Dimensionality Reduction Models for Language Data

Overview and Applications

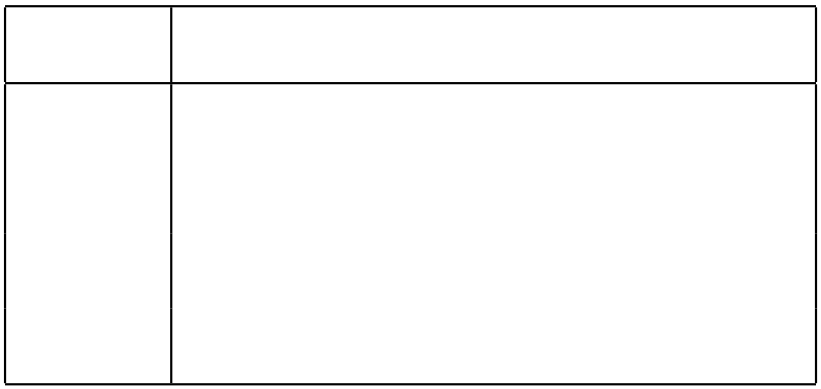
Tim Van de Cruys

University of Groningen

Statistics Seminar
February 25th, 2009

Matrices

- Capturing co-occurrence frequencies of two entities



Matrices

- Capturing co-occurrence frequencies of two entities

	doc1	doc2	doc3	doc4
word1				
word2				
word3				
word4				

Matrices

- Capturing co-occurrence frequencies of two entities

	word1	word2	word3	word4
word1				
word2				
word3				
word4				

Matrices

- Capturing co-occurrence frequencies of two entities

	dep1	dep2	dep3	dep4
word1				
word2				
word3				
word4				

Matrices

- Capturing co-occurrence frequencies of two entities

	rood	lekker	snel	tweedehands
appel	2	1	0	0
wijn	2	2	0	0
auto	1	0	1	2
vrachtwagen	1	0	1	1

Matrices

- Capturing co-occurrence frequencies of two entities

	rood	lekker	snel	tweedehands
appel	7	9	0	0
wijn	12	6	0	0
auto	7	0	8	4
vrachtwagen	2	0	3	4

Matrices

- Capturing co-occurrence frequencies of two entities

	rood	lekker	snel	tweedehands
appel	56	98	0	0
wijn	44	34	0	0
auto	23	0	31	39
vrachtwagen	4	0	18	29

Matrices

- Capturing co-occurrence frequencies of two entities

	rood	lekker	snel	tweedehands
appel	728	592	1	0
wijn	1035	437	0	2
auto	392	0	487	370
vrachtwagen	104	0	393	293

Introduction

Two reasons for performing dimensionality reduction:

- Intractable computations
  - When number of elements and number of features is too large, similarity computations may become intractable
  - reduction of the number of features makes computation tractable again
- Generalization capacity
  - the dimensionality reduction is able to describe the data better, or is able to capture intrinsic semantic features
  - dimensionality reduction is able to improve the results (counter data sparseness and noise)

Latent Semantic Analysis: Introduction 1/2

- Application of a mathematical/statistical technique to simulate how humans learn the semantics of words
- LSA finds 'latent semantic dimensions' according to which words and documents can be identified
- Words (and passages) are represented as high-dimensional vectors in this semantic space
- Goal: counter data sparseness (poverty of the stimulus) and get rid of noise

Latent Semantic Analysis: Introduction 2/2

What is Latent Semantic Analysis technically speaking?

- The application of **singular value decomposition**
- to a **term-document matrix**
- to improve **similarity calculations**

Bag-of-word semantics

- LSA represents 'bag-of-word' semantics
- Idea that meaning of a passage equals the sum of the meaning of its words
- Meaning = an unordered set of word tokens, syntax is not taken into account
- Done by representing several passages in a **term-document matrix**

Term-document matrix 1/2

Consider two documents:

- België is een koninkrijk in het midden van Europa, met als hoofdstad **Brussel**. **Brussel** heeft een Nederlandstalige en een Franstalige universiteit, maar de grootste studentenstad is **Leuven**. **Leuven** telt 27.000 studenten.
- Nederland is een West-Europees land aan de Noordzee. De hoofdstad van Nederland is **Amsterdam**. **Amsterdam** telt twee universiteiten. **Groningen** is een belangrijke studentenstad. In **Groningen** studeren 37.000 studenten.

Term-document matrix 2/2

$$\begin{bmatrix} & B & NL \\ Groningen & 0 & 2 \\ Leuven & 2 & 0 \\ Amsterdam & 0 & 2 \\ Brussel & 2 & 0 \end{bmatrix}$$

Note: Values for words are transformed to values that represent their **importance** in the passage (entropy, mutual information)

Matrix

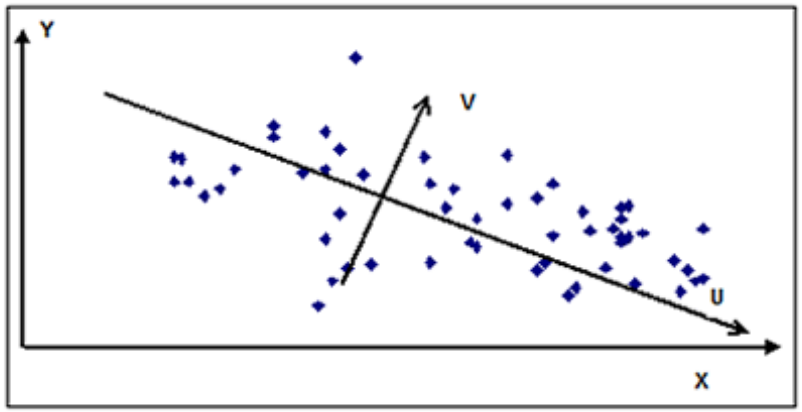
$$\begin{bmatrix} & B & NL \\ Groningen & 0 & 2 \\ Leuven & 2 & 0 \\ Amsterdam & 0 & 2 \\ Brussel & 2 & 0 \end{bmatrix}$$

Apply cosine similarity measure

- $cos(\overrightarrow{x}, \overrightarrow{y}) = \frac{\overrightarrow{x} \cdot \overrightarrow{y}}{|\overrightarrow{x}||\overrightarrow{y}|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2}}$
- Examples:
  - $cos(Groningen, Amsterdam) = \frac{4}{\sqrt{16}} = 1$
  - $cos(Groningen, Brussel) = \frac{0}{\sqrt{16}} = 0$

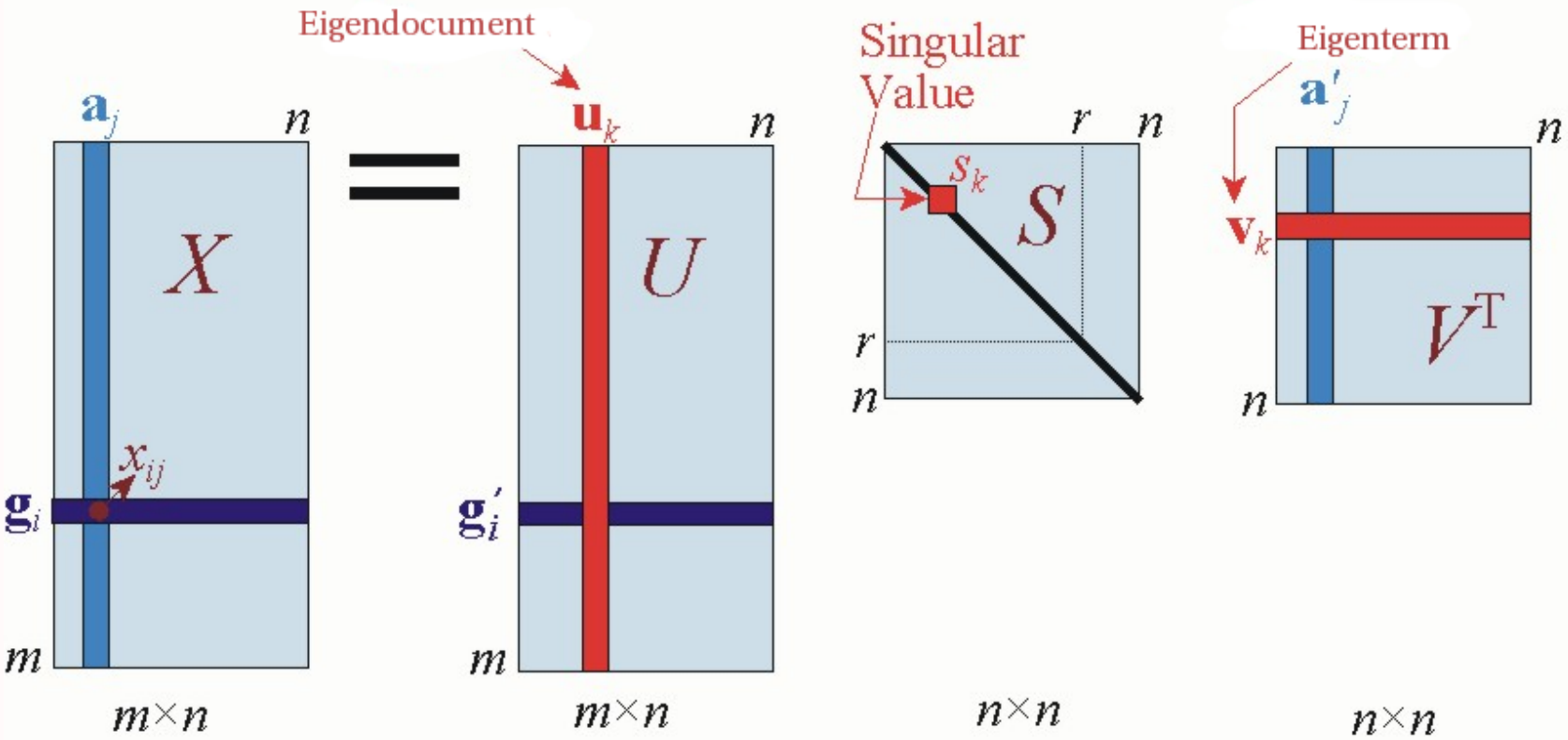
Singular Value Decomposition

- Mathematical/statistical technique closely related to principal component analysis
- SVD: generalization of PCA, computations on the actual data
- rectangular matrix instead of square covariance matrix
- Find the dimensions that explain most variance by solving number of **eigenvector** problems
- Only keep the n most important dimensions ($n = 50 - 300$)



SVD: three matrices

$$X = USV^{\mathrm{T}}$$



Example 1a

$$A \begin{bmatrix} & B & NL \\ Groningen & 0 & 2 \\ Leuven & 2 & 0 \\ Amsterdam & 0 & 2 \\ Brussel & 2 & 0 \end{bmatrix} =$$

$$U \begin{bmatrix} 0.00 & 0.71 \\ -0.71 & 0.00 \\ 0.00 & 0.71 \\ -0.71 & 0.00 \end{bmatrix} * S \begin{bmatrix} 2.83 & 0 \\ 0 & 2.83 \end{bmatrix} * V^T \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

Example 1b

$$\begin{bmatrix} & B & NL & B \\ Groningen & 0 & 2 & 0 \\ Leuven & 2 & 0 & 0 \\ Amsterdam & 0 & 2 & 0 \\ Brussel & 2 & 0 & 1 \end{bmatrix} =$$

$$U \begin{bmatrix} 0.00 & -0.71 & \emptyset \\ -0.66 & 0.00 & \emptyset \\ 0.00 & -0.71 & \emptyset \\ -0.75 & 0.00 & \emptyset \end{bmatrix} * S \begin{bmatrix} 2.92 & 0 & \emptyset \\ 0 & 2.83 & \emptyset \\ \emptyset & \emptyset & \emptyset \end{bmatrix} *$$

$$V^T \begin{bmatrix} -0.97 & 0.00 & 0.26 \\ 0 & -1.00 & 0.00 \\ \emptyset & \emptyset & \emptyset \end{bmatrix} \cong A' \begin{bmatrix} 0.0 & 2.0 & 0.0 \\ 1.9 & 0.0 & 0.5 \\ 0.0 & 2.0 & 0.0 \\ 2.1 & 0 & 0.6 \end{bmatrix}$$

Methodological remarks

- LSA in (part of) Twente Nieuws Corpus: 10 years of Dutch newspaper texts (AD, NRC, TR, VK, PAR)
- terms = nouns
  documents = paragraphs
- 20,000 terms * 2,000,000 documents matrix
- reduced to 300 dimensions

Dimension 4

politiek	0.23	Amerikaans	0.08
land	0.22	kabinet	0.08
minister	0.20	Europa	0.08
partij	0.17	verkiezing	0.08
president	0.15	Nederlands	0.08
regering	0.14	internationaal	0.07
Europees	0.12	steun	0.07
Nederland	0.12	leider	0.07
premier	0.11	economisch	0.07
militair	0.09	VVD	0.07

Dimension 38

maand	0.22	cd	0.09
muziek	0.19	wereld	0.09
politie	0.15	geld	0.08
publiek	0.15	programma	0.08
school	0.14	groei	0.08
hoor	0.14	zing	0.08
volg	0.14	lied	0.08
minister	0.12	miljoen	0.08
lijk	0.11	win	0.08
nummer	0.11	val	0.08

Clustering with LSA 1/2

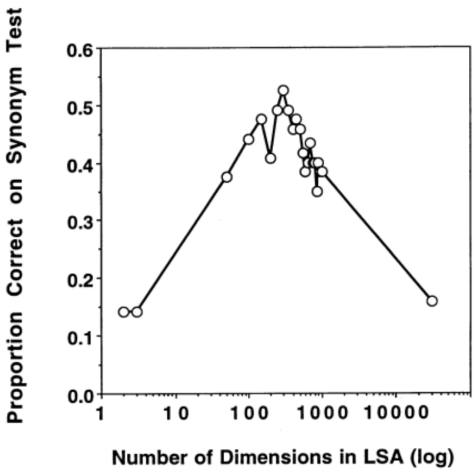
- azijn bak bestrijk bestrooi boter bouillon bout deeg dek_af deksel eierdooier folie gaar garde giet goudbruin gril hak keukenpapier knapperig knoflook koekepan koel_af koelkast kook korrel lepel meng mengsel oven pan part pel pit rijst roer roer_door roerend room rooster royaal saus schep schep_om schik schil serveer smelt smoor snijd snipper strooi sudder vergiet verhit verwarm voeg_toe voorverwarmd vork vruchtvlees vuur wok wrijf_in zacht zeef zout

Clustering with LSA 2/2

- Breuker cellist cello Coleman compositie Concert concertzaal conservatorium contra_bas dj Duke Ensemble Es Gardiner house improvisatie instrument interval jazz Jazz klank klarinet luister Marsalis Monk musiceer musicus muziek_liefhebber muzikaal orgel pianist piano Productions recital Reinbert Rieu saxofonist saxofoon slagwerk slagwerker solist strijker symfonische trompet trompettist twintigste_eeuws violist viool Zappa zweep_op

LSA & synonym tests

- LSA trained on Grolier Encyclopedia, and given synonym test (TOEFL test).
- LSA scores 65%, identical to the average score of a large sample of students applying for college entrance in the United States from non-English speaking countries.



LSA & essay grading

- LSA was used to assign scores to essay questions
- Trained on instructional text read by students, and a few example essays (or pre-graded essays)
- LSA scores found to be as good as expert assigned scores
- Moreover: LSA correlated significantly better with individual expert graders than one expert correlated with another
- LSA grades essays as good as expert **and** more objective

Introduction

- LSA criticized for a number of reasons:
  - dimensionality reduction is best fit in least-square sense, but this is only valid for normally distributed data; language date is not normally distributed
  - Dimensions may contain negative values; it is not clear what negativity on a semantic scale should designate
- Shortcomings are remedied by subsequent techniques (PLSA, LDA, NMF, . . .)

Technique 1/2

- Given a non-negative matrix V, find non-negative matrix factors W and H such that:

$$V_{nxm} \approx W_{nxr} H_{rxm} \tag{1}$$

- Choosing $r \ll n, m$ reduces data
- Constraint on factorization: all values in three matrices need to be *non-negative values* (≥ 0)
- Constraint brings about a *parts-based* representation: only additive, no subtractive relations are allowed

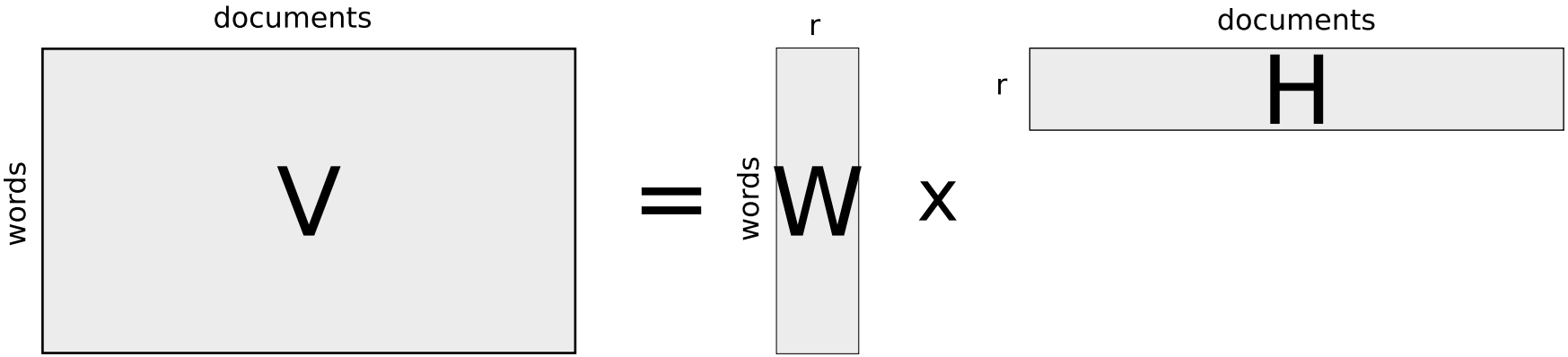
Technique 2/2

- Different kinds of NMF's that minimize different cost functions:
  - Square of Euclidean distance
  - Kullback-Leibler divergence
    $\Rightarrow$ better suited for language phenomena
- To find NMF is to minimize $D(V \| WH)$ with respect to W and H, subject to the constraints $W, H \geq 0$
- This can be done with *update rules*

$$H_{a\mu} \leftarrow H_{a\mu} \frac{\sum_i W_{ia} \frac{V_{i\mu}}{(WH)_{i\mu}}}{\sum_k W_{ka}} \qquad W_{ia} \leftarrow W_{ia} \frac{\sum_\mu H_{a\mu} \frac{V_{i\mu}}{(WH)_{i\mu}}}{\sum_v H_{av}} \tag{2}$$

- these update rules find a *local minimum* in the minimization of KL divergence

Graphical Representation

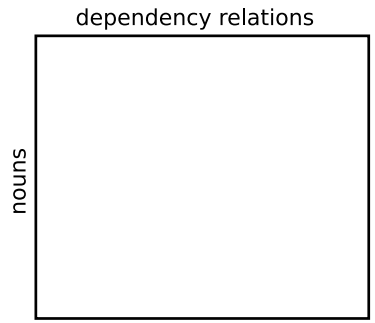


Results

- Context vectors (5k nouns $\times$ 2k co-occurring nouns) extracted from CLEF corpus
- NMF is able to capture 'semantic' dimensions
- Examples:
  - *bus* 'bus', *taxi* 'taxi', *trein* 'train', *halte* 'stop', *reiziger* 'traveler', *perron* 'platform', *tram* 'tram', *station* 'station', *chauffeur* 'driver', *passagier* 'passenger'
  - *bouillon* 'broth', *slagroom* 'cream', *ui* 'onion', *eierdooier* 'egg yolk', *laurierblad* 'bay leaf', *zout* 'salt', *deciliter* 'decilitre', *boter* 'butter', *bleekselderij* 'celery', *saus* 'sauce'

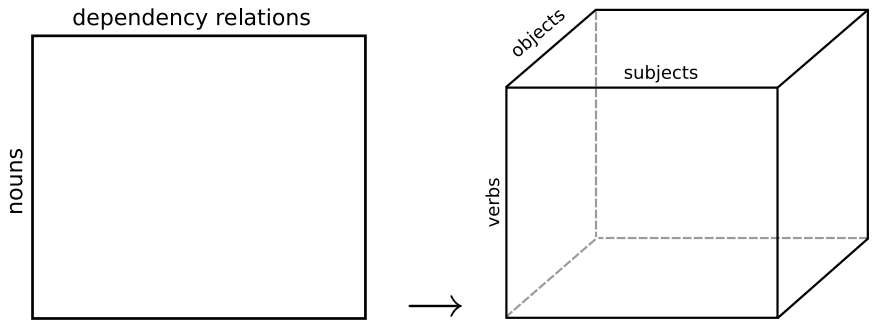
Two-way vs. three-way

- all methods use two way co-occurrence frequencies $\longrightarrow$ matrix
- suitable for two-way problems
  - words $\times$ documents
  - nouns $\times$ dependency relations
- not suitable for *n*-way problems
  - words $\times$ documents $\times$ authors
  - verbs $\times$ subjects $\times$ direct objects



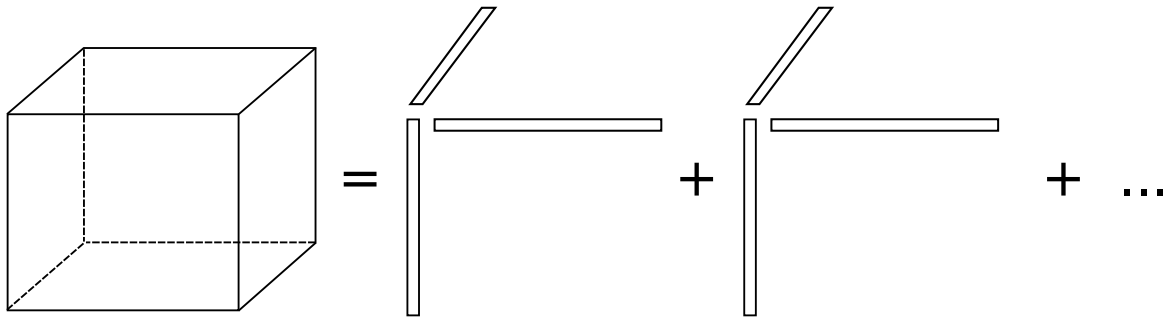
Two-way vs. three-way

- all methods use two way co-occurrence frequencies $\longrightarrow$ matrix
- suitable for two-way problems
  - words $\times$ documents
  - nouns $\times$ dependency relations
- not suitable for *n*-way problems $\longrightarrow$ tensor
  - words $\times$ documents $\times$ authors
  - verbs $\times$ subjects $\times$ direct objects

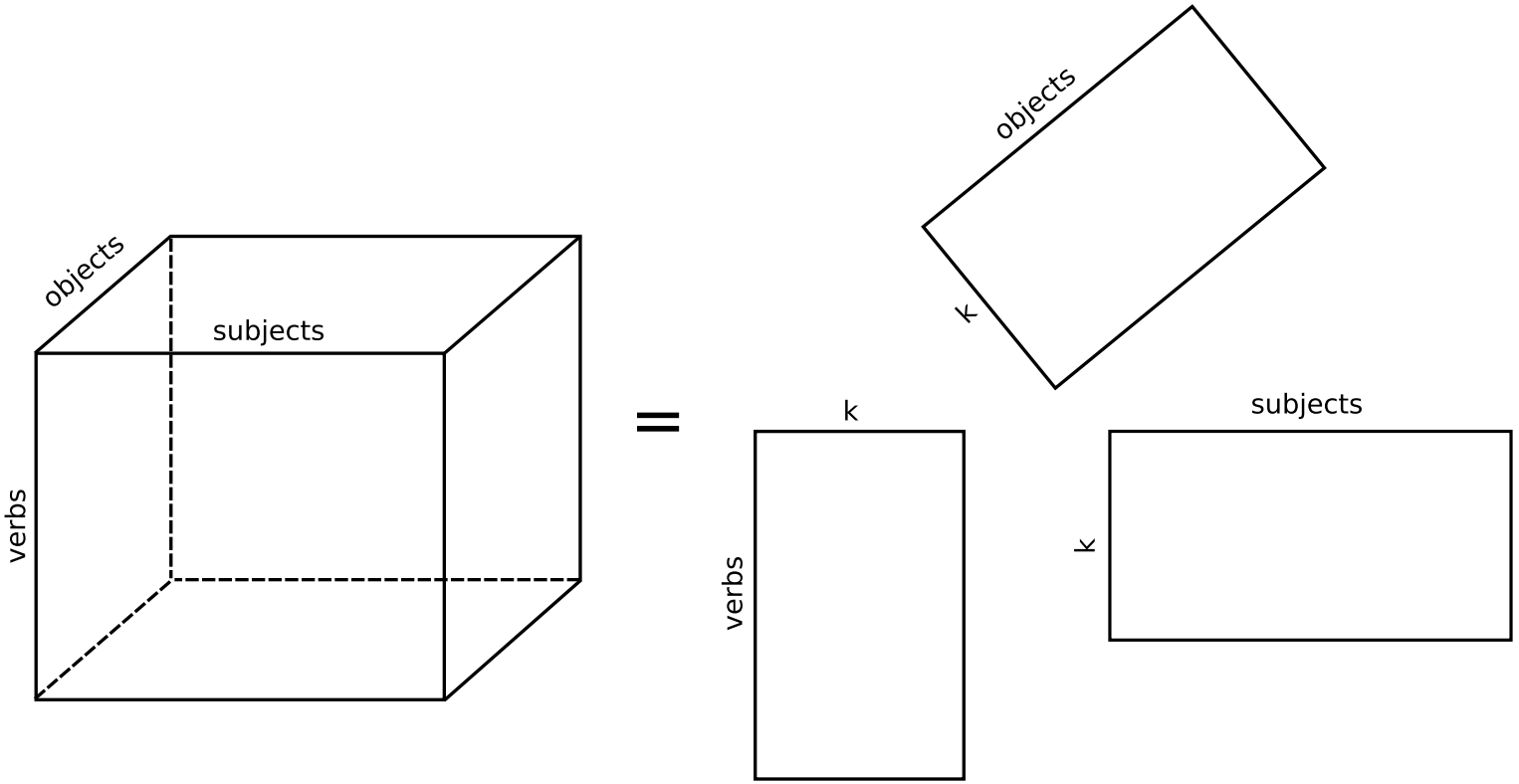


Technique

- Idea similar to non-negative matrix factorization
- Calculations are different
- $\min_{x_i \in \mathbb{R}^{D1}_{\geq 0}, y_i \in \mathbb{R}^{D2}_{\geq 0}, z_i \in \mathbb{R}^{D3}_{\geq 0}} \| T - \sum_{i=1}^{k} x_i \circ y_i \circ z_i \|_F^2$



Graphical representation



Methodology

- Three-way extraction of selectional preferences
- Approach applied to Dutch, using TWENTE NIEUWS CORPUS (500M words)
- parsed with Alpino
- three-way co-occurrence of verbs with subjects and direct objects extracted
- adapted with extension of pointwise mutual information
- Resulting tensor 1K verbs $\times$ 10K subjects $\times$ 10K direct objects
- reduction to k dimensions ($k = 50, 100, 300$)

Example dimension: police action

subjects	su_s	verbs	v_s	objects	obj_s
politie 'police'	.99	*houd_aan* 'arrest'	.64	*verdachte* 'suspect'	.16
agent 'policeman'	.07	*arresteer* 'arrest'	.63	*man* 'man'	.16
autoriteit 'authority'	.05	*pak_op* 'run in'	.41	*betoger* 'demonstrator'	.14
Justitie 'Justice'	.05	*schiet_dood* 'shoot'	.08	*relschopper* 'rioter'	.13
recherche 'detective force'	.04	*verdenk* 'suspect'	.07	*raddraaier* 'instigator'	.13
marechaussee 'military police'	.04	*tref_aan* 'find'	.06	*overvaller* 'raider'	.13
justitie 'justice'	.04	*achterhaal* 'overtake'	.05	*Roemeen* 'Romanian'	.13
arrestatieteam 'special squad'	.03	*verwijder* 'remove'	.05	*actievoerder* 'campaigner'	.13
leger 'army'	.03	*zoek* 'search'	.04	*hooligan* 'hooligan'	.13
douane 'customs'	.02	*spoor_op* 'track'	.03	*Algerijn* 'Algerian'	.13

Example dimension: legislation

subjects	su_s	**verbs**	v_s	**objects**	obj_s
meerderheid 'majority'	.33	*steun* 'support'	.83	*motie* 'motion'	.63
VVD	.28	*dien_in* 'submit'	.44	*voorstel* 'proposal'	.53
D66	.25	*neem_aan* 'pass'	.23	*plan* 'plan'	.28
Kamermeerderheid	.25	*wijs_af* 'reject'	.17	*wetsvoorstel* 'bill'	.19
fractie 'party'	.24	*verwerp* 'reject'	.14	*hem* 'him'	.18
PvdA	.23	*vind* 'think'	.08	*kabinet* 'cabinet'	.16
CDA	.23	*aanvaard* 'accepts'	.05	*minister* 'minister'	.16
Tweede Kamer	.21	*behandel* 'treat'	.05	*beleid* 'policy'	.13
partij 'party'	.20	*doe* 'do'	.04	*kandidatuur* 'candidature'	.11
Kamer 'Chamber'	.20	*keur_goed* 'pass'	.03	*amendement* 'amendment'	.09

Example dimension: exhibition

subjects	su_s	**verbs**	v_s	**objects**	obj_s
tentoonstelling 'exhibition'	.50	*toon* 'display'	.72	*schilderij* 'painting'	.47
expositie 'exposition'	.49	*omvat* 'cover'	.63	*werk* 'work'	.46
galerie 'gallery'	.36	*bevat* 'contain'	.18	*tekening* 'drawing'	.36
collectie 'collection'	.29	*presenteer* 'present'	.17	*foto* 'picture'	.33
museum 'museum'	.27	*laat* 'let'	.07	*sculptuur* 'sculpture'	.25
oeuvre 'oeuvre'	.22	*koop* 'buy'	.07	*aquarel* 'aquarelle'	.20
Kunsthal	.19	*bezit* 'own'	.06	*object* 'object'	.19
kunstenaar 'artist'	.15	*zie* 'see'	.05	*beeld* 'statue'	.12
dat 'that'	.12	*koop_aan* 'acquire'	.05	*overzicht* 'overview'	.12
hij 'he'	.10	*in huis heb* 'own'	.04	*portret* 'portrait'	.11