Multimedia Retrieval

4 Transforms for Feature Extraction

4.4 Singular Value Decomposition

Prof. Dr. Marcin Grzegorzek

Research Group for Pattern Recognition
www.pr.informatik.uni-siegen.de

Institute for Vision and Graphics
University of Siegen, Germany

Table of Contents

1 Introduction

1.1 Fundamental Concept

1.2 Search in a MMDBS

1.3 Applications of MMDBMS

2 Fundamentals of Information Retrieval

2.1 Introduction

2.2 Information Retrieval Models

2.3 Relevance Feedback

2.4 Evaluation of Retrieval Systems

2.5 User Profiles

Table of Contents

3 Fundamentals of Multimedia Retrieval

3.1 Characteristics of MM Management and Retrieval

3.2 Processing Pipeline of a Multimedia Retrieval Systems

3.3 Data of a Multimedia Retrieval System

3.4 Features

3.5 Applicability of Different Retrieval Models

3.6 Multimedia Similarity Model

4 Transforms for Feature Extraction

4.1 Fourier Transform

4.2 Wavelet Transform

4.3 Principal Component Analysis

▶ 4.4 Singular Value Decomposition

Table of Contents

5 Distance Functions

5.1 Properties and Classification

5.2 Distance Functions for Points

5.3 Distance Functions for Binary Data

5.4 Distance Functions for Sequences

5.5 Distance Functions for Sets

6 Similarity Measures

6.1 Introduction

6.2 Distance versus Similarity

6.3 Range of Similarity Measures

6.4 Concrete Similarity Measures

6.5 Aggregation of Similarity Values

6.6 Conversion of Distances into Similarity Values

6.7 Partial Similarity

Table of Contents

7 Efficient Algorithms and Data Structures

7.1 High-Dimensional Index Structures

7.2 Algorithms for Aggregation of Similarity Values

8 Query Processing

8.1 Introduction

8.2 Concepts of Query Processing

8.3 Database Model

8.4 Languages

9 Summary and Conclusions

LSI - Allgemeines

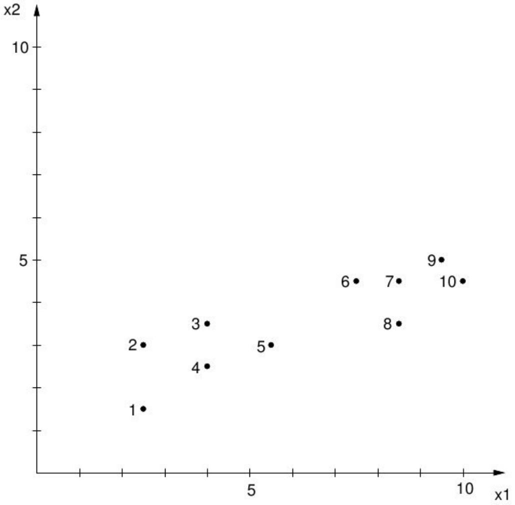
- ähnlich zur KLT: Erkennung und Entfernen linearer Abhängigkeiten durch Lösen von Eigenwertproblem
- allerdings Zerlegung der Feature-Matrix $F = \{f_{ij}\} \in \mathbb{R}^{m \times n}$
- Zerlegung von F in $U * L * V^T$
  - Matrizen U, V enthalten orthonormale Spaltenvektoren
  - Matrix L ist Diagonalmatrix
  - reduzierte, zerlegte Matrizen bedeuten Speichereinsparung
- Zerlegung entspricht Abbildung auf minimale, „schlummernde“ (latente), künstliche Konzepte

LSI - Zerlegung der Feature-Matrix

$r \leq \min(m, n)$ entspricht dem Rang der Matrix F

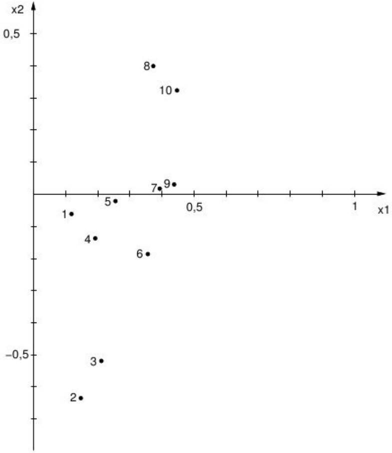
$$\underset{m \times n}{F} = \underset{m \times r}{U} \times \underset{r \times r}{L} \times \underset{r \times n}{V^T}$$

LSI - Beispiel



LSI - Beispiel

Vektoren in Matrix V^T:

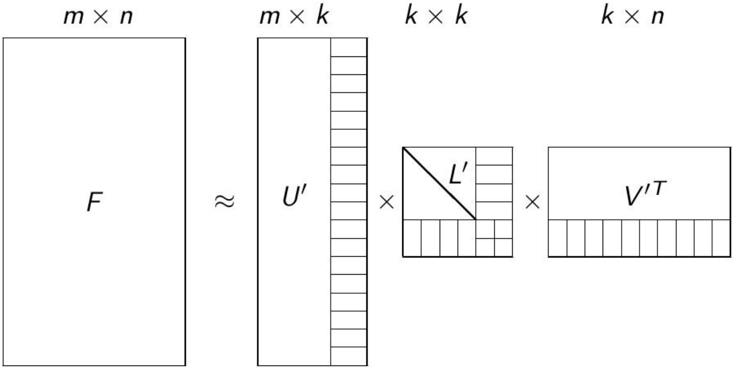


LSI - Analyse der Matrix **L**

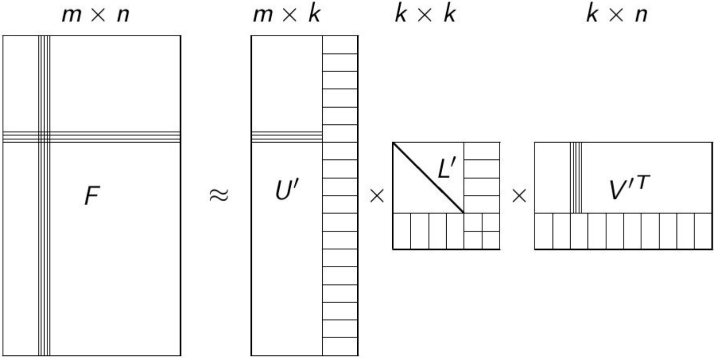
Diagonalwerte der Matrix L

- geben Relevanz der einzelnen Konzepte an
  $\longrightarrow$ niedrige Werte entsprechen geringer Relevanz und umgekehrt
- absteigende Sortierung der Diagonalelemente durch geschicktes Tauschen der Spalten/Zeilen
- Dimensionreduzierung: Entfernen der Konzepte mit den kleinsten Diagonalwerten
  $\longrightarrow$ minimierter Approximationsfehler
- reduzierte Matrizen bedeuten häufig reduzierten Speicheraufwand

LSI - Dimensionsreduzierung graphisch



LSI - Korrespondenz von Spalten und Zeilen



LSI - Ähnlichkeitsvergleiche

Ähnlichkeitsberechnung auf der Basis der 3 Matrizen

- Vergleich von Feature-Vektoren:
  - Skalarprodukt auf Matrizen V'^T und L berechenbar
  - daher Kosinusmaß und euklidsche Distanz leicht berechenbar
- Vergleich der Dimensionen
  - Skalarprodukt auf Matrizen U' und L berechenbar
    $\longrightarrow$ z.B. zur Synonymerkennung in Texten
  - Skalarprodukt ähnlich der Kovarianz zweier Dimensionen

LSI - Dynamische Feature-Matrix

- ständige Neuberechnung der Zerlegung ist zu aufwändig
- Lösungsansatz: Zerlegung einer repräsentativen, statischen Untermenge der Feature-Vektoren
- neue Feature-Vektoren werden dann mit U' und L' multipliziert

LSI - Bewertung

- Bewertung ähnlich zur KLT
- Hauptunterschiede:
  - Zerlegung der Feature-Matrix an Stelle der Kovarianzmatrix
  - Speichern und Manipulieren der zerlegten Matrizen an Stelle Rücktransformation nach Reduktion

LSI - Berechnung

- Ausgangsbasis ist $m \times n$-Feature-Matrix $F = \{f_{ij}\} \in \mathbb{R}^{m \times n}$ für n Feature-Vektoren
- Spalte f_{*j} entspricht Feature-Vektor mit m Werten

LSI - Beispiel

- Beispiel 2×10-Feature-Matrix:

$$F = \begin{pmatrix} 2{,}5 & 2{,}5 & 4 & 4 & 5{,}5 & 7{,}5 & 8{,}5 & 8{,}5 & 9{,}5 & 10 \\ 1{,}5 & 3 & 3{,}5 & 2{,}5 & 3 & 4{,}5 & 4{,}5 & 3{,}5 & 5 & 4{,}5 \end{pmatrix}$$

- Erzeugung einer dritter Dimensionen durch Summierung der ersten beiden plus 0,5:

$$F = \begin{pmatrix} 2{,}5 & 2{,}5 & 4 & 4 & 5{,}5 & 7{,}5 & 8{,}5 & 8{,}5 & 9{,}5 & 10 \\ 1{,}5 & 3 & 3{,}5 & 2{,}5 & 3 & 4{,}5 & 4{,}5 & 3{,}5 & 5 & 4{,}5 \\ 4{,}5 & 6 & 8 & 7 & 9 & 12{,}5 & 13{,}5 & 12{,}5 & 15 & 15 \end{pmatrix}$$

SVD - Allgemeines

$F = U * L * V^T$

- U ist spaltenorthonormale $m \times r$-Matrix
- L ist $r \times r$-Diagonalmatrix
- V^T ist zeilenorthonormale $r \times n$-Matrix
- r ist Rang der Matrix F

SVD - Allgemeines

Berechnung durch Ausnutzung folgender Gesetze

$$\begin{aligned} F * F^T &= U * L * V^T * (U * L * V^T)^T \\ &= U * L * V^T * V * L * U^T \\ &= U * L^2 * U^T \end{aligned}$$

$$\begin{aligned} F^T * F &= (U * L * V^T)^T * U * L * V^T \\ &= V * L * U^T * U * L * V^T \\ &= V * L^2 * V^T \end{aligned}$$

SVD - Beispiel

$$U = \begin{pmatrix} 0{,}5084 & 0{,}6794 & -0{,}5291 \\ 0{,}2732 & -0{,}7099 & -0{,}6491 \\ 0{,}8166 & -0{,}1855 & 0{,}5465 \end{pmatrix}$$

$$L = \begin{pmatrix} 42{,}3264 & 0 & 0 \\ 0 & 2{,}4346 & 0 \\ 0 & 0 & 0{,}2295 \end{pmatrix}$$

$$V = \begin{pmatrix} 0{,}1265 & -0{,}0825 & 0{,}7104 \\ 0{,}1652 & -0{,}6341 & 0{,}0397 \\ 0{,}2250 & -0{,}5136 & -0{,}0697 \\ 0{,}1992 & -0{,}1459 & 0{,}3774 \\ 0{,}2591 & -0{,}0254 & 0{,}2680 \\ 0{,}3603 & -0{,}1712 & -0{,}2505 \\ 0{,}3916 & 0{,}0317 & -0{,}1744 \\ 0{,}3659 & 0{,}3995 & 0{,}2728 \\ 0{,}4358 & 0{,}0507 & -0{,}3218 \\ 0{,}4386 & 0{,}3361 & -0{,}0602 \end{pmatrix}$$

SVD - Reduzierung der Matrizen

- Zeilen-/Spaltentausch damit Diagonalwerte von L absteigen
- Reduzieren heißt Streichen entspr. U, V-Spalten
  $\longrightarrow U', L'$ und V'
- Approximationsfehler ist abhängig von entfernten Diagonalwerten
- Matrizenprodukt in dyadischer Schreibweise:

$$F = l_1(u_1 * v_1^T) + l_2(u_2 * v_2^T) + \ldots + l_r(u_r * v_r^T)$$

SVD - Reduzierung der Matrizen

Beispiel

$$L = \begin{pmatrix} 42{,}3264 & 0 & 0 \\ 0 & 2{,}4346 & 0 \\ 0 & 0 & 0{,}2295 \end{pmatrix}$$

- Wert 0,2295 im Vergleich zu anderen Werten verschwindend klein
- dritte Dimension wurde künstlich erzeugt
- Reduzierung der dritten Dimension

SVD - Transformation neuer Feature-Vektoren

- Annahme: Zerlegung erfolgte auf repräsentativer Feature-Matrix
- Ziel: Erzeugung des entsprechenden V'^T-Spaltenvektoren
- es gilt:

$$
\begin{aligned}
F &\approx U' * L' * V'^T \\
L'^{-1} * U'^{-1} * F &\approx V'^T \\
L'^{-1} * U'^T * F &\approx V'^T
\end{aligned}
$$

SVD - Transformation neuer Feature-Vektoren

- sei f_{*j} zu transformierender Feature-Vektor
- $v'_{*j} \in V'^T$ erzeugt durch Multiplikation mit Matrizen U'^T und L'^{-1}

$$v'_{*j} = L'^{-1} * U'^T * f_{*j}$$

SVD - Transformation neuer Feature-Vektoren

Beispiel

Transformation von

$$f = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

zu

$$v' = L'^{-1} * U'^{T} * \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 0{,}0828 \\ -0{,}5326 \end{pmatrix}$$

SVD - Berechnung auf transformierten Vektoren

Ausnutzung von $F^T * F = V * L^2 * V^T$ zur Berechnung Skalarprodukt

- Kosinusmaß:

$$
\begin{aligned}
sim_{cos}(f_{*1}, f_{*2}) &= \frac{\langle f_{*1}, f_{*2} \rangle}{\sqrt{\langle f_{*1}, f_{*1} \rangle} * \sqrt{\langle f_{*2}, f_{*2} \rangle}} \\
&= \frac{f_{*1}^T * f_{*2}}{\sqrt{f_{*1}^T * f_{*1}} * \sqrt{f_{*2}^T * f_{*2}}} \\
&\approx \frac{v'_{*1} * L' * L' * v'^T_{*2}}{\sqrt{v'_{*1} * L' * L' * v'^T_{*1}} * \sqrt{v'_{*2} * L' * L' * v'^T_{*2}}}
\end{aligned}
$$

SVD - Berechnung auf transformierten Vektoren

Ausnutzung von $F^T * F = V * L^2 * V^T$ zur Berechnung Skalarprodukt

- euklidsche Distanz:

$$\begin{aligned} dissim_{L_2}(f_{*1}, f_{*2}) &= \sqrt{(f_{*1} - f_{*2})^T * (f_{*1} - f_{*2})} \\ &\approx \sqrt{(v'_{*1} - v'_{*2}) * L' * L' * (v'_{*1} - v'_{*2})^T} \end{aligned}$$