CAI Lab, Session 7: Topic models

In this session, we will apply and compare the three topic models we have seen in class (LSA, pLSA, and LDA) to a collection of text documents. The aim of this lab is to be able to apply, and then interpret the kinds of topics we get out of each method. Part of this code, especially the preprocessing part is based on this blog.

The python jupyter notebook provided `topicmodels.ipynb` contains all the code for loading and preprocessing the corpus we will be working on (`20_newsgroups`), as well as the code for training the topic models. In the case of LSA, we use scipy's `SVD` function, for LDA we use the very popular `gensim` package, and for pLSA we code the EM algorithm directly from scratch. The derivation below should serve as the basis for understanding this EM implementation.

pLSA

The code that we provide uses the asymmetric formulation of the model, with paramters $\theta_{d,k}$ being the topic proportions $P(z=k|d)$ and $\beta_{k,w}$ being the word distribution for k-th topic $P(w|z=k)$.

$$P(d,w) = P(d)P(w|d) = P(d)\sum_{z\in\mathcal{Z}} P(z|d)P(w|z) = P(d)\sum_k \theta_{d,k}\beta_{k,w} = P(d)\theta_{d,\cdot}\beta_{\cdot,w}$$

In the asymmetric formulation, the log-likelihood is:

$$\begin{aligned}
\mathcal{L} &= \sum_{d\in\mathcal{D}}\sum_{w\in\mathcal{W}} n(d,w)\log P(d,w) \\
&= \sum_{d\in\mathcal{D}}\sum_{w\in\mathcal{W}} n(d,w)\left(\log P(d) + \log\sum_z P(z|d)P(w|z)\right) \\
&= \sum_{d\in\mathcal{D}}\sum_{w\in\mathcal{W}} n(d,w)\log P(d) + \sum_{d\in\mathcal{D}}\sum_{w\in\mathcal{W}} n(d,w)\log\sum_z P(z|d)P(w|z)
\end{aligned}$$

If we further assume that the distribution over documents $P(d)$ is uniform, then in order to maximize the log-likelihood we need to maximize the following expression:

$$\begin{aligned}
\mathcal{L}' &= \sum_{d\in\mathcal{D}}\sum_{w\in\mathcal{W}} n(d,w)\log\sum_z P(z|d)P(w|z) \\
&= \sum_{d\in\mathcal{D}}\sum_{w\in\mathcal{W}} n(d,w)\log\sum_k P(z=k|d)P(w|z=k) \\
&= \sum_{d\in\mathcal{D}}\sum_{w\in\mathcal{W}} n(d,w)\log\sum_k \theta_{d,k}\,\beta_{k,w}
\end{aligned}$$

In this formulation, the E-step becomes (posterior of latent z for every observation pair (d,w)):

$$P(z|d,w) = \frac{P(d)P(z|d)P(w|z)}{\sum_{z'} P(d)P(z'|d)P(w|z')} = \frac{P(z|d)P(w|z)}{\sum_{z'} P(z'|d)P(w|z')}$$

And so, using the variable names from the code: $P(z=k|d,w) \propto \theta_{d,k}\,\beta_{k,w}$

For the M-step, we need to re-estimate θ and β based on the posterior for z in the following way:

$$\begin{aligned}
\beta_{k,w} = P(w|z=k) &= \frac{\sum_d n(d,w)P(z=k|d,w)}{\sum_{w'}\sum_d n(d,w')P(z=k|d,w')} \\
\theta_{d,k} = P(z=k|d) &= \frac{\sum_w n(d,w)P(z=k|d,w)}{\sum_w n(d,w)}
\end{aligned}$$

Things you can try

The idea of this lab session is that you have some code example showing how to apply these models. If you want to try a few things to deepen your understanding, I propose the following:

- Try to apply the three models on some corpus that interests you, and compare models obtained
- Play with the hyperparameter α and see how the topics obtained change. In the code given, this hyperparameter is set to `'auto'` but you can give it different values and see how the results change.
- Explore the visualization of trained LDA models using library `pyldavis`

Deliverables

Nothing to deliver.