Topic modeling and identification

El Habib NFAOUI (elhabib.nfaoui@usmba.ac.ma)
Faculty of Sciences Dhar Al Mahraz, Fes
Sidi Mohamed Ben Abdellah University, Fes

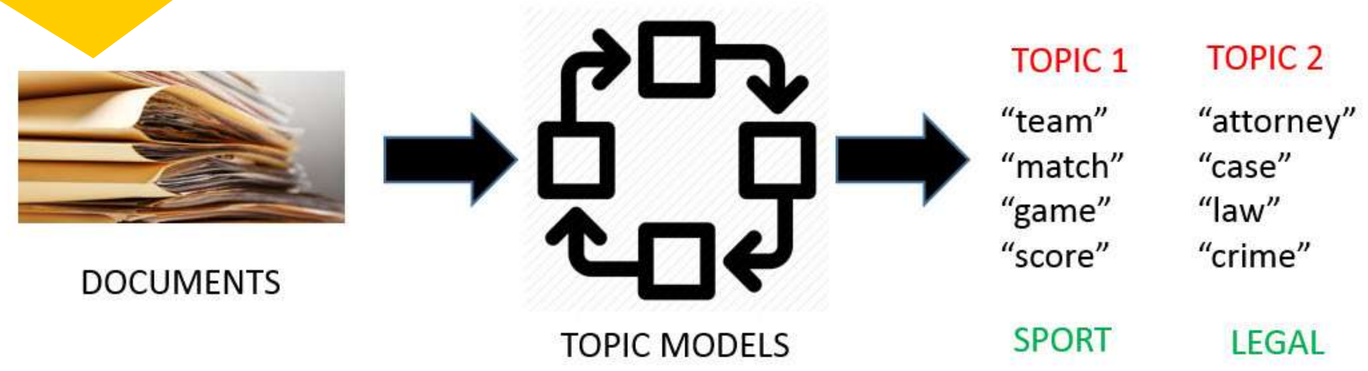
2020-2021

Outline

1. Introduction to Topic Modeling
2. Topic Models
3. Latent Dirichlet Allocation (LDA)

Large text could be feeds from social media, customer reviews of hotels, movies, user feedbacks, news stories, e-mails of customer complaints etc.

opic Modeling



- Topic models are essentially iterative algorithms that work with **document feature matrices**, to use overlapping features to group documents together. Features could simply be all the words in a sentence, or selected features such as nouns or named entities, and so on. To explain in a simplistic manner, we imagine that we have a corpus of documents of mixed subjects and we use words as features to represent a document. If we had to analyze these documents using topic models, and the **topic model would group words like** **"team"**, **"match"**, **"game"**, and **"score"** in a single topic (as these word frequently appear together) which we would name as a **SPORT** topic, while words like **"attorney"**, **"case"**, **"law"**, and **"crime"** in another topic that we would name as a **LEGAL** topic. This example shows that the documents that we used for topic models were essentially containing two topics, **SPORT** and **LEGAL**, as you can see in the following illustration:

2. Topic Models

- Two of the best known topic models are:
  - **Latent Semantic Analysis (LSA)**
  - **Latent Dirichlet Allocation (LDA)**.

LDA is proven to be better suited for clustering with high dimensionality and more accurate in identifying topics than LSA.

- Another topic model is Hierarchical Dirichlet Process (HDP).

3. Latent Dirichlet Allocation

- **Latent Dirichlet Allocation** (**LDA**) is an unsupervised generative model that **assigns topic distributions to documents**. It **assumes each document is a mixture of topics and in turn, each topic is a mixture of words**.

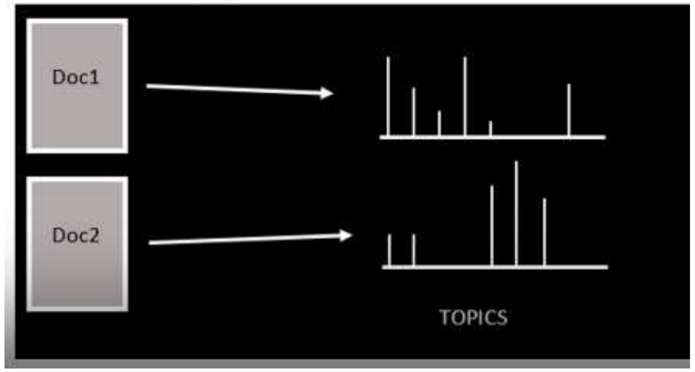
In other words:

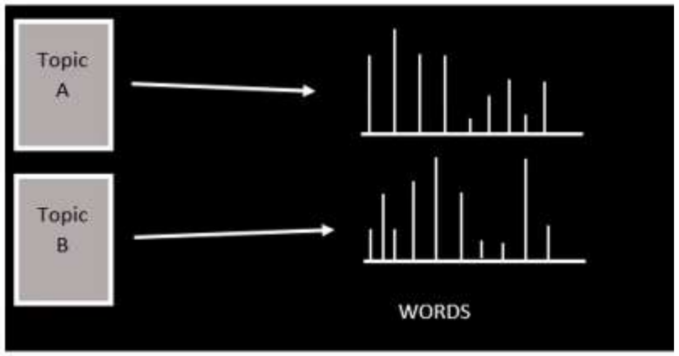
- Documents are probability distributions over latent topics.
- Topics are probability distributions over words.

After training, each document will have a discrete distribution over all topics, and each topic will have a discrete distribution over all words.

- When we say topic, what is it actually and how it is represented?

A topic is a collection of **dominant keywords** that are **typical representative**. Just by looking at the keywords, you can identify what the topic is all about.





3.1 Example

For instance, consider the two following documents describing two companies:

doc1:	**doc2**:
Changing how people search for fashion items and, share and buy fashion via visual recognition, TRUELIFE is going to become the best approach to search the ultimate trends …	Cinema4you enabling any venue to be a cinema is a new digital filmed media distribution company currently in the testing phase. It applies technologies used in Video on Demand and broadcasting to ...

LDA is a way of automatically discovering latent topics (**latent = present but not visible**) that these documents contain. For example, given these documents and asked for two topics, LDA might return the following words associated with each topic:

- **topic 1**: people Video fashion media Cinema …

  Probability of each word
  0.163 * "people" + 0.122 * "Video " + 0.09 * "fashion" + 0.08 * "media" + 0.02 * " Cinema" …

- **topic 2**: Cinema technologies recognition broadcasting…

Therefore, the first topic can be labeled as "***business"*** while the second as "***technology"***.

3.1 Example

Documents are then represented as **mixtures of topics** that spit out words with **certain probabilities**:

- **doc1**: topic 1 (0,36), topic 2 (0,64)
- **doc2**: topic 1 (0,21), topic 2 (0,79)

An important point to note:

Although I have named two topics in the example above, the model itself does not actually **do any "naming"** or classifying of topics. But by visually inspecting the top contributing words of a topic (i.e. the discrete distribution over words for a topic), one can name the topics if necessary after training.

3.2 Model

- Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus. This generative model posits (assumes) that the characteristics of topics and documents are drawn from Dirichlet distributions.

3.2.1 The Dirichlet Distribution

- Lda's generative model posits (assumes) that the characteristics of topics and documents are drawn from Dirichlet distributions.
- The Dirichlet distribution is the multivariate generalization of the beta distribution. The Dirichlet distribution's probability density function is defined as (see for example Blei and Laerty, 2009):

$$p(x|\alpha_1, \ldots, \alpha_K) = \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} x_i^{\alpha_i - 1}$$

With α being a positive K-vector and Γ denoting the Gamma function, which is generalization of the factorial function to real values

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t}\, dt.$$

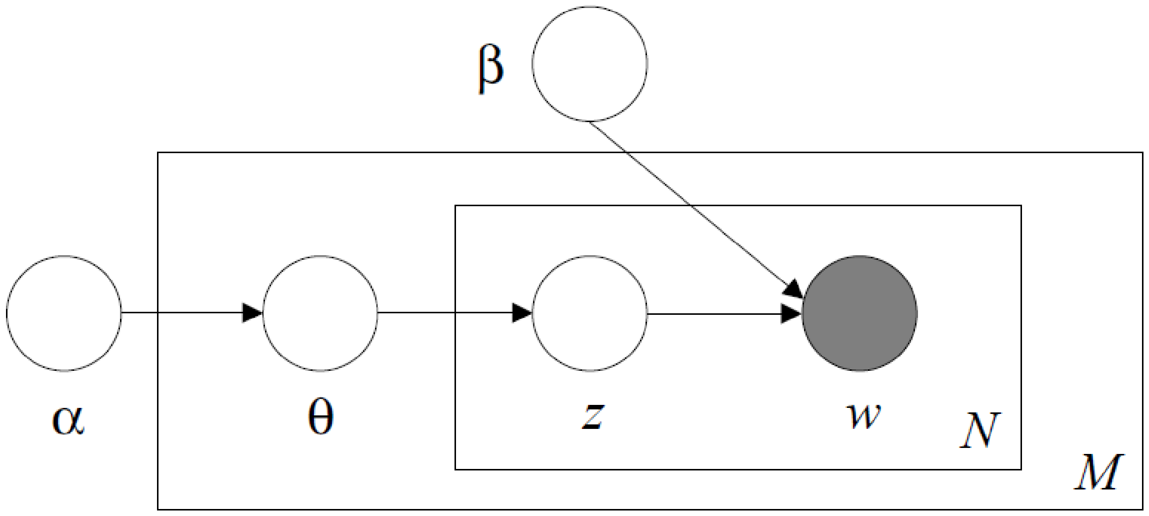
3.2.2 Generative Process

- Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words.
- The Lda model involves drawing samples from **Dirichlet distributions** and from **multinomial distributions**. A generalization of the binomial distribution, the multinomial distribution tells us the probability of observing a count of two or more independent events, given the number of draws and fixed probabilities per outcome that sum to one. In our case the outcomes are terms and topics.
- LDA assumes the following generative process for each document in a corpus:

1. Choose $N \sim \text{Poisson}(\xi)$.
2. Choose $\theta \sim \text{Dir}(\alpha)$.
3. For each of the N words w_n:
   (a) Choose a topic $z_n \sim \text{Multinomial}(\theta)$.
   (b) Choose a word w_n from $p(w_n \mid z_n, \beta)$, a multinomial probability conditioned on the topic z_n.

3.2.3 Plate notation

- The following figure shows LDA in plate notation, where rectangles signify a repetition over enclosed nodes. Plate notation is a concise way of visually representing the relationship among documents, topics, and words.



- **Terminology and Notation**
  - A *word* is the basic unit of discrete data, defined to be an item from a vocabulary indexed by {1,…,*V*}.
  - A *document* is a sequence of ***N*** words
  - A *corpus* is a collection of ***M*** documents

- **Parameters**

α	is the parameter of the Dirichlet prior on the per-document topic distributions,
β	is the parameter of the Dirichlet prior on the per-topic word distribution,
θ_m	is the topic distribution for document m,
z_{mn}	is the topic for the n-th word in document m, and
w_{mn}	is the specific word.

3.2.3 Plate notation

- The **topics distribution per document**, which is drawn from the Dirichlet distribution, given the Dirichlet parameter α which is K-vector with components $\alpha_k > 0$:

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \theta_1^{\alpha_k - 1} \cdots \theta_K^{\alpha_K - 1} = \frac{\Gamma(\alpha.)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1}$$

- **LDA has 2 parameters: alpha (α) and beta (*β*)**

alpha - A low value for α means that documents have only a low number of topics contributing to them. A high value of α yields the inverse, meaning that each document is likely to contain a mixture of most of the topics.

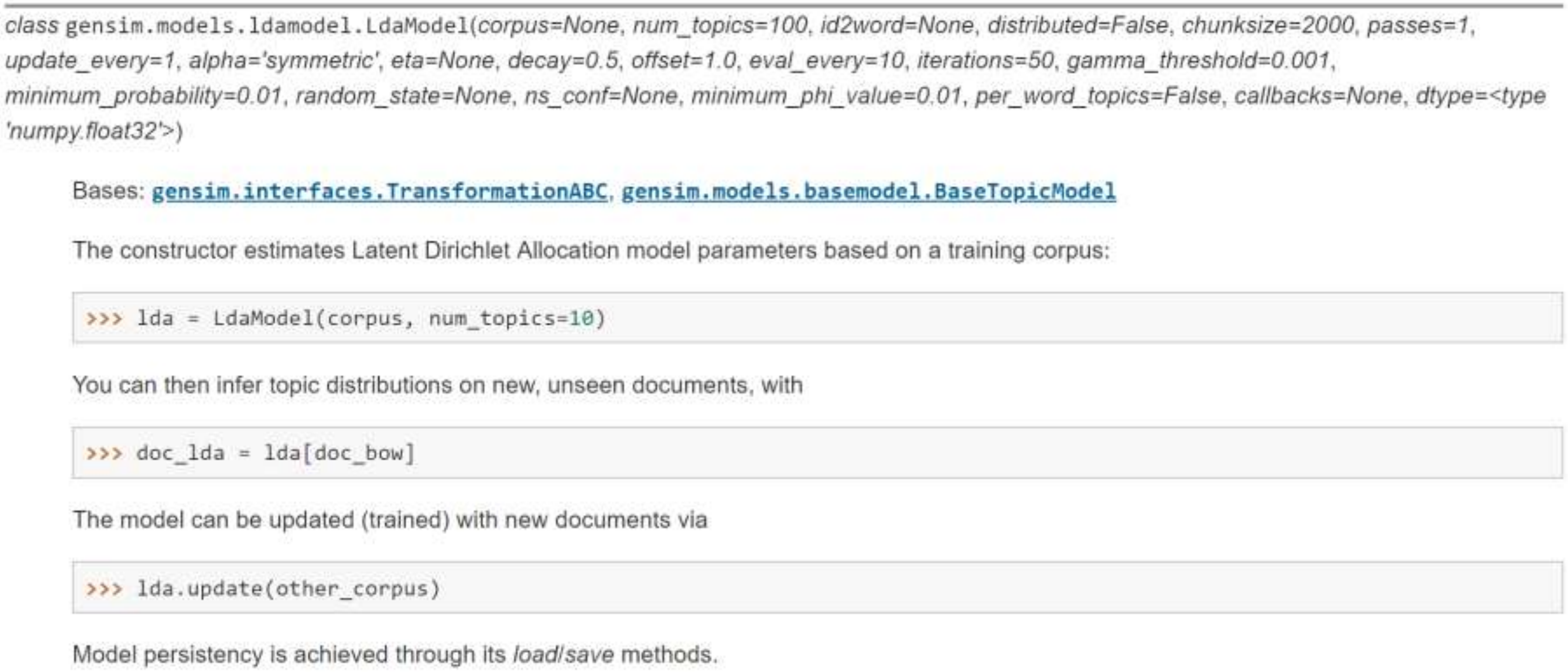
beta - A low value for beta means the topics have a low number of contributing words. A high value for beta means the topics have a high number of contributing words.

The values of alpha and beta really depend on the application, and may need to be tweaked several times before the desired results are found. Even then, LDA is non-deterministic since parameters are randomly initialized, so the outcome of any run of the model can never be known in advance.

3.3 Implementation: gensim Python Library

models.ldamodel – Latent Dirichlet Allocation

https://radimrehurek.com/gensim/models/ldamodel.html



class `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, callbacks=None, dtype=<type 'numpy.float32'>*)

Bases: `gensim.interfaces.TransformationABC`, `gensim.models.basemodel.BaseTopicModel`

The constructor estimates Latent Dirichlet Allocation model parameters based on a training corpus:

```
>>> lda = LdaModel(corpus, num_topics=10)
```

You can then infer topic distributions on new, unseen documents, with

```
>>> doc_lda = lda[doc_bow]
```

The model can be updated (trained) with new documents via

```
>>> lda.update(other_corpus)
```

Model persistency is achieved through its *load*/*save* methods.

Hands-on exercises

- Train an LDA model using a small dataset
  - File: **lda_basics_0.ipynb**

3.4 Setting up a good topic modeling system

- The following are key factors to obtaining good segregation topics:
  - The quality of processed text data.

  This is the single most important step in setting up a **good topic modeling system**. If the preprocessing is not good, the algorithm can't do much since we would be feeding it a lot of noise. We can filter out the noise using the following steps :
    - Stopword removal,
    - Bigram collocation detection (frequently co-occuring tokens). This is an attempt to find some hidden structure in the corpus. You can even try trigram collocation detection.
    - Lemmatization to only keep the nouns. **Lemmatization is generally better than stemming in the case of topic modeling since the words after lemmatization still remain understandable**. However, generally stemming might be preferred if the data is being fed into a vectorizer and isn't intended to be viewed.
  - The variety of topics the text talks about.
  - The choice of topic modeling algorithm.
  - The number of topics fed to the algorithm.
  - The algorithms tuning parameters.

Preprocessing our data. Remember: Garbage In Garbage Out

"NLP is 80% preprocessing."

-Lev Konstantinovskiy

3.5 Notes and Uses of LDA

- LDA does not work well with very short documents, like twitter feeds. Very briefly, this is because the model infers parameters from observations and if there are not enough observations (words) in a document, the model performs poorly. For short texts, it may be best to use other techniques such as biterm model (A biterm topic model for short texts [Xiaohui Yan et al., 2013]

- Unlike the word2vec algorithm, which performs extremely well with full structured sentences, LDA is a bag of words model, meaning word order in a document doesn't count. This also means that **stopwords** and **rare words** should be excluded, so that the model doesn't overcompensate for very frequent words and very rare words, both of which do not contribute to general topics.

References

Siddhartha Chatterjee and Michal Krystyanczuk. Python Social Media Analytics. July 2017, ISBN 978-1-78712-148-5. Packt Publishing Ltd.

Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the Space of Topic Coherence Measures. In Proceedings of the Eighth ACM International Conference on Web Search and Data Mining (WSDM '15). 2015, ACM, New York, NY, USA, 399-408. DOI: http://dx.doi.org/10.1145/2684822.2685324

Martin Ponweiser. Latent Dirichlet Allocation in R. Diploma Thesis. May 2012. Institute for Statistics and Mathematics, Vienna University of Business and Economics

Isoni Andrea. Machine Learning for the Web. Pub. Date: 2016, pages: 299, ISBN: 978-1-78588-660-7. Publisher: Packt Publishing

https://www.machinelearningplus.com/nlp/topic-modeling-gensim-python/

https://markroxor.github.io/gensim/static/notebooks/gensim_news_classification.html

https://rare-technologies.com/what-is-topic-coherence/

https://www.kaggle.com/ktattan/lda-and-document-similarity