Vector semantics -II

Anton Alekseev,
Steklov Mathematical Institute in St Petersburg

NRU ITMO, St Petersburg, 2018
anton.m.alexeyev+itmo@gmail.com

REMINDER

Distributional hypothesis

- Zellig S. Harris: “oculist and eye-doctor... occur in almost the same environments”, “If A and B have almost **identical environments**. . . we say that they are synonyms”

- Most famous, John Firth:
You shall know a word by the company it keeps!

BTW,
Z. Harris is sometimes referred to as Noam Chomsky’s teacher

John Rupert Firth -- the originator of the London school of linguistics

Harris, Z. S. (1954). Distributional structure. Word, 10, 146–162. Reprinted in J. Fodor and J. Katz, The Structure of Language, Prentice Hall, 1964
Z. S. Harris, Papers in Structural and Transformational Linguistics, Reidel, 1970, 775–794

Firth, J. R. (1957). A synopsis of linguistic theory 1930– 1955. In Studies in Linguistic Analysis. Philological Society. Reprinted in Palmer, F. (ed.) 1968. Selected Papers of J. R. Firth. Longman, Harlow

REMINDER

What IS ‘similarity’?

many faces of similarity

- dog -- cat
- dog -- poodle
- dog -- animal
- dog -- bark
- dog -- leash

- dog -- chair — same POS
- dog -- dig — edit distance
- dog -- god — same letters
- dog -- fog — rhyme
- dog -- 6op — shape

https://speakerdeck.com/mlreview/yoav-goldberg-word-embeddings-what-how-and-whither

Reminder

We already know sparse representations:
term-term/term-document counts/weights

1) how to build the matrix
2) a few ways to set weights
3) tricks to tune
4) how to evaluate (extrinsic/intrinsic)

“Dense” vectors

- tens of thousands dimensions to hundreds dimensions
- small number of zeros
- moving away from approach ‘coordinate=term’

But... why would we do it?

Sparse vectors we’ve discussed assign every word a coordinate, hence

- models using sparse vectors as input are hard to train: a large number of parameters sometimes makes machine learning models too complex

- it is hard to ‘grasp’ synonymy as contexts-synonyms simply have different and unrelated coordinates

Main approaches

1. Matrix factorization
2. “Predictive”, “neural” approaches
3. Word clustering

Lecture plan

1. ~~Sparse vectors~~
    a. ~~"Term-document" approach~~
    b. ~~"Term-term" approach~~
        i. ~~Construction~~
        ii. ~~HAL~~
    c. ~~Weighting~~
    d. ~~Semantic similarity estimation~~
    e. ~~Quality evaluation~~
2. Dense vectors
    a. Matrix decomposition
    b. "Predictive" approaches

Matrix decomposition

Intuition:

1) we decrease the number of dimensions hoping to keep the regularities and laws present in the data (e.g., synonymy),

2) one may want to keep only the most ‘important’ coordinates (the ones that have the largest variance in values)

SVD: singular value decomposition

Any matrix can be represented like this

$$A = USV^T$$

where **S** is **a diagonal matrix** (having the same dimensions as A),
values on diagonals are singular values, **U, V are orthogonal**

Eckart-Yang theorem
the best possible **rank k approximation of the matrix A** (in terms of Frobenius norm) is a singular value decomposition, where in the resulting matrix **S** only first **k diagonal elements** are non-zero and are ordered in non-increasing order.

Lower rank approximation

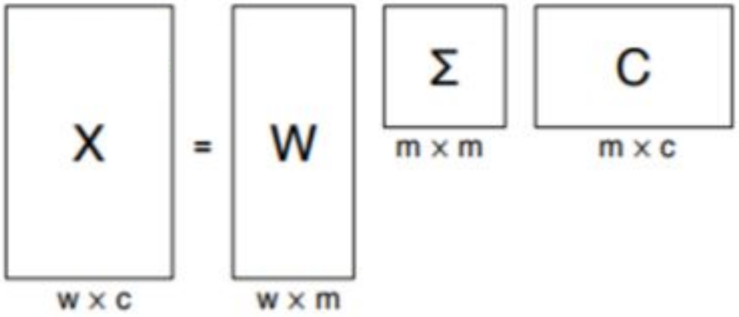
The task can be posed in a different way

W: matrix: **w words** x **m dimensions** of the 'latent space', and

- columns are orthogonal to each other
- columns are ordered in the order of decreasing variance in coordinates in a new space

Σ: diagonal matrix **m** x **m**, where each value on the diagonal reflects the 'importance' of the corresponding dimension

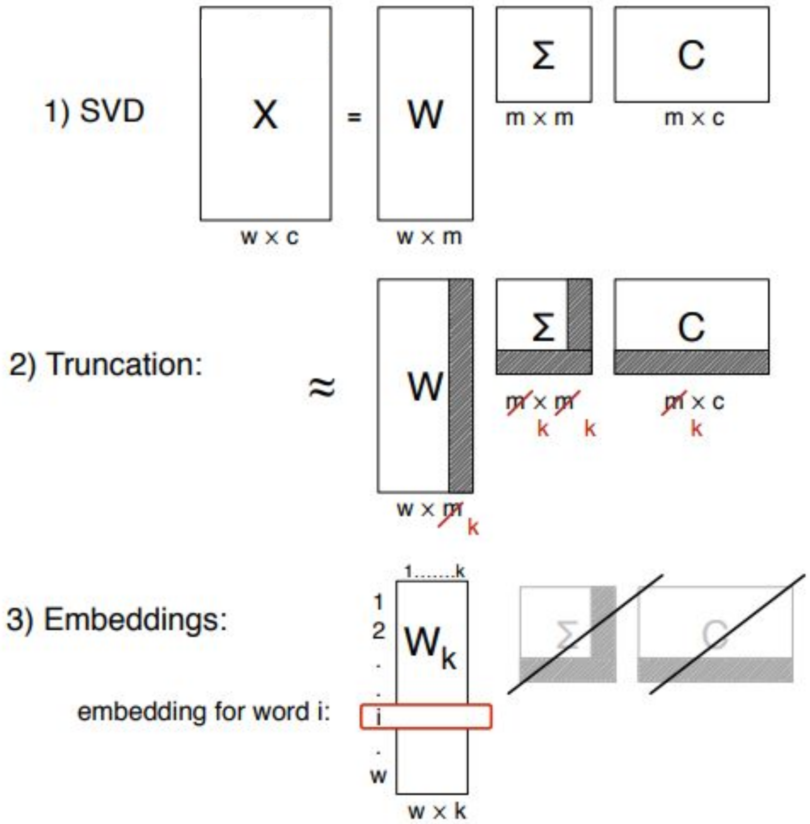
C: matrix: **m** x **c**



Truncated SVD

Letting only top K dimensions live

Then our word vector representations are corresponding rows in matrix W_k , that is, k-dimensional vectors



LSA: Latent Semantic Analysis

	access	document	retrieval	information	theory	database	indexing	computer
Doc 1	x	x	x			x	x	
Doc 2				x*	x			x*
Doc 3			x	x*				x*

Applying SVD (**m** = hundreds) to term-document matrix,
setting weights as a product of:

the local weight

$$\log f(i,j) + 1$$

the global weight

$$1 + \frac{\sum_j p(i,j) \log p(i,j)}{\log D}$$

for all terms **i** in all documents **j**

S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman. 1988. Using latent semantic analysis to improve access to textual information. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '88), J. J. O'Hare (Ed.). ACM, New York, NY, USA, 281-285.

Truncated SVD for term-term PPMI matrix

We simply apply SVD to word-context matrix and cut off some of the dimensions, choosing **k** manually. Sometimes works better than the sparse analogue.

Other notes on SVD as a way of obtaining vector representations of words:

- $(W\Sigma)^T$ can also be treated and used as word vectors (it doesn't work, though)
- Truncating (you never know, but it seems so) helps to generalize and filter out useless information,
- Sometimes throwing away the **first few dimensions** may be helpful

However, it is computationally hard

to be continued...

Used/recommended materials

1. Martin/Jurafsky, Ch. 15
2. Yoav Goldberg: word embeddings what, how and whither
3. Papers on slides
4. Valentin Malykh from ODS/iPavlov on w2v
5. A very cool explanation of what word2vec is
6. Wikipedia

Vector semantics -II

Anton Alekseev,
Steklov Mathematical Institute in St Petersburg

NRU ITMO, St Petersburg, 2018
anton.m.alexeyev+itmo@gmail.com

Many thanks to Denis Kirjanov (who knows some real linguistics) for words of advice