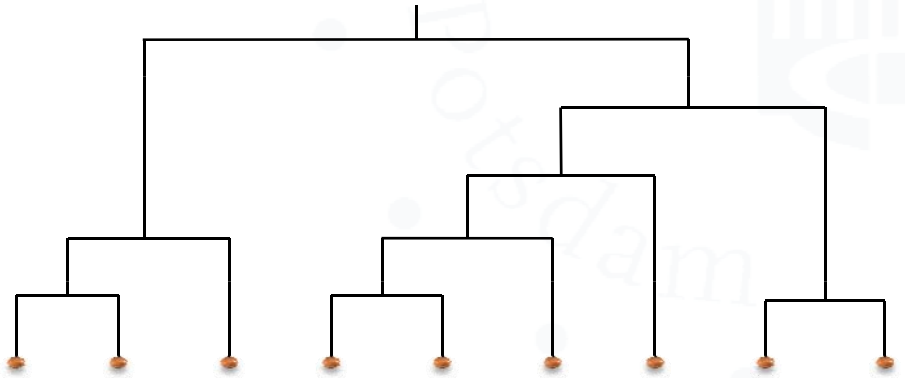
WHILE $D_{i^* j^*} < \varepsilon$

RETURN $\{C_i\}$

Jeder Datenpunkt ein eigener Cluster

Distanz zwischen zwei Clustern?

Mindest-Qualität (Abbruchbedingung)



Ähnlichkeitsbasierendes Clustern:

Verlinkungsbasierte Verfahren: Agglomerative Nesting (AGNES)

- Distanz zwischen zwei Clustern:
  - Single Linkage: $dist(C_i, C_j) = \min_{\mathbf{x} \in C_i, \mathbf{y} \in C_j} \left(dist(\mathbf{x}, \mathbf{y}) \right)$
  - Complete Linkage: $dist(C_i, C_j) = \max_{\mathbf{x} \in C_i, \mathbf{y} \in C_j} \left(dist(\mathbf{x}, \mathbf{y}) \right)$
  - Average Linkage: $dist(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{\mathbf{x} \in C_i, \mathbf{y} \in C_j} dist(\mathbf{x}, \mathbf{y})$
  - Average Group Linkage: $dist(C_i, C_j) = \frac{1}{|C_i \cup C_j|} \sum_{\mathbf{x}, \mathbf{y} \in C_i \cup C_j} dist(\mathbf{x}, \mathbf{y})$
  - Centroid: $dist(C_i, C_j) = dist\left(\frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} \mathbf{x}, \frac{1}{|C_j|} \sum_{\mathbf{y} \in C_j} \mathbf{y} \right)$

Ähnlichkeitsbasierendes Clustern:

Verlinkungsbasierte Verfahren: Agglomerative Nesting (AGNES)

- Vorteile:
  - Einfach zu implementieren
  - Relativ schnell.
  - Iteratives Verfahren, für Online-Clustering geeignet.
  - Für nominale, ordinale und numerische Attribute geeignet.
  - Anzahl Cluster muss nicht vorgeben werden.
- Nachteile:
  - Nur lokales Optimum garantiert.
  - Harte Zuweisung zu Clustern (nicht-probabilistisch).
  - Maß für Cluster-Distanz & Abbruchbedingung muss vorgegeben werden.

Ähnlichkeitsbasierendes Clustern:

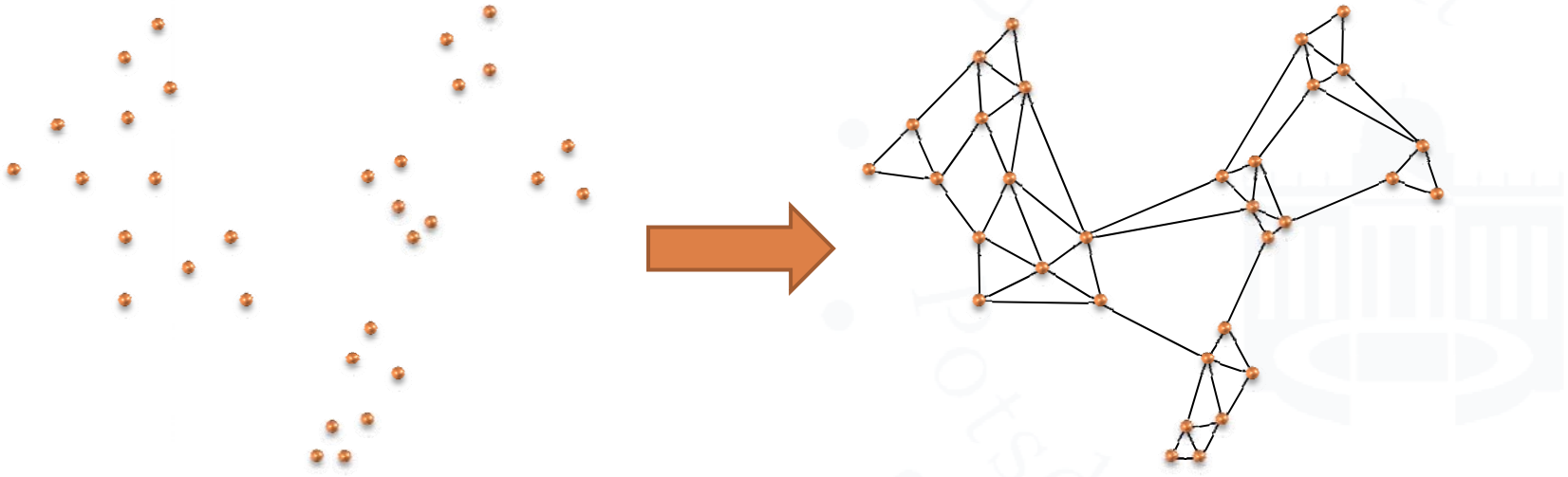
Graphentheoretische Verfahren

- Gegeben:
  - Ungelabelte Daten $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$.
  - Anzahl vermuteter Cluster k mit $1 < k < n$.
  - Ähnlichkeitsmaß (Kernel) sim zwischen Datenpunkten.
- Gesucht: Partitionierung der Daten in k Cluster.
- Ziel: Hohe Ähnlichkeit zw. Punkten im selben Cluster und geringe Ähnlichkeit zw. Punkten verschiedener Cluster.
  - Repräsentation der Daten als Graph und Partitionieren des Graphs.

Ähnlichkeitsbasierendes Clustern:

Graphentheoretische Verfahren

- Ähnlichkeit zwischen Datenpunkten (Knoten) bilden gewichtete Kanten:



Ähnlichkeitsbasierendes Clustern:

Graphentheoretische Verfahren

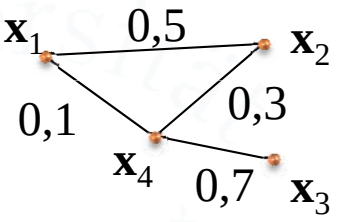
- Konstruktion des Graphen:
  - ε-Neighborhood-Graph: Verbinde Knoten $\mathbf{x}_i$ mit $\mathbf{x}_j$ falls $sim(\mathbf{x}_i, \mathbf{x}_j) > \varepsilon$, und setze Kantengewicht auf 1.
  - k-Nearest-Neighbor-Graph: Verbinde Knoten $\mathbf{x}_i$ mit $\mathbf{x}_j$ falls $\mathbf{x}_i$ einer der k-nächsten Nachbarn von $\mathbf{x}_j$ ist oder/und $\mathbf{x}_j$ einer der k-nächsten Nachbarn von $\mathbf{x}_i$ ist, und setze Kantengewicht auf $sim(\mathbf{x}_i, \mathbf{x}_j)$.
  - Vollständiger Graph: Verbinde alle Knoten $\mathbf{x}_i$ mit $\mathbf{x}_j$, und setze Kantengewicht auf $sim(\mathbf{x}_i, \mathbf{x}_j)$.
- Partitionierung des Graphen:
  - Kanten zwischen Clustern (Teilgraphen) haben geringe Gewichte (geringe inter-cluster similarity).
  - Kanten innerhalb eines Clusters haben hohe Gewichte (hohe intra-cluster similarity).

Ähnlichkeitsbasierendes Clustern:

Graphentheoretische Verfahren

- Repräsentation eines (ungericheten) Graphen:
  - Adjazenzmatrix:
    - Enthält Kantengewichte A_{ij}.

$$\mathbf{A} = \begin{bmatrix} 1 & 0,5 & 0 & 0,1 \\ 0,5 & 1 & 0 & 0,3 \\ 0 & 0 & 1 & 0,7 \\ 0,1 & 0,3 & 0,7 & 1 \end{bmatrix}$$



  - Knotengrad-Matrix:

$$\mathbf{D} = \begin{bmatrix} d_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_n \end{bmatrix} \qquad d_i = \sum_{j=1}^{n} A_{ij}$$

  - Laplace-Matrix:
    - Unnormalisiert: $\mathbf{L}_{un} = \mathbf{D} - \mathbf{A}$
    - Normalisiert (Random Walk): $\mathbf{L}_{rw} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}$
    - Symmetrisch normalisiert: $\mathbf{L}_{sym} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$

Ähnlichkeitsbasierendes Clustern:

Graphentheoretische Verfahren: Spectral Clustering

- Algorithmus:

SpecClust(*Instanzen* $\mathbf{x}_i$, *Clusteranzahl* k)

Konstruiere Graph aus *Instanzen* $\mathbf{x}_i$

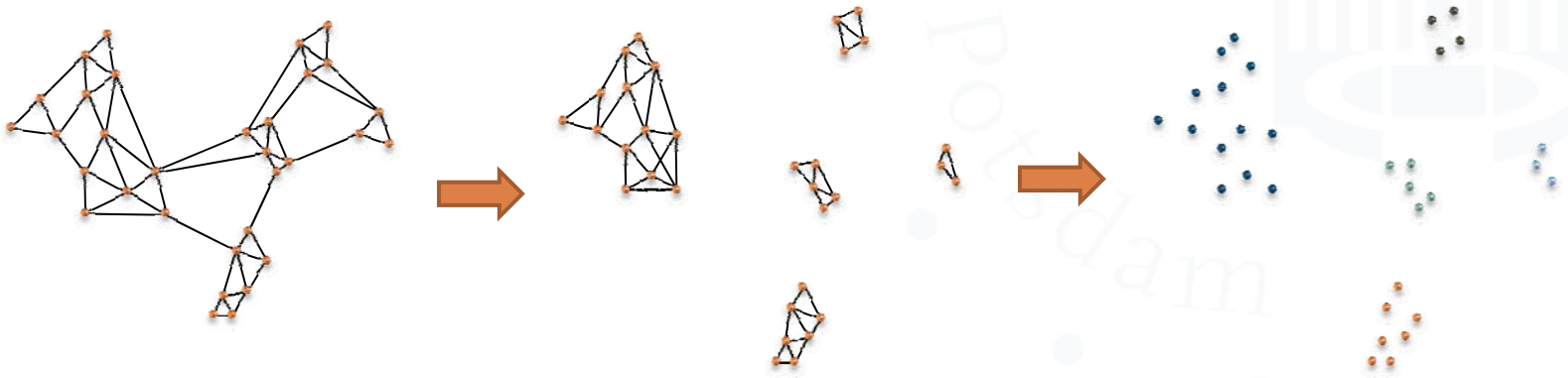
Berechne zugehörige Laplace-Matrix $\mathbf{L}$

Berechne die k Eigenvektoren $\mathbf{v}_i \in \mathbb{R}^n$ mit den k kleinsten Eigenwerten

Setze $\left[\mathbf{x}'_1 \quad \mathbf{x}'_2 \quad \cdots \quad \mathbf{x}'_n\right] = \left[\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_k\right]^{\mathrm{T}}$

$\{\mathbf{r}_1, \ldots, \mathbf{r}_n\}$ = K-Means(*Instanzen* $\mathbf{x}'_i$, *Clusteranzahl* k)

RETURN $\{\mathbf{r}_1, \ldots, \mathbf{r}_n\}$



Ähnlichkeitsbasierendes Clustern:

Graphentheoretische Verfahren: Spectral Clustering

- Vorteile:
  - Cluster können beliebige Form haben.
  - Einfach zu implementieren.
  - Relativ schnell (falls Laplace-Matrix sparse).
  - Meist hohe Qualität.
- Nachteile:
  - Nur lokales Optimum garantiert.
  - Harte Zuweisung zu Clustern (nicht-probabilistisch).
  - Anzahl Cluster muss vorgeben werden.

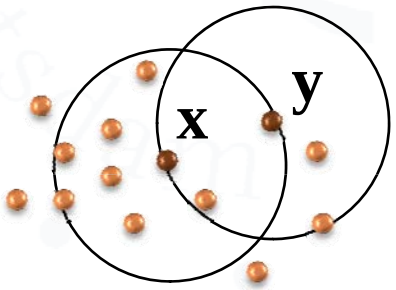
Dichtebasierendes Clustern:

Spatial Clustering

- Idee: Cluster-Grenzen sollen durch Regionen mit geringer Datendichte verlaufen.
  - Zwei Punkte $\mathbf{x}$ und $\mathbf{y}$ gehören zum gleichen Cluster falls es einen „Pfad“ zwischen beiden Punkten gibt.

- *Direkte Erreichbarkeit* des Punktes $\mathbf{y}$ von $\mathbf{x}$:
  - Nachbarschaftsbedingung: $\mathbf{y} \in N_\varepsilon(\mathbf{x}) = \{\mathbf{z} \mid dist(\mathbf{x}, \mathbf{z}) < \varepsilon\}$
  - Core-*Point*-Bedingung: $\#_{\min} \leq |N_\varepsilon(\mathbf{x})|$

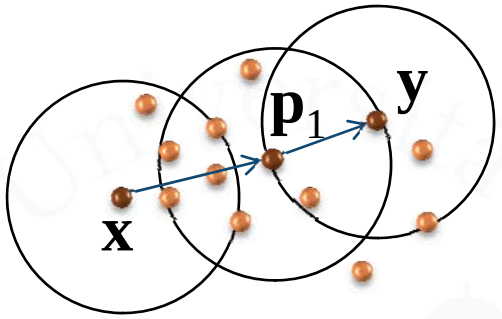
Mindestanzahl von benachbarten Punkten



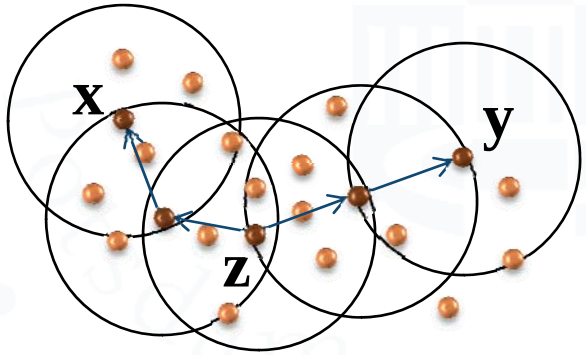
Dichtebasierendes Clustern:

Spatial Clustering

- Punkt $\mathbf{y}$ ist von $\mathbf{x}$ *erreichbar*:
  - Es gibt einen Pfad $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_m$ mit $\mathbf{p}_1 = \mathbf{x}$ und $\mathbf{p}_m = \mathbf{y}$, so dass $\mathbf{p}_{i+1}$ direkt erreichbar von $\mathbf{p}_i$ ist für $i = 1 \ldots m$.



- Punkt $\mathbf{y}$ ist mit $\mathbf{x}$ *verbunden*:
  - Es gibt einen Punkt $\mathbf{z}$, so dass $\mathbf{x}$ und $\mathbf{y}$ erreichbar von $\mathbf{z}$ sind.



Dichtebasierendes Clustern:

Spatial Clustering: DBSCAN

- Algorithmus:

DBSCAN(*Instanzen* $\mathbf{x}_i$, *Radius* ε, *Mindest-Clustergröße* g)

Setze $l = 0, D = \{\mathbf{x}_i\}$

DO

Wähle zufällig $\mathbf{x} \in D$

$P = \{\mathbf{y} \mid \mathbf{y}$ ist von $\mathbf{x}$ erreichbar$\}$

IF $|P| < g$

$D = D \setminus \mathbf{x}$

ELSE

l=l+1

$C_l = P$

$D = D \setminus P$

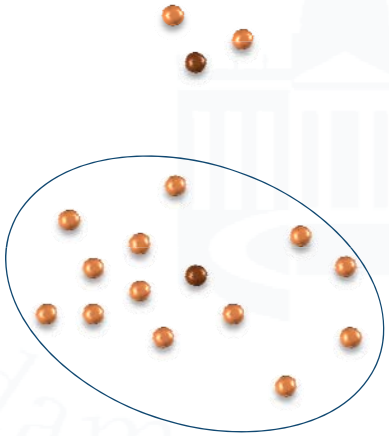
WHILE $D \neq \varnothing$

RETURN $\{C_1, \ldots, C_l\}$

P enthält $\mathbf{x}$

$\mathbf{x}$ ist Outlier (nicht Teil eines Clusters)

$\mathbf{x}$ ist Core-Point eines Clusters



Dichtebasierendes Clustern:

Spatial Clustering: DBSCAN

- Vorteile:
  - Cluster können beliebige Form haben.
  - Für nominale, ordinale und numerische Attribute geeignet.
  - Anzahl Cluster muss nicht vorgeben werden.
- Nachteile:
  - Nur lokales Optimum garantiert.
  - Harte Zuweisung zu Clustern (nicht-probabilistisch).
  - Radius & Mindest-Clustergröße müssen vorgegeben werden.

Zusammenfassung

- Prototyp-Verfahren (z.B. K-Means):
  - Schnell, numerische Attribute, bekannte Clusteranzahl.
- Mischmodelle (z.B. Mixture of Gaussians):
  - Probabilistische Cluster-Zuweisung, numerische Attribute, unbekannte Clusteranzahl.
- Verlinkungsbasierte Verfahren (z.B. AGNES):
  - Iterativ, beliebige Attribute, unbekannte Clusteranzahl.
- Graphbasierte Verfahren (z.B. Spectral Clustering):
  - Beliebige Clusterform, hohe Qualität.
- Spatial Clustering (z.B. DBSCAN):
  - Beliebige Clusterform, unbekannte Clusteranzahl.