Topic Modelling

Tuesday, September 18, 2018 12:49 PM

https://github.com/chibueze07/Machine-Learning-In-Law/blob/master/project.ipynb

Pdfminer for Python 3

```
pip install pdfminer.six
```

From <https://stackoverflow.com/questions/39854841/pdfminer-python-3-5>

https://towardsdatascience.com/topic-modeling-and-latent-dirichlet-allocation-in-python-9bf156893c24

https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05

4 of the most popular techniques today: LSA, pLSA, LDA, and the newer, deep learning-based lda2vec.

From <https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05>

All topic models are based on the same basic assumption:

- each **document** consists of a mixture of ***topics***, and
- each ***topic*** consists of a collection of **words**.

LSA:-

Consequently, LSA models typically replace raw counts in the document-term matrix with a **tf-idf score**. Tf-idf, or term frequency-inverse document frequency, assigns a weight for term j in document i as follows:

Intuitively, the more frequently the term appears in the document, the smaller its weight, and the more *infrequently* it appears across the corpus, the greater its weight.

This dimensionality reduction can be performed using **truncated SVD.**

singular value decomposition

With these document vectors and term vectors, we can now easily apply measures such as cosine similarity to evaluate:

- the similarity of different documents
- the similarity of different words
- the similarity of terms (or "queries") and documents (which becomes useful in information retrieval, when we want to retrieve passages most relevant to our search query).

##333

Document 1: I had a peanut butter sandwich for breakfast.

Document 2: I like to eat almonds, peanuts and walnuts.

Document 3: My neighbor got a little dog yesterday.

Document 4: Cats and dogs are mortal enemies.

Document 5: You mustn't feed peanuts to your dog.

The LDA model discovers the different topics that the documents represent and how much of each topic is present in a document. For example, LDA may produce the following results:

Topic 1: 30% peanuts, 15% almonds, 10% breakfast… (you can interpret that this topic deals with food)

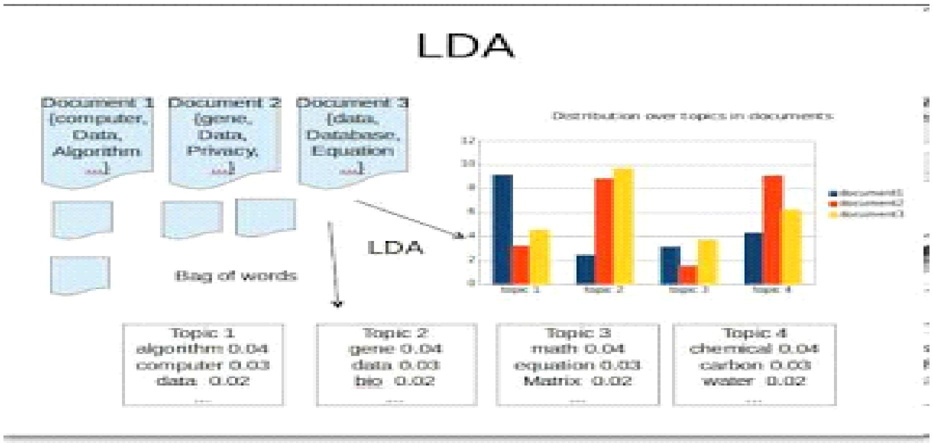
Topic 2: 20% dogs, 10% cats, 5% peanuts… (you can interpret that this topic deals with pets or animals)

Documents 1 and 2: 100% Topic 1

Documents 3 and 4: 100% Topic 2

Document 5: 70% Topic 1, 30% Topic 2

So, how does LDA perform this process?



Collapsed Gibbs sampling is one way the LDA learns the topics and the topic representations of each document. The procedure is as follows:

- Go through each document and randomly assign each word in the document to one of K topics (K is chosen beforehand)
- This random assignment gives topic representations of all documents and word distributions of all the topics, albeit not very good ones

- So, to improve upon them:
- For each document *d*, go through each word *w* and compute:
- p(topic t | document d): proportion of words in document *d* that are assigned to topic *t*
- p(word w| topic t): proportion of assignments to topic t, over all documents d, that come from word w

- Reassign word w a new topic t', where we choose topic t' with probability
  p(topic t' | document d) * p(word w | topic t')
  This generative model predicts the probability that topic t' generated word w
  On repeating the last step a large number of times, we reach a steady state where topic assignments are pretty good. These assignments are then used to determine the topic mixtures of each document.

From <https://www.kdnuggets.com/2016/07/text-mining-101-topic-modeling.html>

http://blog.echen.me/2011/08/22/introduction-to-latent-dirichlet-allocation/