9. Dezember 2015

TEXT MINING

Sebastian Wack

GLIEDERUNG

- Was ist „Text Mining“?
- Primitive Algorithmen
- Vorbereitungen
- Vektormodell
- Latent Semantic Indexing
- Clustering
- Nichtnegative Matrix Faktorisierung
- LGK Bidiagonalisierung
- Zusammenfassung

WAS IST „TEXT MINING“?

- Methoden zur Extrahierung von Informationen aus Texten (oft unstrukturiert)
- Anwendungen
  - Datenbanksysteme

  - Automatisierte Textzusammenfassung

PRIMITIVE ALGORITHMEN

- Gegeben:
  - Index mit allen vorhandenen Büchern (inkl. Autor, Titel, ISBN, Inhalt)
  - Jedes Buch im Index besitzt einen Relevanzwert
- Gesucht:
  - Funktion search()
    - Eingabe: Suchwort / Suchwörter
    - Ausgabe: nach Relevanz aufsteigend sortierte Liste von Büchern

VERSUCH 1

```
Search(String query)
{
        foreach book in index do
        {
                if(book.isbn.contains(query)  ||  …)
                {
                        book.relevance++;
                }
        }
        sort(book.relevance);
        return index;
}
```

GEGENBEISPIEL

- Search(Matrix methods pattern recognition)

→ 0 Ergebnisse

ID	Title	ISBN	…
…	…	…	…
4865	Matrix methods in data mining and pattern recognition	978-0-89871-626-9	…
…	…	…	…

VERSUCH 2

```
Search(String query)
{
        queries = split(query,‘ ‘);
        foreach word in queries do
        {
                foreach book in index do
                {
                        if(book.isbn.contains(query) || …)
                        {
                                book.relevance++;
                        }
                }
        }
        sort(book.relevance);
        return index;
}
```

GEGENBEISPIEL

- Search(Matrix Methods Pattern Recognition)

→ 0 Ergebnisse

ID	Title	ISBN	…
…	…	…	…
4865	Matrix methods in data mining and pattern recognition	978-0-89871-626-9	…
…	…	…	…

VERSUCH 3

```
Search(String query)
{
        queries = split(query,‘ ‘);
        foreach word in queries do
        {
                foreach book in index do
                {
                        if(book.isbn.toLowerCase().contains(query.toLowerCase()) || …)
                        {
                                book.relevance++;
                        }
                }
        }
        sort(book.relevance);
        return index;
}
```

GEGENBEISPIEL

- Search(Matrix Methods Pattern Recognition)

→mind. 1 Ergebnis

ABER:

- Search(computing science engineering)

→ ungenaue Ergebnisse

ID	Title	ISBN	…
…	…	…	…
4865	Matrix methods in data mining and pattern recognition	978-0-89871-626-9	…
…	…	…	…
8913	Computer science engineering	978-0-12345-678-9	…
…	…	…	…

VORBEREITUNGEN

- Für jeden Suchbegriff: Liste von Dokumenten (invertierter Index)
- Schritt 1:
  - Stopp Wörter herausfiltern
  - Beispiele:

    a, a's, able, about, above, according, accordingly, across, actually, after, afterwards, again, against, ain't, all, allow, allows, almost, alone, along, already, also, although, always, am, among, amongst, an, and, another, any, anybody, anyhow, anyone, anything, anyway, anyways, anywhere, apart, appear, appreciate, appropriate, are, aren't, around, as, aside, ask, ...

VORBEREITUNGEN

- Schritt 2:
  - Wortstämme extrahieren:
    - computable → comput
    - computational → comput
    - walked → walk
    - thrown → throw
    - adaptive → adapt

TYPISCHE SUCHANFRAGE

the use of induced hypothermia in heart surgery, neurosurgery, headinjuries, and infectious diseases. (Q1)

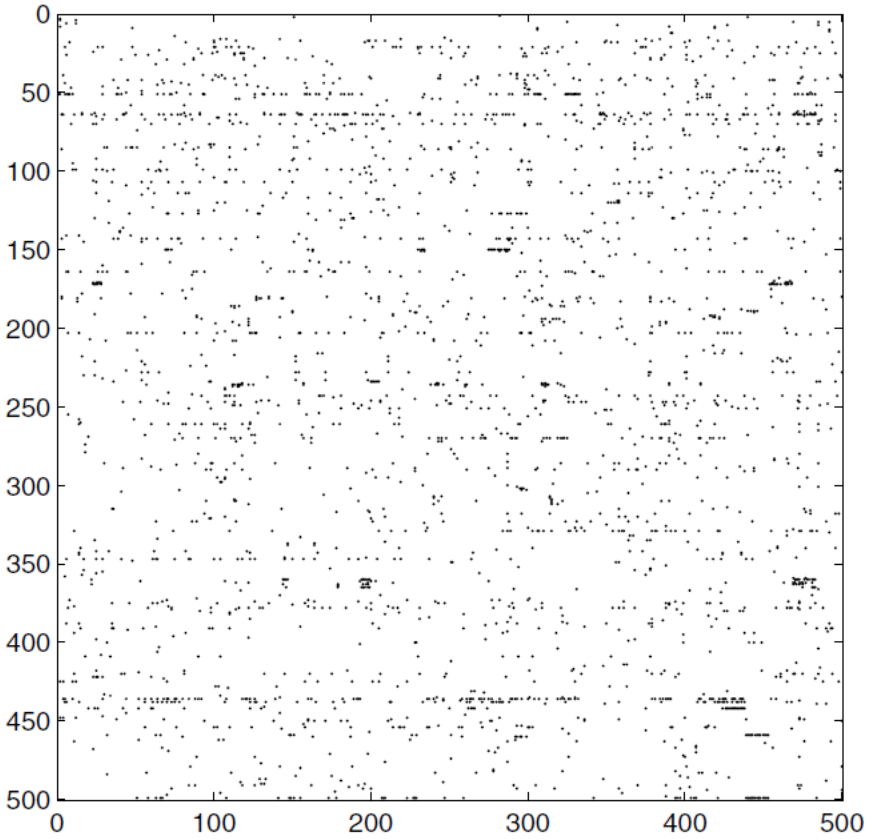
Bildquelle: https://www.iconfinder.com/icons/63121/doctor_icon

VEKTORMODELL

- Suchbegriff – Dokument Matrix
  - Dokumente: Spalten
  - Begriffe: Zeilen

- Text Parser
- Gewichtungsfunktion: $a_{ij} = f_{ij} * \log\left(\frac{n}{n_i}\right)$
- Dokument a_j ist relevant, wenn der Winkel zwischen q und a_j klein genug ist:
- $\cos\left(\theta(q, a_j)\right) = \frac{q^T a_j}{\|q\|_2 * \|a_j\|_2} > \epsilon$

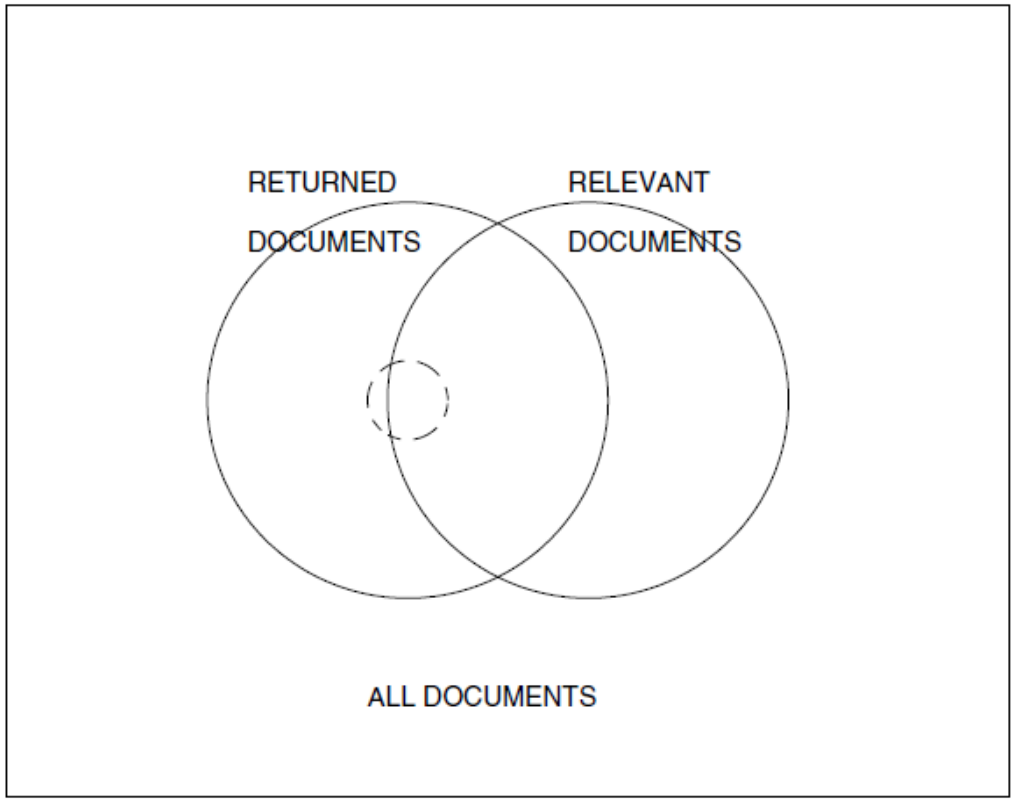
VEKTORMODELL: BEISPIEL



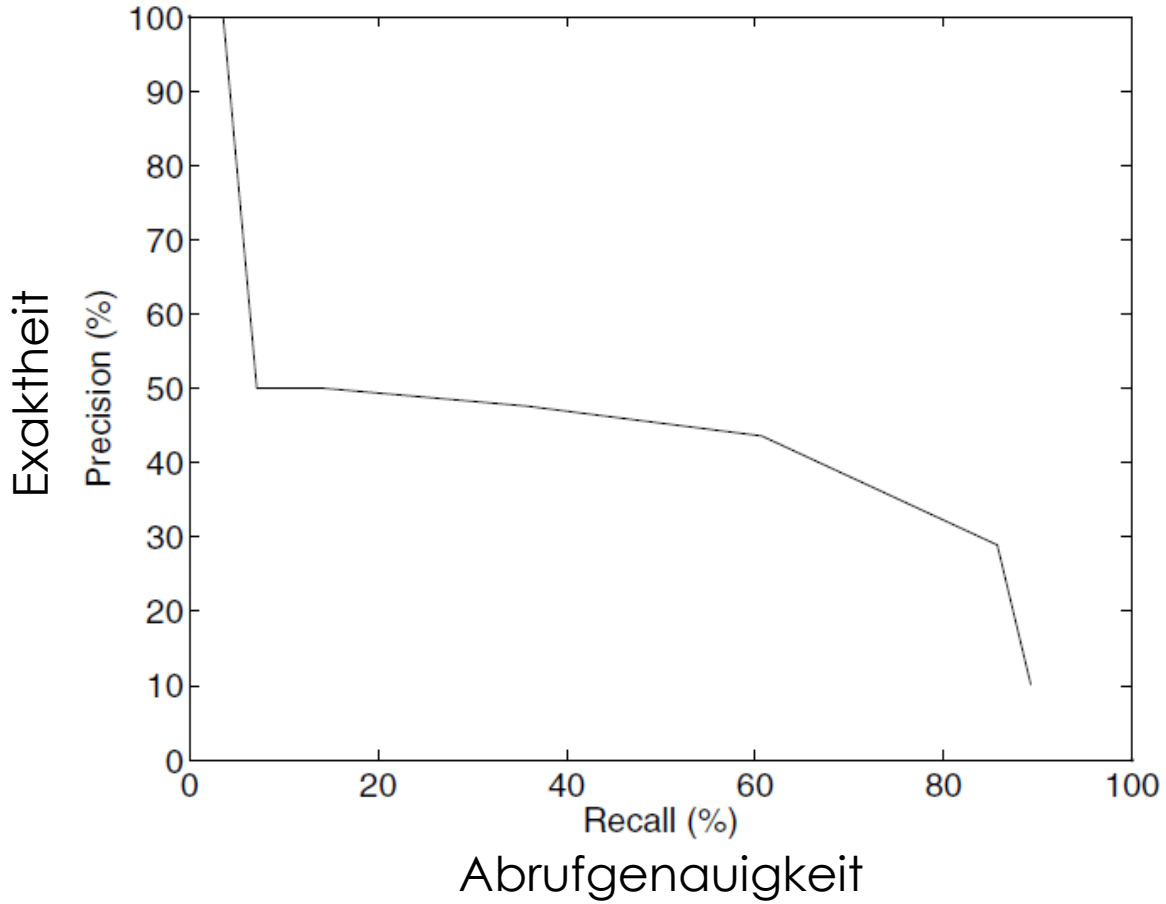
- Ersten 500 Zeilen und Spalten der Q1 Matrix

VEKTORMODELL: GENAUIGKEIT

- Exaktheit: $P = \frac{D_r}{D_t}$
- Abrufgenauigkeit: $R = \frac{D_r}{N_r}$
- D_r := # der erhaltenen Dokumente, die relevant sind
- D_t := # der erhaltenen Dokumente
- N_r := # der Dokumente, die relevant sind



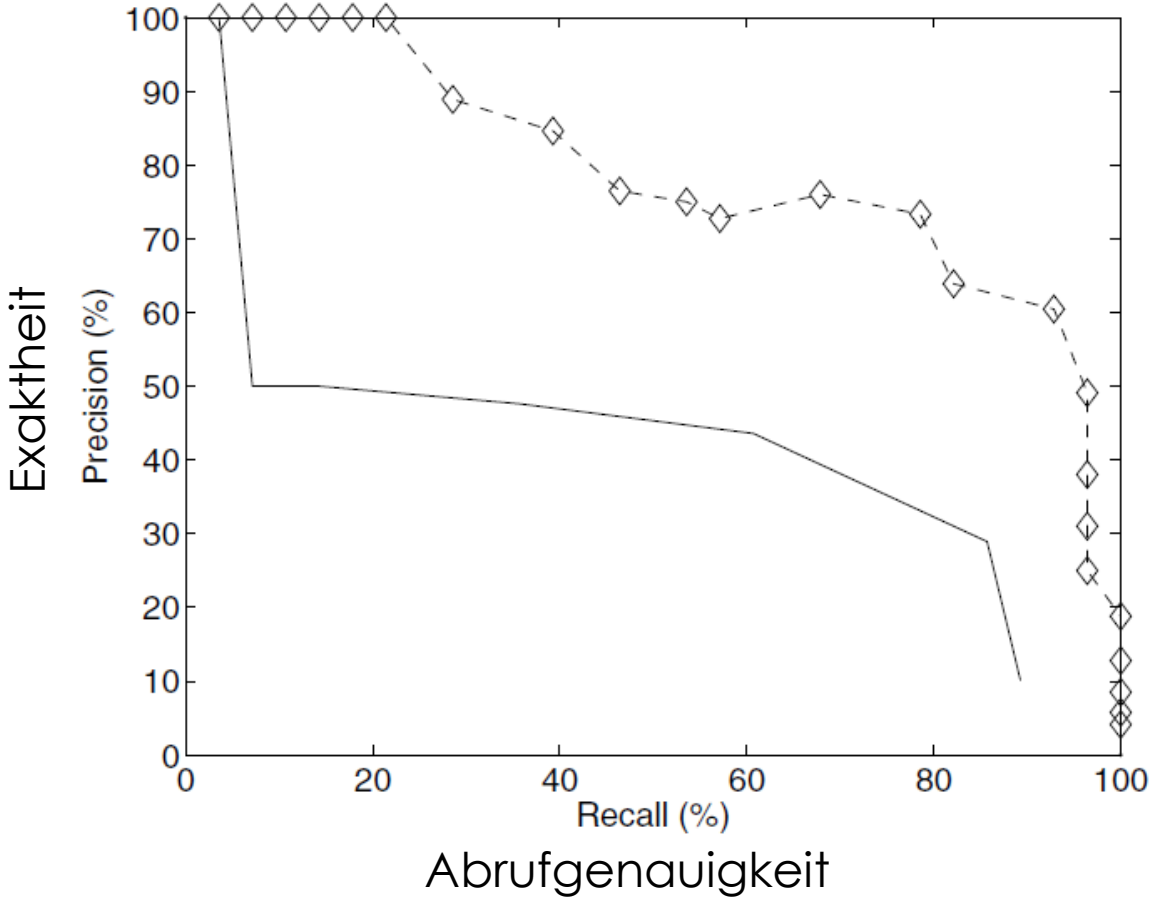
VEKTORMODELL: GENAUIGKEIT



LATENT SEMANTIC INDEXING

- Basiert auf Singulärwertzerlegung
- $A_k \approx U_k H_k \Rightarrow a_j \approx U_k h_j$
- $q^T A_k = q^T U_k H_k = \left(U_k^T q\right)^T H_k$
- $\cos(\theta_j) = \frac{q_k^T h_j}{\|q_k\|_2 \|h_j\|_2}$ $\qquad q_k = U_k^T q$

LATENT SEMANTIC INDEXING: BEISPIEL

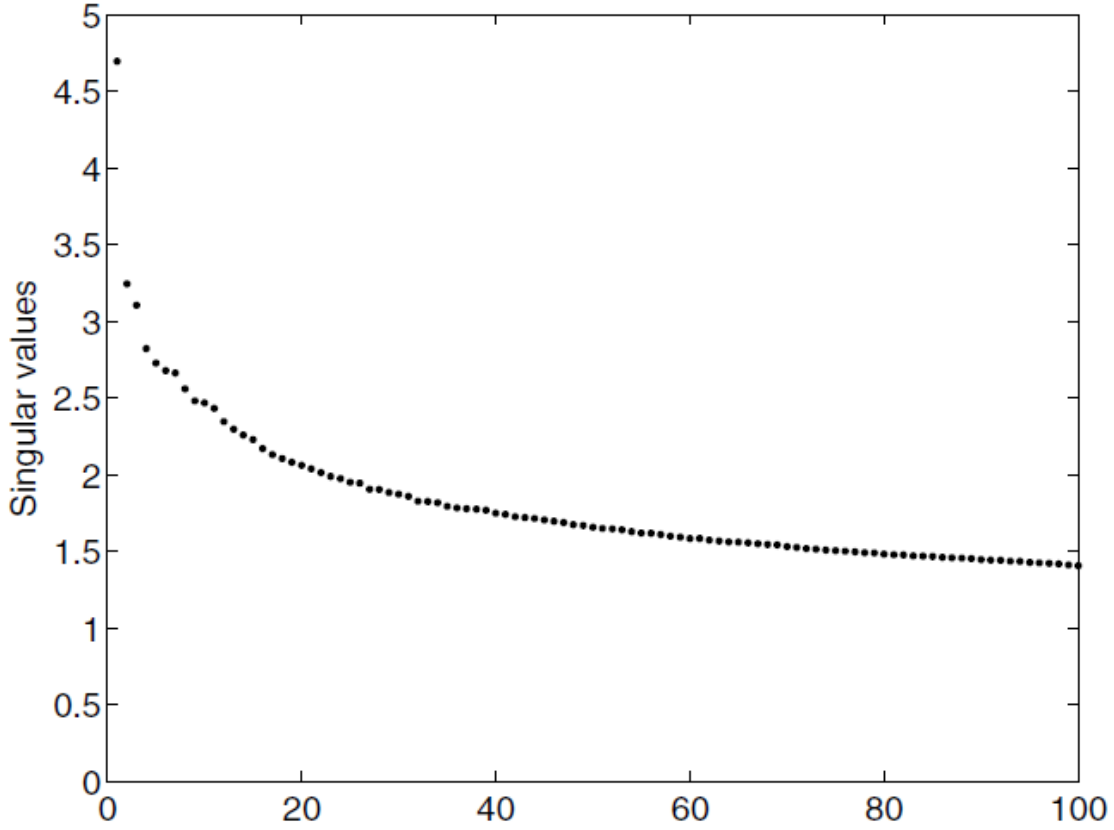


- Rang: 100
- Latent Semantic Indexing: ◇
- Vektormodell: ---

→ In diesem Fall: LSI erheblich besser

LATENT SEMANTIC INDEXING: ABSCHÄTZUNGEN

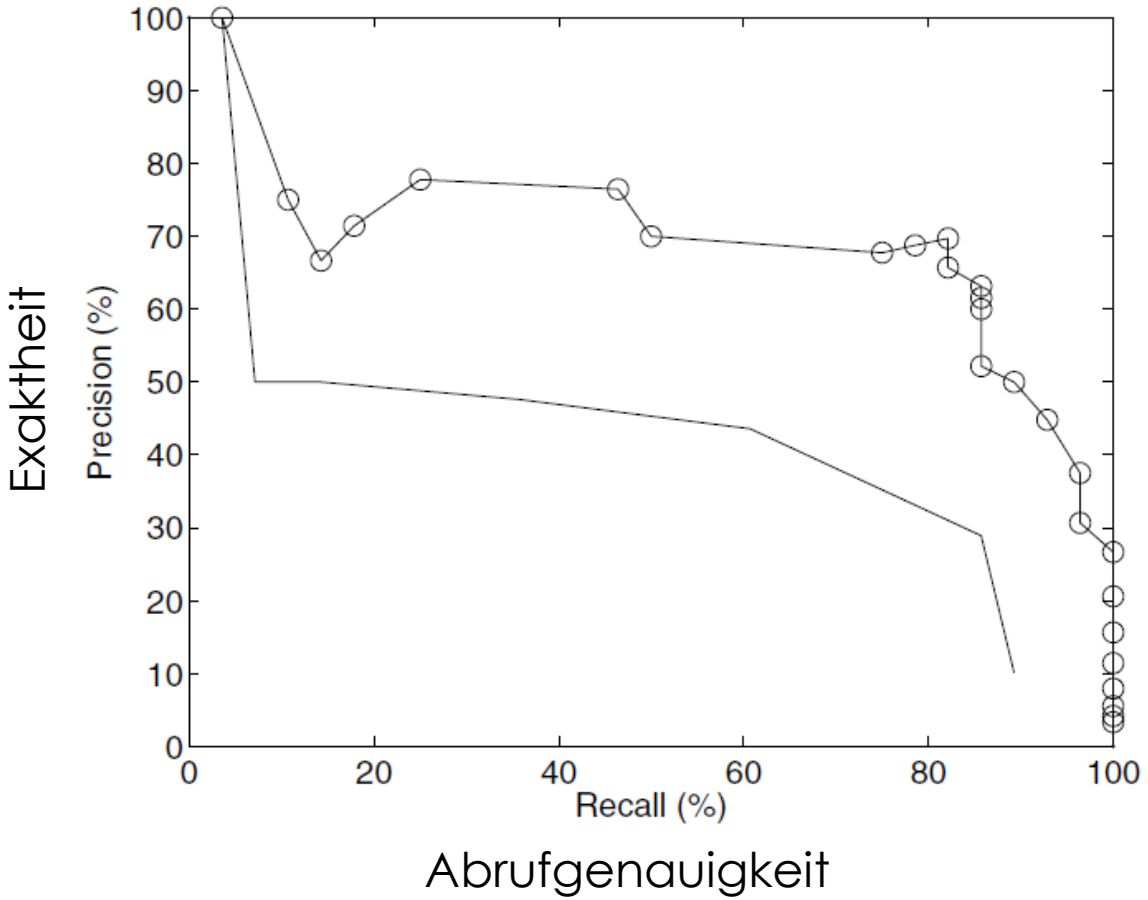
- Gut konditionierte Matrix
- Keine Lücke bei den Singulärwerten
- Näherungsfehler hoch $\frac{\|A-A_k\|_F}{\|A\|_F} \approx 0{,}8 \quad (k = 100)$
- Bessere ODER schlechtere Performance



CLUSTERING

- Dokumentengruppen mit ähnlichem Inhalt
- Jede Gruppe wird durch ihren Durchschnittswert repräsentiert
- Matrix $C_k \in \mathbb{R}^{m \times k}$ als Näherung

CLUSTERING: BEISPIEL



- Clustering:
  - Normierte Spalten (euklidisch)
  - Rang: 50
- Vektormodell: ---

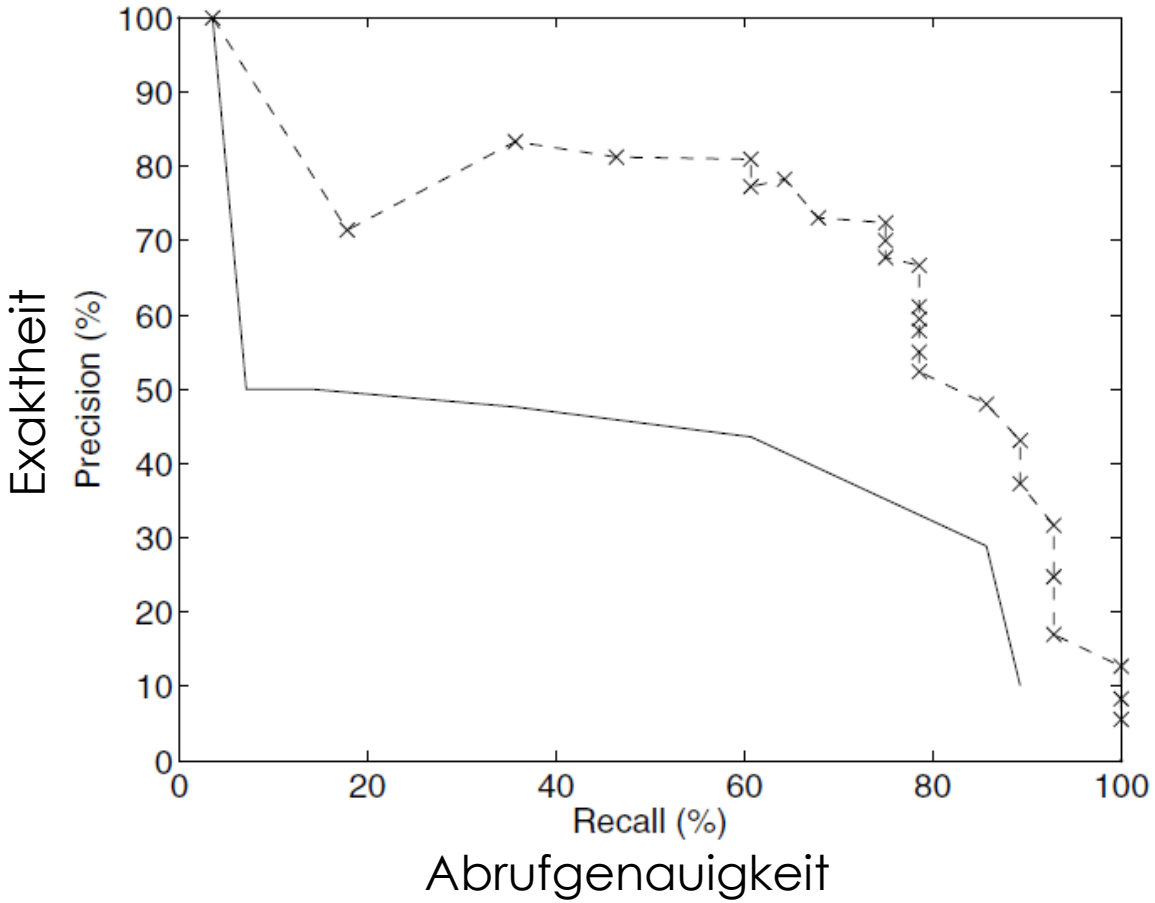
CLUSTERING: ABSCHÄTZUNGEN

- Abschätzungsfehler: $\frac{\|A-P_k G_k\|_F}{\|A\|_F} \approx 0{,}9$
- Aber: Bei unterschiedlichen Anfragen haben LSI ($k = 100$) und Clustering ($k = 50$) ungefähr gleiche Performance

NICHTNEGATIVE MATRIX FAKTORISIERUNG

- $A \approx WH$
- $W = QR$
- $\hat{q} = R^{-1}Q^T q$

NMF: BEISPIEL & ABSCHÄTZUNGEN



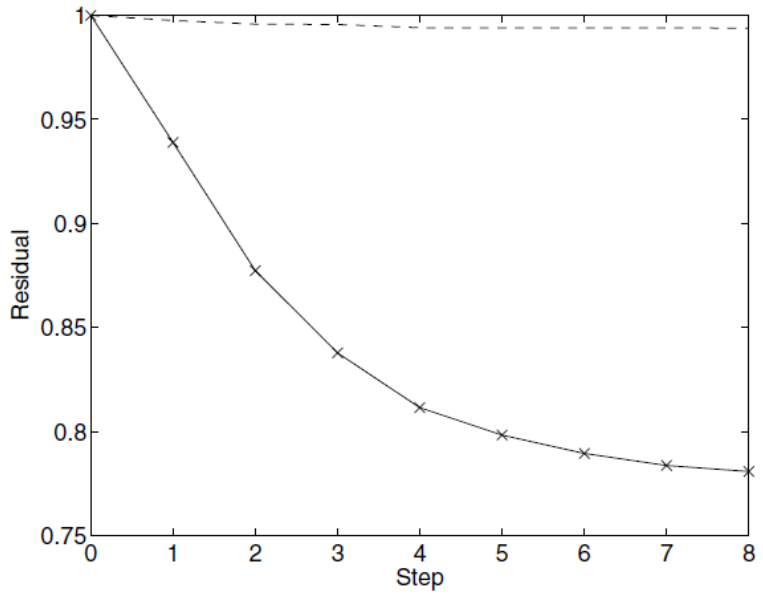
- Nichtnegative Matrix Faktorisierung:
  - $\frac{\|A-WH\|_F}{\|A\|_F} \approx 0{,}89$
- Vektormodell: ---

LGK BIDIAGONALISIERUNG

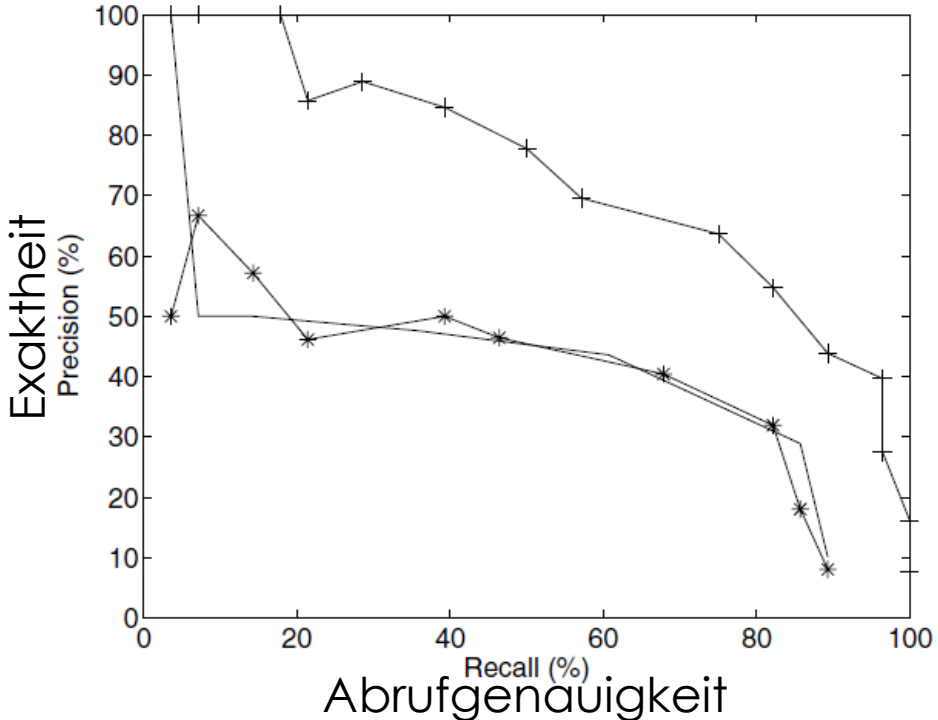
- Updates sind bei den bisher vorgestellten Algorithmen teuer
- Basiert auf der LGK Bidiagonalisierung (Kapitel 7)
- Einzelner Aufruf teurer
- Aber deutlich geringere Updatekosten

LGK BIDIAGONALISIERUNG: BEISPIEL

- Relatives Residuum der LGK Bidiagonalisierung



- Vektormodell: ---
- Bidiagonalisierung (2 Schritte): +
- Bidiagonalisierung (8 Schritte): *



ZUSAMMENFASSUNG

- Perfomanzaussagen: immer unterschiedliche Testdurchläufe mit unterschiedlichen Anfragen
- Ergebnisse abhängig von der Beschaffenheit der Daten
- Neuberechnung notwendig
  - Vektormodell
  - Latent Semantic Indexing
  - Clustering
  - Nichtnegative Matrix Faktorisierung
- In – Place Updates möglich
  - LGK Bidiagonalisierung

VIELEN DANK FÜR IHRE AUFMERKSAMKEIT!

LITERATURVERZEICHNIS

- L. Eldén: **Matrix methods in data mining and pattern recognition.** Volume 4, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2007.

- Martin Porter: **The Porter Stemming Algorithm**
  online abrufbar unter: http://tartarus.org/~martin/PorterStemmer/ (zuletzt überprüft am 5.12.2015)

- Dr. René Witte, Jutta Mülle u.a.: **Text Mining: Wissensgewinnung aus natürlichsprachigen Dokumenten** (2006)
  online abrufbar unter: http://digbib.ubka.uni-karlsruhe.de/volltexte/documents/3230 (zuletzt überprüft am 5.12.2015)

- Dr. Steffen Weißer: Praktische Mathematik: Vorlesungsmitschrift SS 2015