06 – Topic identification, topic modelling

IA161 Advanced Techniques of Natural Language Processing

Adam Rambousek

NLP Centre, FI MU, Brno

October 23, 2019

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

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*

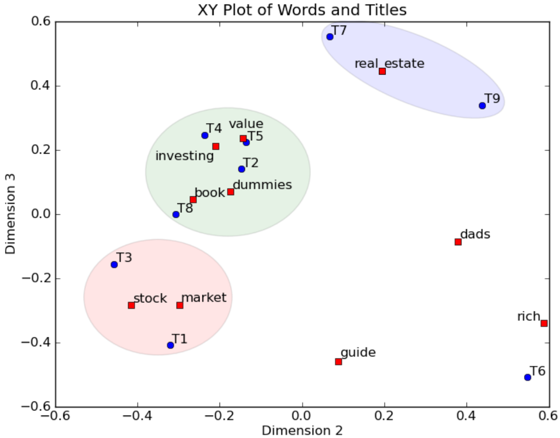
Terms	Documents													
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14
abnormalities	0	0	0	0	0	0	0	1	0	1	0	0	0	0
age	1	0	0	0	0	0	0	0	0	0	0	1	0	0
behavior	0	0	0	0	1	1	0	0	0	0	0	0	0	0
blood	0	0	0	0	0	0	0	1	0	0	1	0	0	0
close	0	0	0	0	0	0	1	0	0	0	1	0	0	0
culture	1	1	0	0	0	0	0	1	1	0	0	0	0	0
depressed	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 occuring 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*
    - $p(\textit{topic } t | \textit{document } d)$ – how many words in document have topic?
    - $p(\textit{word } w | \textit{topic } t)$ – how many assignments to topic for word?
    - new topic: probability $p(\textit{topic } t | \textit{document } d) \times p(\textit{word } w | \textit{topic } t)$
- repeat and reach almost steady state

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]
```

References I

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003).
Latent Dirichlet Allocation.
Journal of Machine Learning Research, 3:993 – 1022.

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.

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.