

06 – Topic identification, topic modelling

IA161 Advanced Techniques of Natural Language Processing

Adam Rambousek

NLP Centre, FI MU, Brno

November 12, 2020

Outline

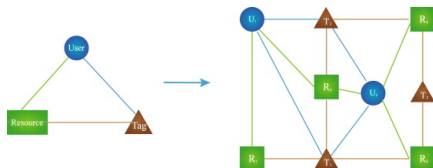
- Introduction to topic modelling
- Latent Semantic Analysis
- Latent Dirichlet Allocation
- Gensim

Topic modelling

- **organize and understand** large collections of documents
- text mining
- discover **topical patterns** in documents
- **topic** – group of words representing the information
- applications
 - ▶ recommender systems
 - ▶ document/book classification
 - ▶ bio-informatics (interpret biological data)
 - ▶ opinion/sentiment analysis
 - ▶ chatbots, topic tracking
 - ▶ text categorization

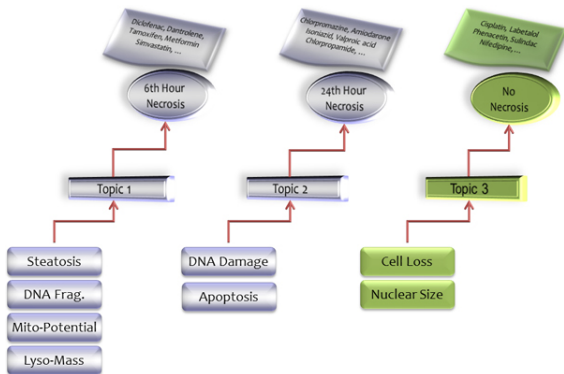
Recommender systems

- recommend the best product for user
- clusters of users, based on preference
- clusters of products
- Netflix prize



Bio-informatics

- categorize patients to risk groups, based on text protocols
- detect common genomic features, based on gene sequence data
- group drugs by diagnosis



Latent Semantic Analysis

- **vector representation** of documents
- compare by vector distance
- **document** = bag of words
- **topic** = set of words
- applications:
 - ▶ data clustering, document classification
 - ▶ term relations (synonymy, polysemy)
 - ▶ cross language document retrieval
 - ▶ word relations in text
 - ▶ similarity in multi choice questions
 - ▶ prior art in patents

LSA – step 1

- count **term-document matrix** (word frequency in documents)
- rows = words, columns = documents
- *sparse matrix*

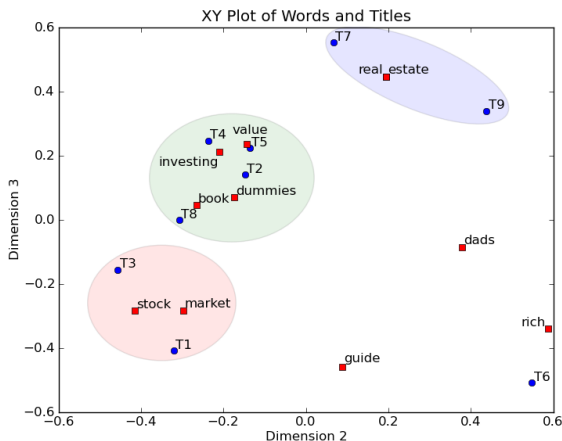
Terms	Documents													
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14
abnormalities	0	0	0	0	0	0	0	1	0	1	0	0	0	0
age	1	0	0	0	0	0	0	0	0	0	0	1	0	0
behavior	0	0	0	0	1	1	0	0	0	0	0	0	0	0
blood	0	0	0	0	0	0	0	1	0	0	1	0	0	0
close	0	0	0	0	0	0	1	0	0	0	1	0	0	0
culture	1	1	0	0	0	0	0	1	1	0	0	0	0	0
depressed	1	0	1	1	1	0	0	0	0	0	0	0	0	0
discharge	1	1	0	0	0	1	0	0	0	0	0	0	0	0
disease	0	0	0	0	0	0	0	0	1	0	1	0	0	0
fast	0	0	0	0	0	0	0	0	0	1	0	1	1	1
generation	0	0	0	0	0	0	0	0	1	0	0	0	1	0
oestrogen	0	0	1	1	0	0	0	0	0	0	0	0	0	0
patients	1	1	0	1	0	0	0	1	0	0	0	0	0	0
pressure	0	0	0	0	0	0	0	0	0	0	1	0	0	1
rats	0	0	0	0	0	0	0	0	0	0	0	0	1	1
respect	0	0	0	0	0	0	0	1	0	0	0	1	0	0
rise	0	0	0	1	0	0	0	0	0	0	0	0	0	1
study	1	0	1	0	0	0	0	0	1	0	0	0	0	0

LSA – step 2

- **weighting** matrix elements
- most popular **tf-idf**
- term occurring in many documents is not interesting for analysis

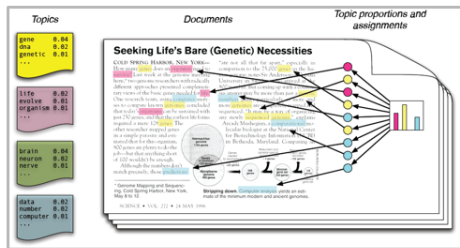
LSA – step 3

- Singular Value Decomposition
- matrix factorization (reduce dimensions, throw away noise)
- cluster close vectors (documents and terms)



Latent Dirichlet Allocation

- statistical model
- each document is a **mix of topics**
- LDA discovers topics and their ratio
- each word in document was **generated** by one of the topics
- applications:
 - ▶ topic relations
 - ▶ content recommendation
 - ▶ group/community overlapping
 - ▶ document topic changes
 - ▶ genetics (ancestral populations)



Example

Document 1: I like to eat **broccoli** and **bananas**.

Document 2: I ate a **banana** and spinach smoothie for **breakfast**.

Document 3: **Chinchillas** and **kittens** are **cute**.

Document 4: My sister adopted a **kitten** yesterday.

Document 5: Look at this **cute hamster munching** on a piece of **broccoli**.

Example

Topic A: 30% broccoli, 15% bananas, 10% breakfast, 10% munching

Topic B: 20% chinchillas, 20% kittens, 20% cute, 15% hamster

Example

Document 1 and 2: 100% Topic A

Document 3 and 4: 100% Topic B

Document 5: 60% Topic A, 40% Topic B

LDA process

- pick fixed number of topics
- for each document, randomly assign topic to each word
- improve, for each document d :
 - ▶ for each word w and topic t count:
 - ▶ *all topic assignments are correct, except for current word*
 - ▶ $p(\text{topic } t | \text{document } d)$ – how many words in document have topic?
 - ▶ $p(\text{word } w | \text{topic } t)$ – how many assignments to topic for word?
 - ▶ new topic: probability $p(\text{topic } t | \text{document } d) \times p(\text{word } w | \text{topic } t)$
- repeat and reach almost steady state

Topic Labeling

represent topic with human-friendly label

- top N words from the list
- find Wikipedia article based on word list
- document summarization from topic documents

Gensim

```
>>> from gensim import corpora, models, similarities
>>>
>>> # Load corpus iterator from a Matrix Market file on disk.
>>> corpus = corpora.MmCorpus('/path/to/corpus.mm')
>>>
>>> # Initialize Latent Semantic Indexing with 200 dimensions.
>>> lsi = models.LsiModel(corpus, num_topics=200)
>>>
>>> # Convert another corpus to the Latent space and index it.
>>> index = similarities.MatrixSimilarity(lsi[another_corpus])
>>>
>>> # Compute similarity of a query vs. indexed documents
>>> sims = index[query]
```

Gensim – LSA

```
gensim.models.lsimodel.LsiModel(corpus=None,  
num_topics=200, id2word=None, chunksize=20000, decay=1.0,  
distributed=False, onepass=True, power_iters=2,  
extra_samples=100)
```



- `chunksize` – number of documents in memory (more documents, more memory)
- `decay` – newly added documents are more important?
- `power_iters` – more iterations improve accuracy, but lower performance
- `onepass` – False to use multi-pass algorithm, for static data increase accuracy

Gensim – LDA

```
gensim.models.ldamodel.LdaModel(corpus=None,  
num_topics=100, id2word=None, distributed=False,  
chunksize=2000, passes=1, update_every=1,  
alpha='symmetric', eta=None, decay=0.5, offset=1.0,  
eval_every=10, iterations=50, gamma_threshold=0.001,  
minimum_probability=0.01, random_state=None, ns_conf=None,  
minimum_phi_value=0.01, per_word_topics=False)
```

- `chunksize` – number of documents in memory (more documents, more memory)
- `update_every` – number of chunks before moving to next step
- `chunksize=100k, update_every=1` equals to `chunksize=50k, update_every=2` (saves memory)
- `decay` – newly added documents are more important?
- `alpha, eta` – preset expected topics and word probability for start
- `eval_every` – log perplexity is estimated after `x` updates (lower number, slower training)

References I

-  Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003).
Latent Dirichlet Allocation.
Journal of Machine Learning Research, 3:993 – 1022.
-  Castellanos, A., Cigarrn, J., and Garca-Serrano, A. (2017).
Formal concept analysis for topic detection.
Inf. Syst., 66(C):24–42.
-  Dumais, S. T., Furnas, G. W., Landauer, T. K., Deerwester, S., and Harshman, R. (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, pages 281–285, New York, NY, USA. ACM.

References II



Lim, K. H., Karunasekera, S., and Harwood, A. (2017).

Clustop: A clustering-based topic modelling algorithm for twitter using word networks.

In *2017 IEEE International Conference on Big Data (Big Data)*, pages 2009–2018. IEEE.



Lü, L., Medo, M., Yeung, C. H., Zhang, Y.-C., Zhang, Z.-K., and Zhou, T. (2012).

Recommender systems.

Physics Reports, 519(1):1 – 49.

Recommender Systems.



Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006).

Hierarchical Dirichlet processes .

Journal of the American Statistical Association, 101:1566 – 1581.

References III



Wan, X. and Wang, T. (2016).

Automatic labeling of topic models using text summaries.

In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2297–2305.



Xie, P. and Xing, E. P. (2013).

Integrating document clustering and topic modeling.

CoRR, abs/1309.6874.