Performanz und Probleme von Sparse Embeddings

Katja Markert (einige Folien von Michael Staniek)

Institut für Computerlinguistik
Uni Heidelberg
markert@cl.uni-heidelberg.de

May 14, 2019

Bisher und heute

1. Bisher: Assoziationsmaße und Sparse Embeddings

	species	*computer*	*animal*
cat	59	5	304
carnivore	21	1	21
feline	2	0	5
airport	4	12	2

2. Bisher: Distanzen und Ähnlichkeitsmaße zur Wortähnlichkeitsbestimmung

Paar	cos_{sim}
cat, carnivore	0.828
cat, feline	0.98
cat, airport	0.227

3. Jetzt (Wiederholung aus ECL) : Umwandlung von frequenzbasierten Kokkurrenzmatrizen in PPMI-Matrizen
4. Jetzt: Performanz und Probleme von sparse embeddings
5. Nächste Folien: Vorbereitung von Singular Value Decomposition mit Hintergrund Unterräumen und Matrizen

Übersicht

1. Umwandlung der Frequenzmatrizen in PPMI-Matrizen
2. Performanz von sparse embeddings
3. Probleme bei Sparse Embeddings und die Idee der Singular Value Decomposition

Übersicht

PPMI Beispiel I

Term-Term-Matrix mit Frequenzen (aus Jurafsky und Martin, Edition 3)

	computer	data	pinch	result	sugar	
apricot	0	0	1	0	1	2
pineapple	0	0	1	0	1	2
digital	2	1	0	1	0	4
information	1	6	0	4	0	11
	3	7	2	5	2	19

- Die Randhäufigkeiten entsprechen nicht den Unigramfrequenzen der Wörter (Warum nicht?)
- Im Unterschied zur Kollokationsberechnung für Bigramme, enspricht die Gesamthäufigkeit N der Beobachtungen (hier 19) im Normalfall nicht der Korpusgröße (Warum nicht?)

PPMI Beispiel II

Term-Term-Matrix mit Frequenzen:

	computer	data	pinch	result	sugar	
apricot	0	0	1	0	1	2
pineapple	0	0	1	0	1	2
digital	2	1	0	1	0	4
information	1	6	0	4	0	11
	3	7	2	5	2	19

$$ppmi(information, data) = max(\log_2 \frac{\frac{6}{19}}{\frac{11}{19} \cdot \frac{7}{19}}, 0) = 0.57$$

$$ppmi(information, computer) = max(\log_2 \frac{\frac{1}{19}}{\frac{11}{19} \cdot \frac{3}{19}}, 0) = max(\log_2 \frac{19}{33}, 0) = 0$$

$$ppmi(apricot, computer) = max(\log_2 \frac{\frac{0}{19}}{\frac{2}{19} \cdot \frac{3}{19}}, 0) = max(\log_2 0, 0) = 0$$

$$ppmi(apricot, pinch) = max(\log_2 \frac{\frac{1}{19}}{\frac{2}{19} \cdot \frac{2}{19}}, 0) = 2.25$$

PPMI Beispiel II

Term-Term-Matrix mit Frequenzen:

	computer	data	pinch	result	sugar	
apricot	0	0	1	0	1	2
pineapple	0	0	1	0	1	2
digital	2	1	0	1	0	4
information	1	6	0	4	0	11
	3	7	2	5	2	19

$$ppmi(information, data) = max(\log_2 \frac{\frac{6}{19}}{\frac{11}{19} \cdot \frac{7}{19}}, 0) = 0.57$$

$$ppmi(information, computer) = max(\log_2 \frac{\frac{1}{19}}{\frac{11}{19} \cdot \frac{3}{19}}, 0) = max(\log_2 \frac{19}{33}, 0) = 0$$

$$ppmi(apricot, computer) = max(\log_2 \frac{\frac{0}{19}}{\frac{2}{19} \cdot \frac{3}{19}}, 0) = max(\log_2 0, 0) = 0$$

$$ppmi(apricot, pinch) = max(\log_2 \frac{\frac{1}{19}}{\frac{2}{19} \cdot \frac{2}{19}}, 0) = 2.25$$

PPMI Beispiel II

Term-Term-Matrix mit Frequenzen:

	computer	data	pinch	result	sugar	
apricot	0	0	1	0	1	2
pineapple	0	0	1	0	1	2
digital	2	1	0	1	0	4
information	1	6	0	4	0	11
	3	7	2	5	2	19

$$ppmi(information, data) = max(\log_2 \frac{\frac{6}{19}}{\frac{11}{19} \cdot \frac{7}{19}}, 0) = 0.57$$

$$ppmi(information, computer) = max(\log_2 \frac{\frac{1}{19}}{\frac{11}{19} \cdot \frac{3}{19}}, 0) = max(\log_2 \frac{19}{33}, 0) = 0$$

$$ppmi(apricot, computer) = max(\log_2 \frac{\frac{0}{19}}{\frac{2}{19} \cdot \frac{3}{19}}, 0) = max(\log_2 0, 0) = 0$$

$$ppmi(apricot, pinch) = max(\log_2 \frac{\frac{1}{19}}{\frac{2}{19} \cdot \frac{2}{19}}, 0) = 2.25$$

PPMI Beispiel II

Term-Term-Matrix mit Frequenzen:

	computer	data	pinch	result	sugar	
apricot	0	0	1	0	1	2
pineapple	0	0	1	0	1	2
digital	2	1	0	1	0	4
information	1	6	0	4	0	11
	3	7	2	5	2	19

$$ppmi(information, data) = max(\log_2 \frac{\frac{6}{19}}{\frac{11}{19} \cdot \frac{7}{19}}, 0) = 0.57$$

$$ppmi(information, computer) = max(\log_2 \frac{\frac{1}{19}}{\frac{11}{19} \cdot \frac{3}{19}}, 0) = max(\log_2 \frac{19}{33}, 0) = 0$$

$$ppmi(apricot, computer) = max(\log_2 \frac{\frac{0}{19}}{\frac{2}{19} \cdot \frac{3}{19}}, 0) = max(\log_2 0, 0) = 0$$

$$ppmi(apricot, pinch) = max(\log_2 \frac{\frac{1}{19}}{\frac{2}{19} \cdot \frac{2}{19}}, 0) = 2.25$$

PPMI Beispiel II

Term-Term-Matrix mit Frequenzen:

	computer	data	pinch	result	sugar	
apricot	0	0	1	0	1	2
pineapple	0	0	1	0	1	2
digital	2	1	0	1	0	4
information	1	6	0	4	0	11
	3	7	2	5	2	19

$$ppmi(information, data) = max(\log_2 \frac{\frac{6}{19}}{\frac{11}{19} \cdot \frac{7}{19}}, 0) = 0.57$$

$$ppmi(information, computer) = max(\log_2 \frac{\frac{1}{19}}{\frac{11}{19} \cdot \frac{3}{19}}, 0) = max(\log_2 \frac{19}{33}, 0) = 0$$

$$ppmi(apricot, computer) = max(\log_2 \frac{\frac{0}{19}}{\frac{2}{19} \cdot \frac{3}{19}}, 0) = max(\log_2 0, 0) = 0$$

$$ppmi(apricot, pinch) = max(\log_2 \frac{\frac{1}{19}}{\frac{2}{19} \cdot \frac{2}{19}}, 0) = 2.25$$

PPMI-Beispiel III

Term-Term-Matrix mit PPMI

	computer	data	pinch	result	sugar
apricot	0	0	2.25	0	2.25
pineapple	0	0	2.25	0	2.25
digital	1.66	0	0	0	0
information	0	0.57	0	0.47	0

Ein Problem: PPMI überschätzt seltene Kontextwörter (siehe *pinch*).
Wie löst man das? (Smoothing siehe Jurafsky und Martin, Kapitel 6)

PPMI-Beispiel IV: Laplace Smoothing

Originalmatrix:

	computer	data	pinch	result	sugar
apricot	0	0	1	0	1
pineapple	0	0	1	0	1
digital	2	1	0	1	0
information	1	6	0	4	0

Nach Add-2 Smoothing:

	computer	data	pinch	result	sugar
apricot	2	2	3	2	3
pineapple	2	2	3	2	3
digital	4	3	2	3	2
information	3	8	2	6	2

Übung

Konvertiere unsere Standardmatrix in PPMI

	species	*computer*	*animal*
cat	59	5	304
carnivore	21	1	21
feline	2	0	5
airport	4	12	2

Lösung

	species	*computer*	*animal*	
cat	59	5	304	368
carnivore	21	1	21	43
feline	2	0	5	7
airport	4	12	2	18
	86	18	332	436

	species	*computer*	*animal*
cat	0	0	0.111
carnivore	1.3	0	0
feline	0.52	0	0
airport	0.16	4	0

Warum hat dies nicht gut funktioniert?

Übersicht

Beispielperformanz

Agirre et al (NAACL 2009): A study on similarity and relatedness using distributional and wordnet-based approaches

Für WordSim353. Stopwörter gefiltered. Korpus 4 Milliarden Webdokumente (1.6 Terawords). Assoziationsmaß χ^2.

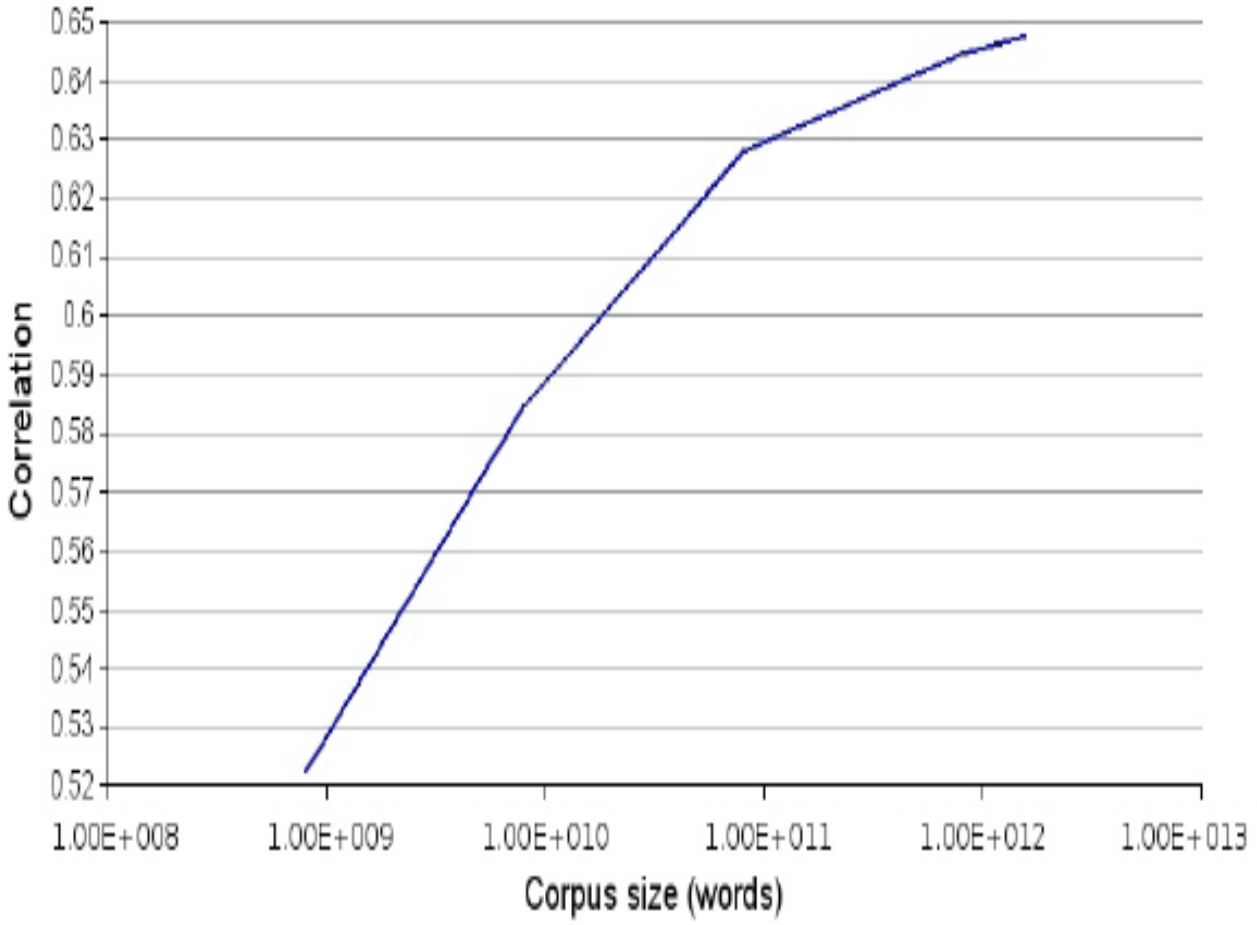
Fenstergröße	Spearman Rank
1	0.64
4	0.65
6	0.64

1. Warum ändert die Fenstergröße so wenig?
2. Warum sind Ihre Ergebnisse im Übungsblatt so viel schlechter?

Lernkurve (Learning curve)

Agirre et al (NAACL 2009): A study on similarity and relatedness using distributional and wordnet-based approaches

Abhängigkeit der Performanz von Korpusgröße



Übersicht

Probleme bei sparse embeddings

- Overfitting durch zu viele Tokens mit geringen Frequenzen: Alle Assoziationsmaße haben Probleme mit seltenen Wörtern
- Zu viele fälschlich unterschiedliche Dimensionen:
  - *Nasa* kommt mit *cosmonaut* vor
  - *Roscosmos* kommt mit *astronaut* vor
  - *cosmonaut* und *astronaut* sind unterschiedliche Dimensionen. Damit lässt sich die “Ahnlichkeit zwischen *NASA* und *roscosmos* schwer fassen.

Beispiel aus erster Vorlesung:

	astronaut	*cosmonaut*	*tomato*
NASA	4	0	1
Roscosmos	0	4	0
avocado	0	0	7
salad	0	1	10

Ziel

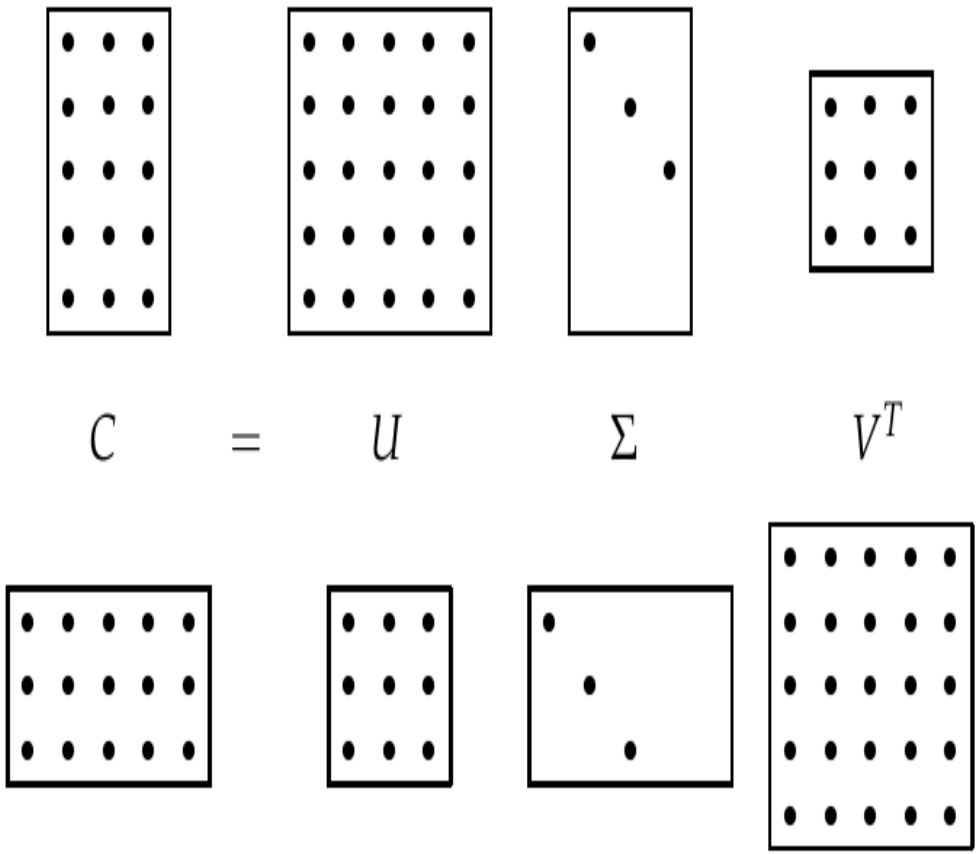
- Approximiere den n-dimensionalen Raum mit weniger Dimensionen
- Indem wir Achsen rotieren, so dass wir einen Raum erhalten, in dem die erste Dimension die meiste Varianz in den Originaldaten erklärt

Motivation für SVD bzw SVD ohne Details

- Gegeben: Matrix M der Dimension $m \times n$
- Gesucht Matrix $\tilde{M}$ mit der Dimension $m \times n$, die **ähnlich** zu M ist, aber niedrigeren **Rang** hat
- Methode: **Singular Value Decomposition (SVD)**: Zerlege M in drei Matrizen: $M_{m \times n} = U_{m \times m} \Sigma_{m \times n} V^T_{n \times n}$, die besonders schöne Eigenschaften haben
- Methode: Aus dieser Zerlegung können nun geschickt unwichtige Dimensionen “herausgenommen” werden, um dichte Matrizen niedrigeren Ranges zu erhalten, mit denen bessere Ähnlichkeitsberechnungen möglich sind.

SVD Illustration

Aus Manning et al, Figure 18.1



Da bei $m > n$ die letzten Zeilen von Σ Nullzeilen sind, fallen die letzten Spalten von U nicht ins Gewicht und man kann diese ignorieren und z. B. U auf $m \times \min(m, n)$ beschränken.

SVD Beispiel

Eine Wort-Dokument-Matrix M aus dem $\mathbb{R}^{5\times 3}$

	$d1$	$d2$	$d3$
ship	1	1	0
boat	0	0	1
ocean	1	0	1
motor	1	0	1
wood	0	1	0

$$cos_{sim}(ship, boat) = 0$$

$$cos_{sim}(ship, ocean) = \frac{1}{2}$$

$$cos_{sim}(boat, ocean) = \frac{1}{\sqrt{2}} = 0.7$$

SVD Beispiel

Die Matrix U ist eine 5×3 Matrix:

ship	-0.41	0.7	0.24
boat	-0.29	-0.33	-0.72
ocean	-0.61	-0.2	0.14
motor	-0.61	-0.2	0.14
wood	-0.1	0.57	-0.62

- Eine Reihe pro Wort
- Matrix ist **orthonormal**, d.h. die Spaltenvektoren haben alle die Länge 1 und stehen alle aufeinander senkrecht (Skalarprodukt zweier Spaltenvektoren =0)
- Spalten sind nach Höhe der Varianz geordnet

SVD Beispiel

Die Matrix Σ ist eine Diagonalmatrix der Dimension 3×3

$$\begin{matrix} 2.27 & 0 & 0 \\ 0 & 1.49 & 0 \\ 0 & 0 & 0.78 \end{matrix}$$

Σ enthält die Wurzeln der **Eigenwerte** von MM^T in absteigender Reihenfolge. Je größer, desto wichtiger ist eine Dimension.

Die Matrix V^T

V^T ist eine 3×3-Matrix mit einer Spalte pro Dokument

$d1$	$d2$	$d3$
-0.72	-0.23	-0.66
0.19	0.85	-0.5
0.67	-0.48	-0.56

- Eine Spalte pro Dokument
- Matrix ist **orthonormal**, d.h. die Spaltenvektoren haben alle die Länge 1 und stehen alle aufeinander senkrecht (Skalarprodukt zweier Spaltenvektoren =0)

Stimmt die Zerlegung?

Berechnen wir $U\Sigma V^T$:

$$\begin{bmatrix} -0.41 & 0.7 & 0.24 \\ -0.29 & -0.33 & -0.72 \\ -0.61 & -0.2 & 0.14 \\ -0.61 & -0.2 & 0.14 \\ -0.1 & 0.57 & -0.62 \end{bmatrix} \begin{bmatrix} 2.27 & 0 & 0 \\ 0 & 1.49 & 0 \\ 0 & 0 & 0.78 \end{bmatrix} \begin{bmatrix} -0.72 & -0.23 & -0.66 \\ 0.19 & 0.85 & -0.5 \\ 0.67 & -0.48 & -0.56 \end{bmatrix}$$

$$= \begin{bmatrix} -0.93 & 1.04 & 0.187 \\ -0.658 & -0.49 & -0.56 \\ -1.38 & -0.298 & 0.1092 \\ -1.38 & -0.298 & -.1092 \\ -0.227 & 0.84 & -0.483 \end{bmatrix} \begin{bmatrix} -0.72 & -0.23 & -0.66 \\ 0.19 & 0.85 & -0.5 \\ 0.67 & -0.48 & -0.56 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Niedrigdimensionale Approximation

Die kleinsten Eigenwerte sind die unwichtigsten. Wir können diese "weglassen" = auf Null setzen $\rightarrow$ eine Matrix mit kleinerem Rang, die aber relativ ähnlich zur Ausgangsmatrix ist.

$$\begin{bmatrix} -0.41 & 0.7 & 0.24 \\ -0.29 & -0.33 & -0.72 \\ -0.61 & -0.2 & 0.14 \\ -0.61 & -0.2 & 0.14 \\ -0.1 & 0.57 & -0.62 \end{bmatrix} \begin{bmatrix} 2.27 & 0 & 0 \\ 0 & 1.49 & 0 \\ 0 & 0 & \mathbf{0} \end{bmatrix} \begin{bmatrix} -0.72 & -0.23 & -0.66 \\ 0.19 & 0.85 & -0.5 \\ 0.67 & -0.48 & -0.56 \end{bmatrix}$$

$$= \begin{bmatrix} -0.93 & 1.04 & 0 \\ -0.658 & -0.49 & 0 \\ -1.38 & -0.298 & 0 \\ -1.38 & -0.298 & 0 \\ -0.227 & 0.84 & 0 \end{bmatrix} \begin{bmatrix} -0.72 & -0.23 & -0.66 \\ 0.19 & 0.85 & -0.5 \\ 0.67 & -0.48 & -0.56 \end{bmatrix}$$

$$= \begin{bmatrix} 0.87 & 1.09 & 0.11 \\ 0.38 & -0.27 & 0.68 \\ 0.93 & 0.05 & 1.06 \\ 0.93 & 0.05 & 1.06 \\ 0.32 & 0.77 & -0.27 \end{bmatrix}$$

Die niedrigdimensionale Approximation

Wir interessieren uns für die Matrix $U_2 = U\Sigma_2$, also die Matrix mit dem niedrigerem Rang:

$$\begin{bmatrix} -0.93 & 1.04 & 0 \\ -0.658 & -0.49 & 0 \\ -1.38 & -0.298 & 0 \\ -1.38 & -0.298 & 0 \\ -0.227 & 0.84 & 0 \end{bmatrix}$$

Man kann diese nun als die Repräsentation unserer 5 Wörter mit zwei versteckten Dimensionen auffassen:

	$h1$	$h2$
ship	-0.93	1.04
boat	-0.658	-0.49
ocean	-1.38	-0.298
motor	-1.38	-0.298
wood	-0.227	0.84

Neue Ähnlichkeitsberechnungen

	$h1$	$h2$
ship	-0.93	1.04
boat	-0.658	-0.49
ocean	-1.38	-0.298
motor	-1.38	-0.298
wood	-0.227	0.84

$$cos_{sim}(ship, boat) = \frac{(-0.93) \cdot (-0.658) + 1.04 \cdot (-0.49)}{\sqrt{(0.93^2 + 1.04^2)} \cdot \sqrt{(0.658^2 + 0.49^2)}} = 0.09$$

$$cos_{sim}(ship, ocean) = \frac{(-0.93) \cdot (-1.38) + 1.04 \cdot (-0.29)}{\sqrt{(0.93^2 + 1.04^2)} \cdot \sqrt{(1.38^2 + 0.29^2)}} = 0.49$$

$$cos_{sim}(boat, ocean) = 0.9$$

Vorsicht: Habe auch schon oft Benutzung von U_2' (einfach die ersten zwei Spalten von U abgeschnitten ohne Sigmamultiplikation) gesehen.

Intuition

- Vektoren in U und V sind nach Variation in den Originaldaten geordnet
- Löschen von Dimensionen, die keine wesentliche Variation beitragen, reduziert “Rauschen”
- Wortvektoren sind nun kürzer und enthalten nur Elemente, die die wichtigsten versteckten Dimensionen aufzeigen
- Im Normalfall: von Tausenden von Dimensionen zu wenigen 100

Was brauchen wir nun hierfür?

Wie berechne ich diese Zerlegung und finde die Matrizen? Und was bedeuten die Fachbegriffe? Und warum funktioniert das?

- Basis, Unterräume
- Orthonormalisierungen
- Matrizenhintergrund
- Eigenvektoren, Eigenwerte sowie Berechnungsmethoden von Eigenvektoren und Eigenwerten
- Ränge
- Ähnlichkeiten von Matrizen (Normen und Distanzen für Matrizen)
- ...

Anwendungen von SVD

- Dichte Embeddings von Wörtern
- Image Compression
- Recommendersysteme
- Information Retrieval

Literatur

- Jurafsky und Martin (Edition 3): Introduction to Natural Language Processing. (Kapitel 6.)
- Agirre et al (NAACL 2009): A study on similarity and relatedness using distributional and wordnet-based approaches
- Manning et al: Introduction to Information Retrieval (Kapitel 18)
- SVD Tutorial (ohne vollständigen Hintergrund) : Kirk Baker (2005): Singular Value Decomposition Tutorial `https://datajobs.com/data-science-repo/SVD-Tutorial-[Kirk-Baker].pdf`