# An easy introduction to LDA and NMF

17.05.2022 Yuyin Lang

Faculty of Informatics
Technische Universität München
wwwmatthes.in.tum.de

# Outline

LDA

- Basic idea
- Assumptions
- How does it work?
- An example

NMF

- Idea

# LDA – basic idea

- Each document is made up of various words
- Each topic also has various words belonging to it
- Aim: Finding topics a document belongs to, based on the words in it. (Note that a document can have multiple topics)

An example:

Suppose we have 5 documents and 3 topics:

```
Doc1: word1, word3, word5, word45, word11, word 62, word88 ...
Doc2: word9, word77, word31, word58, word83, word 92, word49 ...
Doc3: word44, word18, word52, word36, word64, word 11, word20 ...
Doc4: word85, word62, word19, word4, word30, word 94, word67 ...
Doc5: word19, word53, word74, word79, word45, word 39, word54 ...
```

This tells us which words are in which documents.

| | Word1 | word2 | word3 | word4 | ..... |
|---|---|---|---|---|---|
| Topic1 | 0.01 | 0.23 | 0.19 | 0.03 | |
| Topic2 | 0.21 | 0.07 | 0.48 | 0.02 | |
| Topic3 | 0.53 | 0.01 | 0.17 | 0.04 | |

This table gives the prob of the words showing up in a specific topic.

# LDA – basic idea

An example:

Now, suppose we have topics “Dog_related” and “Cat_related”

Words in “Dog_related” are more likely: puppy, bark, and bone

Words in “Cat_related” are more likely: milk, meow, and kitten

Hence, the document “Dogs like to chew on bones and fetch sticks. Puppies drink milk. Both like to bark.” should be classified as “Dog_related”.

# LDA – Assumptions

- BOW: order of the words and the grammatical role of the words (subject, object, verbs, …) are not considered in the model.
- Delete irrelevant words (like stop words)
- The number of topics is predefined
- All topic assignments except for the current word in question are correct, and then update the assignment of the current word using our model of how documents are generated

# LDA – How does it work?

We need two parts:

- The words that belong to a document (already known)
- The words that belong to a topic **or the probability** of words belonging into a topic (unknown)

Algorithm:

- Go through each document and **randomly assign** each word in the document to one of k topics (k is chosen beforehand).
- For each document d, go through each word w and compute
  - p(topic t | document d): the proportion of words in document d that are assigned to topic t.

  (#words in d with t +alpha/ #words in d with any topic+ k*alpha) (smoothing)

  - p(word w | topic t): the proportion of assignments to topic t over all documents that come from this word w.

  (Tries to capture how many documents are in topic t because of word w)

- Update the probability for the word w belonging to topic t, as

  p(word w with topic t) = p(topic t | document d) * p(word w | topic t)

# LDA – An example

Suppose we have various photographs(documents) with captions(words)

The task is to categorize them on themes(topics)

- Topics: Nature and city

Steps:

- 1. Assign the photographs which only have nature or city elements in them into their respective categories while randomly assign the rest.
- 2. Notice that tree is related to nature while building is related to city
- 3. Choose the caption “The tree is in front of the building and behind a car”, the word “tree”, the topic “nature”, compute p(topic t | document d) ✉ low, because building and car are more likely to be city
- 4. Compute p(word w| topic t) ✉ high, because many nature photos contain tree
- 5. Multiply these two and we get a lower relation between tree and nature

Reference: https://towardsdatascience.com/latent-dirichlet-allocation-lda-9d1cd064ffa2

# Outline

LDA

- Basic idea
- Assumptions
- How does it work?
- An example

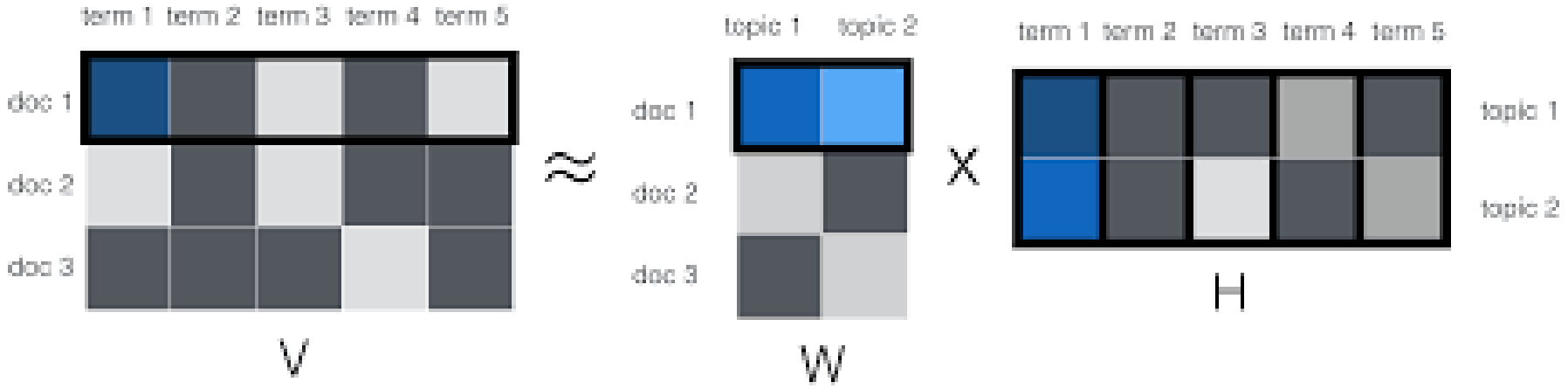
NMF

- Idea

# NMF – Idea

NMF: Non-negative matrix factorization

Given a matrix V, find non-negative W and H so that V = W * H




Reference: https://medium.com/analytics-vidhya/topic-modeling-with-non-negative-matrix-factorization-nmf-3caf3a6bb6da