Topic Modeling & Latent Dirichlet Allocation

MMAI 5400 – lecture 5
Fall 2024

Today's class

Topic models

- Applications

Latent Dirichlet Allocation (LDA)

- Simple
- Dirichlet distribution
- Combined

Learning a topic model

An application

Evaluation

Tutorial

BoW representation

- Number of features is $|V|$

The size of the vocabulary ($|V|$) can easily be > 10 000.

- In non-NLP ML we would do some form of dimensionality reduction (e.g. PCA)

Can we do something similar with BoW?

- PCA for NLP: **Latent Semantic Analysis** (LSA) [not discussed further]
- Topic Modeling: **Latent Dirichlet Allocation** (LDA)

Topic model

A model for discovering the abstract “topics” that occur in a corpus.

The “topics” are not provided labels (that would be *supervised learning*), but discovered by the algorithm (*unsupervised learning*).

Topic model

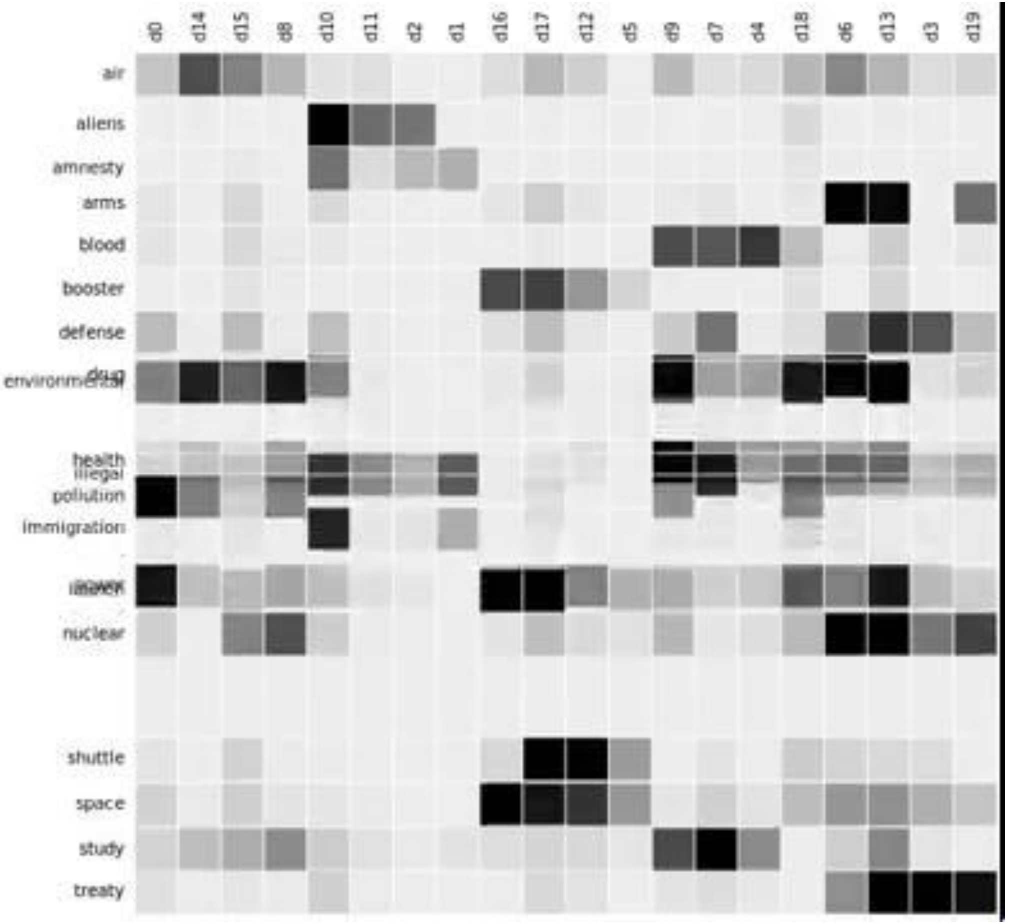
Intuitive description

- If a document is about a particular topic, then we expect particular words to appear with varying frequencies. E.g. "*dog*" & "*bone*" should be frequent in documents about dogs, whereas "*cat*" & "*meow*" should appear in documents about cats.
- A document often mixes topics in different proportions. I.e. a document can be 10% about cats & 90% about dogs.
- The "topics" produced by topic modeling are clusters of similar words.

Question: What does d0 - d19 mean?

Topic model

Amazing visualization!!



Applications of topic modelling

Text-mining

- Organize huge amounts to text (too much for human processing capacity).

Recommendation systems

- Organize documents by similarity.
    - See later slides.

Dynamic text analysis

- Track the change in topics over time.
    - E.g. Science articles from 1880 to 2000.

Population genetics

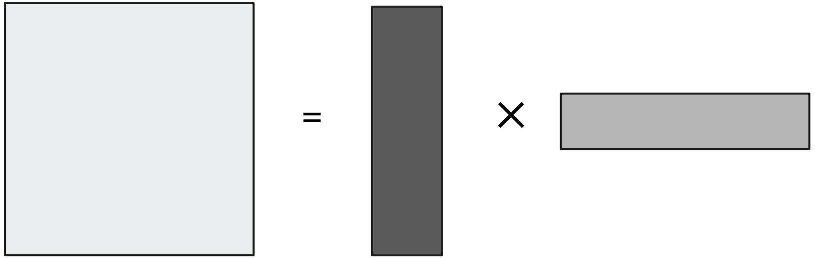
- Scaling probabilistic models of genetic variation to millions of humans

Topic modeling
- Latent Dirichlet Allocation (LDA)

Originally from population genetics by Pritchard, Stephens & Donnelly in 2000.

Applied to ML & text mining in 2003 by Blei, Ng & Jordan.

Decomposes a term-frequency matrix (terms in rows & documents in columns) into the product of a tall & skinny & a short & wide matrix.

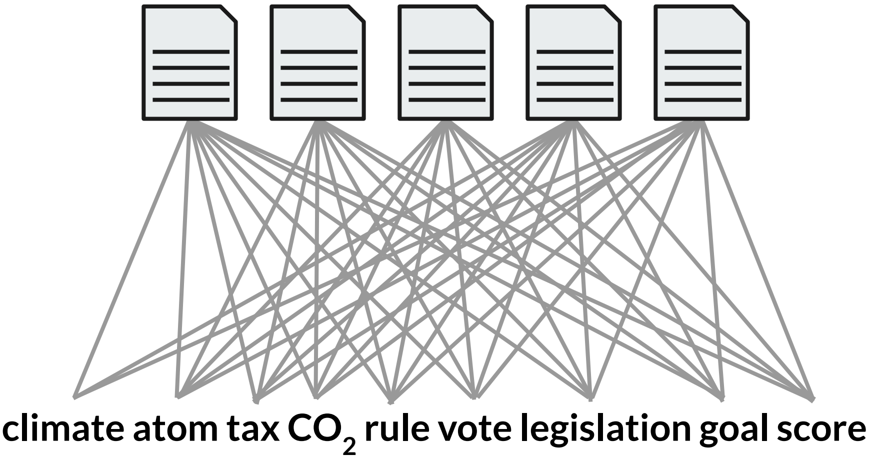


Question: LDA looks more complicated than term-frequency, how can it have fewer weights/probabilities?

LDA - simple

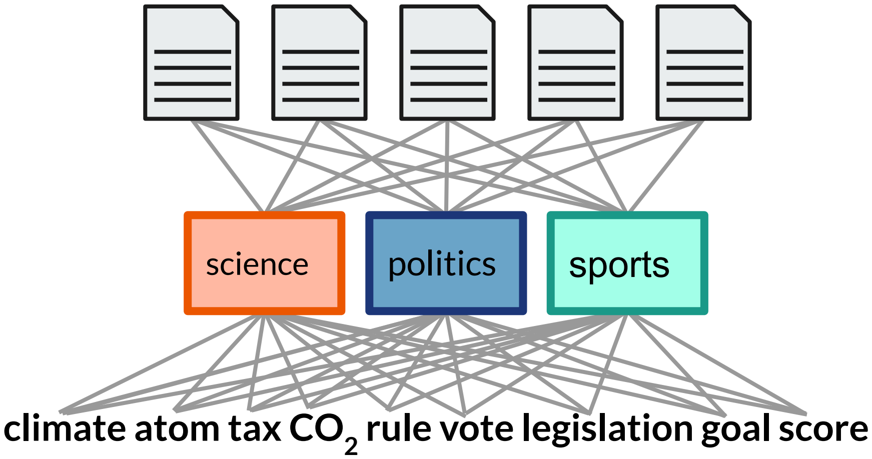
Term-frequency

- A weight/probability for every document-term pair



LDA

- A weight for each document-topic pair
- A weight for each topic-term pair



LDA - intuitive

Two layers of aggregation

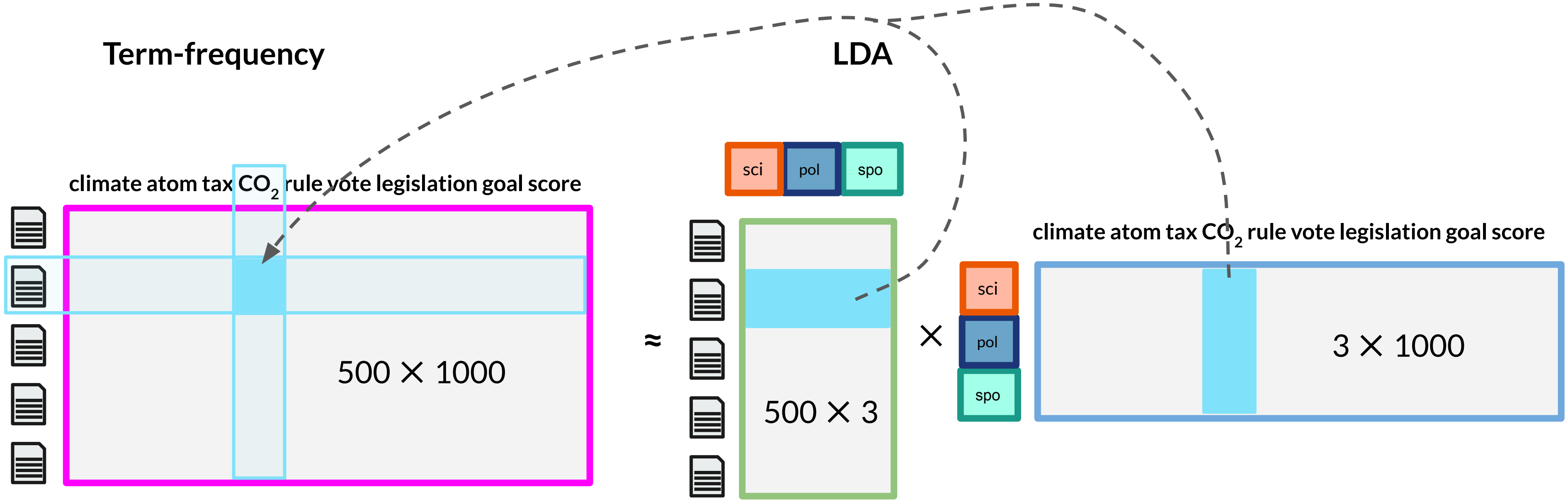
First layer is the distribution of topics

- E.g. finance news, weather news & political news.

Second layer is the distribution of words within the topic

- E.g. “sunny” & “cloud” are common in weather news while “money” & “stock” are common in finance news.

LDA



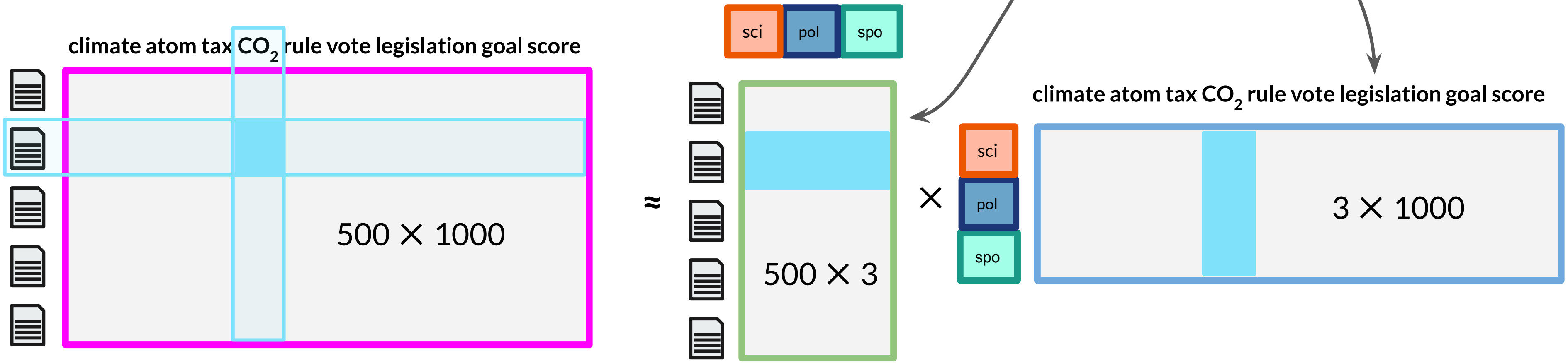
Question:
How many weights/probabilities in a tf-matrix with 500 documents & 1 000 terms?

Question:
How many weights/probabilities in an LDA-model with with 500 documents, 3 topics & 1 000 terms?

LDA

$$P(t|d) = \sum_z P(t|z)P(z|d)$$

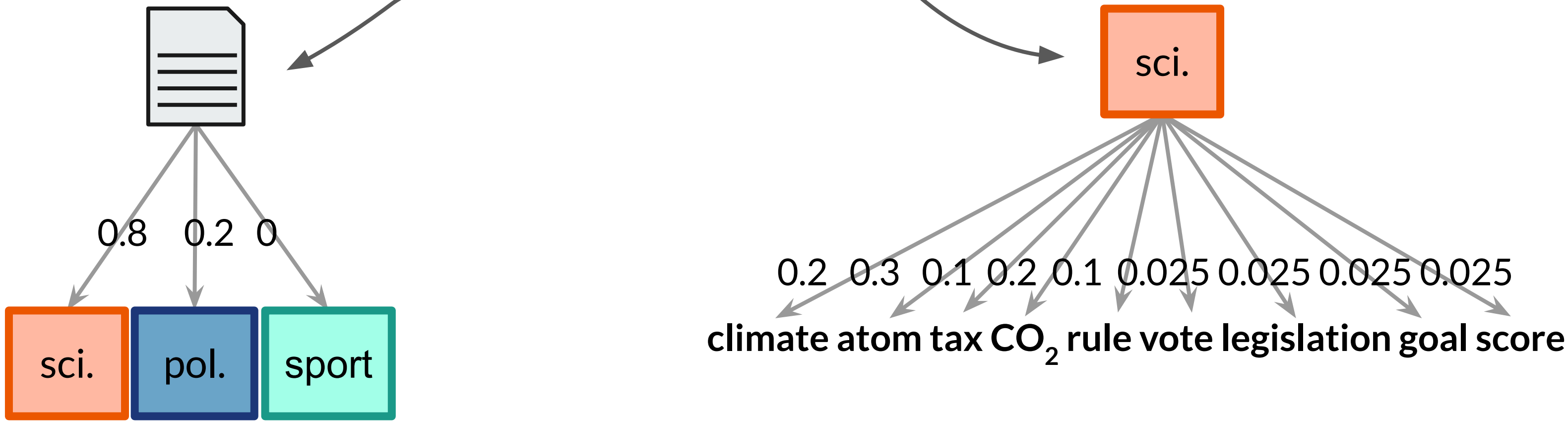
Shorthand for sum over *topics* (z)



t - term/token/word; d - document; z - topic

LDA

$$P(t|d) = \sum_{z} P(t|z)P(z|d)$$

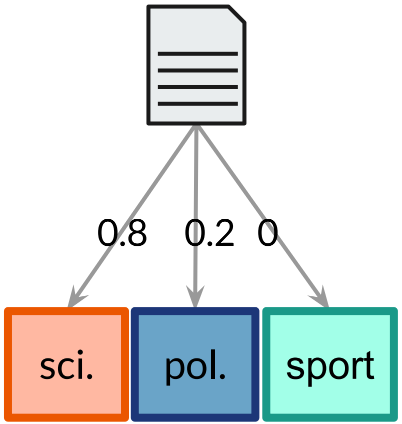


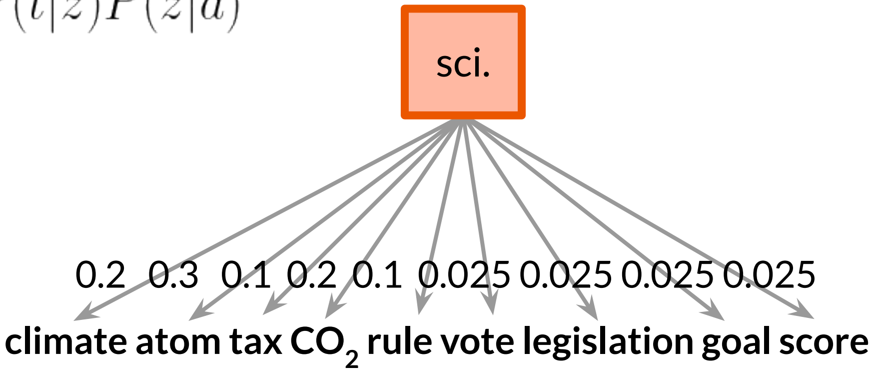
t - term/token/word; d - document; z - topic

LDA

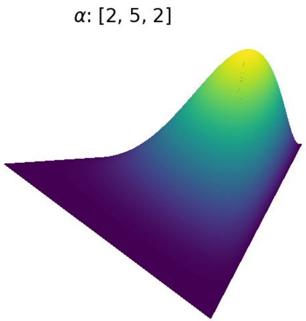
- How do we get the probabilities?

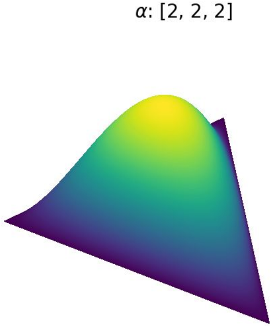
$$P(t|d) = \sum_z P(t|z)P(z|d)$$





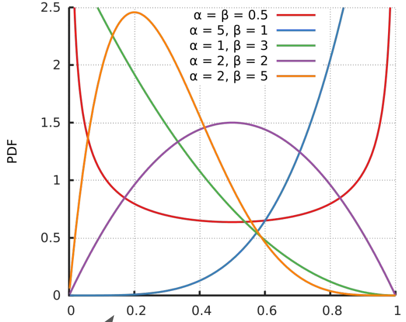
Latent Dirichlet Allocation





To estimate $P(t|z)$ & $P(z|d)$ we will use the **Dirichlet probability distribution**

- Multivariate probability distribution
- A generalization of the univariate *beta* distribution
- Parameterized by a vector α



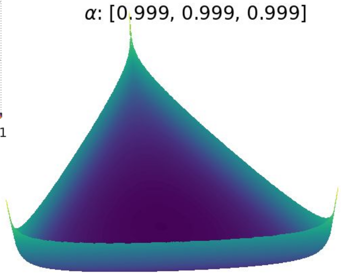
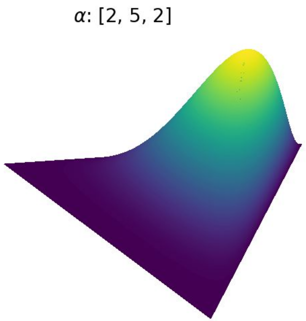
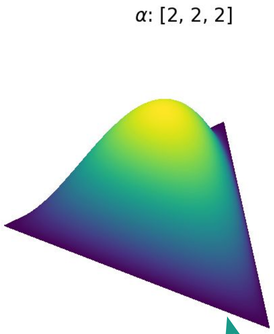


Image credit: https://commons.wikimedia.org/wiki/File:Beta_distribution_pdf.svg

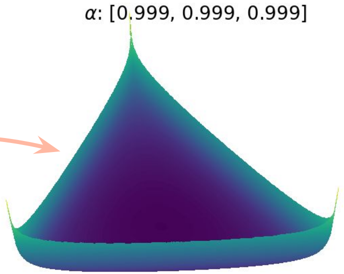
The Dirichlet probability distribution





α controls the shape of the distribution:

- α: a vector of the same dimensionality as the distribution
- E.g. for a probability distribution over 10 topics, $|\alpha| = 10$
- If all elements in α are equal $\rightarrow$ symmetric distribution
- Unequal elements $\rightarrow$ asymmetric distribution
- Elements < 1 $\rightarrow$ peak in corners (concave)
- Elements > 1 $\rightarrow$ peak in middle (convex)



Dirichlet probability distribution in LDA

Topics lie at the corners of the distribution

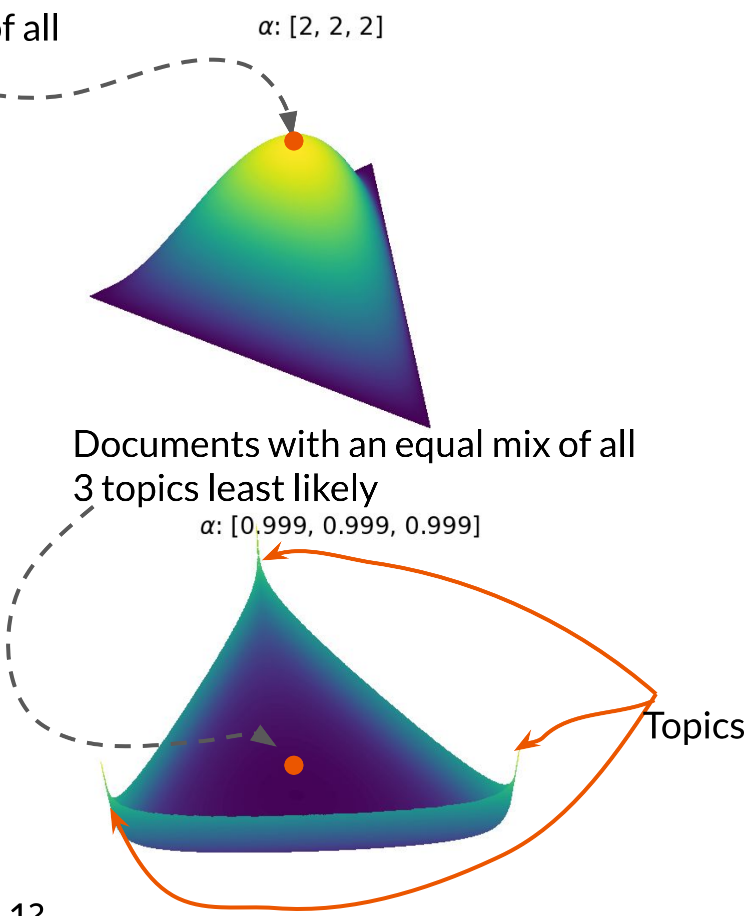
A point on the distribution represents the probability of that combination of topics (colour represents probability)

Elements in $\alpha < 1 \rightarrow$ peak in corners (concave)

- Favours assigning 1 or a few topics per document

Elements $> 1 \rightarrow$ peak in middle (convex)

- Favours assigning many topics per document



Question: For topic modeling: what makes most sense α values > 1 or < 1?

What is a simplex?

The Dirichlet distribution is a simplex

A simplex is a generalization of the triangle

One edge between a corner & any other corner, all edges have equal length

- 0-simplex: a point
- 1-simplex: a line segment
- 2-simplex: a triangle
- 3-simplex: a tetrahedron
- …

Simplexes that can be visualized in 3D

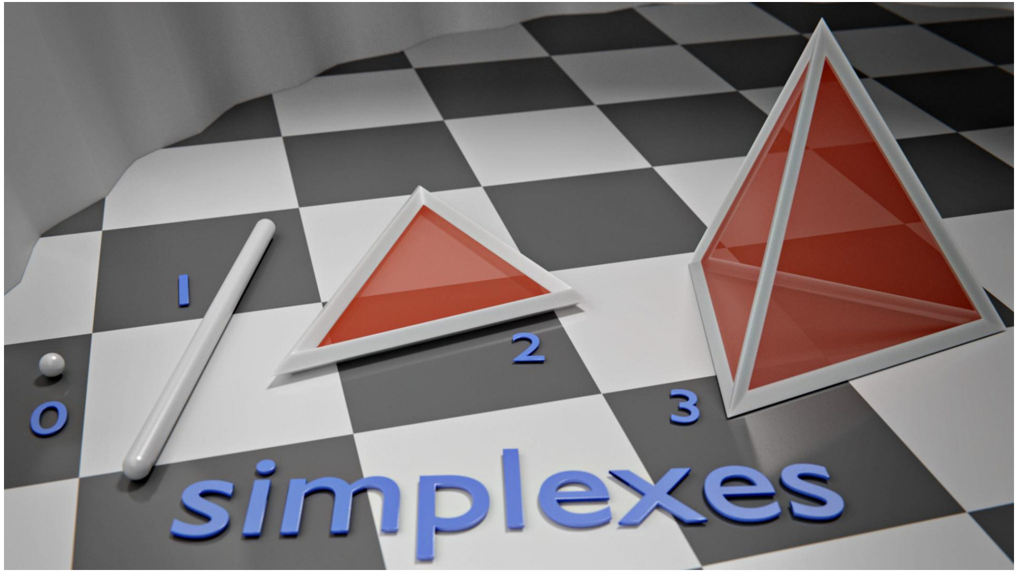


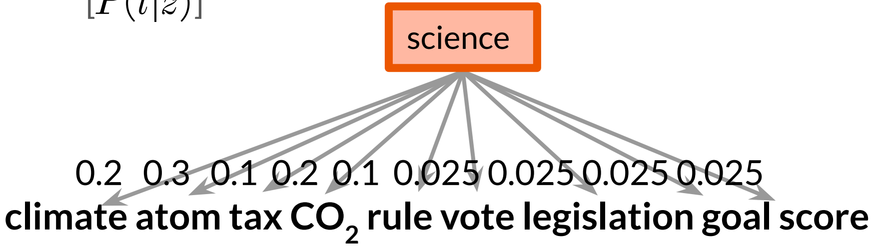
Image credit: Hjhornbeck - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=57828604

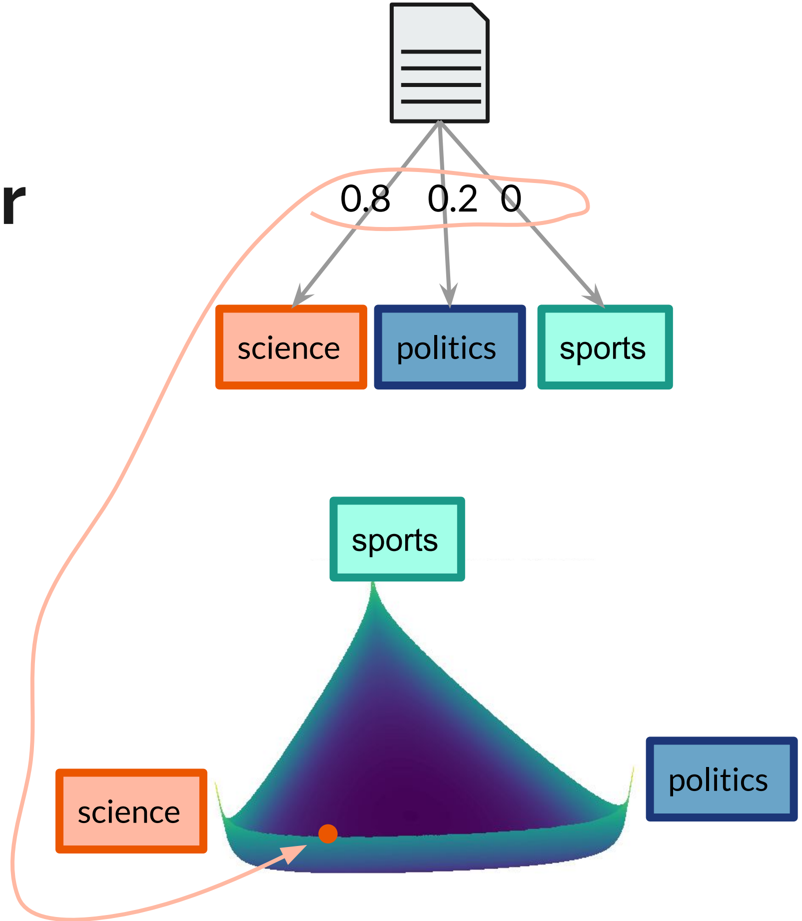
t - term/token/word; d - document; z - topic

Putting the pieces together

A document made up of n topics will be represented as a point in an n-dimensional Dirichlet distribution $[P(z|d)]$

A topic made up of k words will be represented as a point in a k-dimensional Dirichlet distribution $[P(t|z)]$





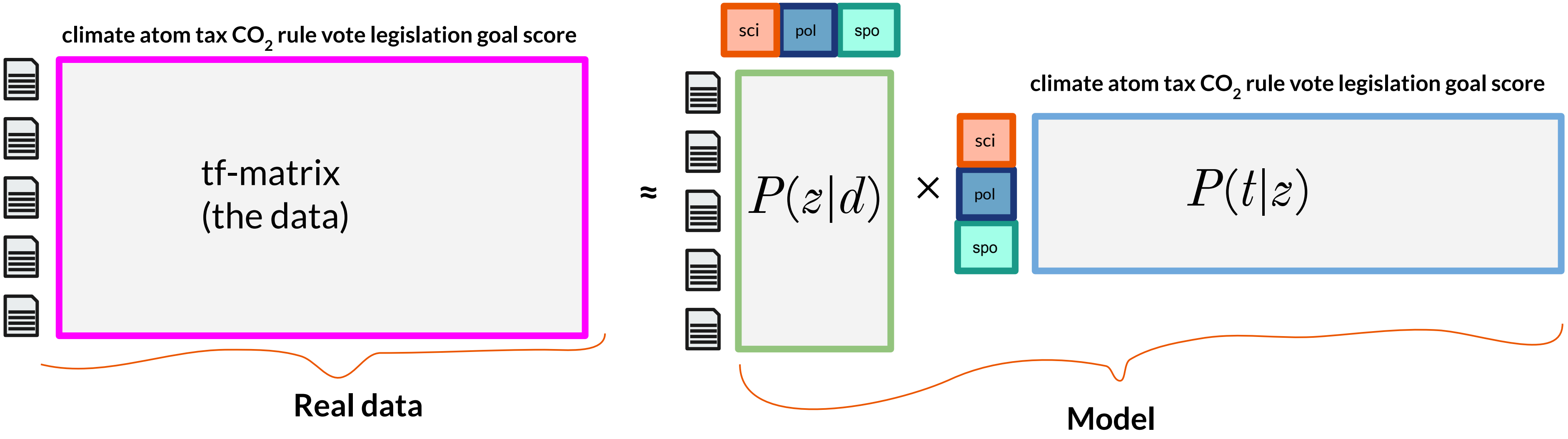
t - term/token/word; d - document; z - topic

Learning the model

$[P(t|z)$ & $P(z|d)]$

There are **several ways** to learn an LDA model, but the general goal is the same.

We want to find $P(t|z)$ & $P(z|d)$ that best approximates the tf-matrix.
I.e. we want to maximize the likelihood of $P(t|z)$ & $P(z|d)$.



Learning
- Generate a fake document

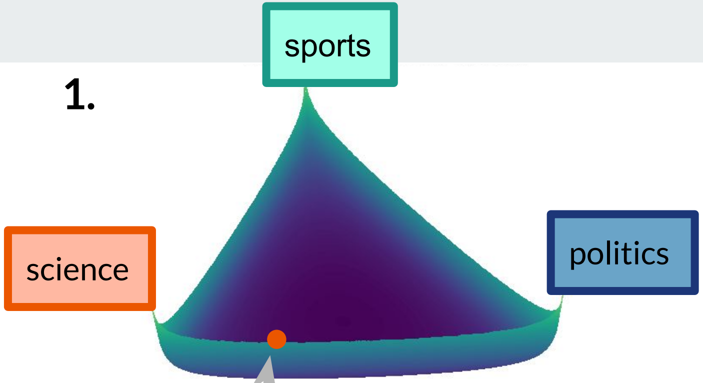
A *generative* model

Generate a fake document (doc 1):

1. Pick a random point in the Dirichlet distribution over topics → topic distribution for the fake doc. E.g. $P(z|d\text{=}1) = [0.8\ 0.2\ 0]$
2. Based on this topic distribution, sample one topic per word in the fake doc. E.g. doc 1 has 5 words.
3. For each topic pick a random point in the Dirichlet distribution over words (not shown). E.g. $P(t|z\text{=}sci) = [0.2\ 0.3\ 0.1\ 0.2\ 0.1\ 0.025\ 0.025\ 0.025\ 0.025]$
4. Populate the fake doc with words by sampling the topic vector from 3), according to $P(t|z\text{=}sci)$

Fake document (doc 1)

1.



2.

science, science, politics, science, science

3. *The Dirichlet distribution over words is not shown because the example uses $|V| = 9$, which requires a 9-dimensional simplex (hard to show in 2D).*

4.

science, science, politics, science, science

0.2 0.3 0.4 0.2 0.2

CO_2, atom, legislation, CO_2, climate

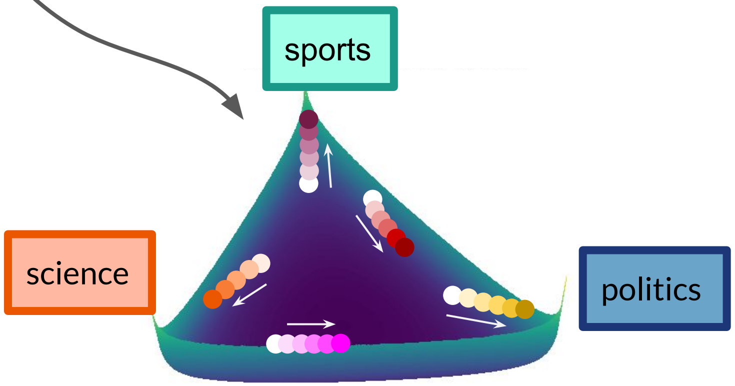
Learning

- A rough overview

5 randomly initialized topic distributions that are iteratively improved until they generate fake documents that are similar to the real documents.

Iterative learning algorithm

- Random initialize n *topic distributions* (i.e. points in the Dirichlet space) $[P(z|d)]$.
- Randomly initialize k fake *word distributions* $[P(t|z)]$.
- Sample the topic & word distributions to populate the fake docs with words.
- Compare the fake docs with the real docs.
- Update topic & words distributions to make the fake documents more similar to the real doc.



Word distributions are not shown because the example uses $|V| = 9$, which requires a 9-dimensional simplex.

Question: What is a trivial example of when assumption 1 fails?

LDA

- Assumptions

1. Documents covering similar topics contain similar words.
2. Documents are probability distributions over latent topics.
3. Topics are probability distributions over words.

Questions

Question: What is a topic in LDA?

Question: How do we know how many topics we have in our corpus?

An application of LDA

Measuring document similarity

Task

- Assume that we have a big corpus of thousands of documents.
- We want to recommend documents based on similar topics.

Document similarity

Dataset for model fitting

- E.g. Wikipedia articles.

Pre-processing

- Remove meta-data, but retain articles
- Remove too common & uncommon words (e.g. retain the middle 90%).
- Text representation
  - TF-IDF
  - N-grams

Fit the LDA model

- Evaluate
- Tune hyper parameters.

Use a similarity metric to compare documents

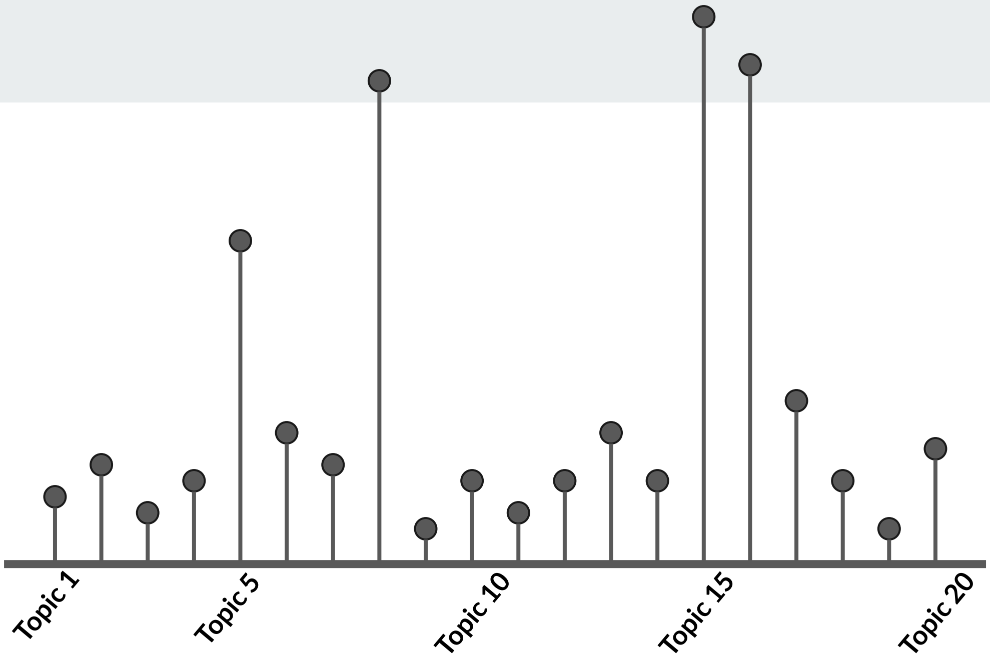
- Jensen-Shannon distance

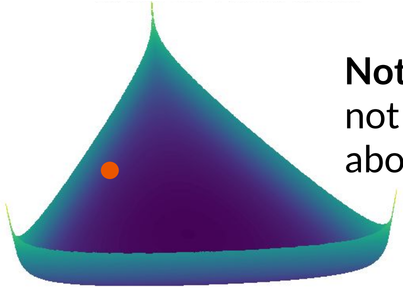
The model's output

For each document the model returns a vector where each value represents the fraction of words from a particular topic (weighted by the probability of the words belonging to that topic).

All values in the vector sum to 1.

It is a probability distribution over topics.





Note: illustration is 3-dim, & not 20-dim as the vector above.

Similarity metric

Measure the distance between probability distributions.

Jensen-Shannon divergence

- Value between 0 & 1.
- 0 indicates that the two distributions are the same.
- 1 indicates that they are completely different.

$$JSD(P_{doc1} \parallel P_{doc2}) = \sqrt{\frac{D_{KL}(P_{doc1} \parallel M) + D_{KL}(P_{doc2} \parallel M)}{2}}$$

D_{KL} - Kullback-Leibler Divergence

$$M = \frac{P_{doc1} + P_{doc2}}{2}$$

Evaluation

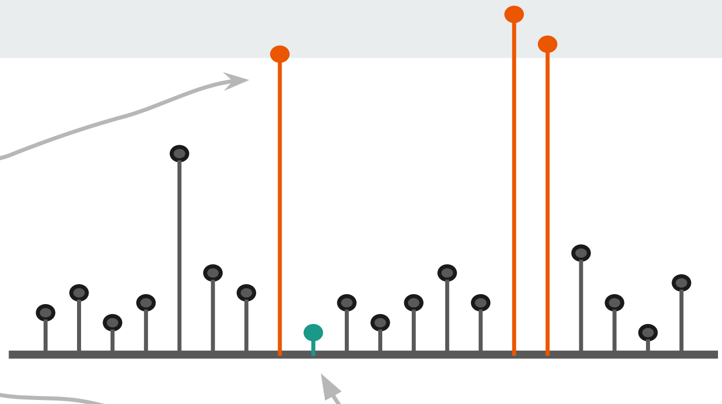
Evaluation
- Unsupervised learning

Automatic evaluation

- *Intuitively*: words from the same topic should occur in the same documents.
- Evaluate either on the training corpus (*intrinsic evaluation*) or on an external corpus like Wikipedia (*extrinsic evaluation*).

Manual evaluation

- Human subjects are used to evaluate the quality of topic & word distributions.



Manual evaluation

Word intrusion

For a given topic

- Pick the 5 most probable words.
- Pick 1 of the least probable words that is also a top word in another topic.
- Ask a (human) subject to pick out the intruding word.

Topic intrusion

For a given document

- Pick the 3 most probable topics (actually, the most probable words from those topics).
- Pick 1 low probability topic (most probable words from the low-probability topic).
- Show subjects a snippet of the document & ask them to pick the intruding (low-probability) topic.

Manual evaluation

Question: What are the intruding words?

Question: Which is the intruding topic?

Word intrusion

floppy alphabet computer processor memory disk
molecule education study university school student
linguistics actor film comedy director movie
islands island bird coast portuguese mainland

Score: Model Precision (MP) - the fraction of subjects agreeing with the model

Topic intrusion

"Douglas Richard Hofstadter (born February 15, 1945 in New York, New York) is an American academic whose research focuses on consciousness, thinking and creativity. He is best known for ", first published in"

student school study education research university science learn
human life scientific science scientist experiment work idea
play role good actor star career show performance
write work book publish life friend influence father

Score: Topic Log Odds (TLO) - conceptually similar, but more complicated

Automatic evaluation
- Topic coherence

Sum pairwise scores

$$Coherence = \sum_{i<j} score(w_i, w_j)$$

Number of docs containing word w_i: $D(w_i)$
Number of docs containing both w_i & w_j: $D(w_i, w_j)$
Total number of docs in corpus: D

Pairwise scores of the words $w_1, w_2, \ldots, w_n$ used to describe a topic, usually the top n words.

Extrinsic UCI score

$$score_{UCI} = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}$$

$$p(w_i) = \frac{D_{\text{Wikipedia}}(w_i)}{D_{\text{Wikipedia}}}$$

$$p(w_i, w_j) = \frac{D_{\text{Wikipedia}}(w_i, w_j)}{D_{\text{Wikipedia}}}$$

Intrinsic UMass score

$$score_{UMass}(w_i, w_j) = \log \frac{D(w_i, w_j) + 1}{D(w_i)}$$

Question: Suggest one way of addressing the issue with word order for BOW-based LDA models.

Shortcomings of LDA

Limitations of BoW

- Word order is lost.
  - Important for short documents, e.g. single sentences.
- Word representation.
  - Words relatedness, e.g. synonyms & antonyms, are not represented.

Alternatives

- Topic Modeling in Embedding Spaces
- Hybrid lda2vec

Tutorial

`MMAI5400_class05_evalLDA.ipynb`