LMU LUDWIG-MAXIMILIANS-UNIVERSITY MUNICH
DEPARTMENT INSTITUTE FOR INFORMATICS
DATABASE SYSTEMS GROUP

Skript zur Vorlesung:

Einführung in die Informatik: Systeme und Anwendungen

Sommersemester 2013

Kapitel 4: Data Mining

Vorlesung: PD Dr. Peer Kröger
Übungen: Johannes Niedermayer

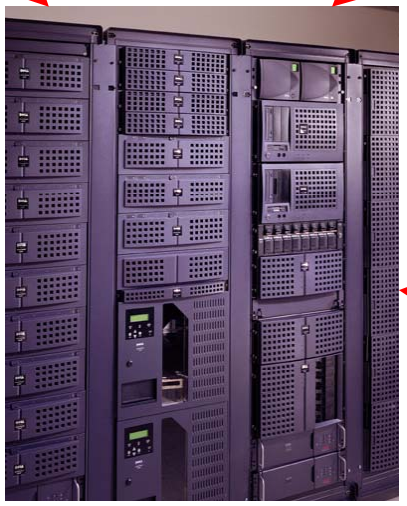
http://www.dbs.ifi.lmu.de/cms/ Einführung_in_die_Informatik_Systeme_und_Anwendungen

Überblick

Motivation

Kreditkarten

Scanner-Kassen



Datenbanken

Telefongesellschaft

Astronomie

- Riesige Datenmengen werden in Datenbanken gesammelt
- Analysen können nicht mehr manuell durchgeführt werden

Big Data

- Das Schlagwort "Big Data" geht zurück auf einen Report von McKinsey vom May 2011
  (http://www.mckinsey.com/insights/business_technology/big_data_the_next_frontier_for_innovation)
- "The amount of data in our world has been exploding, and analyzing large data sets—so-called big data—will become a key basis of competition, underpinning new waves of productivity growth, innovation, and consumer surplus [...]"
- "Data have swept into every industry and business function and are now an important factor of production, alongside labor and capital"
  - Wertschöpfungspotential im US Healthcare Sektor: > $300 Millionen
  - Wertschöpfungspotential im öffentlichen Sektor der EU: > €100 Millionen
- "There will be a shortage of talent necessary for organizations to take advantage of big data. By 2018, the United States alone could face a shortage of 140,000 to 190,000 people with deep analytical skills as well as 1.5 million managers and analysts with the know-how to use the analysis of big data to make effective decisions."

Big Data

- Data Mining ist offensichtlich eine der entscheidenden Technologien um Big Data in den Griff zu bekommen

- Im Übrigen gibt es seit ca. 40 Jahren (!!!) die Konferenz on Very Large Databases – schönen Gruss Herr McKinsey ;-)

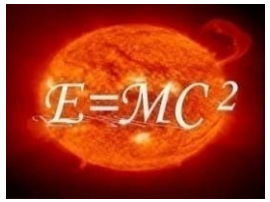
  (http://www.vldb.org/)

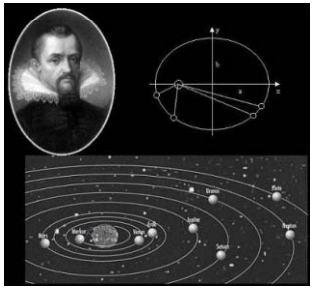
- Achtung: Big Data ist nicht notwendigerweise nur "einfach groß"

  => Die drei V's (the three V's characterizing big data)
  - Volume
  - Velocity
  - Variety

A Paradigm Shift in Science?

- Some 1,000 years ago, science was empirical (describing natural phenomena)
- Last few hundred years, science was theoretical (Models, generalizations)





- Last few decades, science became computational (data intensive)
  - Computational methods for simulation
  - Automatic data generation, high-throughput methods, …

$$\frac{\partial T}{\partial t} + \operatorname{div}(\vec{v}T) = \frac{k_1}{\rho_1 c_1}\Delta T$$
$$\rho_1\left(\frac{\partial \vec{v}}{\partial t} + \operatorname{div}(\vec{v}\otimes\vec{v})\right) = -\operatorname{grad} p + \mu\Delta\vec{v} + \rho_1(1-\beta(T-T_m))\vec{g}$$
$$\operatorname{div}\vec{v} = 0$$
$$\frac{\partial T}{\partial t} = \frac{k_2}{\rho_2 c_2}\Delta T$$
$$\left[k\frac{\partial T}{\partial n}\right]_s + \rho_1 c_1(T-T_m)(\vec{V}_\Gamma - \vec{v})\cdot\vec{n} - \rho_s c_s(T-T_m)\vec{V}_\Gamma\cdot\vec{n} = -\rho_s L V_\Gamma$$
$$\frac{T-T_m}{T_m} = -\frac{a\rho_s T_m V_\Gamma}{L} = \frac{\gamma_1 \nu_1}{L\rho_1}\left[\frac{1}{\rho}\right]_s^{-1}\frac{p_1 - p^*}{L}$$
$$\vec{v}(\vec{x}) = \left(1 - \frac{\rho_s}{\rho_1}\right)\vec{V}_\Gamma, \qquad \vec{x}\in\Gamma_1$$
$$T|_\Gamma = T|_s$$
$$k\frac{\partial T}{\partial n}(\vec{x}) = -h(T(\vec{x}) - T^*), \qquad \vec{x}\in\Gamma_2$$
$$-p\vec{n} + \tau\vec{n} = -\gamma_2\kappa_2\vec{n} - p^*\vec{n}$$
$$\vec{v}(\vec{x}) = 0, \qquad \vec{x}\in\Gamma_3$$
$$T(\vec{x}) = T_{\Gamma_3}(\vec{x}), \qquad \vec{x}\in\Gamma_3$$
$$k\frac{\partial T}{\partial n}(\vec{x}) = q_{\Gamma_4}(\vec{x}), \qquad \vec{x}\in\Gamma_4$$

- Data Sciene

A Paradigm Shift in Science?

- Wissenschatler produzieren eine riesige Menge an Daten über die Phänomene, die sie untersuchen
- Bottom-up und top-down Modelle:
  - Bottom-up:
    - Von Experten über Jahre hinweg erforscht
    - Meist aus wenigen Beobachtungen abgeleitet
  - Top-down:
    - Mit Hilfe von Data Mining Methoden erzeugt
    - Meist aus großen Datenmengen abgeleitet
- Fragestellungen für die Informatik:
  - Wie können top-down-Modelle erzeugt werden
  - Wie können sich mehrere Modell sinnvoll verknüpfen

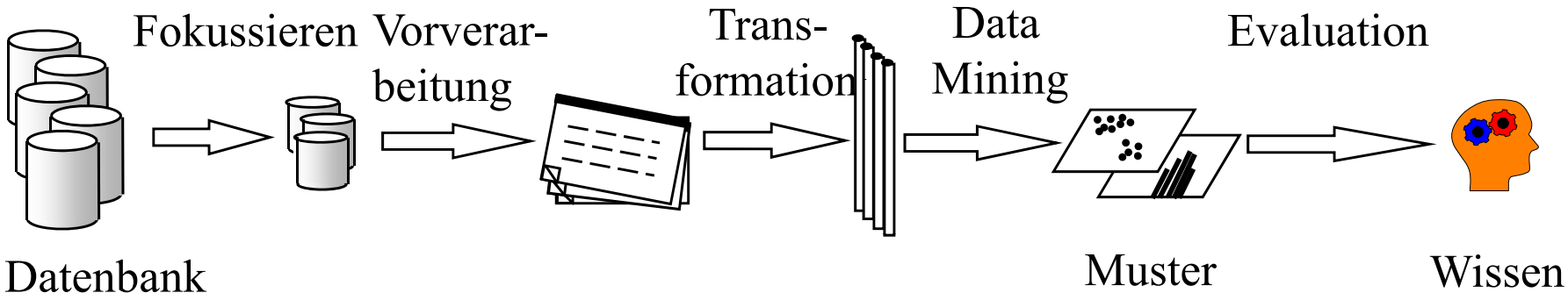
Knowledge Discovery in Databases

- *Knowledge Discovery in Databases* (*KDD*) ist der Prozess der (semi-) automatischen Extraktion von Wissen aus Datenbanken, das
  - *gültig*
  - *bisher unbekannt*
  - und *potentiell nützlich* ist.

- Bemerkungen:
  - (*semi-*) *automatisch*: im Unterschied zu manueller Analyse.
    Häufig ist trotzdem Interaktion mit dem Benutzer nötig.
  - *gültig*: im statistischen Sinn.
  - *bisher unbekannt*: bisher nicht explizit, kein „Allgemeinwissen“.
  - *potentiell nützlich*: für eine gegebene Anwendung.

Der KDD-Prozess (Modell)

Prozessmodell nach Fayyad, Piatetsky-Shapiro & Smyth



Fokussieren:
- Beschaffung der Daten
- Verwaltung (File/DB)
- Selektion relevanter Daten

Vorverarbeitung:
- Integration von Daten aus unterschiedlichen Quellen
- Vervollständigung
- Konsistenzprüfung

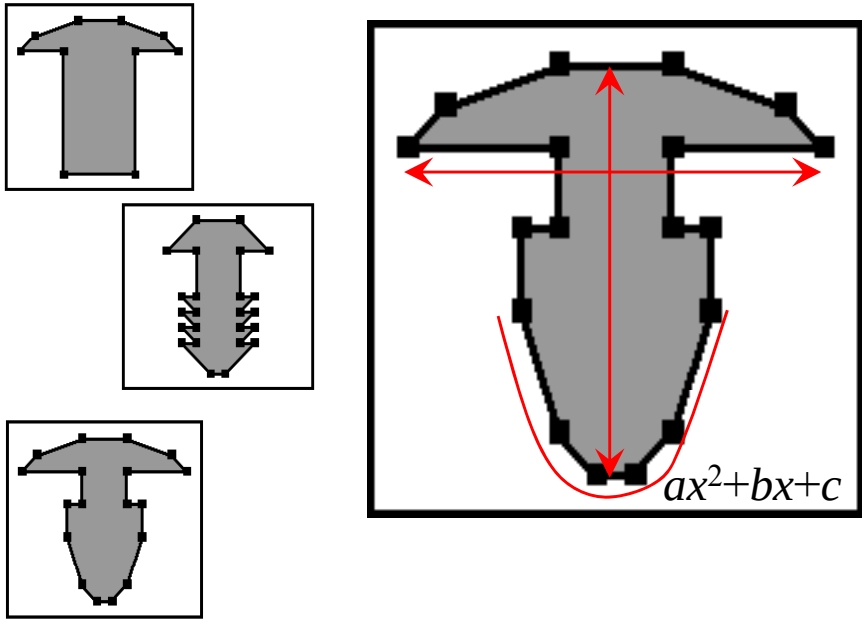
Transformation
- Diskretisierung numerischer Merkmale
- Ableitung neuer Merkmale
- Selektion relevanter Merkm.

Data Mining
- Generierung der Muster bzw. Modelle

Evaluation
- Bewertung der Interessantheit durch den Benutzer
- Validierung: Statistische Prüfung der Modelle

Objekt-Merkmale (Feature)

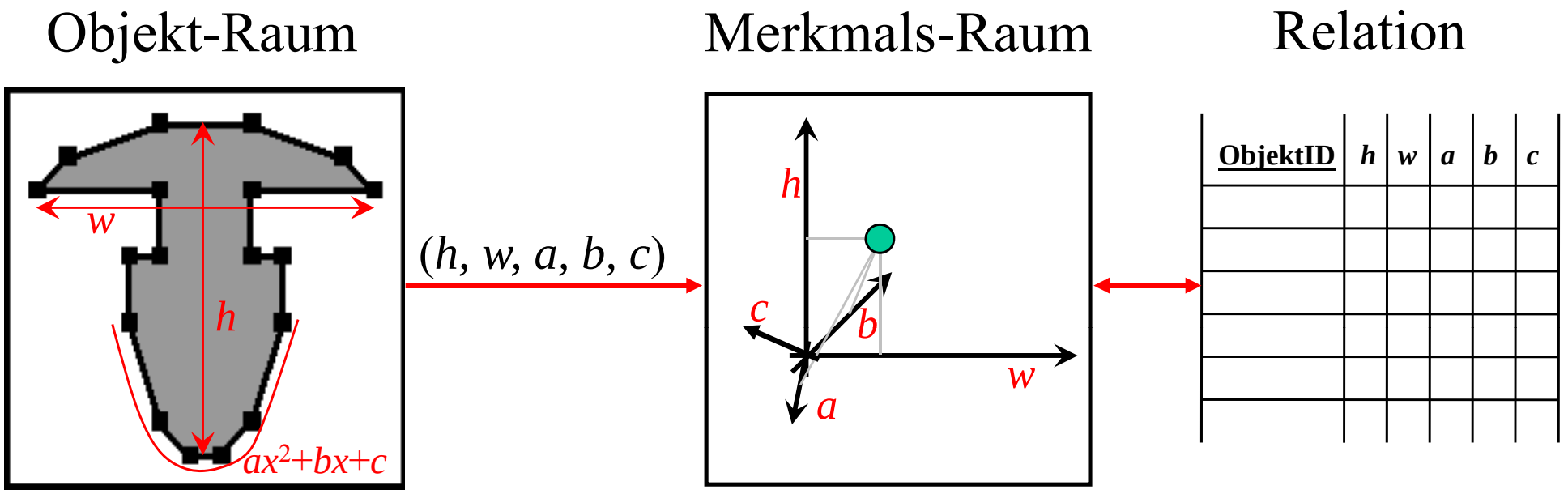
- Oft sind die betrachteten Objekte komplex
- Eine Aufgabe des KDD-Experten ist dann, geeignete Merkmale (*Features*) zu definieren bzw. auszuwählen, die für die Unterscheidung (Klassifikation, Ähnlichkeit) der Objekte relevant sind.
- Diese Merkmale können dann z.B. in einer Tabelle in einem (rel.) DBMS verwaltet werden
- Beispiel: CAD-Zeichnungen:



Mögliche Merkmale:

- Höhe h
- Breite w
- Kurvatur-Parameter (a,b,c)

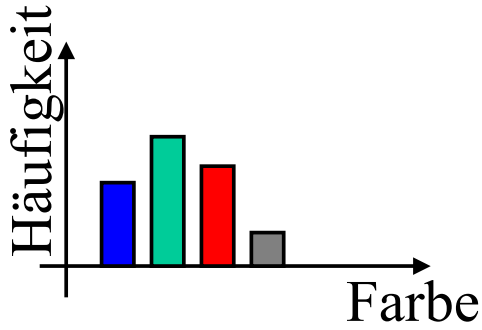
Feature-Vektoren



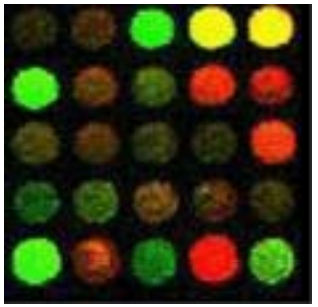
- Im Kontext von statistischen Betrachtungen werden die Merkmale häufig auch als *Variablen* bezeichnet
- Die ausgewählten Merkmale werden zu Merkmals-Vektoren (*Feature Vector*) zusammengefasst
- Der Merkmalsraum ist häufig hochdimensional (im Beispiel 5-dim.)
- Merkmale sind oft nummerisch (=> Euklidischer Vektorraum), können aber auch kategorisch, ordinal, etc. sein (siehe übernächste Folie)

Feature-Vektoren (weitere Beispiele)

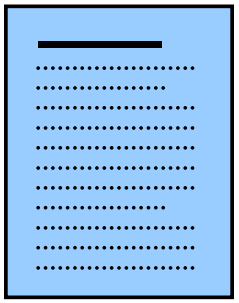
Bilddatenbanken:
Farbhistogramme



Gen-Datenbanken:
Expressionslevel



Text-Datenbanken:
Begriffs-Häufigkeiten



Data	25
Mining	15
Feature	12
Object	7
...	

Der Feature-Ansatz ermöglicht einheitliche Behandlung von Objekten verschiedenster Anwendungsklassen

Feature: verschiedene Kategorien

Nominal (kategorisch)

Charakteristik:

Nur feststellbar, ob der Wert gleich oder verschieden ist. Keine Richtung (besser, schlechter) und kein Abstand.
Merkmale mit nur zwei Werten nennt man *dichotom*.

Beispiele:

Geschlecht (dichotom)
Augenfarbe
Gesund/krank (dichotom)

Ordinal

Charakteristik:

Es existiert eine Ordnungsrelation (besser/schlechter) zwischen den Kategorien, aber kein einheitlicher Abstand.

Beispiele:

Schulnote (metrisch?)
Güteklasse
Altersklasse

Metrisch

Charakteristik:

Sowohl Differenzen als auch Verhältnisse zwischen den Werten sind aussagekräftig. Die Werte können diskret oder stetig sein.

Beispiele:

Gewicht (stetig)
Verkaufszahl (diskret)
Alter (stetig oder diskret)

Ähnlichkeit von Objekten

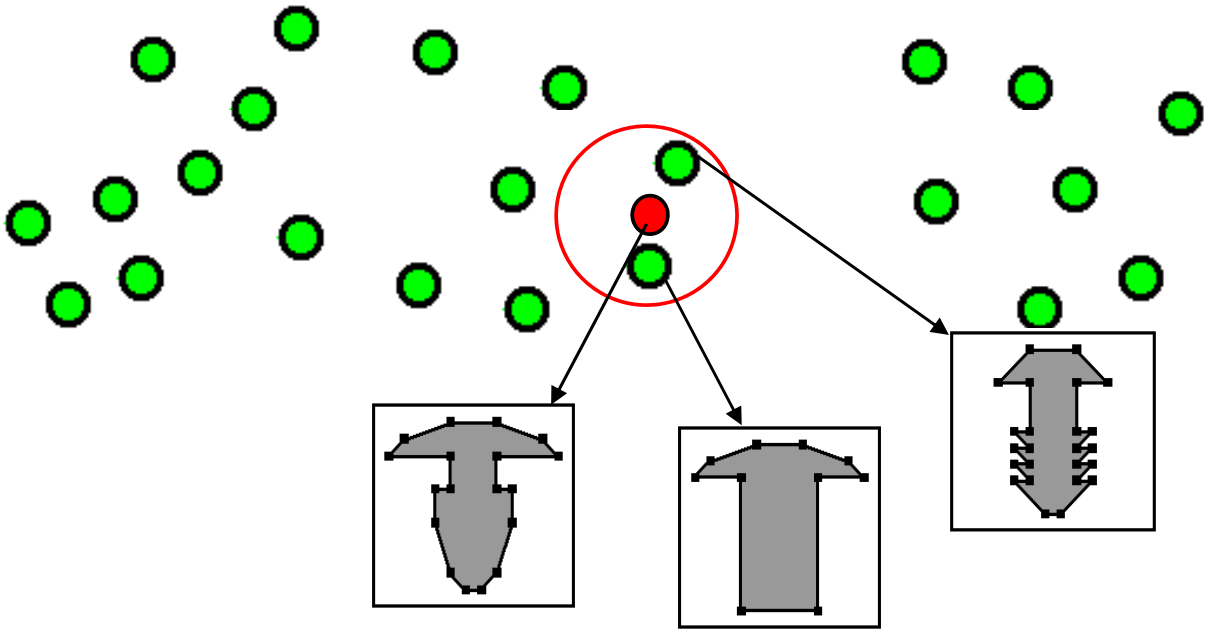
- Definiere ein Distanzmaß δ auf den Feature Vektoren
- Zu einem Anfrage-Objekt $q \in DB$ kann man dann …
  - … ähnliche Objekte suchen – Range-Query (Radius ε)

$$RQ(q,\varepsilon) = \{ o \in DB \mid \delta(q,o) \leq \varepsilon \}$$

  - … die k ähnlichsten Objekte suchen– Nearest Neighbor Query

$NN(q,k) \subseteq DB$ mit k Objekten, sodass $\forall o \in NN(q,k), p \in DB \backslash NN(q,k) : \delta(q,o) \leq \delta(q,p)$

alternative Schreibweise für Mengendifferenz:
$A \backslash B = A - B$

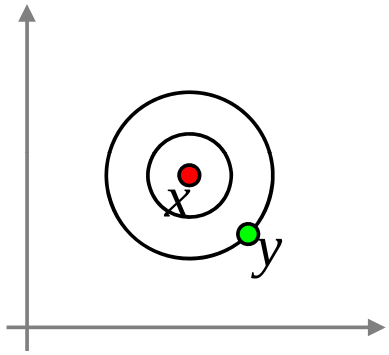


Distanz im Feature-Raum ≈ (Un-)Ähnlichkeit der ursprünglichen Objekte

Distanzmaße für nummerische Features

Euklidische Norm (L_2):

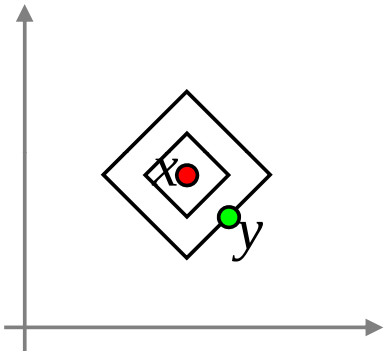
$\delta_1(x,y) = ((x_1-y_1)^2+(x_2-y_2)^2+...)^{1/2}$



Abstand in Euklidischen Raum (natürliche Distanz)

Manhattan-Norm (L_1):

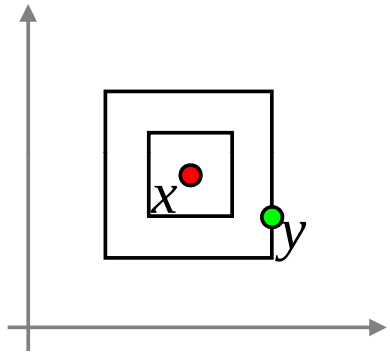
$\delta_2(x,y) = |x_1-y_1|+|x_2-y_2|+...$



Die Unähnlichkeiten der einzelnen Merkmale werden direkt addiert

Maximums-Norm (L_∞):

$\delta_\infty(x,y) = \max\{|x_1-y_1|, |x_2-y_2|,...\}$



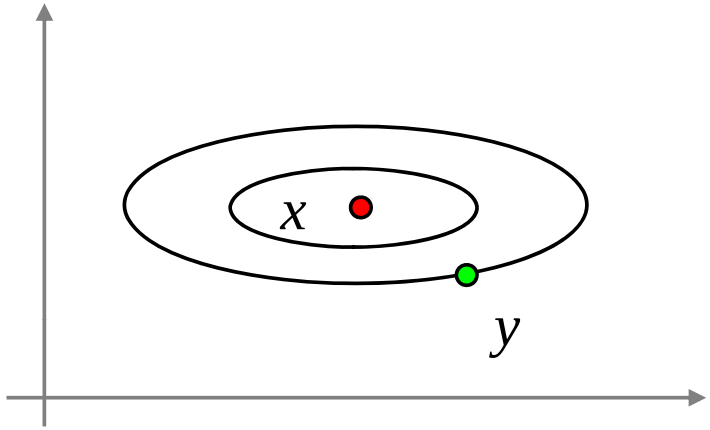
Die Unähnlichkeit des am wenigsten ähnlichen Merkmals zählt

Verallgemeinerung L_p-Abstandsmaß:

$$\delta_p(x,y) = (|x_1-y_1|^p + |x_2-y_2|^p + ...)^{1/p}$$

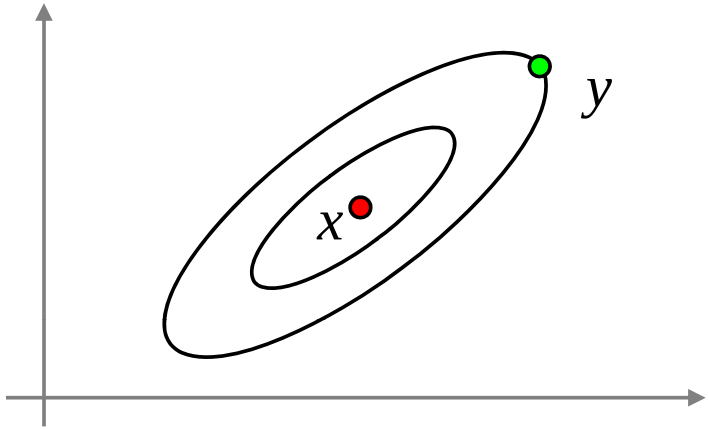
Gewichtete Distanzmaße

- Viele Varianten gewichten verschiedene Merkmale unterschiedlich stark.



Gewichtete Euklidische Distanz

$$\delta_{p,w}(x,y) = \sqrt[p]{\sum_{i=1}^{d} w_i \cdot |x_i - y_i|^p}$$



Mahalanobis Distanz

$$\delta_{\Sigma}(x,y) = \sqrt{(x-y)^T \cdot \Sigma^{-1} \cdot (x-y)}$$

Σ = Kovarianz-Matrix

- Statt Distanzmaßen verwendet man auch manchmal Ähnlichkeitsmaße
  - Distanzmaß: je höher der Wert, desto unähnlicher
  - Ähnlichkeitsmaß: je höher der Wert, desto ähnlicher

Kategorien von Data Mining

- Klassen von Data-Mining-Verfahren
  - Clustering
  - Outlier Detection
  - Klassifikation
  - Regression
  - Frequent Pattern Mining

  } normalerweise unsupervised

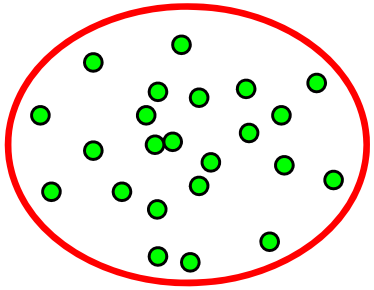
  } normalerweise supervised

- Supervised: Trainingsphase erforderlich, der Lernerfolg kann überwacht werden.
- Unsupervised: Die Methode lernt nicht, sondern findet Muster, die einem bestimmten Modell entsprechen.
- Viele Methoden arbeiten auf Merkmalsvektoren, meist explizit auf Euklidischen Räumen (nur nummerische Merkmale)

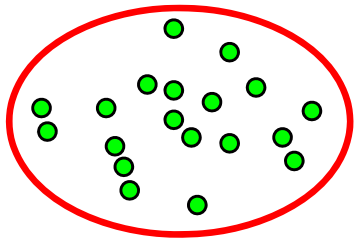
  => wir konzentrieren uns hier auf Euklidische Merkmalsräume
- Darüber hinaus gibt es zahlreiche Verfahren, die nicht auf Merkmalsvektoren, sondern direkt auf Texten, Mengen, Graphen usw. arbeiten.

Überblick

Clustering



Cluster 1: Klammern



Cluster 2: Nägel

- Ein Grundmodell des Clustering ist:
  Zerlegung (Partitionierung) einer Menge von Objekten (bzw. Feature-Vektoren) in Teilmengen (Cluster), so dass
  - Objekte im *gleichen* Cluster möglichst ähnlich sind
  - Objekte aus *verschiedenen* Clustern möglichst unähnlich sind

  Idee: Die verschiedenen Cluster repräsentieren meist unterschiedliche Klassen von Objekten; bei evtl. unbek. Anzahl und Bedeutung der Klassen

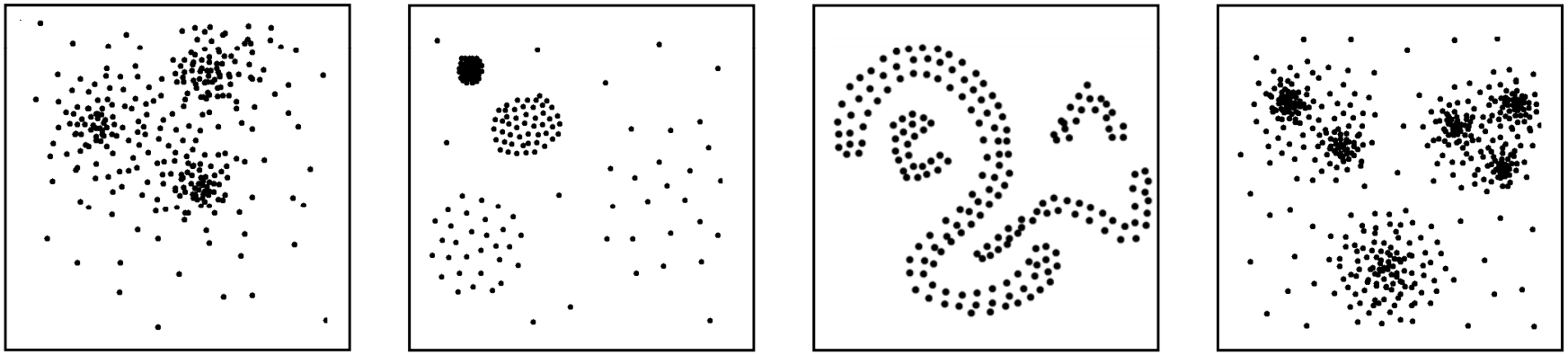
⇒ Im Euklidischen Raum:
Finde Regionen, in denen die Punkte näher zusammen liegen (dichter sind?)

Ziel des Clustering

Herausforderungen:

- Cluster unterschiedlicher Größe, Form und Dichte
- hierarchische Cluster
- Rauschen (Noise)

  => unterschiedliche Clustering-Algorithmen



Hier: Partitionierendes Verfahren

- Parameter: Anzahl *k* der Cluster
- sucht eine Partitionierung in *k* Cluster
- Cluster sind daher meist sphärisch, Rauschen wird nicht berücksichtigt

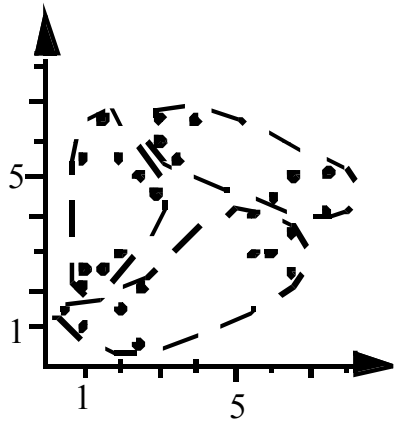
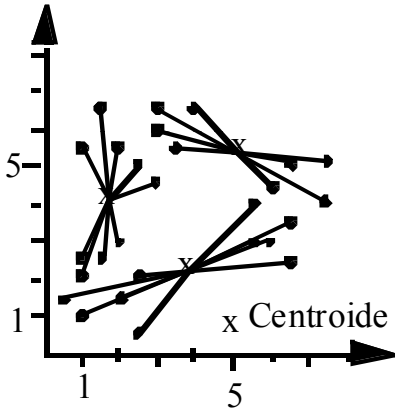
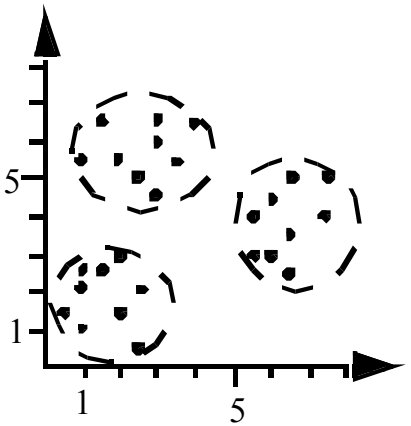
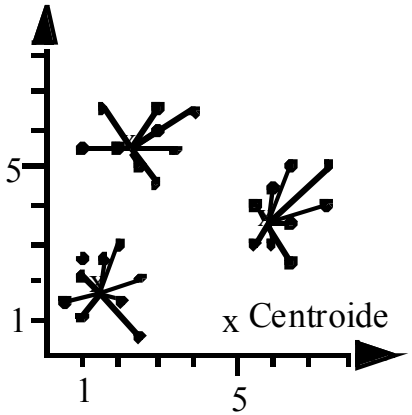
Partitionierende Verfahren

Grundlagen

- Ziel
  - Partitionierung in *k* Cluster so dass eine Kostenfunktion minimiert wird (Gütekriterium)
- Lokal optimierendes Verfahren
  - wähle *k* initiale Cluster-Repräsentanten
  - optimiere diese Repräsentanten iterativ
  - ordne jedes Objekt seinem ähnlichsten Repräsentanten zu
- Typen von Cluster-Repräsentanten
  - Mittelwert des Clusters (*Centroid*)
  - Element des Clusters (*Medoid*)
  - Wahrscheinlichkeitsverteilung des Clusters (*Erwartungsmaximierung*)

Partitionierende Verfahren

Beispiel

	Cluster	Cluster-Repräsentanten
schlechtes Clustering		
optimales Clustering		

Partitionierende Verfahren: k-means

Grundbegriffe

- Objekte sind Punkte $p=(p_1, ..., p_d)$ in einem d-dimensionalen euklidischen Vektorraum

Datenstrukturen:

Für Punkte:

```
RECORD Punkt =
(
    ID : Nat
    f1 : Real
    ...
    fd : Real
)
```

Für eine Menge von Punkten:

PunktList

- δ = euklidische Distanz auf dem Record-Typ Punkte, d.h.

δ : **Punkt** $\times$ **Punkt** $\rightarrow$ **Real**

Partitionierende Verfahren: k-means

- *Centroid* μ_C: Mittelwert aller Punkte im Cluster *C (ist selbst wieder vom Typ* **Punkt***)*

- *Maß für die Kosten* (Kompaktheit) *eines Clusters C*

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

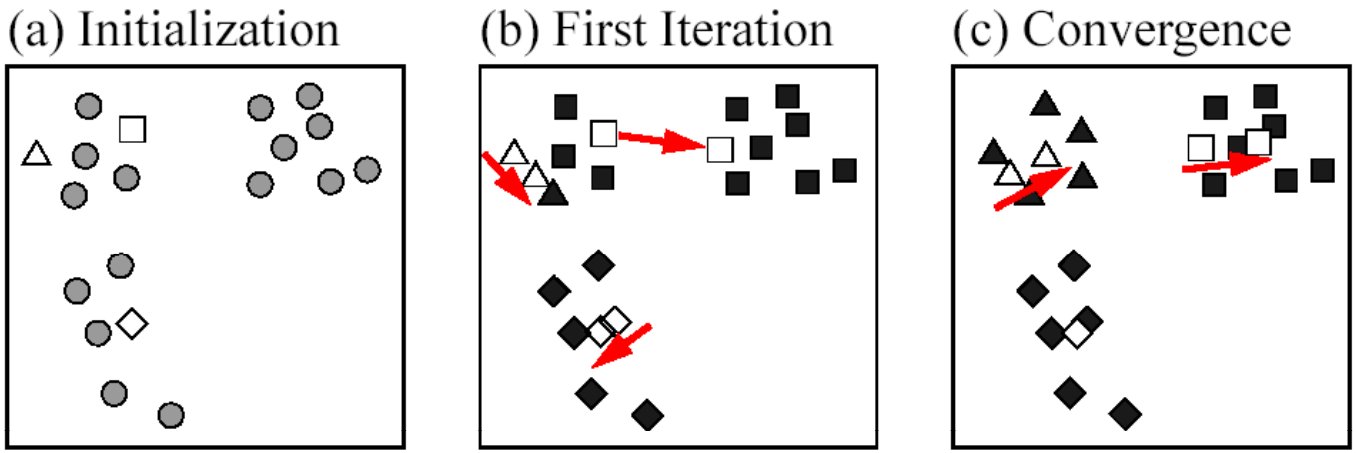
- *Maß für die Kosten* (Kompaktheit) *eines Clustering (aus* k *Clustern)*

$$TD^2 = \sum_{i=1}^{k} TD^2(C_i)$$

Partitionierende Verfahren: k-means

Idee des Algorithmus

- Algorithmus startet z.B. mit 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



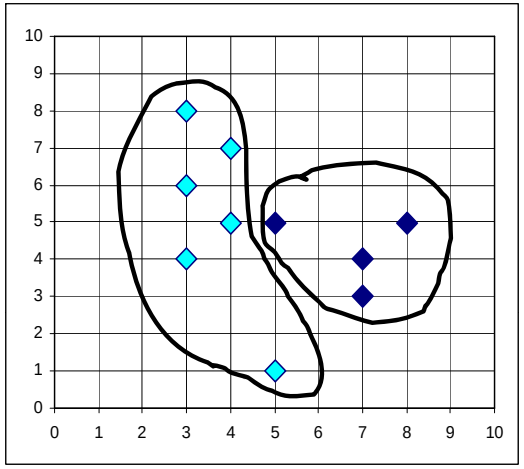
Partitionierende Verfahren: k-means

Algorithmus

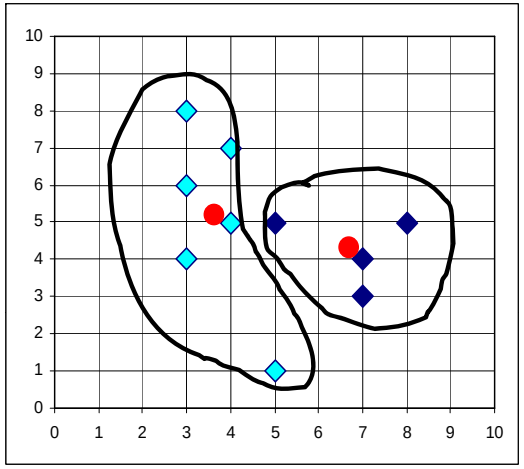
```
algorithmus  kMeansClustering
input        k: Nat, D: PunktList     // Anzahl der Cluster, Punktemenge, die geclustert wird
output       C: PunktList             // Liste der finalen Centroide
variables    C1, ..., Ck: Punkt,      // die Centroide der k Cluster
             C': PunktList            // Liste der neu bestimmten Centroide
begin
    // Erzeuge eine „initiale“ Zerlegung von D in k Klassen;
    Ziehe zufällig k Elemente aus der Liste D und speichere sie in C1, ..., Ck;
    Berechne die Menge C'={C1, ..., Ck} der Zentroide für die k Klassen;
    C = {};
    while  C ≠ C'                     // es ändert sich noch etwas
    do  {
        C = C'                        // die aktuellen Centroide
        Ordne alle Punkte aus D neu zum nächstliegenden Centroid aus C zu;
                                      // für jeden Punkt p in D und alle 1≤i ≤k muss δ(p, Ci) berechnet werden
        Bestimme die Menge C'={C'1, ..., C'k} der Centroide neu;
        }
    return C;
```

Partitionierende Verfahren: k-means

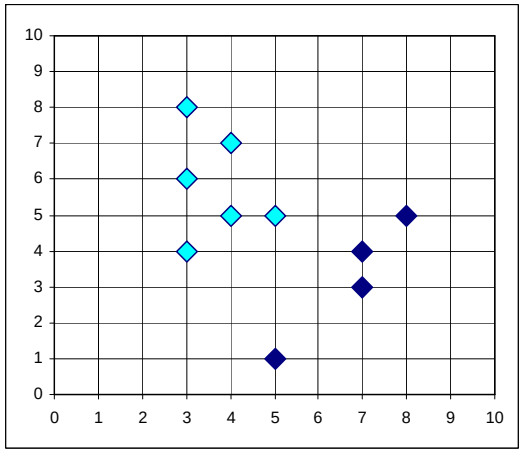
Beispiel



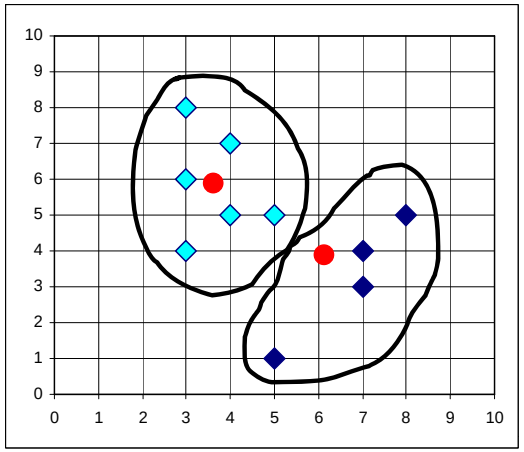
Berechnung der neuen Zentroide



Zuordnung zum nächsten Zentroid



Berechnung der neuen Zentroide



Partitionierende Verfahren: k-means

Diskussion

+ Effizienz:
  - Anzahl der Iterationen ist im allgemeinen klein (~ 5 - 10).
+ einfache Implementierung:
  - *k*-means ist das populärste partitionierende Clustering-Verfahren

– Anfälligkeit gegenüber Rauschen und Ausreißern
(alle Objekte gehen ein in die Berechnung des Centroids)
– Cluster müssen konvexe Form haben
– die Anzahl *k* der Cluster muss bekannt sein
– starke Abhängigkeit von der initialen Zerlegung
(sowohl Ergebnis als auch Laufzeit)

Erweiterung auf kategorische Merkmale

- *k*-means-Verfahren nicht direkt für kategorische Attribute anwendbar
  => gesucht ist ein Analogon zum Centroid eines Clusters
- Idee
  - Centroid einer Menge *C* von Objekten minimiert die Distanzen aller Elemente aus *C* zu sich
  - Diese Intuition kann auf kategorische Daten angewandt werden
  - Der Centroid heißt nun *Mode*
  - Der Mode *m* einer Menge *C* von Objekten minimiert ebenfalls die Distanzen aller Elemente aus *C* zu sich
  - (*m* ist nicht unbedingt ein Element der Menge *C*)

- Eine Distanzfunktion *dist* für kategorische Attribute kann z.B. wie folgt definiert werden

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

Erweiterung auf kategorische Merkmale

- 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), (a,d), (b,c)}

    (a, b) ist ein Mode

    (a, c) ist ein Mode

id	A1	A2
1	a	b
2	a	c
3	c	b
4	a	d
5	b	c

- Anpassungen des Algorithmus
  - Initialisierung
    - keine zufällige Partitionierung
    - sondern k Objekte aus der Datenmenge als initiale Modes
  - Cluster-Repräsentanten
    - Mode anstelle des Centroids
  - Distanzfunktion für kategorische Daten

Clustering: Schlussbemerkungen

- Clustering wird angewendet um Gruppen von ähnlichen Objekten zu entdecken (ohne die Gruppen vorher zu kennen)
- Intuition aus der Statistik: es gibt für jeden Cluster einen (statistischen) Prozess, der die Objekte des Clusters erzeugt hat.
  (jeder Prozess beschreibt ein spezielles (natur-)wissenschaftliches Phänomen)
- Beispiele:
  - Im Marketing werden Kundengruppen identifiziert, die ähnliche Präferenzen haben
  - Analog: ähnliches Surf-Verhalten im Web
  - Analog: Benutzergruppen in sozialen Netzwerken
  - In der Geographie werden z.B. thematische Karten mittels Clusterverfahren erstellt
  - In der Biologie sucht man gruppen von Genen, deren Produkte sich ähnlich verhalten
  - Teilnehmer einer Umfrage werden in Gruppen mit ähnlichem Antwortverhalten eingeteilt
  - …
- Orthogonales Problem: welche Objekte sind nicht in Clustern (weil sie sich „abnormal“ verhalten)