

5.4 Latent Semantic Indexing und Singulärwertzerlegung

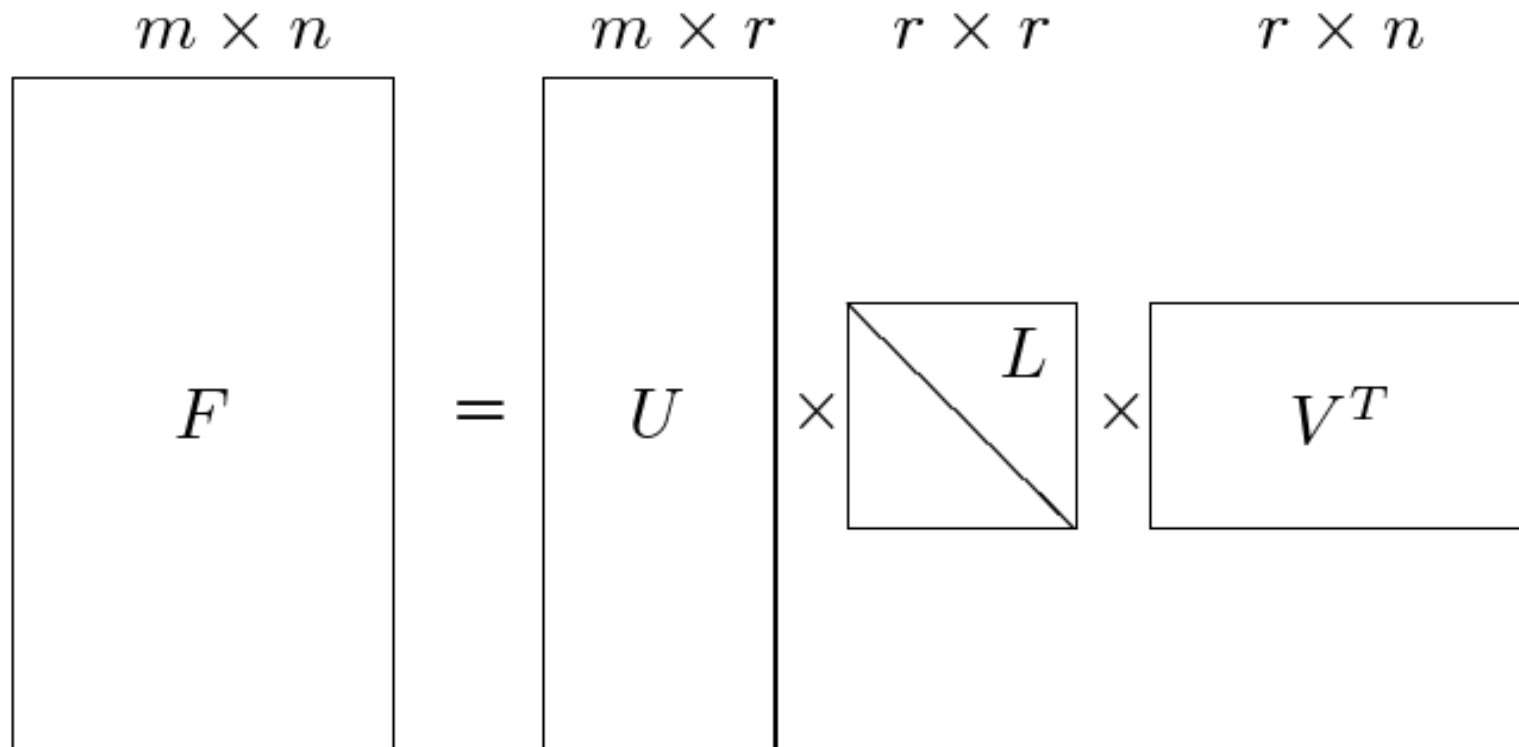
Zerlegung von F in $U * L * V^T$

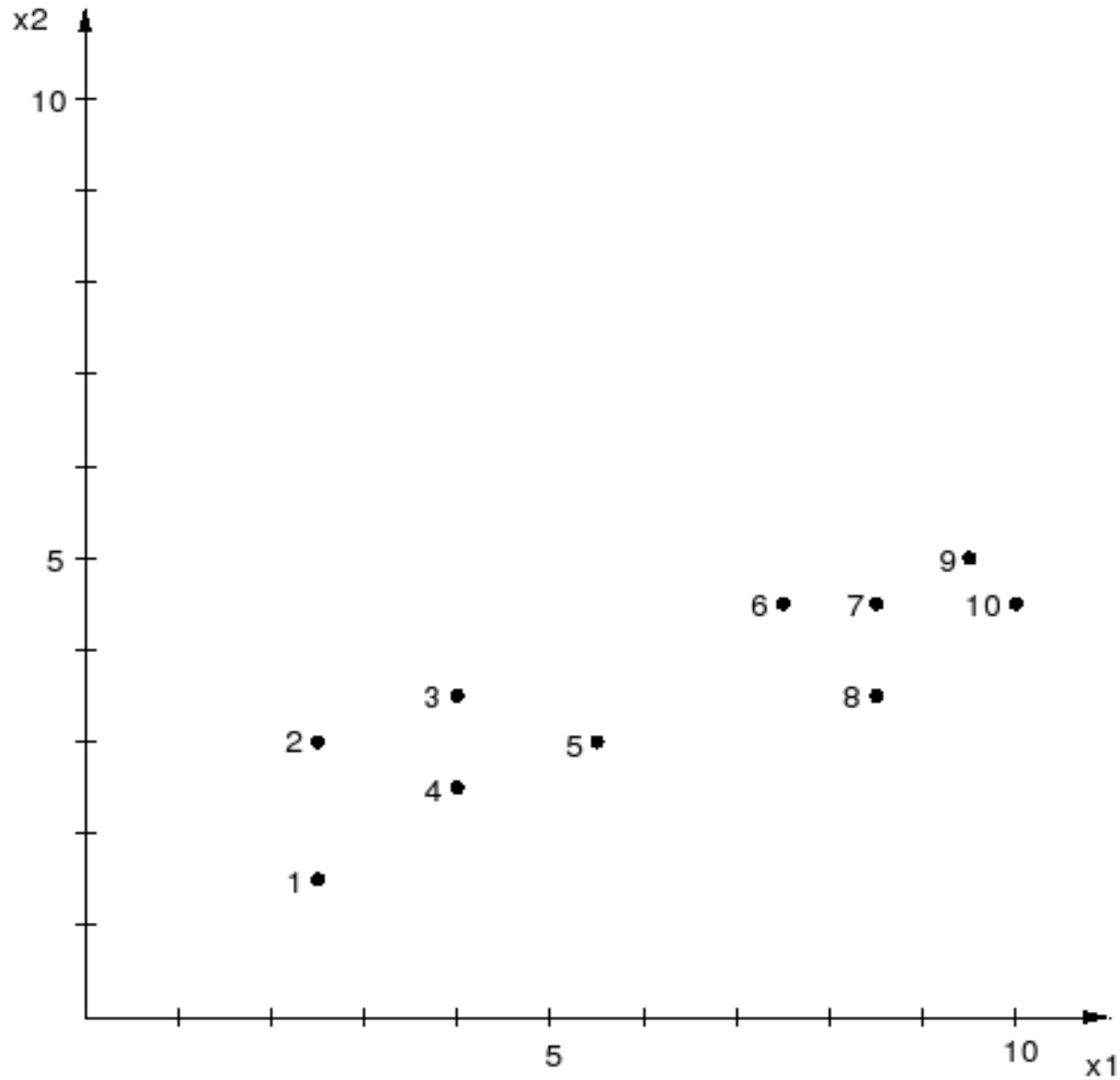
- ◆ Matrizen U, V enthalten orthonormale Spaltenvektoren
- ◆ Matrix L ist Diagonalmatrix
- ◆ reduzierte, zerlegte Matrizen bedeuten Speichereinsparung

Zerlegung entspricht Abbildung auf minimale, „schlummernde“ (latente), künstliche Konzepte

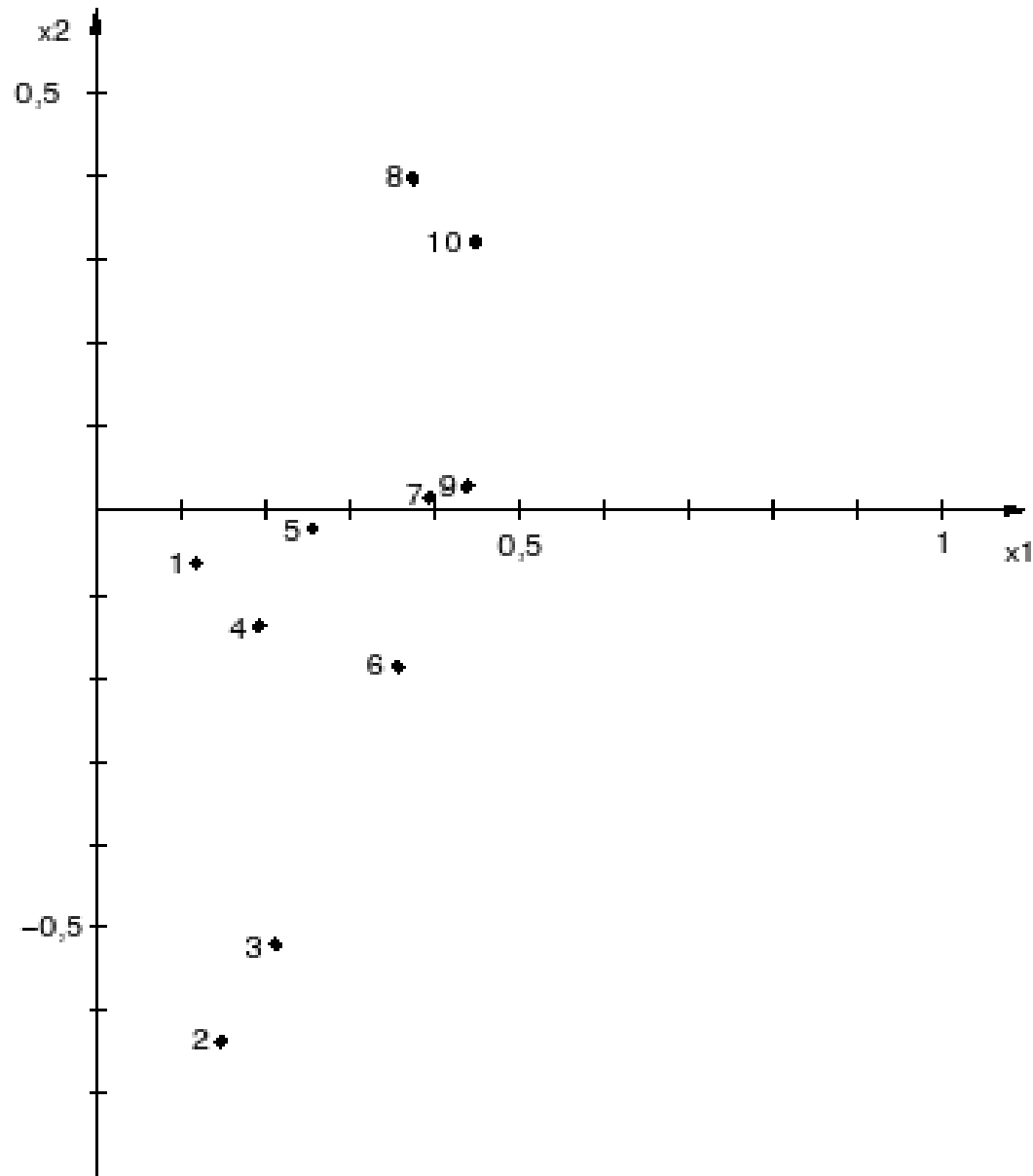
entspricht dem Rang der Matrix F

$$r \leq \min(m, n)$$





Vektoren
in Matrix V^T :



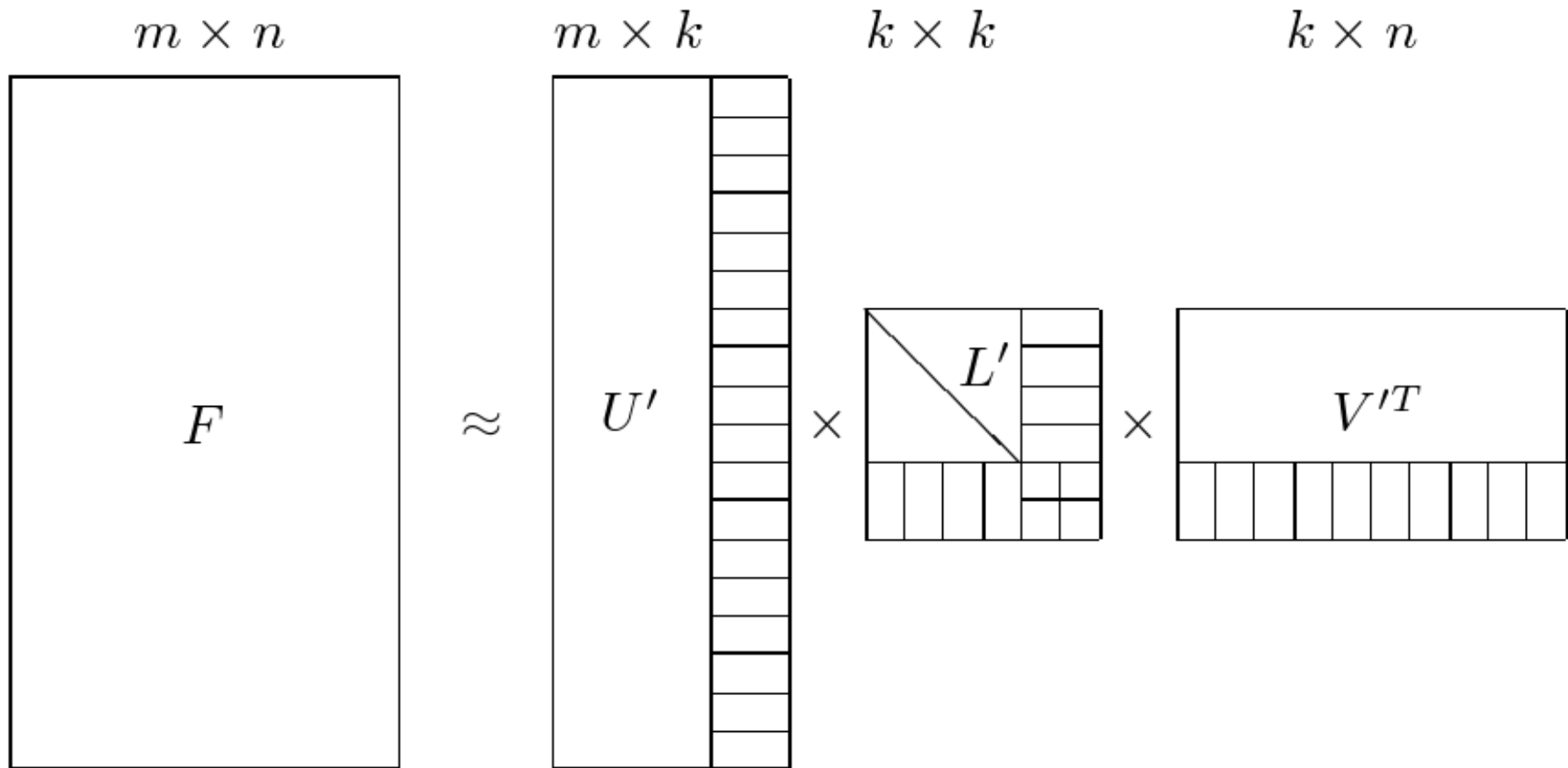
Diagonalwerte der Matrix L

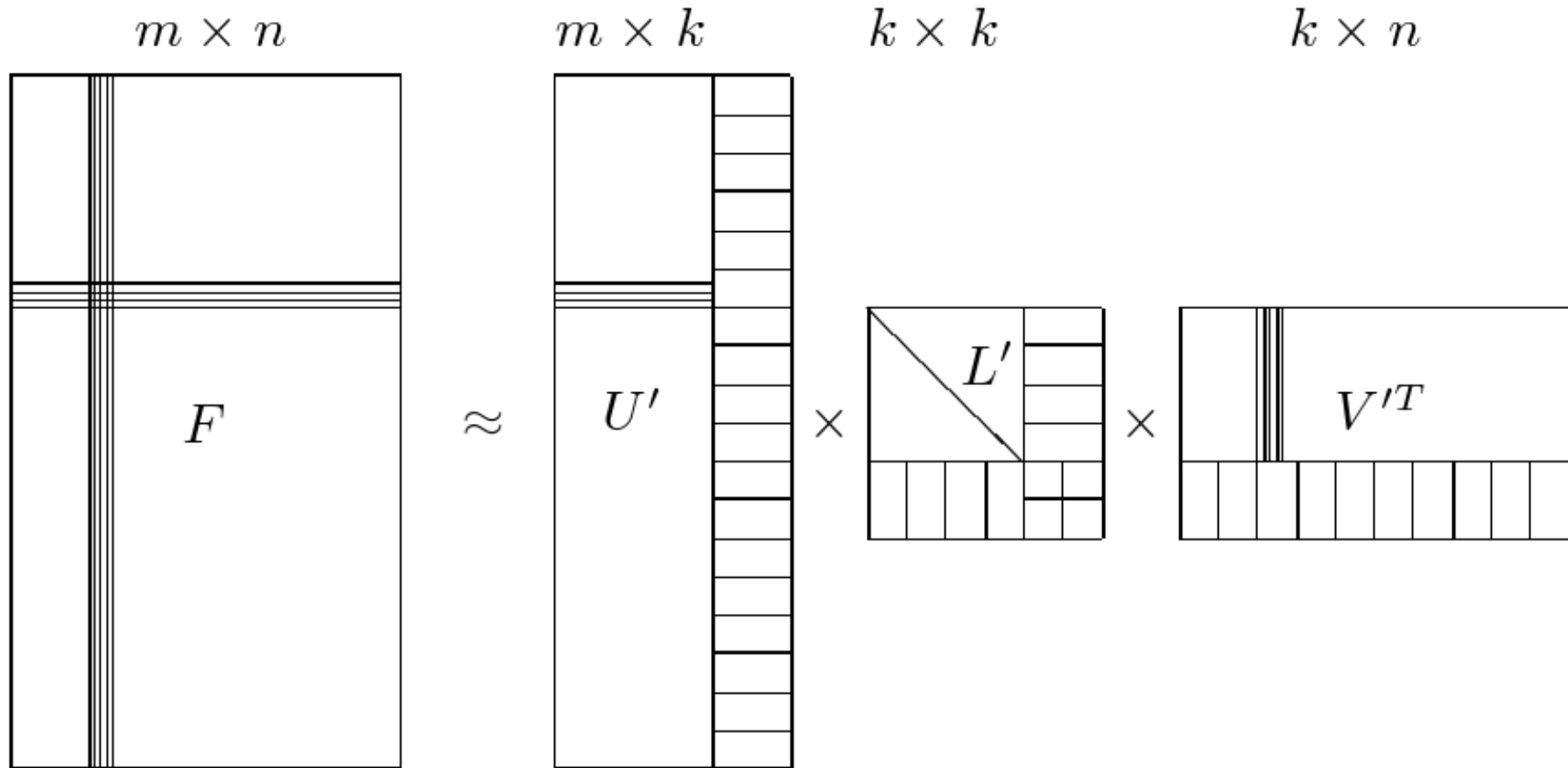
geben Relevanz der einzelnen Konzepte an

→ niedrige Werte entsprechen geringer Relevanz und
umgekehrt

absteigende Sortierung der Diagonalelemente durch
geschicktes Tauschen der Spalten/Zeilen

Dimensionsreduzierung: Entfernen der Konzepte mit den kleinsten Diagonalwerten
→ minimierter Approximationsfehler
reduzierte Matrizen bedeuten häufig reduzierten Speicheraufwand





Ähnlichkeitsberechnung auf der Basis der drei Matrizen

Vergleich von Feature-Vektoren:

- ◆ Skalarprodukt auf Matrizen V'^T und L berechenbar
- ◆ daher Kosinusmaß und euklidische Distanz leicht berechenbar

Vergleich der Dimensionen

- ◆ Skalarprodukt auf Matrizen U' und L berechenbar
→ z.B. Synonymerkennung in Texten
- ◆ Skalarprodukt ähnlich der Kovarianz zweier Dimensionen

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

Bewertung ähnlich zur KLT

Hauptunterschiede:

- ◆ Zerlegung der Feature-Matrix an Stelle der Kovarianzmatrix
- ◆ Speichern und Manipulieren der zerlegten Matrizen an Stelle Rücktransformation nach Reduktion

Ausgangsbasis ist $m \times n$ -Feature-Matrix
für n Feature-Vektoren

Spalte f_{*j} entspricht Feature-Vektoren mit
 m Werten

$$F = \{f_{ij}\} \in \mathbb{R}^{m \times n}$$

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 dritten Dimension durch Summierung der ersten beiden plus 0,5:

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

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

$$F = U * L * V^T$$

Berechnung durch Ausnutzung folgender Gesetze:

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

$$U = \begin{pmatrix} 0,5084 & 0,6794 & -0,5291 \\ 0,2732 & -0,7099 & -0,6491 \\ 0,8166 & -0,1855 & 0,5465 \end{pmatrix} \quad L = \begin{pmatrix} 42,3264 & 0 & 0 \\ 0 & 2,4346 & 0 \\ 0 & 0 & 0,2295 \end{pmatrix}$$
$$V = \begin{pmatrix} 0,1265 & -0,0825 & 0,7104 \\ 0,1652 & -0,6341 & 0,0397 \\ 0,2250 & -0,5136 & -0,0697 \\ 0,1992 & -0,1459 & 0,3774 \\ 0,2591 & -0,0254 & 0,2680 \\ 0,3603 & -0,1712 & -0,2505 \\ 0,3916 & 0,0317 & -0,1744 \\ 0,3659 & 0,3995 & 0,2728 \\ 0,4358 & 0,0507 & -0,3218 \\ 0,4386 & 0,3361 & -0,0602 \end{pmatrix}$$

Zeilen-/Spaltentausch damit Diagonalwerte von L absteigen

Reduzieren heißt Streichen entspr.

U, V -Spalten

→ U', L' und V'

Approximationsfehler ist abhängig von entfernten Diagonalwerten
siehe Matrizenproduktion in dyadischer Schreibweise:

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

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

Wert 0,2295 im Vergleich zu anderen Werten verschwindend klein

dritte Dimension wurde künstlich erzeugt

Reduzierung der dritten Dimension

Annahme: Zerlegung erfolgte auf repräsentativer Feature-Matrix

Ziel: Erzeugung der entsprechenden V'^T -Spaltenvektoren

es gilt:

$$F \approx U' * L' * V'^T$$

$$L'^{-1} * U'^{-1} * F \approx V'^T$$

$$L'^{-1} * U'^T * F \approx V'^T$$

sei f_{*j} zu transformierender Feature-Vektor

$v'_{*j} \in V'^T$ erzeugt durch Multiplikation mit Matrizen U'^T
und L'^{-1}

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

Transformation von

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

zu

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

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

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

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

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