Ludwig-Maximilians-Universität München
Institut für Informatik
Lehr- und Forschungseinheit für Datenbanksysteme

Skript zur Vorlesung

Knowledge Discovery in Databases

im Sommersemester 2013

Kapitel 3: Clustering

Vorlesung: Dr. Arthur Zimek
Übungen: Erich Schubert

http://www.dbs.ifi.lmu.de/cms/Knowledge_Discovery_in_Databases_I_(KDD_I)

Ziel des Clustering

Identifikation einer endlichen Menge von Kategorien, Klassen oder Gruppen (*Cluster*) in den Daten.

Ähnliche Objekte sollen im *gleichen* Cluster sein, *unähnliche* Objekte sollen in *unterschiedlichen* Clustern sein.

Einleitung

- Clustering ist unsupervised, d.h. wir haben keine äußeren Anhaltspunkte zur Steuerung/Überwachung (supervision) des Verfahrens:
  - keine Regeln zur Einordnung der Punkte in Cluster lernbar
  - wir wissen nicht, wie viele Cluster vorhanden sind
  - wir wissen nicht, wie die einzelnen Cluster charakterisiert sind
  - keine eindeutige Beurteilung der Qualität eines gefundenen Clusterings (Evaluation)

Einleitung

- Herausforderungen
  - gegeben: eine hypothetische Funktion f, die für eine Menge von n Datenobjekten entscheidet, ob sie einen (guten) Cluster bilden
  - naive Methode: werte Funktion f aus für alle möglichen Partitionierungen in k Teilmengen von Objekten ($2 \leq k \leq ?$)
  - Problem:
    - Es gibt $O(k^n)$ viele Partitionierungen in k Teilmengen.
    - Wie sieht diese Funktion f überhaut aus?
  - Lösung: wir brauchen ***Heuristiken*** für beide Teilprobleme
    - Effiziente Suche nach Lösungen
    - Effiziente und effektive Modellierung der hypothetischen Funktion f

  => es gibt sehr viele verschiedene Clustering-Algorithmen

Typen von Clustering Verfahren

- Partitionierende Verfahren
  - Modell: Cluster sind kompakt zusammenliegende Datenobjekte
  - Parameter: (meist) Anzahl k der Cluster (d.h. Annahme: Anzahl der Cluster bekannt), Distanzfunktion
  - sucht ein „flaches“ Clustering (Partitionierung in k Cluster mit maximaler Kompaktheit)
- Dichte-basierte Verfahren
  - Modell: Cluster sind Räume mit hoher Punktdichte separiert durch Räume niedriger Punktdichte
  - Parameter: minimale Dichte in einem Cluster, Distanzfunktion (d.h. Annahme: erwartete Dichte für Cluster bekannt)
  - sucht flaches Clustering: typischer Ansatz erweitert Punkte um ihre Nachbarn solange Dichte groß genug

Typen von Clustering Verfahren

- Hierarchische Verfahren
  - Modell: Kompaktheit, Dichte, …
  - Parameter: Distanzfunktion für Punkte und für Cluster
  - bestimmt Hierarchie von Clustern (z.B. in Form eines Baumes darstellbar), mischt jeweils die ähnlichsten Cluster
  - Flaches Clustering kann durch Abschneiden des Baumes erzeugt werden.
- Andere Clustering-Verfahren
  - Fuzzy Clustering
  - Graph-theoretische Verfahren
  - Neuronale Netze
  - …

Partitionierende Verfahren

Grundlagen

Ziel: Partitionierung in k Cluster, so dass eine Kostenfunktion minimiert wird (Gütekriterium: Kompaktheit)

Zentrale Annahmen:

- Anzahl k der Cluster bekannt (Eingabeparameter)
- Clustercharakteristik: Kompaktheit
- Kompaktheit: Abweichung aller Objekte im Cluster von einem ausgezeichneten Cluster-Repräsentanten ist minimal
- Kompaktheitskriterium führt meistens zu sphärisch geformten Clustern

Partitionierende Verfahren

Grundlagen

Erschöpfende (globale) Suche ist zu ineffizient (WARUM?)

Daher: Lokal optimierende Verfahren

- wähle k initiale Cluster-Repräsentanten
- optimiere diese Repräsentanten iterativ
- ordne jedes Objekt seinem ähnlichsten Repräsentanten zu

Typen von Cluster-Repräsentanten

- Mittelwert des Clusters (*Konstruktion zentraler Punkte*)
- Element des Clusters (*Auswahl repräsentativer Punkte*)
- Wahrscheinlichkeitsverteilung des Clusters (*Erwartungsmaximierung*)

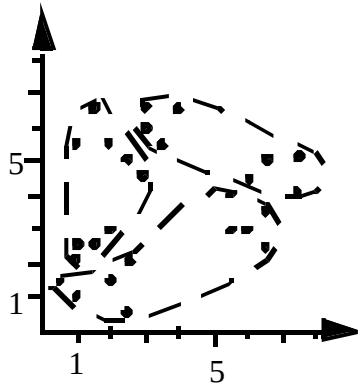
Partitionierende Verfahren

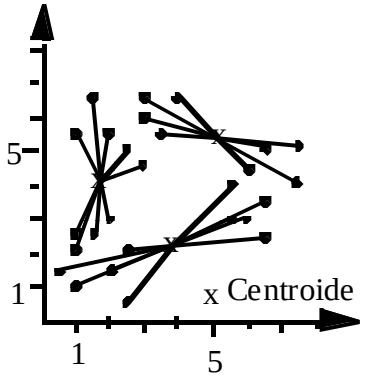
Konstruktion zentraler Punkte (Beispiel)

Cluster

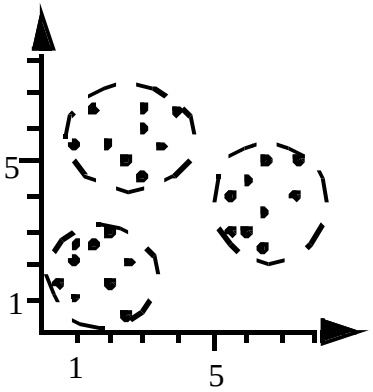
Cluster-Repräsentanten

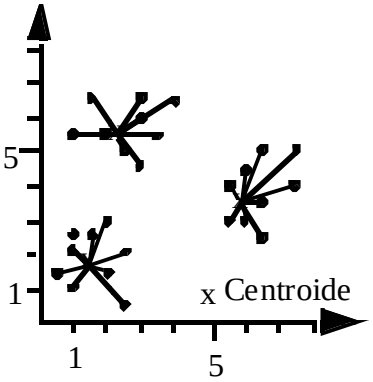
schlechtes Clustering





optimales Clustering





Konstruktion zentraler Punkte (Grundbegriffe)

[Forgy 1965]

- Objekte sind Punkte $x=(x_1,\ldots,x_d)$ in einem euklidischen Vektorraum

  dist = euklidische Distanz (L_2-Norm)
- *Centroid* μ_C: Mittelwert aller Punkte im Cluster *C*
- *Maß für die Kosten* (Kompaktheit) *eines Clusters C*

$$TD^2(C) = \sum_{p \in C} dist(p, \mu_C)^2$$

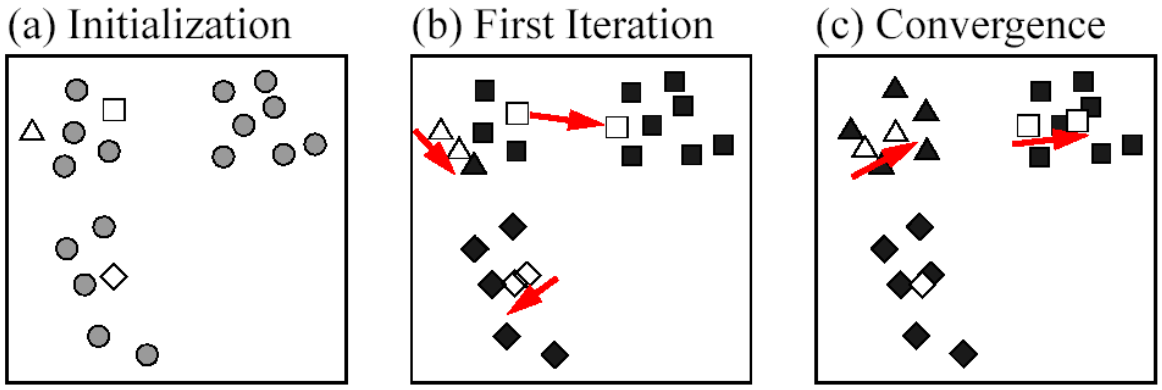
- *Maß für die Kosten* (Kompaktheit) *eines Clustering*

$$TD^2(C_1,\ldots,C_k) = \sum_{i=1}^{k} TD^2(C_i)$$

Partitionierende Verfahren

Idee des Algorithmus

- Algorithmus startet mit (z.B. zufällig gewählten) Punkten als Cluster-Repräsentanten
- Der Algorithmus besteht aus zwei alternierenden Schritten:
  - Zuordnung jedes Datenpunktes zum räumlich nächsten Repräsentanten
  - Neuberechnung der Repräsentanten (Centroid der zugeordneten Punkte)
- Diese Schritte werden so lange wiederholt, bis sich keine Änderung mehr ergibt



Algorithmus [Lloyd 1957]

ClusteringDurchVarianzMinimierung(Punktmenge D, Integer k)

Erzeuge eine „initiale“ Zerlegung der Punktmenge D in k Klassen;

Berechne die Menge $C'=\{C_1, ..., C_k\}$ der Centroide für die k Klassen;

C = {};

repeat

C = C’;

Bilde k Klassen durch Zuordnung jedes Punktes zum nächstliegenden Centroid aus C;

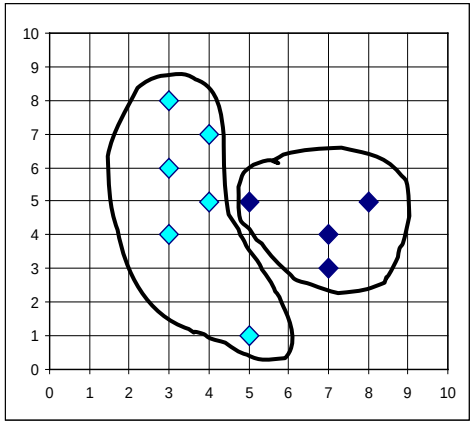
Berechne die Menge $C'=\{C'_1, ..., C'_k\}$ der Centroide für die neu bestimmten Klassen;

until C = C’;

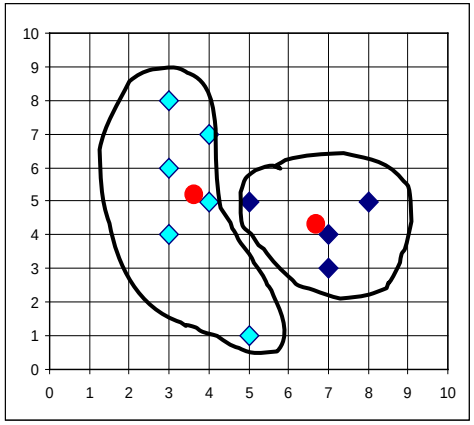
return C;

Partitionierende Verfahren

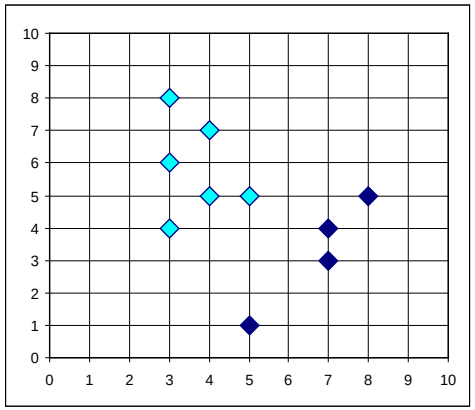
Beispiel



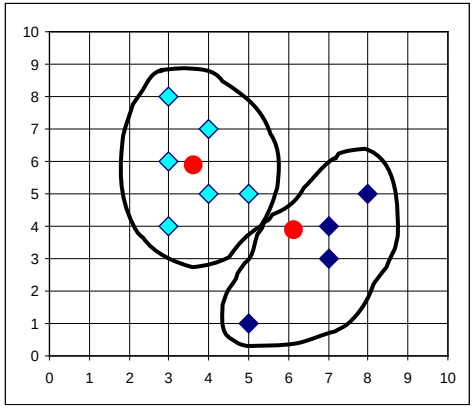
Berechnung der neuen Centroide



Zuordnung zum nächsten Centroid



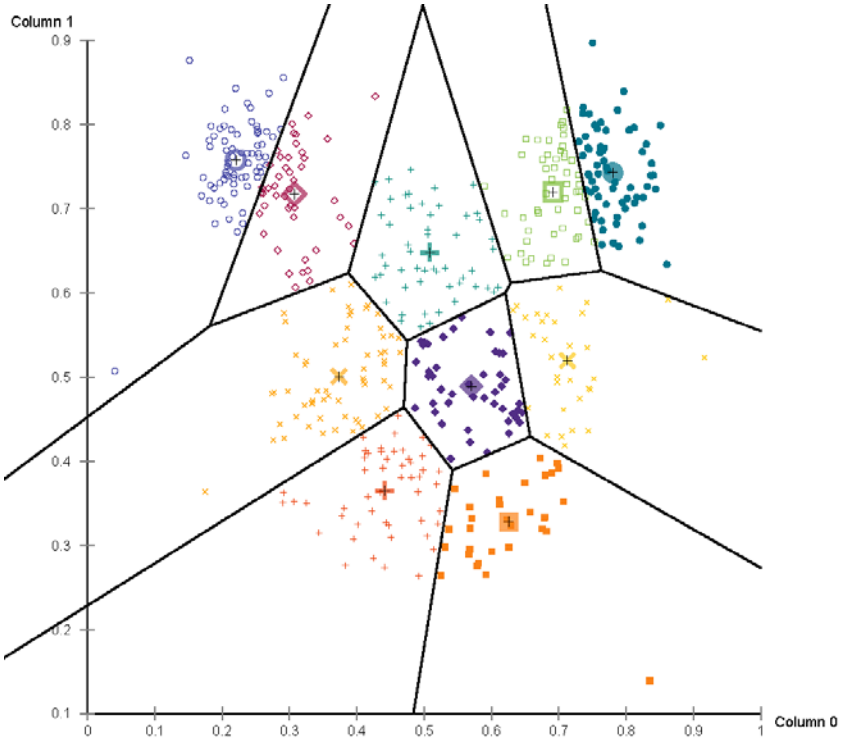
Berechnung der neuen Centroide

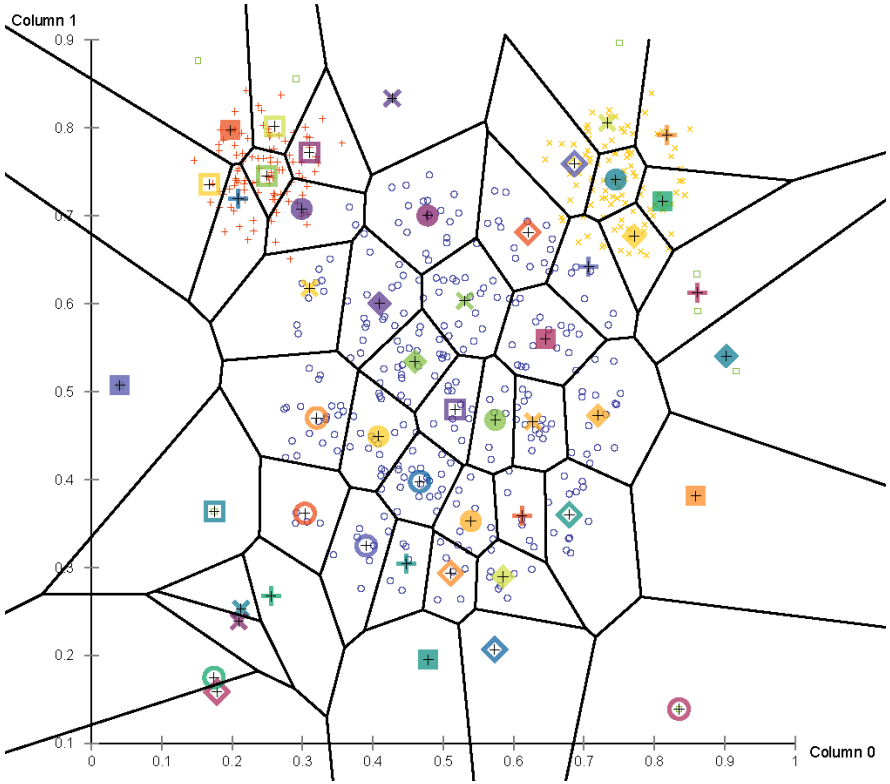


Partitionierende Verfahren

Model:

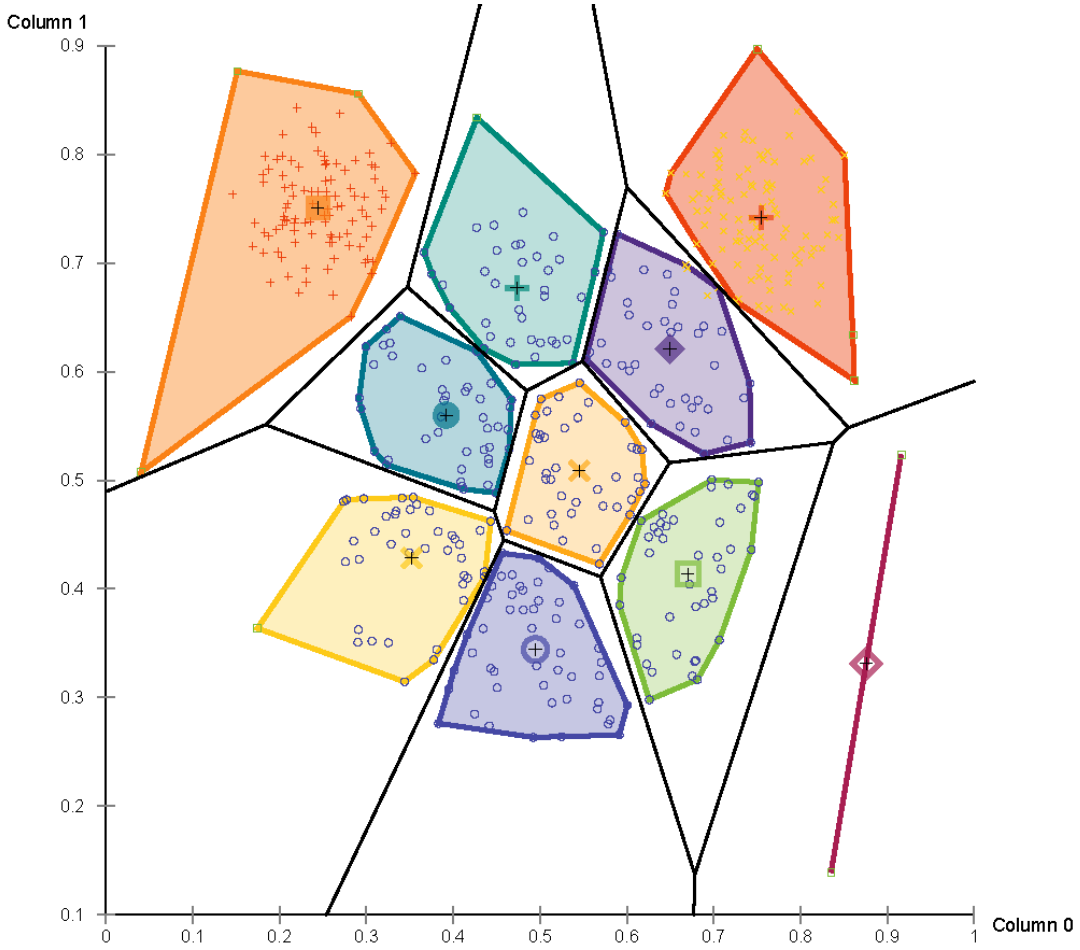
- Zuordnung der Punkte zum nächst-gelegenen Cluster-Repräsentanten
- entspricht Voronoi-Parzellierung





Partitionierende Verfahren

Voronoi-Parzellierung ≠ konvexe Hülle



Bekannteste Variante des Basis-Algorithmus

k-means [MacQueen 67]

Idee: die betroffenen Centroide werden direkt aktualisiert, wenn ein Punkt seine Clusterzugehörigkeit ändert

- *k*-means hat im wesentlichen die Eigenschaften des Basis-Algorithmus
- *k*-means ist aber reihenfolgeabhängig

Achtung:

Der Name „*k*-means“ wird oft undifferenziert für verschiedene Varianten der Grundidee verwendet, insbesondere auch für den Algorithmus von Lloyd.

Partitionierende Verfahren

Diskussion

+ Effizienz
  Aufwand: O($k \cdot n$) für eine Iteration (k kann für kleine Anzahl von Clustern vernachlässigt werden), Anzahl der Iterationen ist im allgemeinen klein (~ 5 - 10).
+ einfache Implementierung

  $\Rightarrow$ populärstes partitionierendes Clustering-Verfahren

– Anfälligkeit gegenüber Rauschen und Ausreißern
  (alle Objekte gehen in die Berechnung des Zentroids ein)
– Cluster müssen konvexe Form haben
– die Anzahl k der Cluster ist oft schwer zu bestimmen
– starke Abhängigkeit von der initialen Zerlegung
  (sowohl Ergebnis als auch Laufzeit)

Auswahl repräsentativer Punkte

[Kaufman & Rousseeuw 1990]

- setze nur Distanzfunktion (*dist*) für Paare von Objekten voraus
- *Medoid*: ein zentrales Element des Clusters (repräsentatives Objekt)
- Maß für die Kosten (Kompaktheit) eines Clusters C mit Medoid m_C

$$TD(C) = \sum_{p \in C} dist(p, m_C)$$

- Maß für die Kosten (Kompaktheit) eines Clustering

$$TD(C_1, \ldots, C_k) = \sum_{i=1}^{k} TD(C_i)$$

Überblick über k-medoid Algorithmen

PAM [Kaufman & Rousseeuw 1990]

- Greedy-Algorithmus:
  in jedem Schritt wird nur ein Medoid mit einem Nicht-Medoid vertauscht
- vertauscht in jedem Schritt das Paar (Medoid, Nicht-Medoid), das die größte Reduktion der Kosten TD bewirkt

CLARANS [Ng & Han 1994]

- zwei zusätzliche Parameter: *maxneighbor* und *numlocal*
- höchstens *maxneighbor* viele von zufällig ausgewählten Paaren (Medoid, Nicht-Medoid) werden betrachtet
- die erste Ersetzung, die überhaupt eine Reduzierung des *TD*-Wertes bewirkt, wird auch durchgeführt
- die Suche nach *k* „optimalen“ Medoiden wird *numlocal* mal wiederholt

Algorithmus PAM

```
PAM(Punktmenge D, Integer k)
   Initialisiere die k Medoide;
   TD_Änderung := −∞;
   while TD_Änderung < 0 do
      Berechne für jedes Paar (Medoid M, Nicht-Medoid N)
        den Wert TD_{N↔M};
      Wähle das Paar (M, N), für das der Wert
        TD_Änderung := TD_{N↔M} − TD minimal ist;
      if TD_Änderung < 0 then
         ersetze den Medoid M durch den Nicht-Medoid N;
         Speichere die aktuellen Medoide als die bisher beste
           Partitionierung;
   return Medoide;
```

Algorithmus CLARANS

```
CLARANS(Punktmenge D, Integer k,
        Integer numlocal, Integer maxneighbor)
   for r from 1 to numlocal do
      wähle zufällig k Objekte als Medoide; i := 0;
      while i < maxneighbor do
         Wähle zufällig (Medoid M, Nicht-Medoid N);
         Berechne TD_Änderung := TD_{N↔M} – TD;
         if TD_Änderung < 0 then
           ersetze M durch N;
           TD := TD_{N↔M}; i := 0;
         else i:= i + 1;
      if TD < TD_best then
         TD_best := TD; Speichere aktuelle Medoide;
   return Medoide;
```

Vergleich von PAM und CLARANS

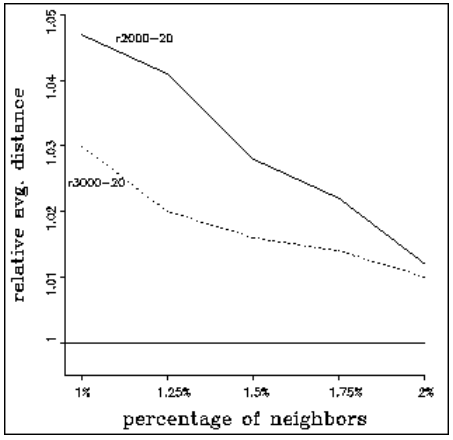
Laufzeitkomplexitäten

PAM: $O(n^3 + k(n-k)^2 * \#Iterationen)$

CLARANS: $O(numlocal * maxneighbor * \#Ersetzungen * n)$
praktisch $O(n^2)$

Experimentelle Untersuchung

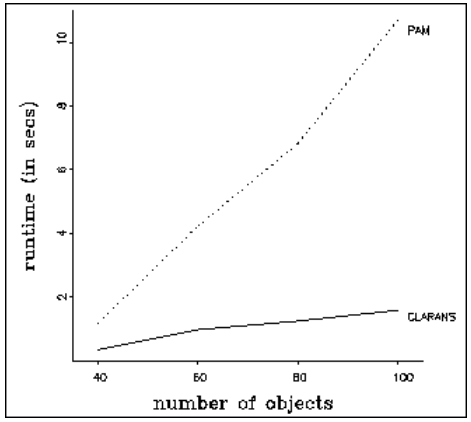
Qualität



TD(CLARANS)

TD(PAM)

Laufzeit



Erwartungsmaximierung (EM)

[Dempster, Laird & Rubin 1977]

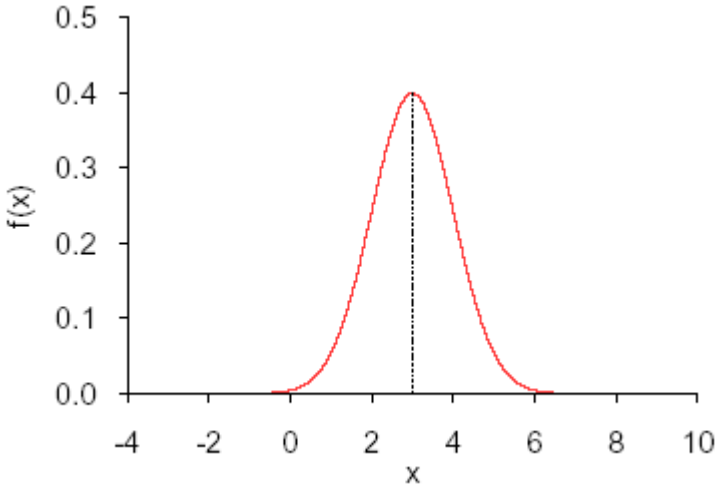
- Objekte sind Punkte x = (x_1, ..., x_d) in einem euklidischen Vektorraum
- Ein Cluster wird durch eine Wahrscheinlichkeitsverteilung beschrieben
- typisch: Modell für einen Cluster ist eine multivariate Normalverteilung
- Repräsentation eines Clusters *C*
  - Mittelwert μ_C aller Punkte des Clusters (Centroid)
  - *d x d* Kovarianzmatrix Σ_C für die Punkte im Cluster *C*
- Wahrscheinlichkeitsdichte eines Clusters *C*

$$P(x \mid C) = \frac{1}{\sqrt{(2\pi)^d \mid \Sigma_C \mid}} e^{-\frac{1}{2}\cdot(x-\mu_C)^T \cdot (\Sigma_C)^{-1} \cdot (x-\mu_C)}$$

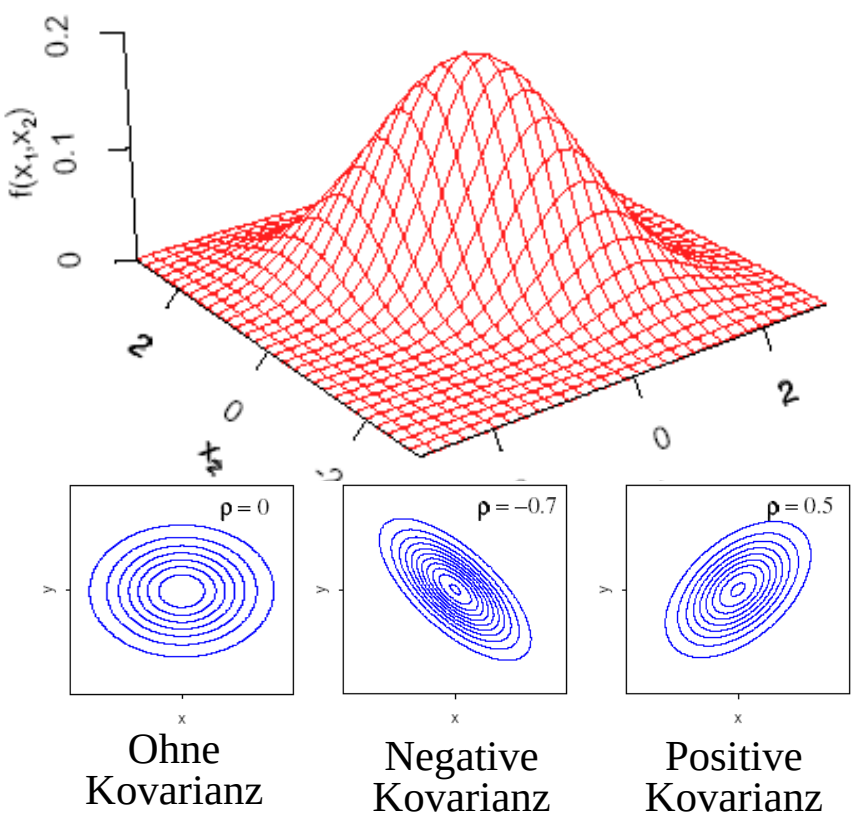
Partitionierende Verfahren

Multivariate Normalverteilung

Univariate Normalverteilung



Bivariate Normalverteilung



Partitionierende Verfahren

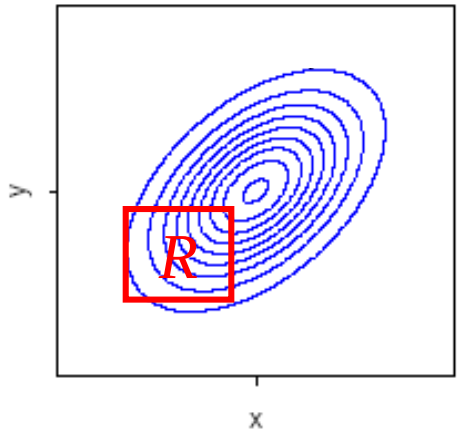
Idee des EM-Algorithmus:

- Jeder Punkt gehört zu mehreren (eigentlich *allen*) Clustern, jeweils mit unterschiedlicher Wahrscheinlichkeit, abh. v. $P(x|C)$
- Algorithmus besteht wieder aus zwei alternierenden Schritten:
  - Zuordnung von Punkten zu Clustern (hier nicht absolut sondern relativ/nach Wahrscheinlichkeit)
  - Neuberechnung der Cluster-Repräsentanten (multivariate Normalverteilungen)
- Alles muss auf eine stochastische Grundlage gestellt werden:
  - Bei Berechnung der Clusterzentren (μ_C) muss berücksichtigt werden, dass Punkte Clustern nicht absolut, sondern nur relativ zugeordnet sind
  - Wie groß ist die Wahrscheinlichkeit der Clusterzugehörigkeit?

Partitionierende Verfahren

Jeder Cluster C_i wird durch eine Wahrscheinlichkeits-Dichte-Funktion (Normalverteilung) modelliert:

$$P(x \mid C_i) = \frac{1}{\sqrt{(2\pi)^d \left|\Sigma_{C_i}\right|}} e^{-\frac{1}{2}\cdot(x-\mu_{C_i})^T \cdot (\Sigma_{C_i})^{-1} \cdot (x-\mu_{C_i})}$$

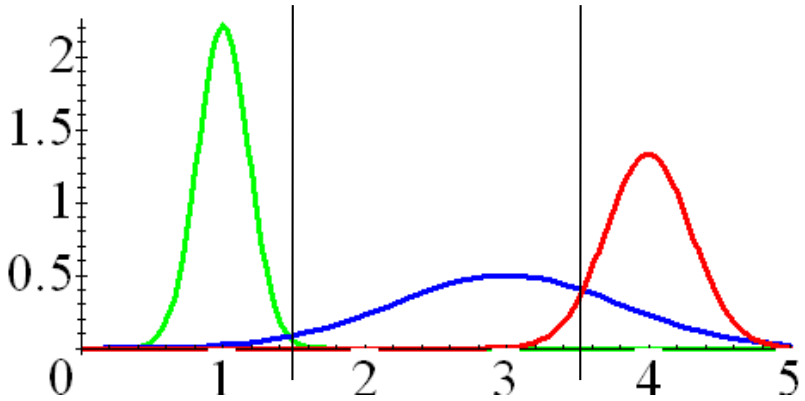


Dichtefunktion:

- Integral über den Gesamtraum ergibt 1
- Integral über Region *R* ergibt Wahrscheinlichkeit, dass ein fiktiver Punkt des Clusters in dieser Region des Clusters liegt, bzw. den relativen Anteil (z.B. 30%) der Punkte des Clusters, die in *R* liegen

Partitionierende Verfahren

$$P(x \mid C_i) = \frac{1}{\sqrt{(2\pi)^d \left|\Sigma_{C_i}\right|}} e^{-\frac{1}{2}\cdot(x-\mu_{C_i})^{\mathbf{T}}\cdot\left(\Sigma_{C_i}\right)^{-1}\cdot(x-\mu_{C_i})}$$



Interpretation der Wahrscheinlichkeit:

- Dies würde unter der Voraussetzung gelten, dass der Punkt *x* ausschließlich dem Cluster C_i zugeordnet wäre (was nicht stimmt)
- Deshalb Notation als *bedingte* Wahrscheinlichkeit

Partitionierende Verfahren

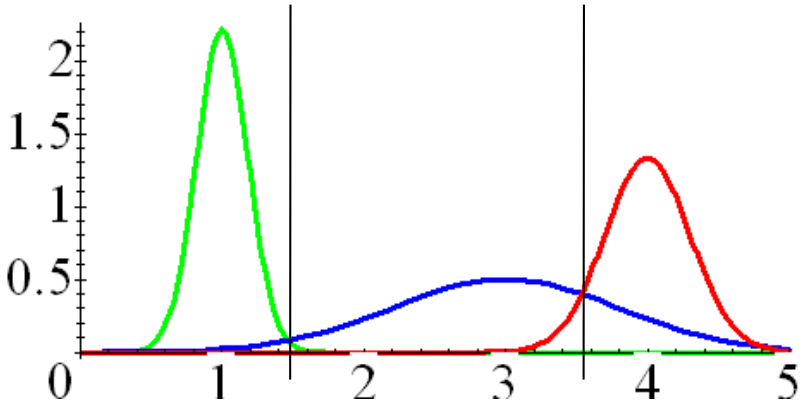
Bei *k* Gauß-Verteilungen (durch *k* Cluster) ergibt sich folgende Gesamt-Wahrscheinlichkeitsdichte:

$$P(x) = \sum_{i=1}^{k} W_i \cdot P(x \mid C_i)$$

wobei W_i der relative Anteil der Datenpunkte ist, der zum Cluster C_i gehört (z.B. 5%), was man auch als Gesamt-Wahrscheinlichkeit des Clusters $P(C_i)$ interpretieren kann.

Satz von Bayes:

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$



Mit dem *Satz von Bayes* kann man die Wahrscheinlichkeit bestimmen, dass ein gegebener Punkt *x* zum Cluster C_i gehört, geschrieben als bedingte Wahrscheinlichkeit $P(C_i|x)$

$$P(C_i \mid x) = W_i \cdot \frac{P(x \mid C_i)}{P(x)}$$

Partitionierende Verfahren

- Maß für die Güte eines Clustering *M*

$$E(M) = \sum_{x \in D} \log(P(x))$$

$\Rightarrow$ *E*(*M*) soll maximiert werden.

- Anteil des Clusters an der Datenmenge:

$$W_i = P(C_i) = \frac{1}{n} \sum_{j=1}^{n} P(C_i \mid x_j)$$

- Mittelwert und Kovarianzmatrix der Gaußverteilung:

$$\mu_i = \frac{\sum_{x \in D} x \cdot P(C_i \mid x)}{\sum_{x \in D} P(C_i \mid x)} \qquad \Sigma_i = \frac{\sum_{x \in D} (x - \mu_i)(x - \mu_i)^{\mathrm{T}} \cdot P(C_i \mid x)}{\sum_{x \in D} P(C_i \mid x)}$$

Partitionierende Verfahren

Algorithmus

ClusteringDurchErwartungsmaximierung(Punktmenge D, Integer k)

Erzeuge ein „initiales“ Modell M’ = (C_1’, ..., C_k’);

repeat // „Neuzuordnung“

Berechne P(x|C_i), P(x) und P(C_i|x) für jedes Objekt aus D und jede Gaußverteilung/jeden Cluster C_i;

// „Neuberechnung des Modells“

Berechne ein neues Modell M ={C_1, ..., C_k} durch Neuberechnung von W_i, μ_C und Σ_C für jedes i;

M’ := M;

until |E(M) - E(M’)| < ε;

return M;

Diskussion

- Aufwand:

  O(*n* ∗ |M| * *#Iterationen*)

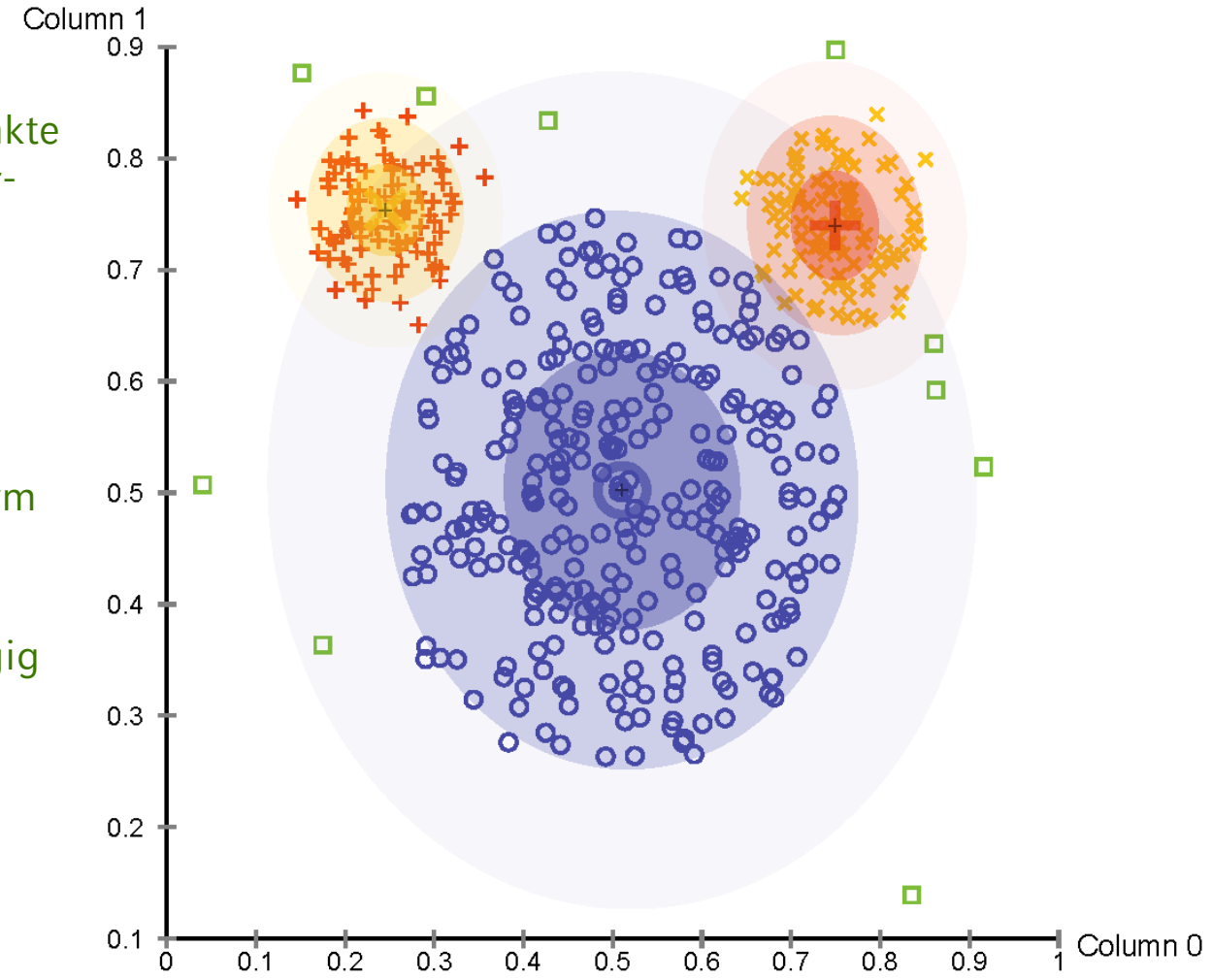
  Anzahl der benötigten Iterationen im allgemeinen sehr hoch

- Ergebnis und Laufzeit hängen (wie beim *k-means* und *k-medoid*) stark ab
  - von der initialen Zuordnung
  - von der „richtigen“ Wahl des Parameters *k*
- Modifikation für Partitionierung der Daten in *k disjunkte* Cluster:
  jedes Objekt nur demjenigen Cluster zuordnen, zu dem es am wahrscheinlichsten gehört.

Partitionierende Verfahren

Model:

- (anteilige) Zuordnung der Punkte zum nächst-gelegenen Cluster-Repräsentanten
- „nächst-gelegen“: bestimmt durch Mahalanobis-Distanz:
  - Distanz quadratischer Form
  - Distanz-Matrix vom jeweiligen Cluster abhängig (Kovarianz-Matrix der zugeordneten Punkte)



Wahl des initialen Clusterings

Idee

- Clustering einer kleinen Stichprobe liefert im allgemeinen gute initiale Cluster einzelne Stichproben sind evtl. deutlich anders verteilt als die Grundgesamtheit

Methode [Fayyad, Reina & Bradley 1998]

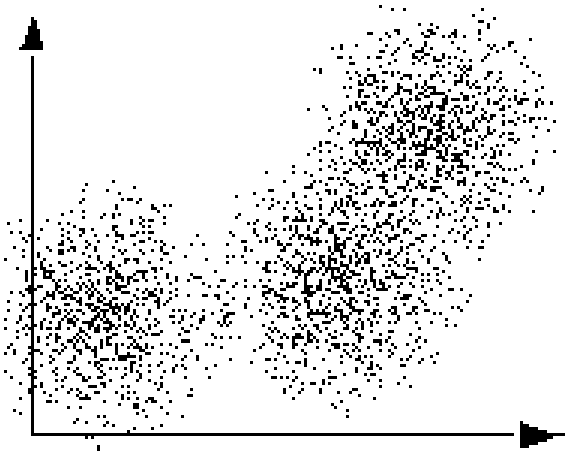
- ziehe unabhängig voneinander m verschiedene Stichproben
- clustere jede der Stichproben

  $\Longrightarrow$ m verschiedene Schätzungen für k Clusterzentren

  $A = (A_1, A_2, \ldots, A_k)$, $B = (B_1, \ldots, B_k)$, $C = (C_1, \ldots, C_k)$, $\ldots$
- Clustere nun die Menge $DB = A \cup B \cup C \cup \ldots$ mit m verschiedenen Schätzungen der Zentren A, B, C, $\ldots$ als Startkonfiguration
- Wähle von den m Clusterings dasjenige mit dem besten Wert bezüglich des zugehörigen Maßes für die Güte eines Clusterings

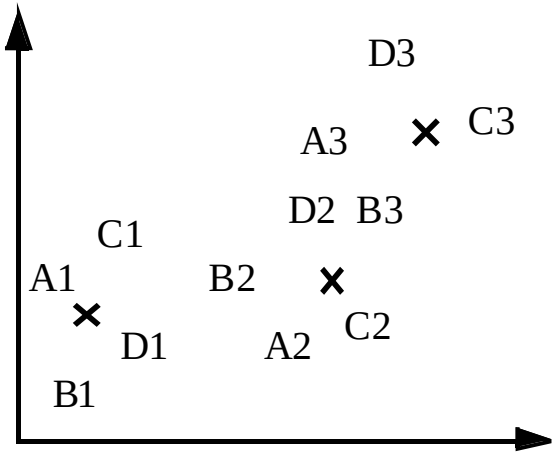
Partitionierende Verfahren

Beispiel



Grundgesamtheit

$k = 3$ Gauß-Cluster



Clusterzentren

von $m = 4$ Stichproben

× *wahre Clusterzentren*

Wahl des Parameters k

Methode

- Bestimme für $k = 2, \ldots, n\text{-}1$ (oder kleinere Grenze, z.B. $n/2$, $\sqrt{n}$) jeweils ein Clustering
- Wähle aus der Menge der Ergebnisse das „beste“ Clustering aus

Maß für die Güte eines Clusterings

- muss unabhängig von der Anzahl k sein
- bei k-means und k-medoid: TD^2 und TD sinken monoton mit steigendem k
- bei EM: E steigt monoton mit steigendem k (=> „overfitting“)
- wir brauchen ein von k unabhängiges Gütemaß für die k-means- und k-medoid-Verfahren

⇒ *Silhouetten-Koeffizient*

Silhouetten-Koeffizient [Kaufman & Rousseeuw 1990]

- sei *a*(*o*) der Abstand eines Objekts *o* zum Repräsentanten seines Clusters und *b*(*o*) der Abstand zum Repräsentanten des „zweitnächsten" Clusters
- Silhouette *s*(*o*) von *o:*

$$s(o) = \frac{b(o) - a(o)}{\max\{a(o), b(o)\}}$$

$-1 \leq s(o) \leq +1$

$s(o) \approx -1 / 0 / +1$: *schlecht / indifferent / gute Zuordnung*

- Silhouettenkoeffizient s_C eines Clustering durchschnittliche Silhouette aller Objekte
- Interpretation des Silhouettenkoeffizients

$s_C > 0{,}7$: starke Struktur,

$s_C > 0{,}5$: brauchbare Struktur, . . .

Partitionierende Verfahren

k-modes Verfahren [Huang 1997]

- *k*-medoid-Algorithmus wesentlich langsamer als *k*-means-Algorithmus
- *k*-means-Verfahren nicht direkt für kategorische Attribute anwendbar

=> gesucht ist ein Analogon zum Centroid eines Clusters

- Numerische Attribute
  Centroid $\bar{x}$ einer Menge *C* von Objekten minimiert $TD(C,\bar{x}) = \sum_{p \in C} dist(p,\bar{x})$
- Kategorische Attribute
  Mode *m* einer Menge *C* von Objekten minimiert $TD(C,m) = \sum_{p \in C} dist(p,m)$
  (*m* ist nicht unbedingt ein Element der Menge *C*)
- $m = (m_1, ..., m_d)$, *dist* eine Distanzfunktion für kategorische Attribute, z.B.

$$dist(x,y) = \sum_{i=1}^{d} \delta(x_i, y_i) \text{ mit } \delta(x_i, y_i) = \begin{cases} 0, & falls\ x_i = y_i \\ 1, & sonst \end{cases}$$

Bestimmung des Modes

- Die Funktion $TD(C,m) = \sum_{p \in C} dist(p,m)$ wird genau dann

  minimiert, wenn für $m = (m_1, ..., m_d)$ und für alle Attribute A_i, $i = 1,..., d$, gilt:

  Es gibt in A_i keinen häufigeren Attributwert als m_i

- Der Mode einer Menge von Objekten ist nicht eindeutig bestimmt.

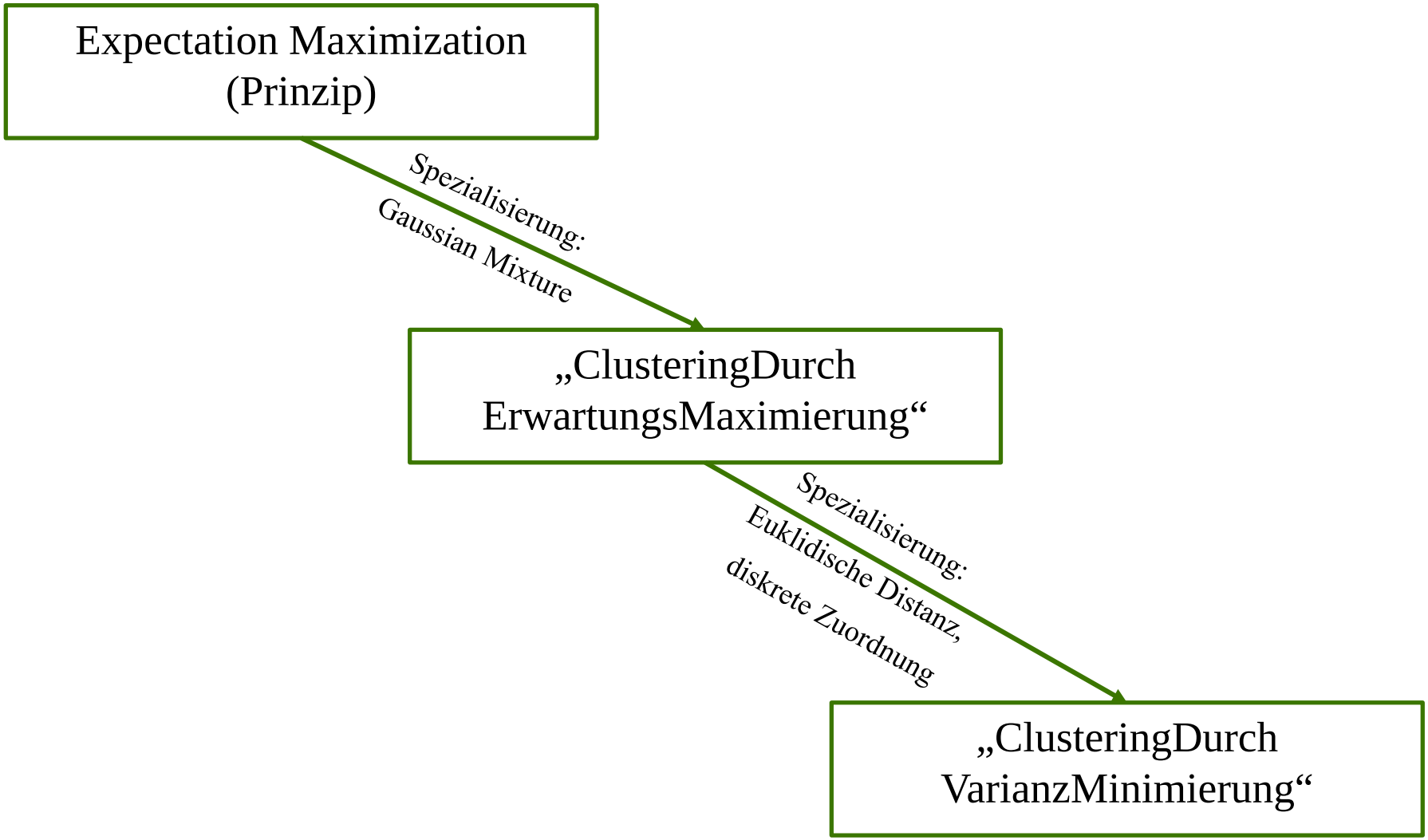
- Beispiel

  Objektmenge {(a, b), (a,c), (c, b), (b,c)}
  (a, b) ist ein Mode
  (a, c) ist ein Mode

Algorithmus k-modes

- Initialisierung
  - keine zufällige Partitionierung
  - sondern *k* Objekte aus der Datenmenge als initiale *Modes*
- Cluster-Repräsentanten
  - Mode anstelle des Centroids
- Distanzfunktion
  - anstelle der quadrierten euklidischen Distanz
  - Distanzfunktion für Datensätze mit kategorischen Attributen

Partitionierende Verfahren



Partitionierende Verfahren

Literatur

A. P. Dempster, N. M. Laird, and D. B. Rubin: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B, 39(1):1–31, 1977.

Usama M. Fayyad, Cory Reina, Paul S. Bradley: Initialization of Iterative Refinement Clustering Algorithms. KDD 1998: 194-198

E. W. Forgy: Cluster analysis of multivariate data: efficiency vs interpretability of classifications. Biometrics 21, 768–769, 1965

Z. Huang: A fast clustering algorithm to cluster very large categorical data sets in data mining. Workshop on Research Issues on Data Mining and Knowledge Discovery. 1997.

L. Kaufman, P. J. Rousseeuw: Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley 1990

S. P. Lloyd: Least square quantization in PCM. In: Bell Telephone Laboratories Paper. 1957

J. MacQueen. Some methods for classification and analysis of multivariate observations. In 5th Berkeley Symposium on Mathematics, Statistics, and Probabilistics, volume 1, pages 281–297, 1967.

R. Ng and J. Han. Efficient and effective clustering methods for spatial data mining. In Proceedings of the 20th International Conference on Very Large Data Bases (VLDB), Santiago de Chile, Chile, 1994.

Partitionierende Verfahren: Was haben Sie gelernt?

- Was ist „Clustering“?
- Warum Heuristiken zur Identifikation von Clustern?
- grundlegende Heuristiken zur „Partitionierung“ in *k* Cluster:
  - Auswahl zentraler Punkte (Repräsentanten)
  - Optimierungsalgorithmen zur Zuordnung der Daten zu den Repräsentanten
    - Varianz-Minimierung
    - k-means-Verfahren
    - k-medoid-Verfahren
    - Erwartungs-Maximierung (Gaussian Mixture Modeling)
    - k-modes
  - Gemeinsamkeiten/Unterschiede dieser Verfahren
  - Vorteile/Nachteile der Verfahren