1 Latent Dirichlet Allocation

The classical LDA (Blei et al., 2003) assumes distributions of latent topics for each text. If K denotes the total number of modeled topics, the set of topics is given by $\boldsymbol{T} = \{T_1, \ldots, T_K\}$. We define $W_n^{(m)}$ as a single token at position n in text m. The set of possible tokens is given by the vocabulary $\boldsymbol{W} = \{W_1, \ldots, W_V\}$ with $V = |\boldsymbol{W}|$, the vocabulary size. Then, let

$$\boldsymbol{D}^{(m)} = \left(W_1^{(m)}, \ldots, W_{N^{(m)}}^{(m)}\right),$$

be text (or document) $m = 1, \ldots, M$, of a corpus consisting of M texts. Each text in turn consists of $N^{(m)}$ word tokens $W_n^{(m)} \in \boldsymbol{W}$, $n = 1, \ldots, N^{(m)}$. Topics are referred to as $T_n^{(m)} \in \boldsymbol{T}$ for the topic assignment of token $W_n^{(m)}$. Then, analogously the topic assignments of every text m are given by

$$\boldsymbol{T}^{(m)} = \left(T_1^{(m)}, \ldots, T_{N^{(m)}}^{(m)}\right).$$

When $n_k^{(mv)}, k = 1, \ldots, K, v = 1, \ldots, V$ describes the number of assignments of word v in text m to topic k, we can define the cumulative count of word v in topic k over all documents by $n_k^{(\bullet v)}$ and, analogously, the cumulative count of topic k over all words in document m by $n_k^{(m\bullet)}$, while $n_k^{(\bullet\bullet)}$ indicates the total count of assignments to topic k.

Using these definitions, the underlying probability model (Griffiths and Steyvers, 2004) can be written as

$$\begin{aligned}
W_n^{(m)} \mid T_n^{(m)}, \boldsymbol{\phi}_k &\sim \text{Discrete}(\boldsymbol{\phi}_k),\\
\boldsymbol{\phi}_k &\sim \text{Dirichlet}(\eta),\\
T_n^{(m)} \mid \boldsymbol{\theta}_m &\sim \text{Discrete}(\boldsymbol{\theta}_m),\\
\boldsymbol{\theta}_m &\sim \text{Dirichlet}(\alpha).
\end{aligned}$$

For a given parameter set $\{K, \alpha, \eta\}$, LDA assigns one of the K topics to each token. Here K denotes the number of topics and α, η are parameters of a Dirichlet distribution defining the type of mixture of topics in every text and the type of mixture of words in every topic.

Estimators for topic distributions per text $\boldsymbol{\theta}_m = (\theta_{m,1}, \ldots, \theta_{m,K})^T \in (0,1)^K$ and word distributions per topic $\boldsymbol{\phi}_k = (\phi_{k,1}, \ldots, \phi_{k,V})^T \in (0,1)^V$ can be derived through the Collapsed Gibbs Sampler procedure (Griffiths and Steyvers, 2004) by

$$\hat{\theta}_{m,k} = \frac{n_k^{(m\bullet)} + \alpha}{N^{(m)} + K\alpha}, \quad \hat{\phi}_{k,v} = \frac{n_k^{(\bullet v)} + \eta}{n_k^{(\bullet\bullet)} + V\eta}.$$

2 LDAPrototype

The Gibbs sampler in the modeling procedure of LDA is sensitive to the random initialization of topic assignments. To overcome this issue, the selection algorithm LDAPrototype can be used. The method selects the LDA as prototype model of a set of LDAs that maximizes its mean pairwise similarity to all other models (Rieger et al., 2020). Thus, the LDAPrototype method increases the reliability of conclusions drawn from the resulting prototype model. The approach is implemented in the R package ldaPrototype (Rieger, 2020).

3 Similarity Measures

Self-similarities of topics over time (NOTE: In the DMC competition, we are more interested in document similarities rather than topic similarities. This means that our compared vectors have the length K instead of V - and are thus significantly shorter.) are useful as indicators for the stability of topics. Using the notation from Sect. 1 the word count vector for topic $k = 1, \ldots, K$ is given by

$$\boldsymbol{n}_k = \left(n_k^{(\bullet 1)}, \ldots, n_k^{(\bullet V)}\right)^T \in \mathbb{N}_0^V.$$

Extending the notation to account for different temporal aggregations t leads to $\boldsymbol{n}_{k|t}$. Since k is constant within a similarity calculation because self-similarities of topics over time are considered, the notation simplifies to

$$\begin{aligned}
\boldsymbol{n}_{k|t} = \boldsymbol{n}_t &= (n_{t,1}, \ldots, n_{t,V})^T,\\
\boldsymbol{p}_t &= (n_{t,1}, \ldots, n_{t,V})^T / \sum_v n_{t,v}.
\end{aligned}$$

We consider two different types of similarity measures, one based on word count vectors $\boldsymbol{n}_i, \boldsymbol{n}_j$ and one based on word distribution vectors $\boldsymbol{p}_i, \boldsymbol{p}_j$. Then, cosine similarity and a thresholded version of the Jaccard coefficient, respectively, are defined as

$$\text{COS} = \frac{\sum_v n_{i,v} n_{j,v}}{\sqrt{\sum_v n_{i,v}^2}\sqrt{\sum_v n_{j,v}^2}}, \tag{1}$$

$$\text{TJ} = \frac{\sum_v \mathbb{1}_{\{n_{i,v} > c_i \,\wedge\, n_{j,v} > c_j\}}}{\sum_v \mathbb{1}_{\{n_{i,v} > c_i \,\vee\, n_{j,v} > c_j\}}}. \tag{2}$$

The thresholds c_i, c_j may be chosen as an absolute lower bound $c_{\text{abs}} \in \mathbb{N}_0$ or as relative lower bound $c_{\text{rel}} \in [0,1]$ in dependence of $\boldsymbol{n}_t$. A combination of both bounds can be constructed by

$c_t = \max\{c_{\text{abs}}, c_{\text{rel}} \sum_v n_{t,v}\}$. The default value is $c_{\text{rel}} = 0.002$ as proposed by Rieger et al. (2020). The distributional similarity measures based on the Manhattan distance, χ^2 distance, Hellinger distance and Jensen Shannon divergence, respectively, are given by

$$\text{MH} = 1 - \frac{1}{2} \sum_v |p_{i,v} - p_{j,v}|, \tag{3}$$

$$\chi^2 = 1 - \frac{1}{2} \sum_v \frac{(p_{i,v} - p_{j,v})^2}{p_{i,v} + p_{j,v}}, \tag{4}$$

$$\text{HL} = 1 - \sqrt{\frac{1}{2} \sum_v \left(\sqrt{p_{i,v}} - \sqrt{p_{j,v}}\right)^2}, \tag{5}$$

$$\begin{aligned} \text{JS} = 1 &- \sum_v p_{i,v} \log \frac{2p_{i,v}}{p_{i,v} + p_{j,v}} \\ &- \sum_v p_{j,v} \log \frac{2p_{j,v}}{p_{i,v} + p_{j,v}} \end{aligned} \tag{6}$$

For numerical reasons a small value $\epsilon = 10^{-6}$ is added to the word counts $\boldsymbol{n}_t$ before calculating $\boldsymbol{p}_t$ to determine the similarity using χ^2 and JS.

References

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

Thomas L. Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235.

Jonas Rieger. 2020. ldaPrototype: A method in R to get a Prototype of multiple Latent Dirichlet Allocations. *Journal of Open Source Software*, 5(51):2181.

Jonas Rieger, Jörg Rahnenführer, and Carsten Jentsch. 2020. Improving Latent Dirichlet Allocation: On Reliability of the Novel Method LDAPrototype. In *NLDB: Natural Language Processing and Information Systems*, volume 12089 of *LNCS*, pages 118–125. Springer.