Prof. Dr. Adrian Ulges

Empolis Workshop “Machine Learning”

Clustering

Hochschule RheinMain
Department DCSM *(Design, Computer Science, Media)*

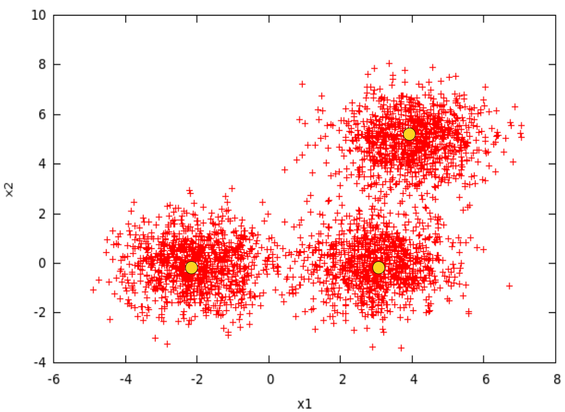
Überblick

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

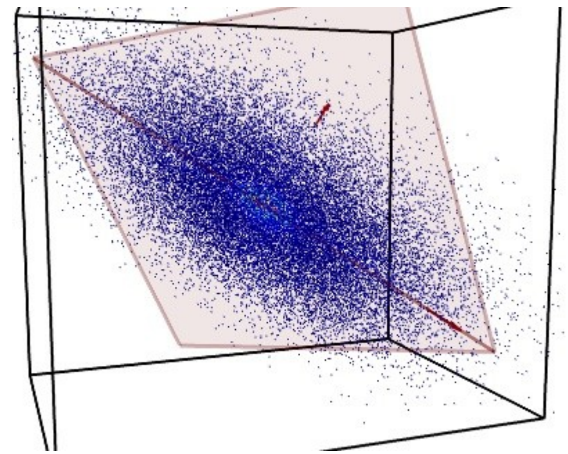
- Clustering
  - Clustering im Vektorraum: K-Means, EM
  - Model Selection (→ Anzahl der Cluster)
  - Agglomeratives Clustering, Topic Modeling
  - Python-Beispiel: News-Clustering

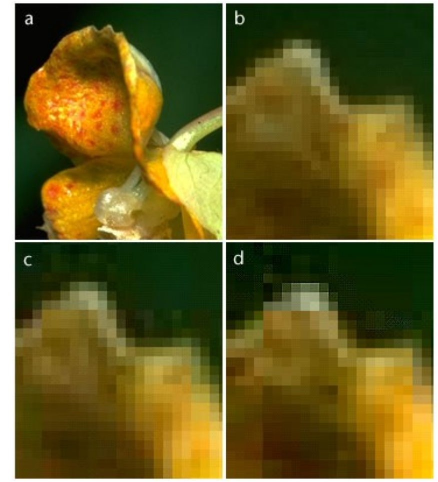
Unüberwachtes Lernen

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain



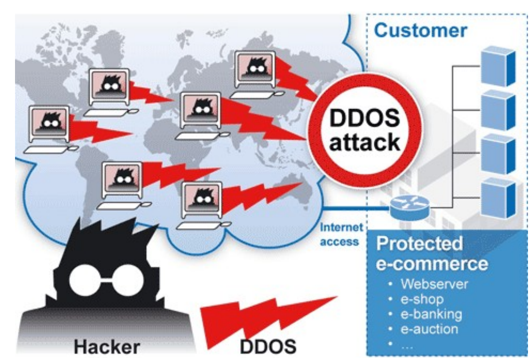
Clustering: Teile die Daten in kohärente Gruppen ein





Dimensionality Reduction: Komprimiere die Daten

Itemset Mining: Finde häufige Substrukturen in den Daten.

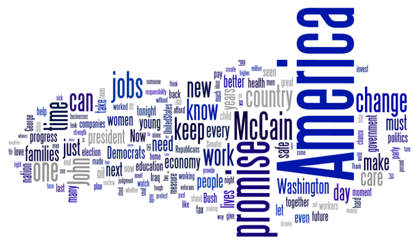


Anomaly Detection: Finde Outlier / Ausreißer in den Daten

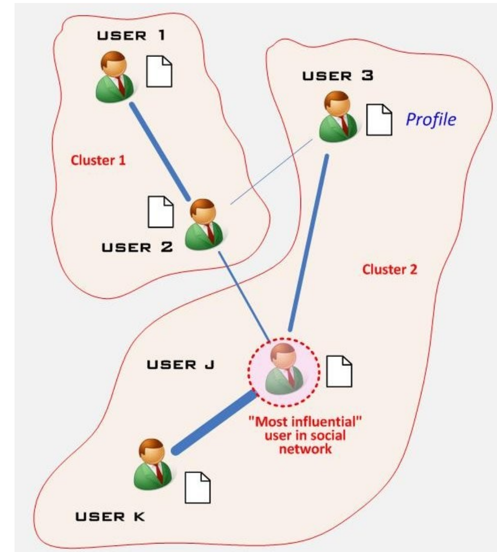
Clustering: Anwendungen

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

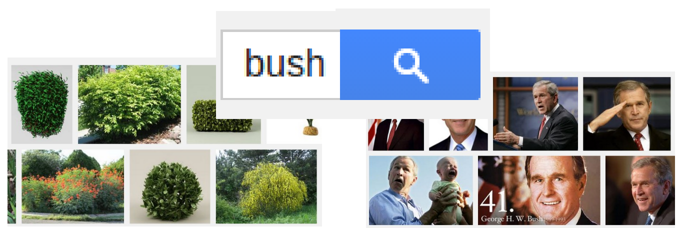
- Clustering-Verfahren finden zahlreiche Anwendungen in den ***verschiedensten Gebieten***
  - Marktforschung
  - Information Retrieval
  - Computer Vision
  - Social Networks
  - Data Mining



Text clustern

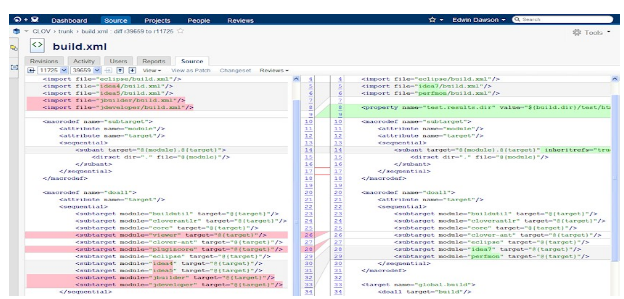


Benutzer clustern

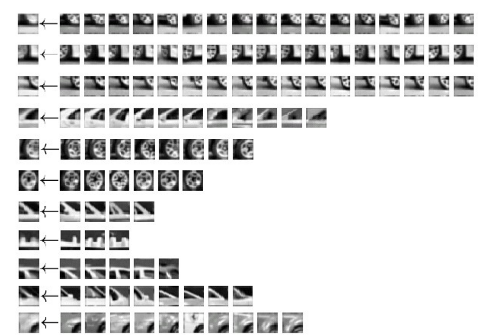


Suchergebnisse clustern

Produkte clustern



Sourcecode clustern



Bildstücke clustern

Überblick

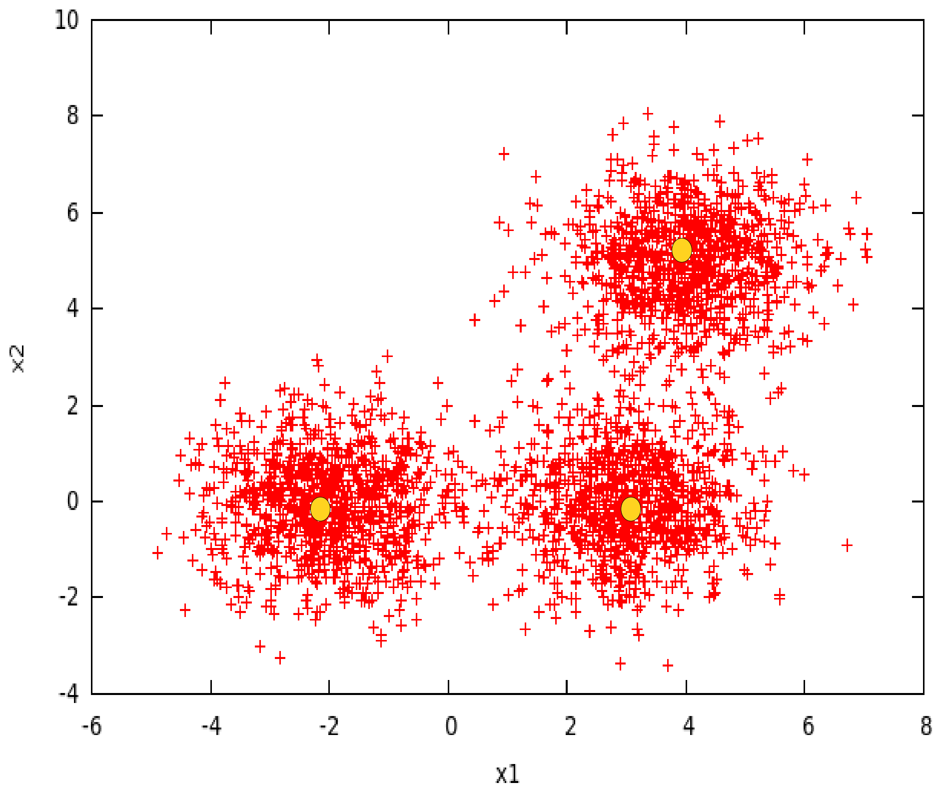
Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

- Grundlagen
- **K-Means**
- Expectation Maximization
- Model Selection
- Agglomeratives Clustering
- Topic Models

K-Means: Grundidee

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

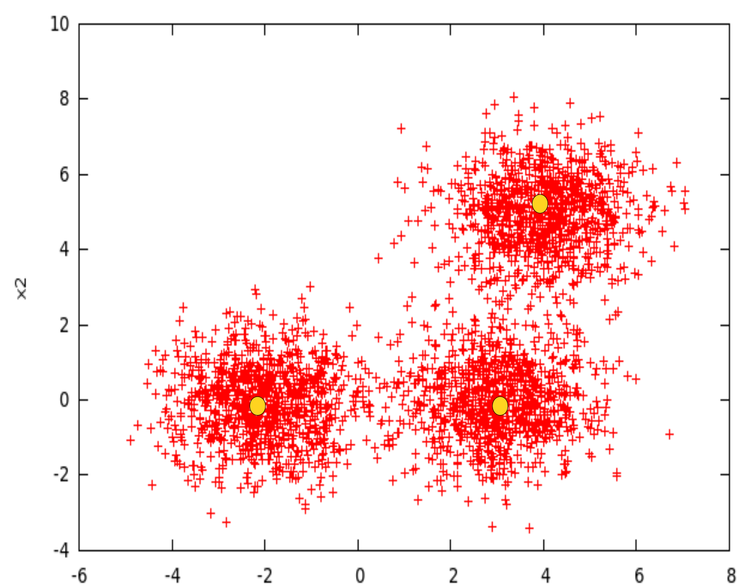
- ***K-Means*** ist ein sehr einfaches und gängiges Verfahren zur Cluster-Analyse
- ***Gegeben***: Samples $\boldsymbol{x_1, ..., x_n} \in \boldsymbol{IR^m}$
- ***Grundannahmen***:
  - Es existieren ***K*** Zentren $\boldsymbol{\mu_1, ..., \mu_K} \in \boldsymbol{IR^m}$ („***K means***")
  - Jedes Sample $\boldsymbol{x_i}$ ist einem Zentrum **k(i)** zugeordnet
  - Die Verteilung zu jedem Zentrum ist ***sphärisch***



K-Means: Grundidee

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

- Es liegt ein ***Henne-Ei-Problem*** vor
- Wüssten wir die ***Cluster-Zuordnung*** ***k(i)*** jedes Samples $\boldsymbol{x}_i$, könnten wir die Zentren ermitteln (z.B. mittels ***ML-Schätzung***)
- Wüssten wir die ***Zentren***, könnten wir die Cluster-Zuordnung ***k(i)*** ermitteln (wir wählen zu jedem Datenpunkt das ***nächstgelegene Zentrum***)
- ***Ansatz***: Wir **fixieren** jeweils einen Aspekt und **optimieren** den anderen!



K-Means: Pseudo-Code

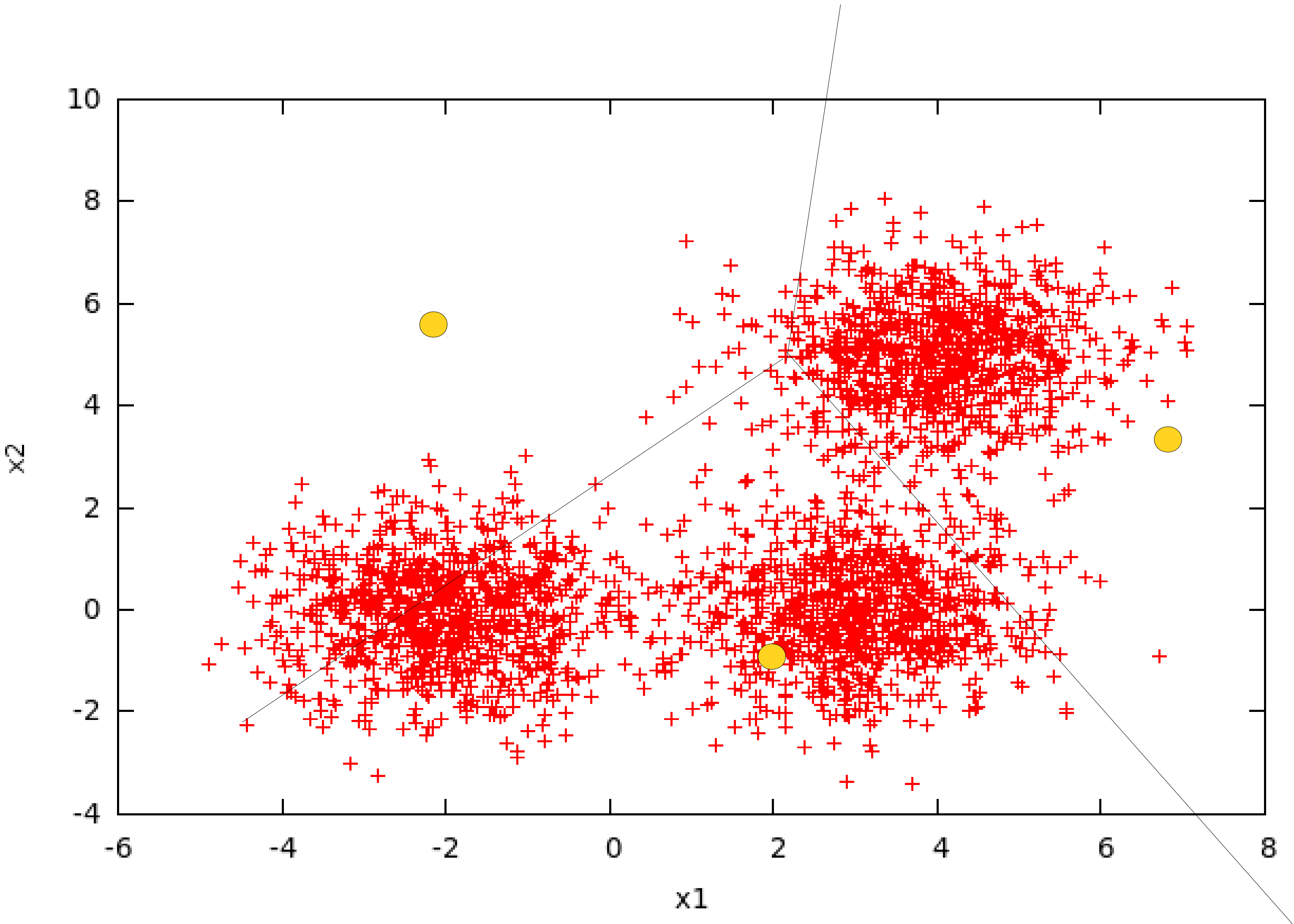
Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

Gegeben: Samples $x_1, .., x_n$

1. Initialisiere $\mu_1, .., \mu_K$ mit zufälligen Samples
2. Iteriere bis zur Konvergenz
   (a) Für $i = 1, .., n$:
   $$k(i) := \arg\min_k ||x_i - \mu_k||_2$$
   (b) Für $k = 1, .., K$:
   $$X_k := \{x_i \mid k(i) = k\}$$
   $$\mu_k := \frac{1}{|X_k|} \cdot \sum_{x \in X_k} x$$

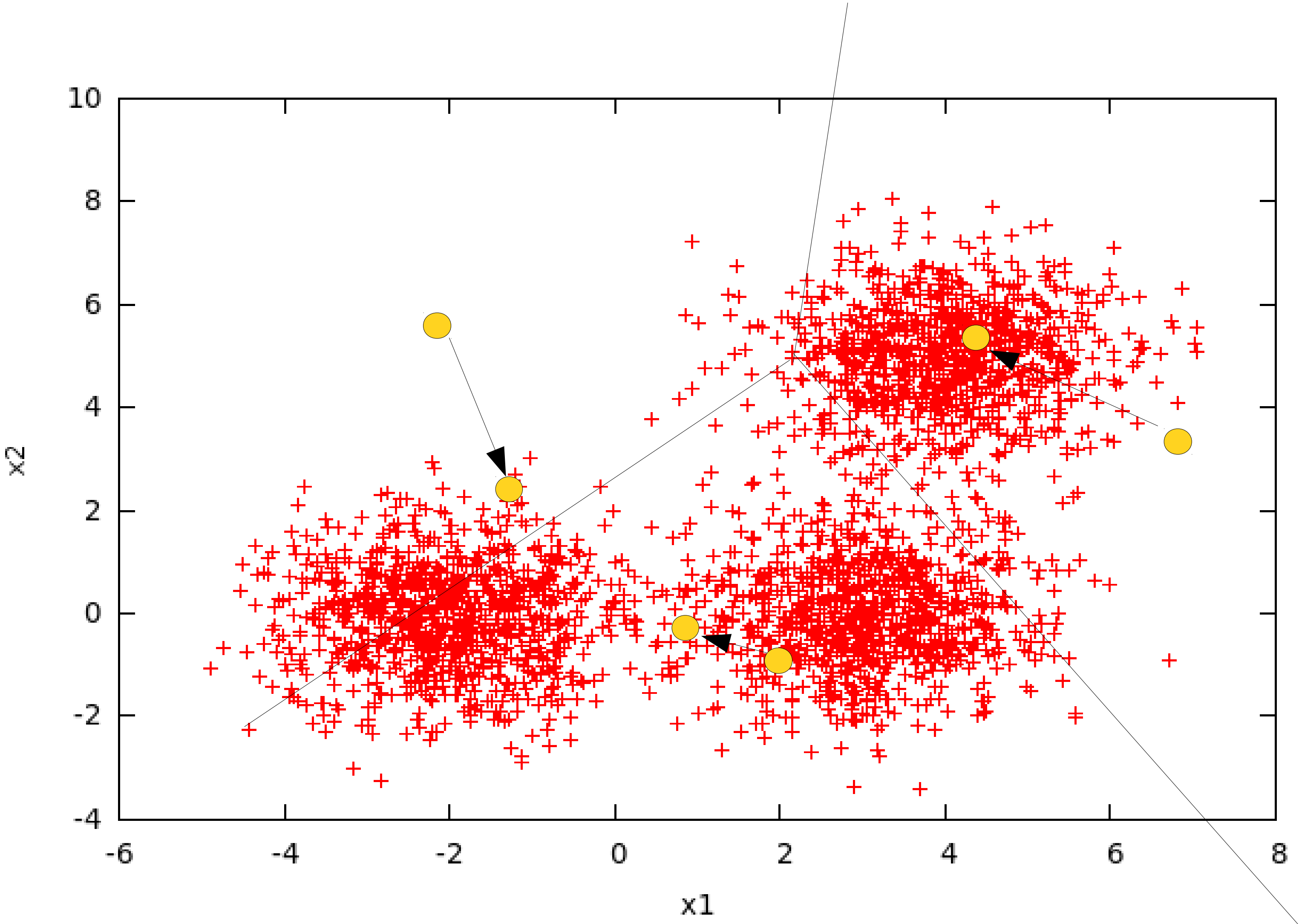
K-Means: Beispiel

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain



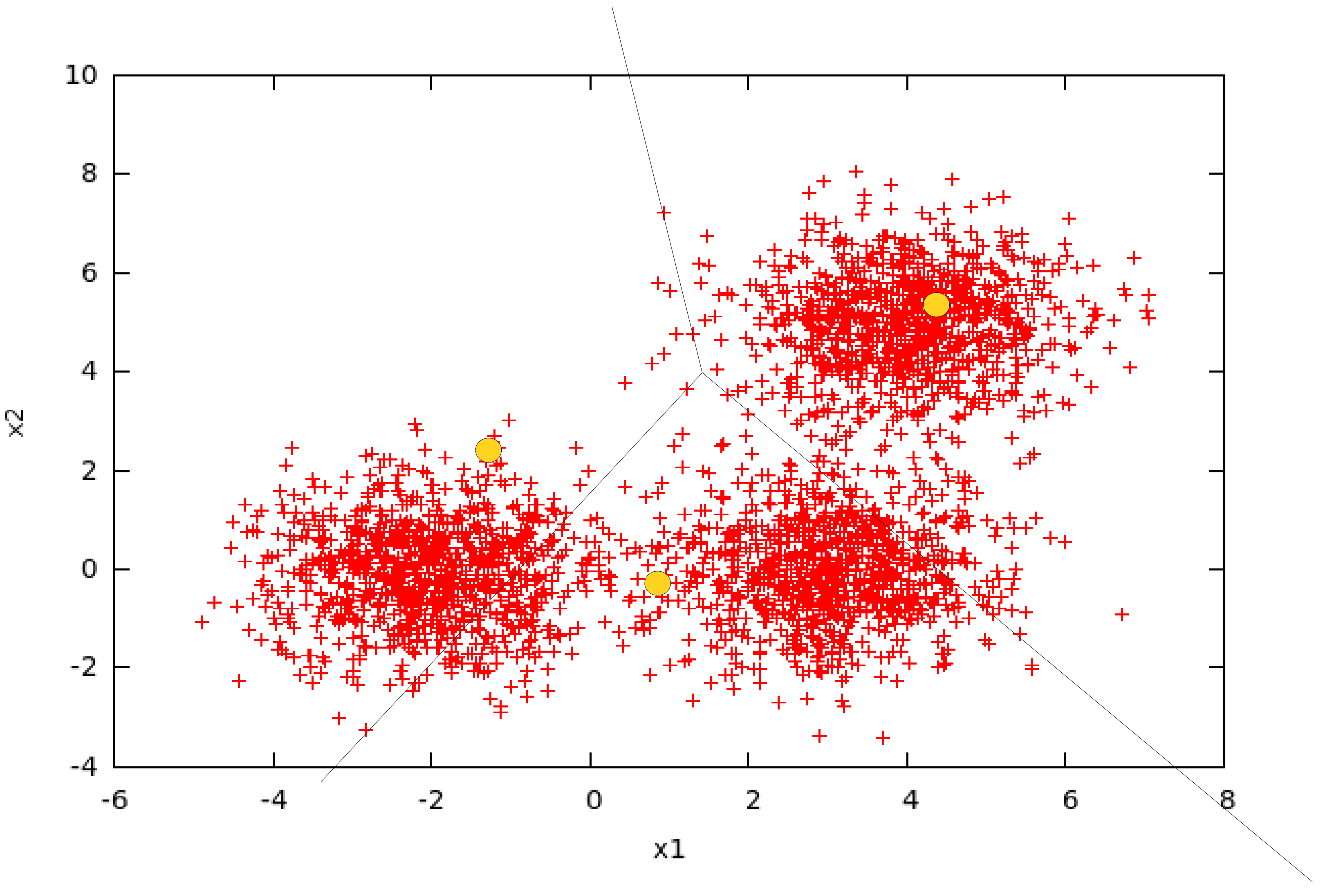
K-Means: Beispiel

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain



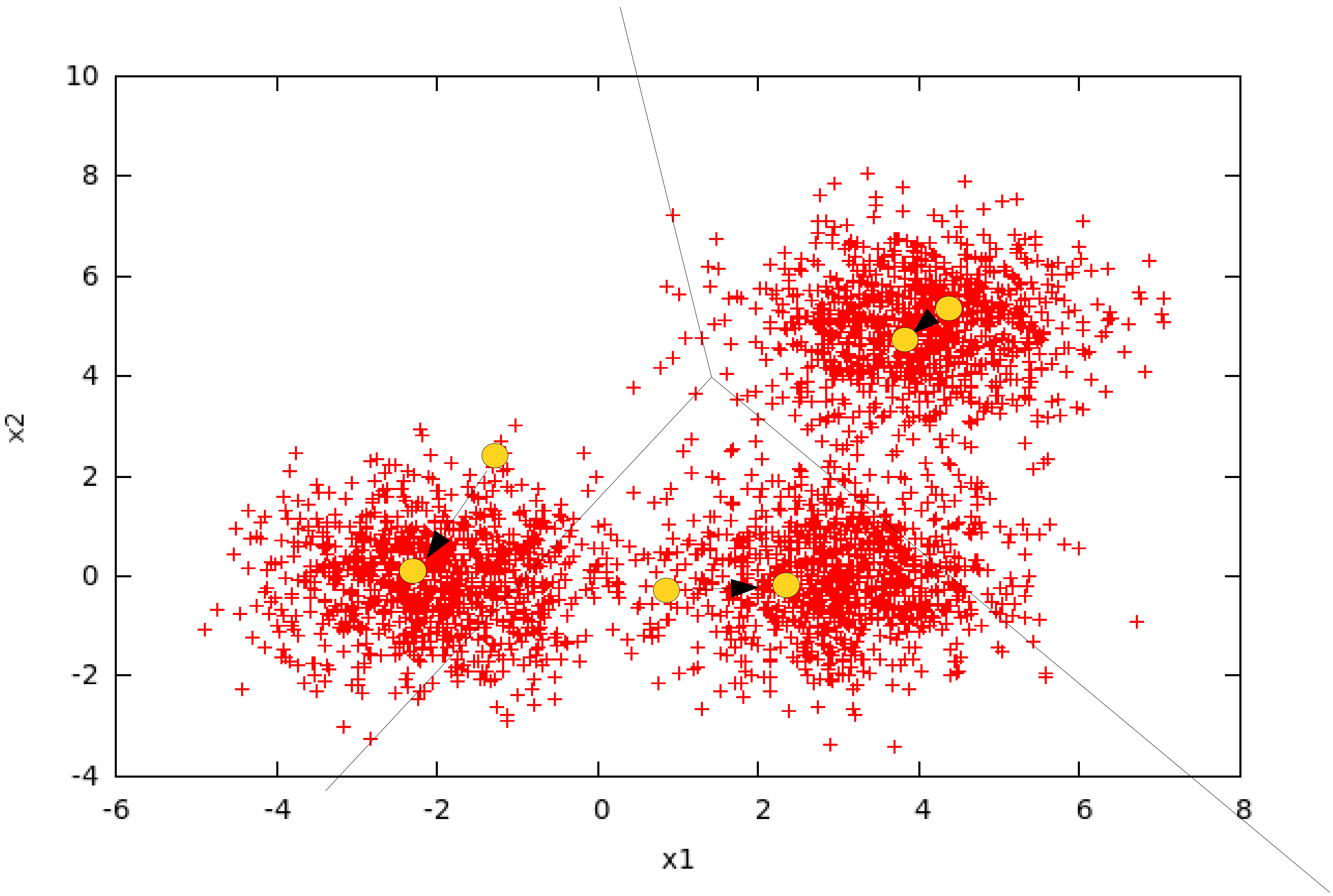
K-Means: Beispiel

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain



K-Means: Beispiel

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

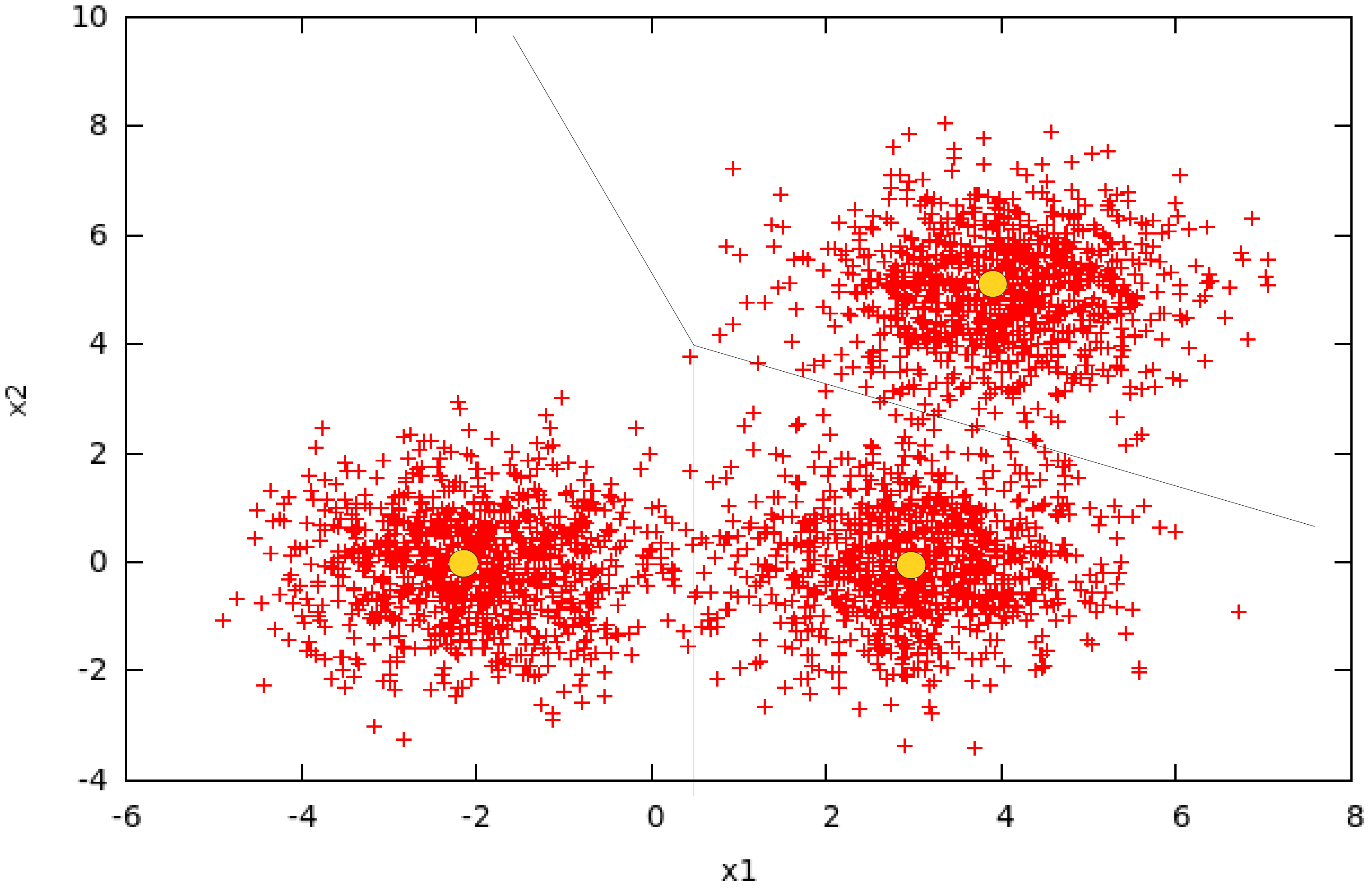


K-Means: Beispiel

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

...

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain



K-Means: Eigenschaften

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

- K-Means entspricht einer Minimierung des ***quadratischen Fehlers E***

$$E = \sum_{i=1}^{n} \Big(x_i - \mu_{k(i)} \Big)^2$$

- ***Berechnungsaufwand?***
  - Pro Iteration: ***O(k·n·m)***
  - Die ***Anzahl*** der benötigten Iterationen ist üblicher Weise moderat
- ***Konvergiert*** K-Means immer?
  - Ja. **Begründung**: Die Folge der Fehlerwerte pro Iteration, $\mathbf{E_0}$, $\mathbf{E_1}$, $\mathbf{E_2}$, … ist …
    - (a) … monoton fallend
    - (b) … nach unten beschränkt (≥ **0**)
  - Also ist die **Folge** (und somit das **Verfahren**) **konvergent**

K-Means: Eigenschaften

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

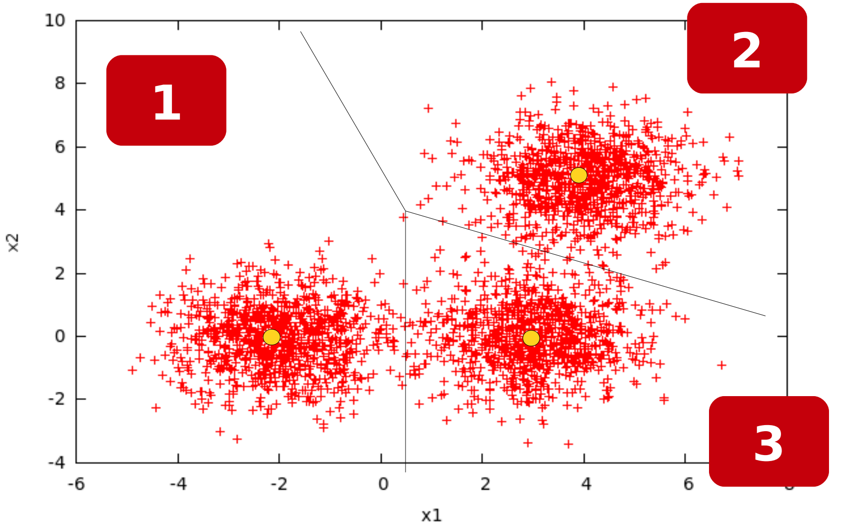
- Gibt K-Means immer ***dasselbe Ergebnis***?
- Nein – K-Means ist ein lokales Suchverfahren (ähnlich dem ***Gradientenabstieg***)
- **Problem 1**: Die Means können ***vertauscht*** sein:
  - $\mu_1=(0,0)$, $\mu_2=(1,1)$, $\mu_3=(5,3)$
  - $\mu_1=(5,3)$, $\mu_2=(0,0)$, $\mu_1=(1,1)$
- **Problem 2**: Die Mittelwerte können ***komplett andere*** sein **(lokales Minimum)**
- **Ansatz**: Starte iterativ neu, behalte das Ergebnis mit minimalem quadratischen Fehler **E**
- Außerdem können während des Algorithmus ***leere Cluster*** auftreten. **Ansatz**: initialisiere das zugehörige Zentrum zufällig neu und iteriere weiter

K-Means: Vektor-Quantisierung

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

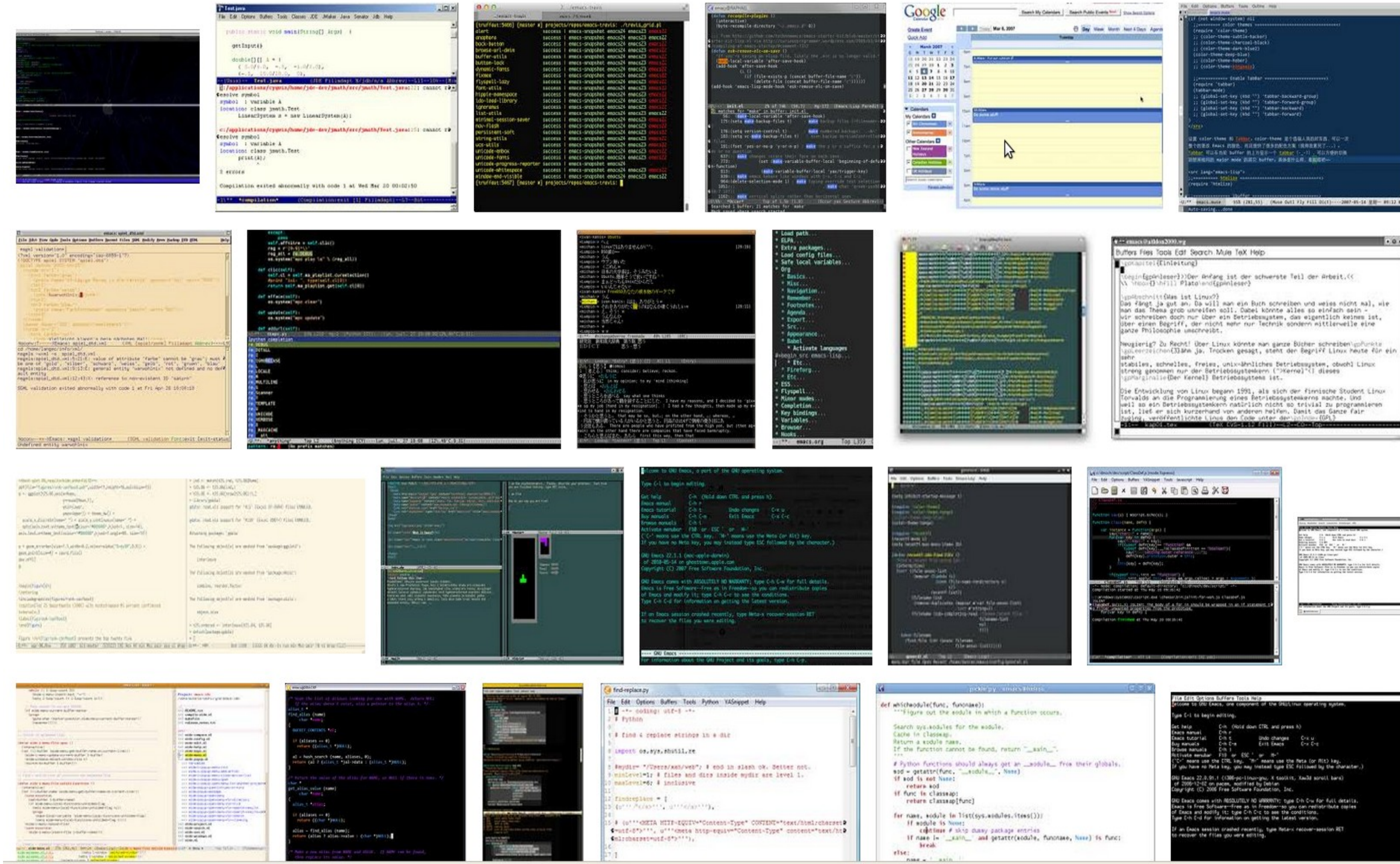
- ***Nach der Clusteranalyse*** können wir den Mittelwerten auch ***neue Samples x*** zuordnen, die nicht zur Eingabe der Clusteranalyse gehörten (***Vektor-Quantisierung***)

$$k(x) = \arg\min_k ||x - \mu_k||$$



K-Means: Code-Beispiel

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain



Überblick

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

Grundlagen

K-Means

Expectation Maximization

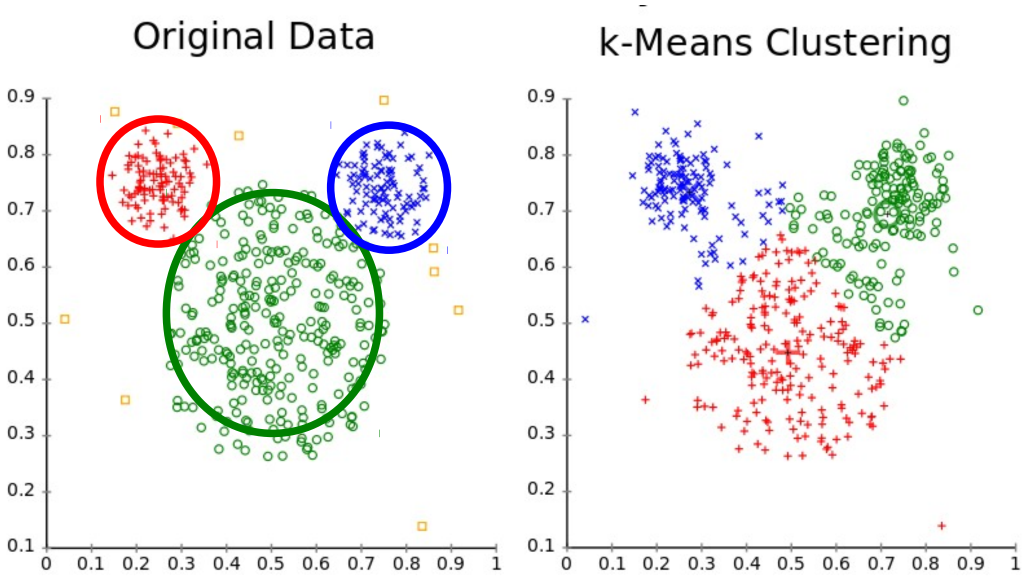
Model Selection

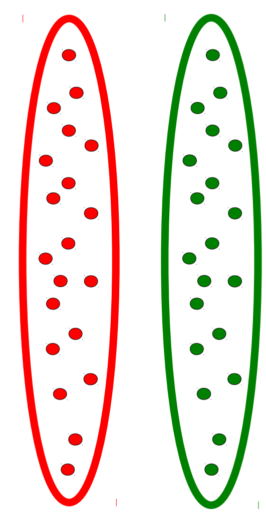
Agglomeratives Clustering

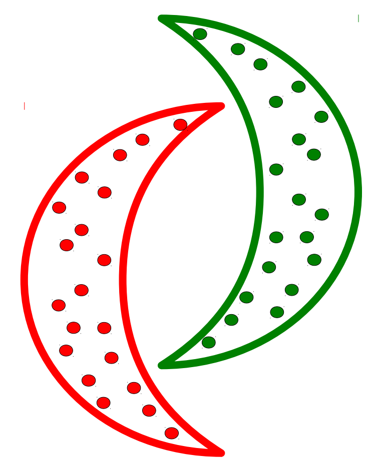
Topic Models

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

- ***Einschränkungen***: Welche Daten können wir mit K-Means nicht richtig clustern?
  → Cluster mit unterschiedlicher ***Varianz***
  → Cluster mit ***anisotroper*** Verteilung
  → ***Nicht normalverteilte*** Cluster



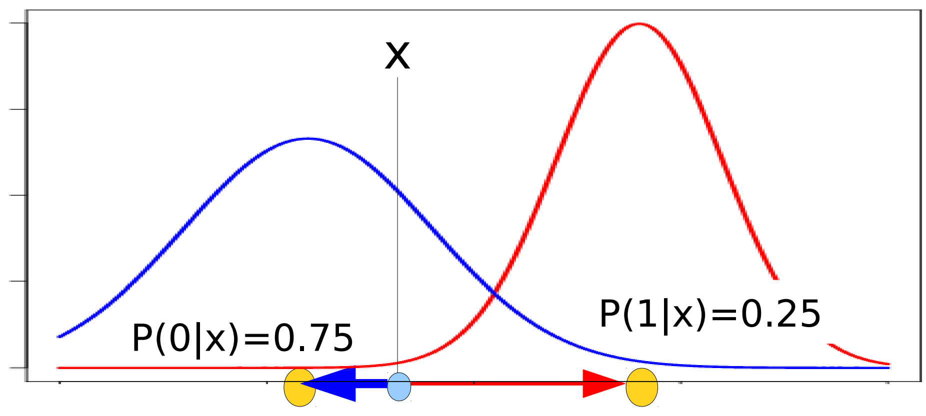




Von K-Means zu Expectation Maximization

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

- Wir können einige der vorgenannten Schwächen durch eine ***Verallgemeinerung des K-Means***-Ansatzes beheben
- Wie bei K-Means ***alternieren wir zwei Schritte***:
  - Zuweisung von Samples zu Clustern (***„E-Schritt“***)
  - Parameterschätzung der Cluster (***„M-Schritt“***)
- Der resultierende Algorithmus heißt ***Expectation Maximization***
- ***Unterschiede zu K-Means***:
  - „***E-Schritt***“: Keine „harte“ Zuweisung von Samples zu Zentren, sondern Schätzung einer ***Wahrscheinlichkeit*** $P(k|x_i)$
  - „***M-Schritt***“: Neben dem Zentrum wird auch die Varianz / Kovarianz geschätzt



EM vs. K-Means

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

„E-Schritt“

$$P(k|x_i) = \frac{\mathcal{N}(x_i; \mu_k, \sigma_k)}{\sum_{k'} \mathcal{N}(x_i; \mu_{k'}, \sigma_{k'})}$$

EM

$$k(i) = \arg\min_k ||x_i - \mu_k||^2$$

K-Means

„M-Schritt“

$$\mu_k := \frac{\sum_i P(k|x_i) \cdot x_i}{\sum_i P(k|x_i)}$$

$$\sigma_k^2 := \frac{\sum_i P(k|x_i) \cdot (x_i - \mu_k)^2}{\sum_i P(k|x_i)}$$

EM

$$\mu_k^{t+1} := \frac{\sum_i \mathbf{1}_{k(i)=k} \cdot x_i}{\sum_i \mathbf{1}_{k(i)=k}}$$

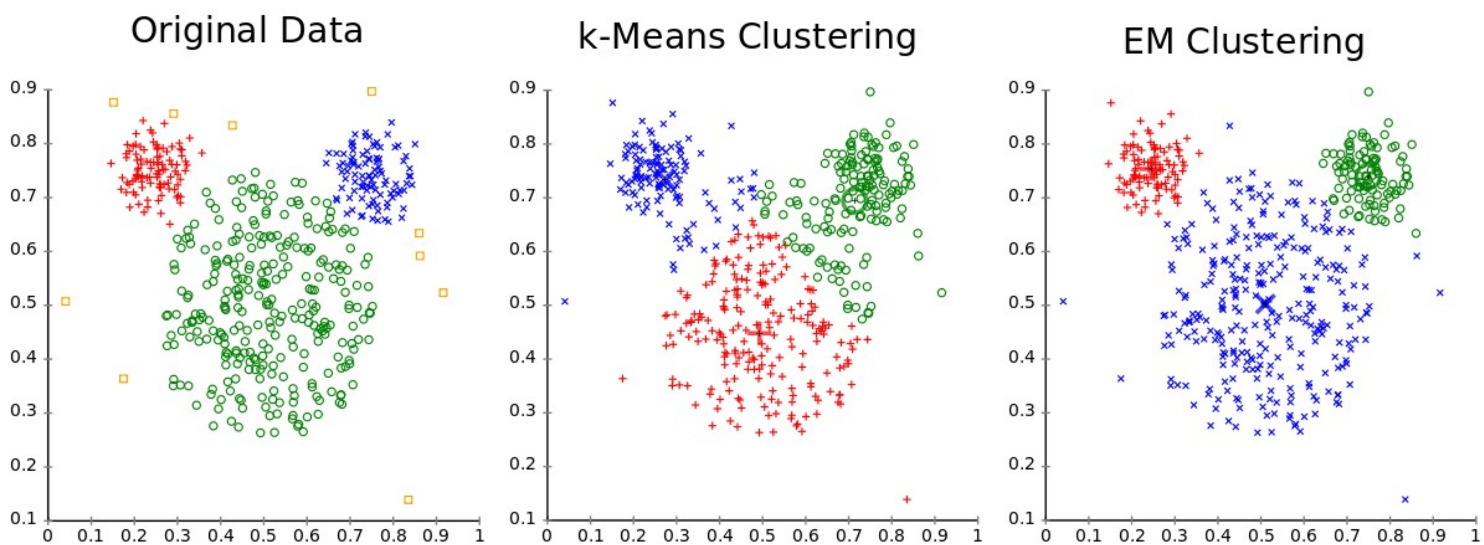
—

K-Means

EM: Diskussion

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

- Expectation Maximization kann Normalverteilungen mit ***beliebigen Varianzen*** lernen
- **Konvergenz** ist garantiert!
- Aber: EM ist immer noch ein ***lokales Suchverfahren***
- Das heisst: Verschiedenen Neustarts können immer noch zu unterschiedlichen (potenziell suboptimalen) Ergebnissen führen.



Überblick

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

Grundlagen

K-Means

Expectation Maximization

Model Selection

Agglomeratives Clustering

Topic Models

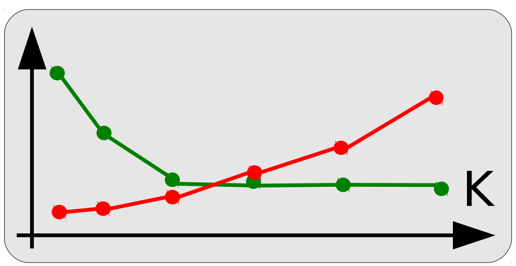
Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

- ***Offene Frage***: ***Anzahl der Cluster K?***

  → ***K zu klein***: Höherer Fehler, Untersegmentierung

  → ***K zu groß***: Komplexes Modell, instabile Optimierung, Übersegmentierung

- **Ansatz**: ***Bayes'sches Informationskriterium***



Dimension der Daten

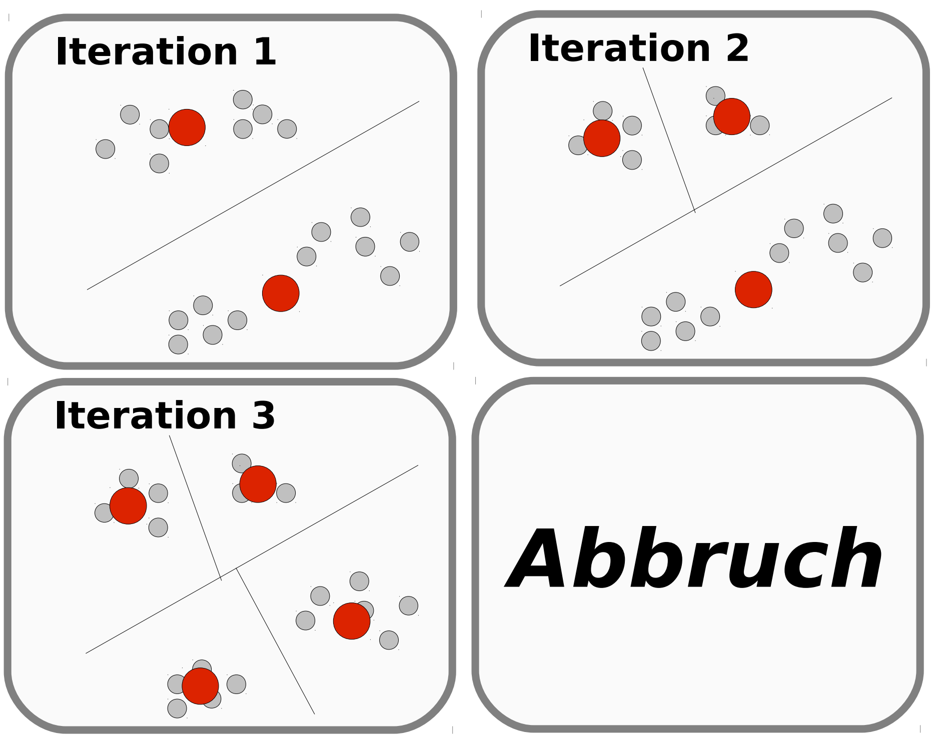
#Samples

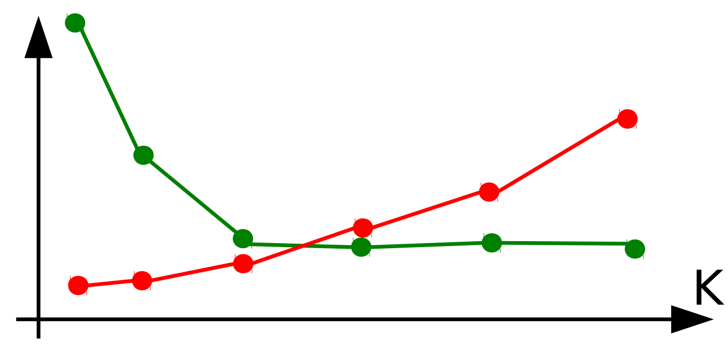
$$K^* = \arg\min_K \underbrace{\sum_{i=1}^{n} \Big(x_i - \mu_{k(i)}\Big)^2}_{E(k)} + \underbrace{m \cdot K \cdot \log(n)}_{\text{"Modell-Komplexität"}}$$

Hierarchischer K-Means

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

- **Hierarchischer Algorithmus**: **Wiederhole** ...
  - Füge ein neues Zentrum hinzu (z.B. durch Split des größten Clusters)
  - Wende K-Means rekursiv auf den einzelnen Clustern an
- **Breche ab**, wenn die Güte sich nicht mehr verbessert

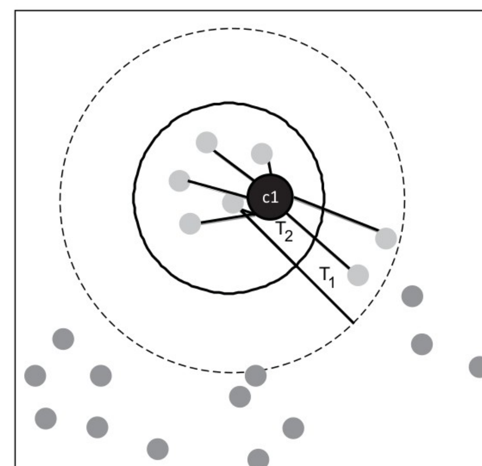




Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

- **Ansatz 3**: ***Canopy Clustering***
  - Eine ***Greedy-Strategie***, um auf großen Datenmengen (potenziell suboptimale) Cluster zu finden
  - ***Anwendung***: Schätzer für **K** und Initialisierung für K-Means
  - Die entstehenden Cluster können überlappen!
  - Zwei ***Schwellwerte***: T_1 (bestimmt die Anzahl der Cluster), T_2 (bestimmt die Überlappung)
  - Oft einfaches Distanzmaß d(.) als Approximation (Aufwand $O(n^2)$)

<u>Gegeben: Menge von Samples X</u>
(1) $C \leftarrow \{\}$ // Menge von Zentren/„canopies"
(2) Wähle zufällig ein Sample $x \in X$
(3) $Y \leftarrow \{y \mid d(y\text{-}x) \leq T_1\}$ // „sehr ähnliche" Samples
(4) $Z \leftarrow \{z \mid T_1 < d(z\text{-}x) \leq T_2\}$ // „ähnliche" Samples
(5) $C \leftarrow C \cup \{x\}$
(6) $X \leftarrow X \setminus Y$
(7) Falls X<>{}: gehe zu (2)
(8) return C



Überblick

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

- Grundlagen
- K-Means
- Expectation Maximization
- Model Selection
- **Agglomeratives Clustering**
- Topic Models

Agglomeratives Clustering

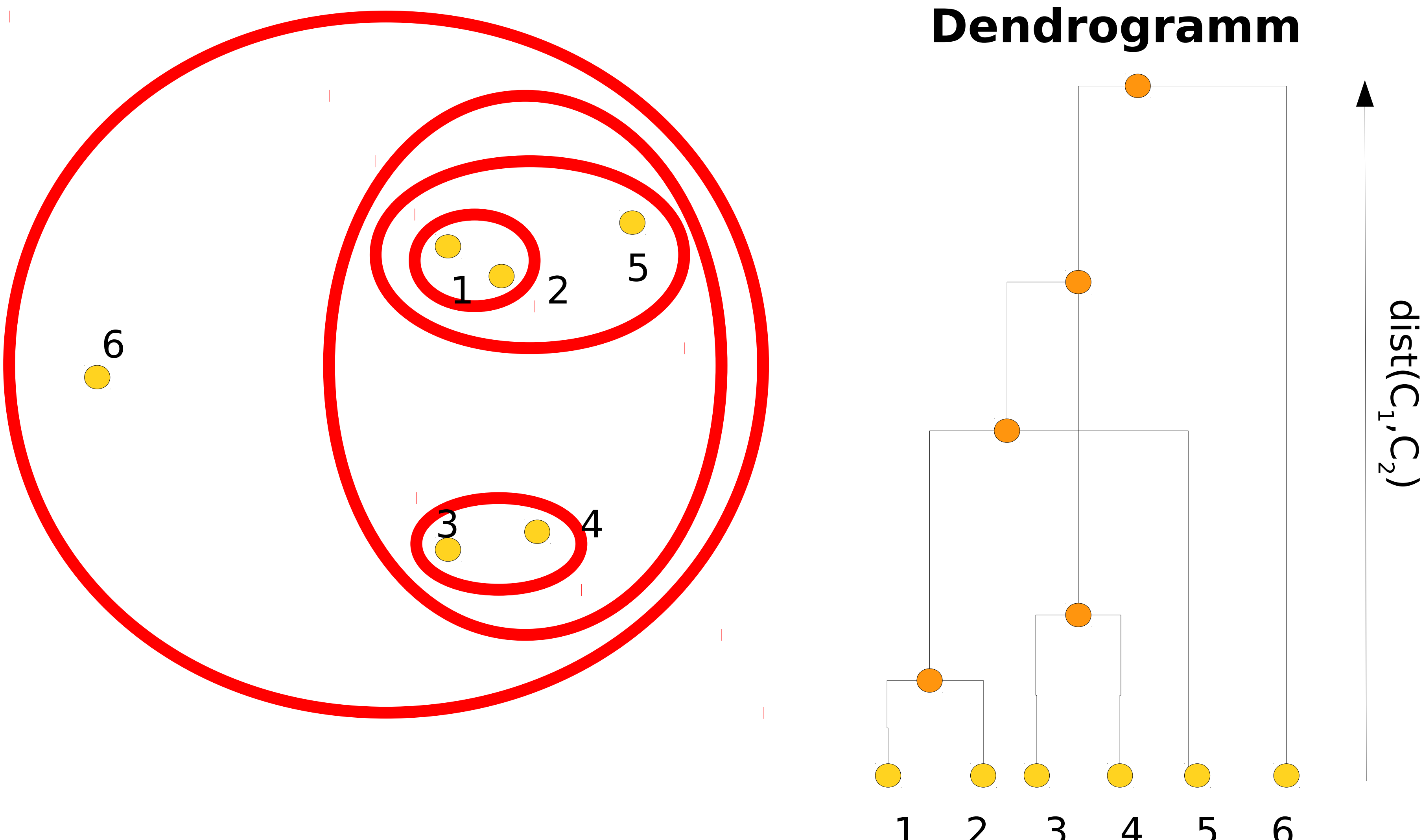
Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

- Wir bezeichnen K-Means/EM als ***divisive Clustering-Verfahren***, weil sie (***top-down***) die Datenmenge ***unterteilen***.
- Alternative: ***Agglomeratives Clustering***
  - ***Startzustand***: Jedes Sample befindet sich in seinem „eigenen“ Cluster („***singletons***“)
  - ***Iterativ***: ***Mische*** die beiden „ähnlichsten“ Cluster zu einem neuen Cluster
- Das Ergebnis kann mittels eines sogenannten ***Dendogramms*** in Baumform illustriert werden.

Agglomeratives Clustering: Beispiel

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

Dendrogramm



Agglomeratives Clustering: Distanzmaß

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

- Zwei offene ***Fragen***:
    1. Distanzmaß
    2. Abbruchbedingung
- ***Abbruchbedingung***: Heuristiken (siehe „Model Selection" / K-Means)
- ***Distanzmaß***: Drei gängige Alternativen (seien X,Y Cluster)

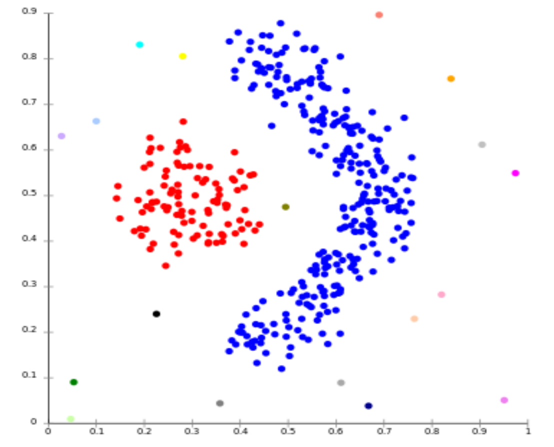
Single Linkage:	$dist(X,Y) = \min_{x \in X, y \in Y} \|\|x-y\|\|_2$
Complete Linkage:	$dist(X,Y) = \max_{x \in X, y \in Y} \|\|x-y\|\|_2$
Average Linkage:	$dist(X,Y) = \frac{1}{\|X\| \cdot \|Y\|} \sum_{x \in X, y \in Y} \|\|x-y\|\|_2$

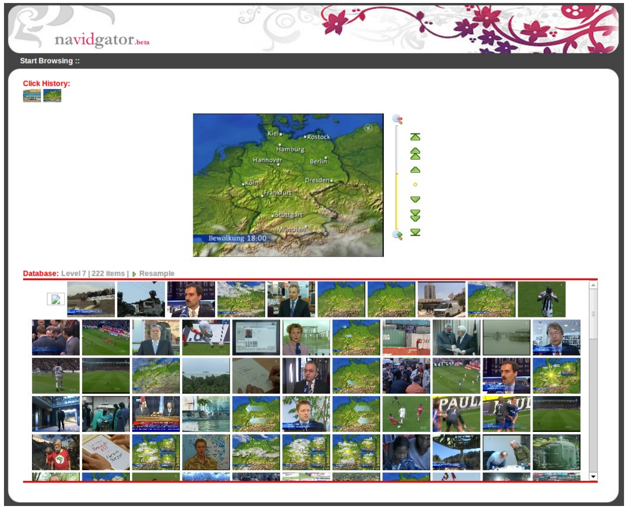
Agglomeratives Clustering: Diskussion

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

- ***Nachteil: Berechnungsaufwand***
  - Aufbau Ähnlichkeitsmatrix: $\mathbf{O(n^2)}$
  - Anzahl der Fusion-Schritte: $\mathbf{O(n)}$
  - ***Pro Fusion-Schritt***: Neuberechnung der Ähnlichkeit des entstehenden Clusters zu den anderen Clustern: $\mathbf{O(n)}$
    (oder noch teurer)

    → Insgesamt: $\mathbf{O(n^2)}$:-(
  - Häufig grobes ***Pre-Clustering*** und Anwendung agglomerativer Verfahren in den einzelnen Clustern
- ***Vorteile***
  - Auch geeignet für nicht-normalverteilte Cluster





Überblick

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

- Grundlagen
- K-Means
- Expectation Maximization
- Model Selection
- Agglomeratives Clustering
- **Topic Models**

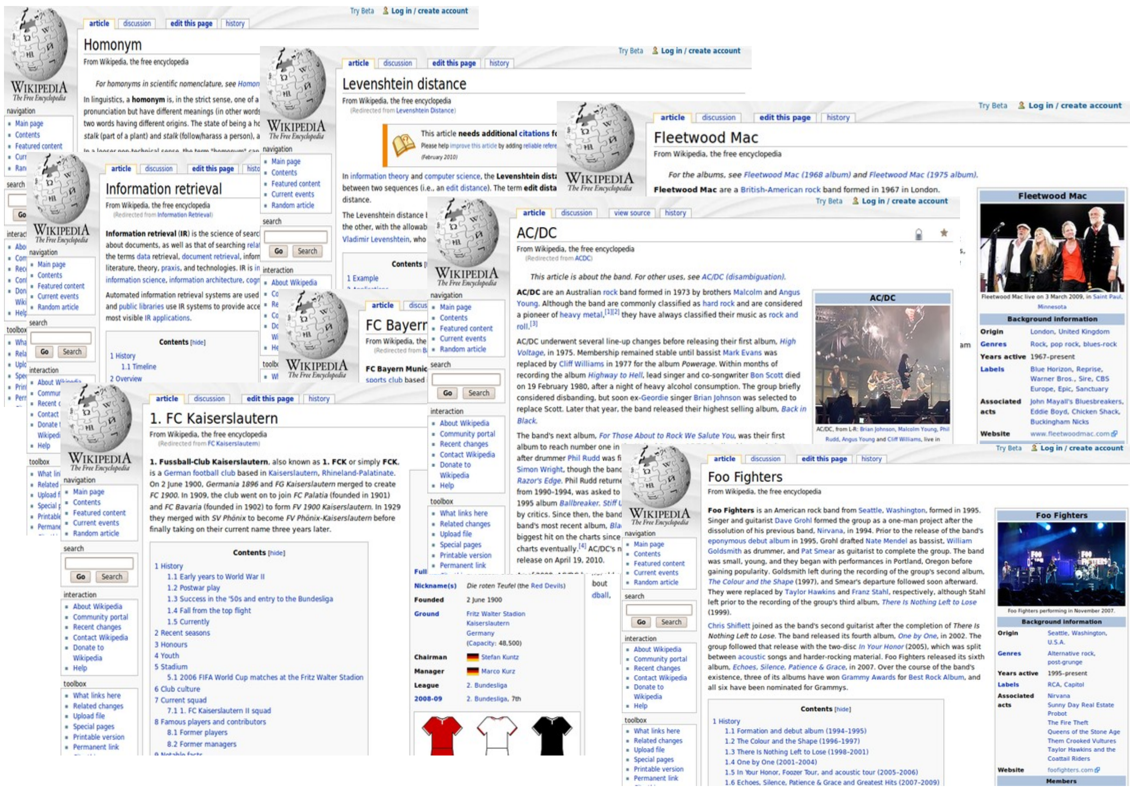
EM Anwendung: Text Clustering

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

- Ein weiteres Verfahren zum Clustering von Text heißt ***PLSA*** (*Probabilistic Latent Semantic Analysis*).
- Wir modellieren Dokumente als ***„bag-of-words“***
- EM liefert zwei interessante Resultate:
  - ***Cluster*** von Dokumenten
  - Die Cluster-Zentren entsprechen prototypischen Schlagwort-Verteilungen (oder „***topics***“), in denen jeweils bestimmte Terme besonders häufig vorkommen.
- Wir nennen Modelle dieser Art ***„topic models“***.

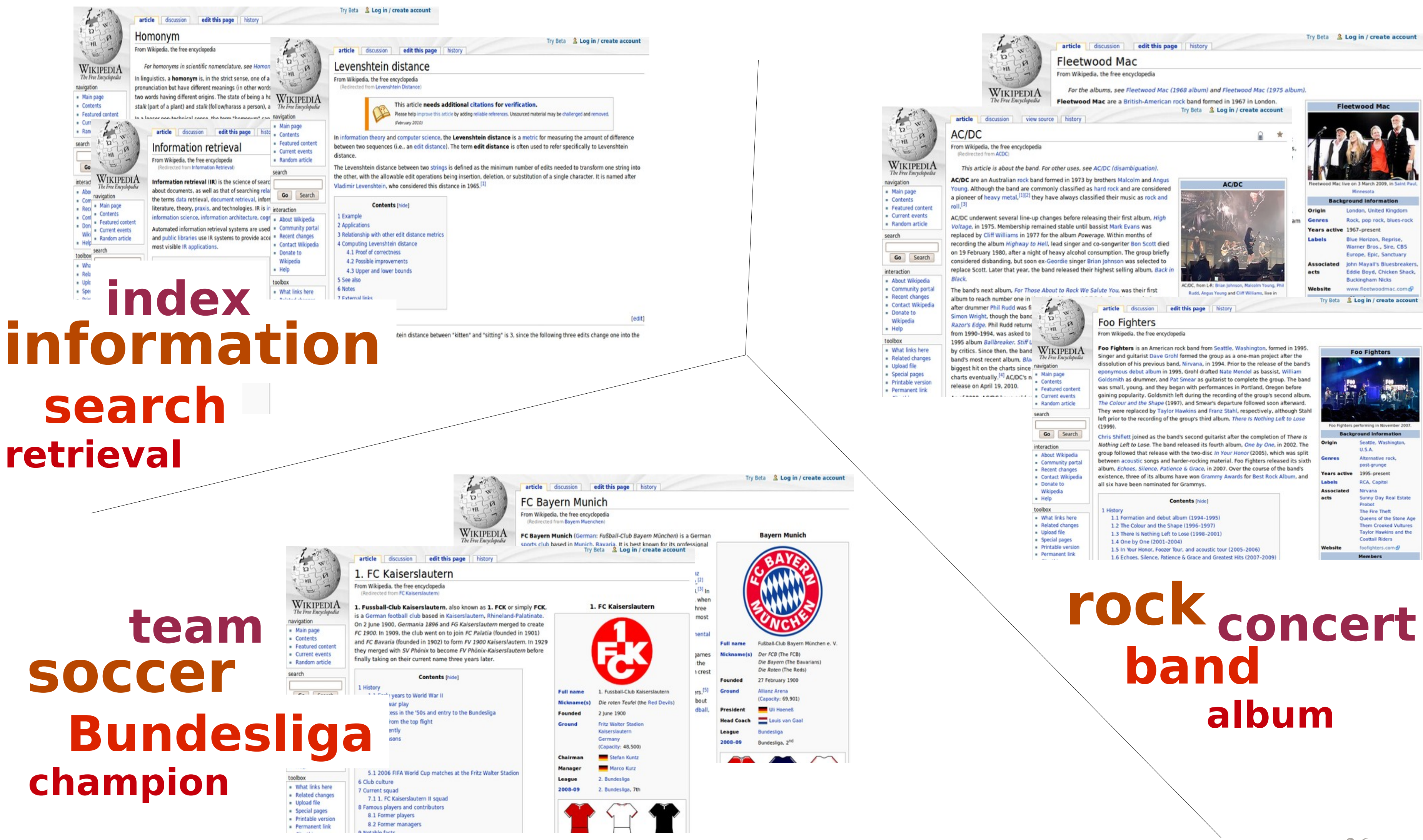
EM Anwendung: Text Clustering

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain



EM Anwendung: Text Clustering

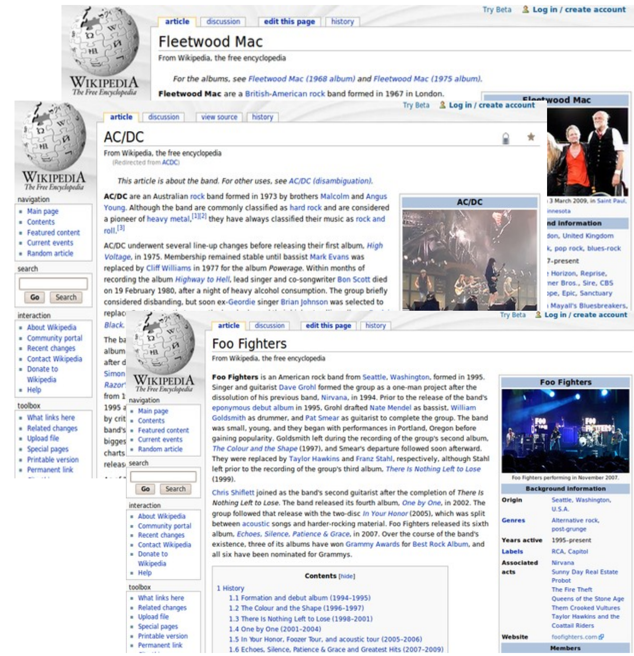
Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain



Das PLSA-Modell

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

- Eingabe: ***Dokumente*** $\boldsymbol{d_1, \ldots, d_M}$
- Jedes Dokument **d** wird repräsentiert durch sein ***Bag-of-Words*** Feature: Für jedes Wort **w** kennen wir die Wahrscheinlichkeit **P(w|d)**.
- Diese Wahrscheinlichkeiten speichern wir in einem ***Vektor*** (siehe IR).
- ***PLSA*** nimmt außerdem an, dass in der Dokumentsammlung **K** ***Themen (topics)*** $\boldsymbol{z_1, \ldots, z_k}$ auftreten.
- Jedes Topic hat wieder eine ***Wortverteilung*** **P(w|z)**.
- Die Topics sollen mittels unüberwachten Lernens ***automatisch ermittelt*** werden!



rock **concert** **band** **album**

Das PLSA-Modell

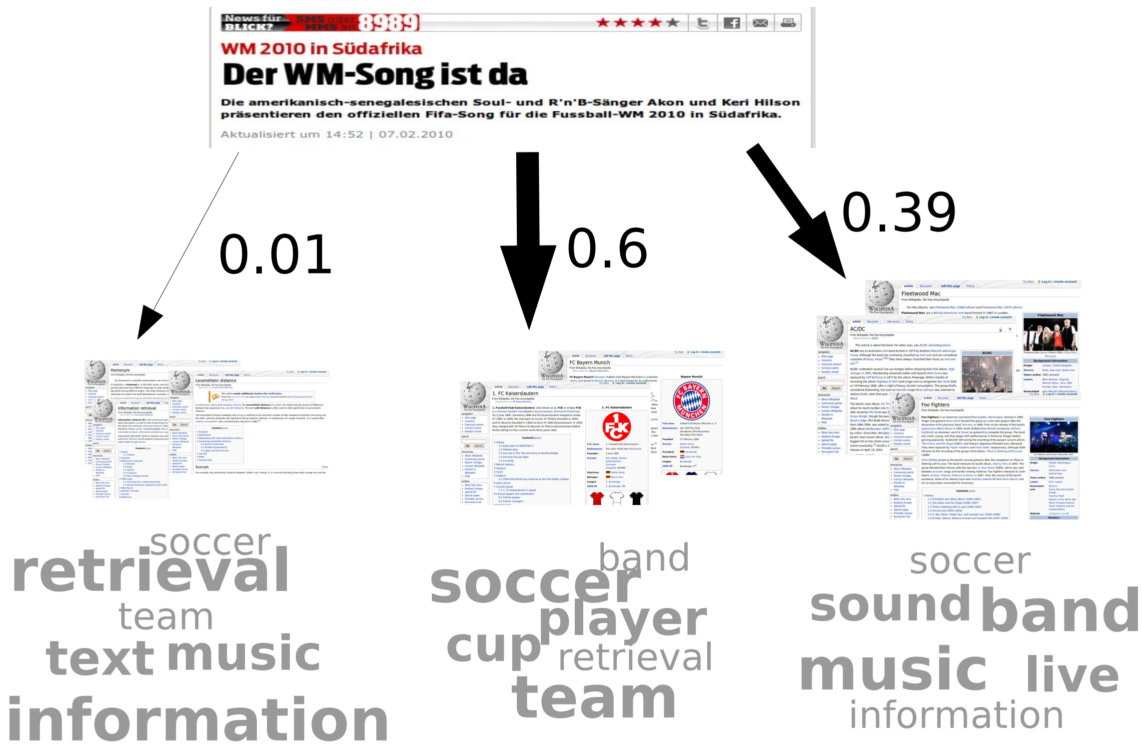
Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

- ***Annahme 1***: Ein Dokument ist eine ***Mischung aus Topics***.
- ***Annahme 2***: Wörter des Dokumentes werden generiert, indem wir (a) ein Topic wählen, und (b) ein zufälliges Wort des gewählten Topics wählen.

1. Wähle topic: P(z|d)

topics

2. Wähle Wort: P(w|z)



Das PLSA-Modell

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

- ***Ziel des PLSA-Clusterings***: Ermittle...
  - Die ***Topics***: **P(w|z)**
  - Die ***Verteilung der Topics*** auf Dokumente: **P(z|d)**
- ***Erinnerung (EM)***: Zwei abwechselnde Schritte
  1. Ordne jedem Trainingssample ein Clustern zu.
  2. Adaptiere Cluster-Parameter (Form der Cluster).
- **EM für PLSA: Dieselben Schritte.**
  1. Ordne jedem Wort in jedem Dokument ein Topic zu
     → P(z|w,d).
  2. Adaptiere Topics und Topicverteilung
     → P(w|z), P(z|d).

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

given: documents $d_1, .., d_M$, words $w_1, .., w_N$.

given: $P(w|d)$, number of topics K.

1. initialize $P(w|z)$, $P(z|d)$ randomly
2. until convergence:
   (a) fit word-topic correspondences:
   for $m = 1, .., M$, $n = 1, .., N$, $k = 1, .., K$:
   $$P(z_k|w_n, d_m) = \frac{P(w_n|z_k) \cdot P(z_k|d_m)}{\sum_{k'} P(w_n|z_{k'}) \cdot P(z_{k'}|d_m)}$$
   (b) re-estimate parameters:
   for $m = 1, .., M$, $n = 1, .., N$, $k = 1, .., K$:
   $$P(z_k|d_m) = \frac{\sum_n n(w_n, d_m) \cdot P(z_k|w_n, d_m)}{n(d_m)}$$
   $$P(w_n|z_k) = \frac{\sum_m n(w_n, d_m) \cdot P(z_k|w_n, d_m)}{\sum_{n'} \sum_m n(w_{n'}, d_m) \cdot P(z_k|w_{n'}, d_m)}$$

$n(w_n,d_m)$ Anzahl der Vorkommen von w_n in d_m.

Unüberwachtes Lernen

PLSA: Diskussion

Prof. Adrian Ulges
Fachbereich DCSM / Informatik
Hochschule RheinMain

- PLSA ist ein EM-Algorithmus, also ein ***lokales Suchverfahren***. Wir erhalten (wieder) unterschiedliche Resultate bei unterschiedlichen Startwerten.
- PLSA löst ***Homonyme*** auf.
  - „bush“ [Garten] vs. „bush“ [Politik]
- PLSA ***„komprimiert“*** Dokumente:
  - Vorher: **P(w|d)**, hochdimensional
  - Hinterher: **P(z|d)**, niedrigdimensional

P(z|d) = (0.01, 0.6, 0.39)