1 Discussed models

This section will introduce the reader into the two Topic modeling approaches which will be compared in this Thesis. The aim of both procedures is to assign one or more topics to different documents. Even if the vocabulary and the notation are similar for both approaches, the notation should be resumed at the beginning of the description of the respective model. The basic structural notation of the data consists of the following variables.

A collection of documents is called corpus $D = (\mathbf{w}_1, \ldots, \mathbf{w}_M)$. It consists out of M documents $\mathbf{w} = (w_1, \ldots, w_N)$ which are itself separated in words w_i. These words are vectors of length V. V refers to the length of a vocabulary which holds all the words occurring in the corpus. The vector for a specific word w_i contains all 0 except for index $j \in \{1, ..., V\}$ which represents this very one word in the vocabulary. This notation may indeed be extended through the addition of indices for documents, but this is not done here or in the standard literature on topic models due to its unnecessary complexity.

1.1 LDA model

Latent Dirichlet Allocation is a Bayesian approach and is often associated with the class of hierarchical models [A. Gelman, 2014]. The idea is based on the representation of exchangeable random variables (acc. to de Finetti) as mixture of distributions. Given that documents $\mathbf{w}$ and words w_i in each document - both considered as random variables in this setting - are exchangeable in such a way, a mixed model such as the LDA model is appropriate [Blei, 2003].

The following notation is used in conjunction with the LDA model. Let z_j be the topics with $j \in \{1, \ldots, k\}$. In the LDA setting we assume for every topic z_j there is a term distribution

$$\beta_j \sim Dir(\delta)$$

We further assume each document w has a distribution of topics.

$$\theta \sim Dir(\alpha)$$

Then each word w_i of $\mathbf{w}$ is generated by the following process:

1. Choose $z_i \sim Mult(\theta)$
2. Choose $w_i \sim Mult(\beta_i)$ This distribution will be referred to as $p(w_i|z_i, \beta)$

You can summarize this setup in a plate diagram as shown in figure 1. The notation above, which is also used within the diagram, coincides with the

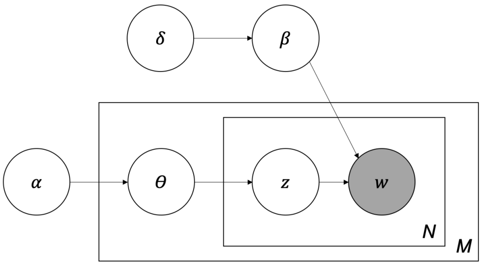

Figure 1: The well-established plate diagram for the standard LDA model extended by the parameter δ. The slightly bigger box represents the generative model of the corporis M documents. The smaller plate represents the iterative generation process of the N words of each document with the aid of the topics. See also "smoothed LDA model" in [Blei, 2003] for comparisons.

notation of [K. Hornik, 2011].

In order to estimate the model parameters, the first task is to calculate the posterior distribution, which consists of the joint distribution in the numerator and the marginal distribution in the denominator.

$$p(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta) = \frac{p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta)}{p(\mathbf{w}|\alpha, \beta)} \tag{1}$$

The joint distribution numerator can be derived straight forward.

$$p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = p(\theta|\alpha) \prod_{i=1}^{N} p(w_i|z_i, \beta)\ p(z_i|\theta) \tag{2}$$

One can obtain the marginal distribution of a document $\mathbf{w}$, by integrating out the parameter θ and summing over the topics z_j. Nevertheless, this expression is intractable.

$$p(\mathbf{w}|\alpha, \beta) = \int p(\theta|\alpha) \left(\prod_{i=1}^{N} \sum_{z_i} p(z_i|\theta) p(w_n|z_i, \beta) \right) d\theta \tag{3}$$

The literature divides the approaches to calculating posterior distribution into two main categories.[Blei, 2012] distinguishes between sampling based algorithms and variational algorithms. [Powieser, 2012] lists a total of 6 algorithms that can be used to estimate parameters in the LDA model. This thesis will be confined to the two most cited and most used members of the two main groups. One approach is to simulate the posterior density by iteratively sampling - the so-called Gibbs sampling. The second approach is a deterministic method, a modified version of the well-known EM algorithm [AP Dempster, 1977]: the Variational EM algorithm (VEM algorithm) [Wainwright and Jordan, 2008]. In the following two sections the both approaches are roughly outlined to give the reader some insight into the Bayesian inference underlying the algorithms.

1.1.1 Variational EM algorithm

In the VEM algorithm for the LDA model is a mean field approach which varies the steps E and M of the EM algorithm in a way such that this algorithm becomes solvable. Note that the main problem of calculating the marginal distribution is, to derive the conditional probability of some hidden variables given some observed values (”evidence”). The variation of the EM algorithms consists mainly in approximating the directly intractable E step. Rewriting the log of the border density of $\mathbf{w}$ as follows in (4), results in the fact that the marginal density can be estimated downwards with the aid of Jensen’s inequality.

$$\log p(\mathbf{w}|\alpha, \beta) = \log \int \sum_z p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) d\theta \tag{4}$$

$$= \log \int \sum_z \frac{p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) q(\theta, \mathbf{z})}{q(\theta, \mathbf{z})} d\theta \tag{5}$$

$$\geq \int \sum_z q(\theta, \mathbf{z}) \log p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) d\theta - \int \sum_z q(\theta, \mathbf{z}) \log q(\theta, \mathbf{z}) d\theta \tag{6}$$

$$= \mathbb{E}_q[\log p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta)] - \mathbb{E}_q[\log q(\theta, \mathbf{z})] \tag{7}$$

Here $q(\theta, \mathbf{z})$ is an arbitrary distribution which can be called the variational distribution.

$$q(\theta, \mathbf{z}) \widehat{=} q(\theta, \mathbf{z}|\gamma, \phi) = q(\theta|\gamma) \prod_{i=1}^{N} q(z_i|\phi_i) \tag{8}$$

The right hand side $L(\gamma, \phi, \alpha, \beta) := \mathbb{E}_q[\log p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta)] - \mathbb{E}_q[\log q(\theta, \mathbf{z})]$ be called ”lower bound”. It can be shown that $\log p(\mathbf{w}|\alpha, \beta) - L(\gamma, \phi, \alpha, \beta)$ is the Kullbak Leibler divergence (D_{KL}) of the true posterior and the variational distribution. From equations (4)-(7) follows that:

$$\log p(\mathbf{w}|\alpha, \beta) = D_{KL}(q(\theta, \mathbf{z}|\gamma, \phi)||p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta)) + L(\gamma, \phi, \alpha, \beta) \tag{9}$$

Since the marginal is fixed, we conclude, that minimizing the KL-divergence is equivalent to maximizing the lower bound (see [M. Jordan, 1999] and [Wainwright and Jordan, 2008], for details of the derivation of the lower bound see [Blei, 2003]).

$$(\gamma^*, \phi^*) = \underset{\gamma,\phi}{\operatorname{argmin}}\, D_{KL}(q(\theta, \mathbf{z}|\gamma, \phi)||p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta)) \tag{10}$$

$$= \underset{\gamma,\phi}{\operatorname{argmax}}\, L(\gamma, \phi, \alpha, \beta) \tag{11}$$

The variation of the EM algorithm thus is to use the variational distribution $q(\theta, \mathbf{z}|\gamma^*(\mathbf{w}), \phi^*(\mathbf{w}))$ instead the posterior distribution $p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta)$. Now the two steps of the VEM algorithm are:

(1) **E step** Optimize the variational parameters θ and ϕ for every document in the corpus. This can be done analytically by deriving the derivatives of the KL divergence. And set them to zero.

(2) **M Step** Maximize the lower bound using the optimized parameter of the E step with respect to α and β.

1.1.2 Gibbs sampling

The second method to approximate the posterior distribution is Gibbs sampling, a so-called monte Carlo method. Instead of calculating the distributions for β and θ, the primary task is to find the posterior distribution over $\mathbf{z}$ given the document $\mathbf{w}$. Gibbs sampling is also known as a Markov Chain Monte Carlo method. The name refers to the simulation process by which a chain of values is simulated whose limiting distribution desirably converges against the true distribution [M. Steyvers, 2006]. (12) shows the distribution, which is sampled from iteratively.

$$p(z_i = j|z_{-i}, w) \propto \frac{n^{(l)}_{-i,j} + \delta}{\sum_t n^{(t)}_{-i,j} + V\delta} \frac{n^{(d_i)+\alpha}_{-i,j}}{n^{(d_i)}_{-i} + k\alpha} \tag{12}$$

$z_i = j$ ... word-topic assignment of word i to topic j
z_{-i} ... vector of word-topic assignments without the entry for word i
$n^{(l)}_{-i,j}$ ... number of times the lth word in the vocabulary is assigned to topic j, not including the assignment for word i
d_i ... document in the corpus which includes word i
δ, α ... parameters of the prior distributions for β and θ

Usually the word-topic distributions $\beta^{(l)}_j$ for the words $l = 1, ..., V$ and topics $j = 1, .., k$ and topic-document distributions $\theta^{(d)}_j$ for the documents $d = 1, ..., D$ and the topics $j = 1, ..., k$ will be of interest. (13) and (14) shows the predictive distributions denoted as ”estimators”.

$$\hat{\beta}_j^{(l)} = \frac{n_{-i,j}^{(l)} + \delta}{\sum_t n_{-i,j}^{(t)} + V\delta} \tag{13}$$

$$\hat{\theta}_j^{(d)} = \frac{n_{-i,j}^{(d_i)+\alpha}}{n_{-i}^{(d_i)} + k\alpha} \tag{14}$$

For derivation and more details regarding the Gibbs sampling procedure see [M. Steyvers, 2006].

1.1.3 Implementation

In this thesis, the implementation of the LDA model and its estimation is mainly based on using the package *topicmodels* of Kurt Hornik. The package *topicmodels* can apply both the VEM algorithm as well as Gibbs sampling in order to fit the model. In addition, the package *tidytext* is being used for text structuring and embedding. Whereby there are other packages besides this implementation of the LDA model, topicmodels is particularly convenient, because tidytext was designed by its developers to work perfectly in combination with topicmodels [Silge and Robinson, 2017, p. 89].

1.2 Artifical Neural Networks

Artificial neural networks (ANN) are much more versatile than the LDA model. There are not only various forms of artificial neural networks, but also a very large number of application areas. Quite as machine learning procedures in general, also deep learning algorithms are divided into two broad categories: supervised learning, where a superset instance provides the algorithm with the output required to learn, and unsupervised procedures that internally train predefined models to find patterns in the input signals. In this chapter we will focus heavily on the former group of ANNs. Also, this chapter is intended to give the reader an overview of the research on neural networks as well as the background of their development.

Research on ANNs dates back to the 1940s, when [McCulloch and Pitts, 1943] introduced the so called “M-P neuron”. Whereby this neuron had only a bivariate input and output, Rosenblatt later extended this idea to a network of M-P neurons, which allowed to set up a simple classification algorithm [Rosenblatt, 1958]. A perceptron in its basic form (single perceptron) is a binary classifier.

Imagine input data of a simple perceptron in the form of a matrix.

$$X = \begin{bmatrix} x_{11} & \dots & x_{1k} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{nn} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{bmatrix}$$

The dependent variable thus is a vector $\mathbf{y} = y_1, \dots, y_n$, with $y_i \in \{0, 1\}$. Consider the lines of the X mtrix as vectors $\mathbf{x}_1, \dots, \mathbf{x}_n$, with $\mathbf{x}_i \in \mathbb{R}^k$. The entries of each of the vectors are weighted with $\mathbf{w} = w_1, \dots, w_k$ with $w_j \in \mathbb{R}$ and aggregated in a function h e.g. a sum.

$$h(\mathbf{x}, \mathbf{w}) = \sum_{j=1}^{k} x_j w_j$$

Using a so called *activation function* h is mapped to the output space, which is in this case $O = \{0, 1\}$. At this point a step function serves as activation function.

$$a \circ h(\mathbf{x}, \mathbf{w}) = \begin{cases} 0 \text{ if } h(\mathbf{x}, \mathbf{w}) \leq 0 \\ 1 \text{ else} \end{cases}$$

The matrix X is passed vector by vector to the percepron and the output is compared with the values for $\mathbf{y}$. During this procedure the weights are iteratively tuned by a simple updating algorithm using the pairs $\mathbf{x}_i$ and y_i.

The algorithm of the simple perceptron is schematically shown in Figure 2. This diagram corresponds to the common representation in education [Mukherjee, 2019], although a horizontal perspective is often chosen.

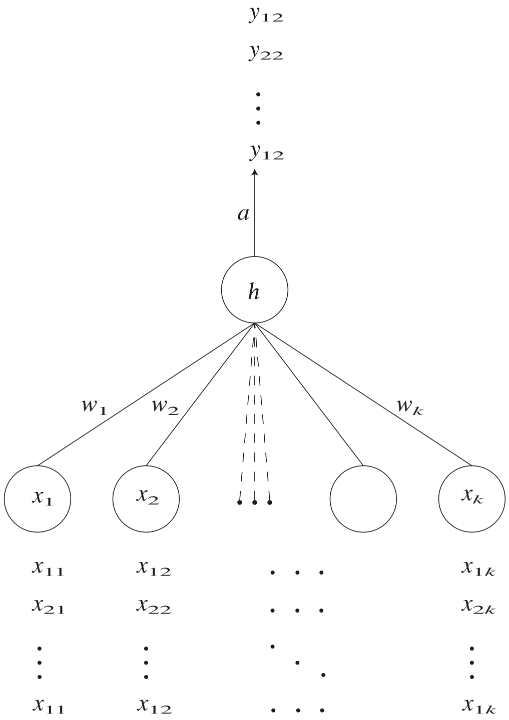

Figure 2: Schematic diagram of a simple perceptron by [Rosenblatt, 1958]

A Appendix

References

[A. Gelman, 2014] A. Gelman, J. Carlin, H. S. D. D. A. V. D. R. (2014). *Bayesian Data Analysis*. Chapman and Hall/CRC.

[AP Dempster, 1977] AP Dempster, NM Laird, D. R. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Rojal Statistical Society*.

[Blei, 2012] Blei, D. (2012). Probabilistic topic models. *Communications of the ACM*, 55(4):77.

[Blei, 2003] Blei, D., N. N. J. M. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3.

[K. Hornik, 2011] K. Hornik, B. G. (2011). topicmodels: An r package for fitting topic models. *Journal of Statistical Software*, 40(13).

[M. Jordan, 1999] M. Jordan, e. a. (1999). An introduction to variational methods for graphical models. *Kluwer Academic Publishers - Machine Learning*, 37:183–233.

[M. Steyvers, 2006] M. Steyvers, T. G. (2006). Probabilistic topic models. *Latent Semantic Analysis: A Road to Meaning.*

[McCulloch and Pitts, 1943] McCulloch, W. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics.*

[Mukherjee, 2019] Mukherjee, S. (2019). Lecture notes for probabilistic machine learning. Duke University.

[Powieser, 2012] Powieser, M. (2012). Latent dirichlet allocation in r. Master's thesis, Vienna University of Economics and Business.

[Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron, a probabilistic model for information storage and organization in the brain. *The Psychological Review.*

[Silge and Robinson, 2017] Silge, J. and Robinson, D. (2017). *Text Mining with R: A Tidy Approach.* O'Reilly Media.

[Wainwright and Jordan, 2008] Wainwright, M. J. and Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305.

List of Figures